LOAD BALANCING FOR BIG DATA COMPUTING WITH DATA LOCALITY AWARENESS

By

Yasir Arfat

A thesis submitted for the requirements of the degree of Master of Science
Computer Science

Faculty of Computing and Information Technology

King Abdulaziz University - Jeddah

Rajab 1438H – April 2017G


بسم الله الرحمن الرحيم

Load Balancing for Big Data Computing with Data Locality Awareness

By

Yasir Arfat

A thesis submitted for the requirements of the degree of Master of Science in Computer Science

Supervised by

Prof. Rashid Mehmood

Dr. Aiiad Albeshri

FACULTY OF COMUTING & INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY
JEDDAH-SAUDI ARABIA
Rajab 1438H – April 2017G

Load Balancing for Big Data Computing with Data Locality Awareness

By

Yasir Arfat

This thesis has been approved and accepted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

EXAMINATION COMMITTEE

	Name	Rank	Field	Signature
Internal Examiner	Dr. Iyad Katib	Associate Professor	Computer Science	
External Examiner	Dr. Abdullah Saad AL-Malaise AL-Ghamdi	Associate Professor	Information Systems	
Advisor	Prof. Rashid Mehmood	Professor	Computer Science	
Advisor	Dr. Aiiad Albeshri	Assistant Professor	Computer Science	

KING ABDULAZIZ UNIVERSITY

Rajab 1438H – April 2017G

Dedication

I would like to dedicate this work to my beloved parents, my brother and sisters, high encouragement and priceless prayers and support for my success throughout this work.

ACKNOWLEDGMENT

In the Name of Allah, the Most Gracious, the Most Merciful. All thanks and praise is due to Allah and peace and blessings be upon His Messenger Muhammad. I thank Allah for giving me the strength, perseverance and ability to accomplish my goals.

Foremost, I would like to acknowledge and thank my supervisor Prof. Rashid Mehmood for his continuous support during my MSc degree. I could not have imagined having a better advisor and mentor for my MSc degree other than him. His motivation, immense knowledge, and mentorship helped me to develop problem solving and research skills. His patient guidance helped me during research and thesis writing. He had spent a tremendous amount of time with me during research when I was stuck with Research problems. His research methodology and mode of teaching has proved to be an inspiration for me. His advice and constructive suggestions kept me motivated and made me stay focused towards my final goals. He continued to believe in me and support me throughout my research, which gave me the confidence to achieve my goals. I am forever indebted to him for his help in completion of this Thesis in required time. I would also like to thank Dr. Aiiad for support throughout this thesis. His moral support and encouragement motivated me to complete this thesis in time. He also facilitated the research and sorted out problems related to research proposal. This research would not have been possible without his help and commitment.

I am also, grateful to all my teachers in the Department Computer Science, in particular, and in King Abdulaziz University (KAU) in general. I am so proud to be a student of KAU.

I would like to express my deepest appreciation and love to all my family “Parents, brother and sisters” for helping me, and for their prayers. I would like to extend heartily thanks to all people who supported me and gave me the strength to go through the project, especially during hard times.

Load Balancing for Big Data Computing with Data Locality Awareness

Yasir Arfat

ABSTRACT

Load balancing of computational tasks and data plays a critical role in big data systems. Load balancing attempts to optimize the overall computation or system performance, such as throughput, solution (or response) time, and resource usage. This is achieved by an optimum allocation of data and workloads to the available resources. Data locality techniques aim to map tasks to the nodes where the relevant data resides. The two performance objectives (i.e. load balancing and data locality) are usually at odds. There is a need to find optimal strategies to maximize the performance.

The aim of this thesis is to develop data locality aware load balancing techniques for big data computing and apply these to practical problems of high significance. A detailed review of the literature is presented to identify major challenges in big data research. These challenges include, among others, load balancing and data locality. A load balancing technique that is also aware of data locality has been developed and is applied to a graph-based road transportation problem. We have modelled the entire US road network data that contains approximately 24 million vertices and 58 million arcs; the specific aim of this research is to identify various points of interests (PoIs) in the regions including living places and healthcare centres. These PoIs are subsequently used to find the shortest paths among them for planning

and operations purposes. The relevant algorithms that we have developed are presented in the thesis. The algorithms have been implemented using the Spark big data platform tools on the Aziz supercomputer, a Top500 supercomputer in the world according to the June and November 2015 rankings. The results are collected in terms of the shortest paths and the road networks are visualised as graphs. The performance of the load balancing and data locality awareness techniques is analysed against a varying number of nodes on the Aziz supercomputer and a good speedup has been reported. Conclusions are drawn with directions for future work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations……..……Meaning

HDFS…….Hadoop File System
YARN……....……Yet Another Resource Negotiator
HLL……..……High Level Language
ETL…….Extraction Transfer and Load
DAG…….Directed Acyclic Graph
BDAS……....Berkeley Data Analytics Stack
HPCC……High Performance Computing Cluster
ECL…….Enterprise Control Language
BI…….Business Intelligence
HPC……..High Performance Computing
HEP……..High-Energy Physics
LaSA……Locality aware Scheduling Algorithms
CCS…….Concurrent Container Slot
APH…….Analytical Process Hierarchy
DDP……..…….Dynamic Data Placement
DALM……..……..Dependency Aware data Locality for MapReduce
FIFO……First In First Out
HPSO……High Performance Scheduling Optimizer
DRAW……Data Group Aware data placement
DGM……Data Group Matrix
ODPA……Optimal Data Placement Algorithm
BLOS……Bipartite Graph Oriented Locality Scheduling
BRDG……Bipartite Request Dependency Graph
DCCM…….Dynamic Concurrency Control Module
LDS……..Local Dense Subgraph
RDBMS……Relational Data Base Management System
BRDG……...…….Bipartite Request Dependency Graph
LSH…….Locality Sensitive Hashing
PAGE……..Partitioning Aware Graph computation Engine
NGS…….Next Generation Sequencing
FPGA……..Filed Programmable Gate Array
BLAST…….Basic Local Alignment Search Tool
D4M……Dynamic Distributed Dimensional Data Model
DIMACS……...Discrete Mathematics & Theoretical Computer Science
POI/PoI…….Points of Interests
OSM……...Open Street Map
NB & HC……Neighbourhood & Healthcare Centre
DC & RI……District of Columbia & Rhode Island
CO & FL……...…….Colorado & Florida
CAL……....California

Chapter 1

Introduction

1.1 An Overview of Big Data

Now days due to the Internet of things (IOT) a large amount of data is being generated. Every day this data is increasing at an exponential rate. It is expected that data will be increase coming years. So with this rapid progression of data and information, we need to process the large amount of data and analysis. There are various sources of big data such as social media, black box data, stock exchange data, power grid data, transport data and search engine data. Data that is generated from these sources as mentioned above is so big that it cannot be able to process by the traditional tools. Any data that comes under following characteristics is called the Big Data. Big Data refers to the emerging technologies that are designed to extract value from data having four Vs characteristics; volume, variety, velocity and veracity. So we can say that 'Big Data' is collection of large amount information with increasing capacity together and store huge volume of data, that can be structured, semi structured, unstructured and time stamped, using the statistical and regression techniques for the analysis of these data.

Now a days there are various applications of big data involves from the medicine, pharmacy, aviation bioinformatics, government, geophysics etc. It also involves many

other disciplines. To manage and process this large amount of data we need tools for the big data that manage large volume of data. There are various tool have been introduced to manage the huge amount of data as these are Hive, Hadoop, Spark etc. Each tool works on the specific levels for mange and process the information. We used these tools for the specific purpose. Hadoop [1] is an open source software to process the big data. It is very popular for the possessing of larger amount of data. It has several characteristics scalability, reliability, fault tolerance, high availability, local processing & storage, distributed and parallel computing faster and cost effective. It uses the simple programming model for the processing of large amount of dataset across clusters of computers. It was developed to scale up from single server to thousands of machines and provides feature like local computing and storage. It library was designed in such a way that it can itself identify failure and recovery from failure by application layer. It has two important components Hadoop file system (HDFS) and MapReduce.

HDFS [2] is very important component of Hadoop. It has master and slave architecture. It consists of two components NameNode and DataNode. It is distributed file system that delivers the high throughput for the getting the application. It is very similar to the exiting distributed file system. It can process large data set and it was designed for low cast system. Main objective of HDFS is to provide the following features, increases the throughput by decreasing the network congestion, fault detection and isolation, access to large data set, accessing the streaming data, simple coherency model, portability among different hardware and software.

MapReduce [3] is an open source framework for the processing of large data set. MapReduce consists two basic components map and reduce phase. Map phase divides

the workload into smaller sub workloads and assigns tasks to Mapper, which processes each unit block of data. The outcome of mapper is a sorted list of (key, value) pairs. This list is passed (also called shuffling) to the next phase. Reduce phase analyzes and merges the input to produce the final output. The final output is written to the HDFS in the cluster. It splits data set into pieces, which will be processing in parallel manner by map task. Mapper output sorted by the framework, that is further send to the reduce task. Both output and input will be store in separate file. This framework also examines the monitoring and scheduling task and task, which failed during execution, re-execute them. It provides feature like reliability and fault tolerance. MapReduce is well liked due it simplicity, scalability, speed, recovery and minimal data motion.

It has several limitations these are: it creates bottleneck due to reading the disk read again and again, It is inefficient for iterative process, it is not primarily designed for the iterative processes. Now a day main challenge that MapReduce is facing is load balancing and data locality in heterogeneous environment. Its performance is suffered in distributed heterogeneous environment. In case of homogenous performance is alike, but in the heterogeneous environment node perforce is quite different because node very the capacity of computation, communication, architecture, memories and power. In case of power performance node will spend more time of the whole MapReduce Job is long. Data skewness takes longer time to finish the task and significant impact on the performance. MapReduce Depends on the slowest running of task in the job, if task takes longer time to finish then other (straggler) it can delay the process of entire job. MapReduce uses the static hash function to partition the Data. Straggling cause by

several factors such as fault in hardware, slow machine, and Speculative execution is solution for the above statement but it decrees the job execution time.

1.2 Recent studies

There are Recent studies suggest that data locality and load balancing are important issues to improve the performance of the system. However, due to inappropriate partition algorithm may result in poor network quality, overloading some reducer and extension of the execution time of job. Using appropriate algorithms to process the skew data will lead negative impact on the system performance.

- All Reduce do have same performance.
- Hash partitioning lead to partitioning skew (it brings another problems).
- Join Algorithm takes long time to balance the data cause by the partitioning skew.
- Moving mapping results to Reduce over network cause another extra overhead.

There is another important issue in MapReduce that is data locality. Data Locality decreases the network traffic and increases the performance. Locality Aware resource allocations in data intensive computing application challenges such as data placement, access and computation can be minimized by data locality. Due to distribute file system data locality problem occurs data intensive applications and cloud environment bring new challenges for the data such as computation, assessment and placement. These exiting challenges can be reduce.

Moreover, MapReduce contains the various nodes which heterogeneous in their computing capacity for the various. It is an important for the partitioning algorithm to place the based the capacity of the nodes in clusters. Data locality is an important factor

on the heterogeneous environment**.** Heterogeneous environment the capacity of the Hard disk is not same. It uses the data locality aware partitioning scheme. Current Hadoop implementation assumes the computing nodes in the cluster are homogenous in nature. Data locality has not been taken into account for launching the speculative map task; it assumes that most maps are local data. Unfortunately, both homogeneity and data locality assumption is not satisfied in the virtual environment. Ignoring the data locality is decrease the performance of the map reduces. To improve the perform of Hadoop MapReduce in heterogeneous environment there is need of new approach that do not only handle the load balancing but also data locality in heterogeneous environment.

1.3 Research Objectives

The aim of this thesis is to develop data locality aware load balancing techniques for big data computing. The proposed techniques shall be implemented and tested using widely used applications or algorithms. The thesis objectives are:

- Carry out a review of big data literature and identify the main challenges.
- Understand the state-of-the-art related to data locality and load balancing research.
- Developed a new approach for data locality aware load balancing.
- Test and evaluate the proposed data locality aware load balancing approach using an application of big data.

1.4 Contribution

In this thesis, a detailed review of the literature is presented to identify major challenges in big data research. These challenges include, among others, load balancing and data locality. A load balancing technique that is also aware of data locality has been developed and is applied to a graph-based road transportation problem. We have modeled the entire US road network data that contains approximately 24 million vertices and 58 million arcs; the specific aim of this research is to identify various points of interests (PoIs) in the regions including living places and healthcare centres. These PoIs are subsequently used to find the shortest paths among them for planning and operations purposes. The relevant algorithms that we have developed are presented in the thesis. The algorithms have been implemented using the Spark big data platform tools on the Aziz supercomputer, a Top500 supercomputer in the world according to the June and November 2015 rankings. The results are collected in terms of the shortest paths and the road networks are visualised as graphs. The performance of the load balancing and data locality awareness techniques is analysed against a varying number of nodes on the Aziz supercomputer and a good speedup has been reported. Conclusions are drawn with directions for future work.

1.5 Thesis Organization

Thesis is organised as follows:

Chapter 1 introduces the research subject, providing a simple overview of the problem, and addressing the objectives.

Chapter 2 provides an overview of background and definitions.

Chapter 3 presents the Literature review.

Chapter 4 provides the implementation of an application transport healthcare.

Chapter 5 presents the load balancing and data locality.

Chapter 6 presents the results and analysis of the proposed technique.

Chapter 7 concludes the work that has been done in this study and proposes recommendations for the futures.

Chapter 2

Background

2.1 Big Data

Big data is the large amount of data, refers to the emerging technologies, it has various characteristics but any data that comes under following characteristics we can call it as “Big Data” as these characteristics are; volume, variety, velocity and veracity. So we can say that ‘Big Data’ is collection of large amount information with increasing capacity together and store huge volume of data, that can be structured, semi structured, unstructured and time stamped, using the statistical and regression techniques for the analysis of these data.

There are various tools has been developed to process the big data. There are various categories and types of big data tools. Each big data tool can use to process the large volume of data; each of them has various characteristics and limitations to process the large volume of big data, these characteristics and limitations we have explored and presented in this section.

2.2 Data Commercial and Open Source Tools

As discussed above, now days there are many tools and framework available in the market for process the big data. Each tool and framework has its own characteristics, limitations and feature according to the nature of application where we are applying it for solving the problems and processing the huge amount of data. We have find out that

we can divide the big tools into two categories open source and commercial. We can also further sub divide these tools into three categories storage, processing and analysis as show in Fig 2.1.

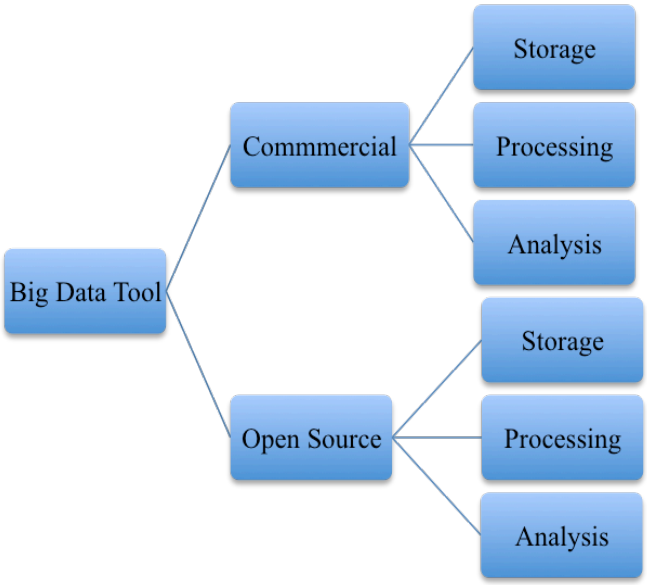


Figure 2.1 Big Data tools and technologies

Hadoop [1] is an open source software to process the big data. It has several characteristics scalability, reliability, fault tolerance, high availability, local processing & storage, distributed and parallel computing faster and cost effective. It uses the simple programming model for the processing of large amount of dataset across clusters of computers. It was developed to scale up from single server to thousands of machines and provides feature like local computing and storage. It library was designed in such a way that it can itself identify failure and recovery from failure by application layer.

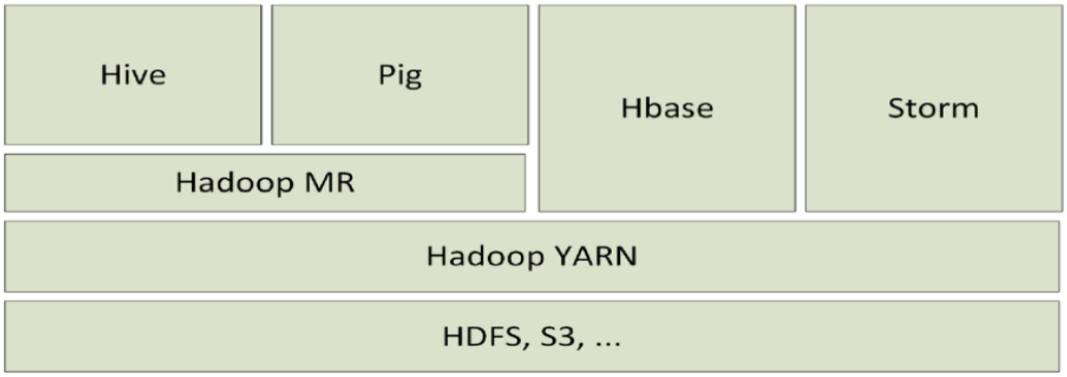


Figure 2.2 Various components of Hadoop [4]

Hadoop file system (HDFS) [2] is very important component of Hadoop. It has master and slave architecture. It consists of two components NameNode and DataNode. It is distributed file system that delivers the high throughput for the getting the application. It is very similar to the exiting distributed file system. It can process large data set and it was designed for low cast system. Main objective of HDFS is to provide the following features, increases the throughput by decreasing the network congestion, fault detection and isolation, access to large data set, accessing the streaming data, simple coherency model, portability among different hardware and software.

YARN [5] is an open source framework for job scheduling and cluster resource management. It is an important feature for the next generation of Hadoop2. Initially designated by the Apache as reformatted resource manager. For the big data application YARN now days categorized as large scale distributed operating system. The essential theme of YARN is to divide the functionalities of resource management and job scheduling and monitoring into distinct daemons. The data computation framework formed by the Node manager and resource manger. The fundamental authority that determines resources between all the applications in the system is resource manger. The main responsibility of the node manager is to monitor the usage of resource like

network, memory CPU and disk, reporting to scheduler and resource manger. It has features like multi-tenancy, compatibility, scalability, and cluster utilization.

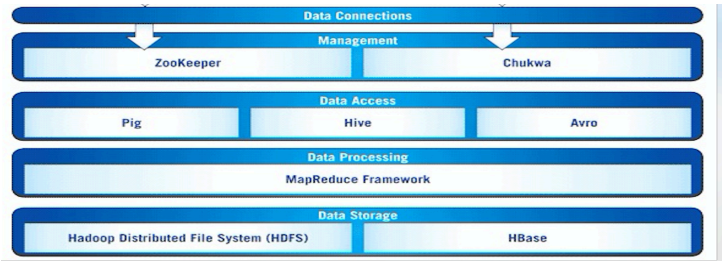


Figure 2.3 Layer of big data tools [6]

[Processing, Map, Reduce, Parallel processing] MapReduce [3] an open source framework for the processing of large amount of data. MapReduce consists two basic components map and reduce phase. Map phase divides the workload into smaller sub workloads and assigns tasks to Mapper, which processes each unit block of data. The outcome of mapper is a sorted list of (key, value) pairs. This list is passed (also called shuffling) to the next phase. Reduce phase analyzes and merges the input to produce the final output. The final output is written to the HDFS in the cluster. It splits data set into pieces, which will be processing in parallel manner by map task. Mapper output sorted by the framework, that is further send to the reduce task. Both output and input will be store in separate file. This framework also examines the monitoring and scheduling task and task, which failed during execution, re-execute them. It provides feature like reliability and fault tolerance. MapReduce is well liked due it simplicity, scalability, speed, recovery and minimal data motion. It has several limitations these are it creates bottleneck due to reading the disk again and again, It is inefficient for iterative process, it is not primarily designed for the iterative processes.

Apache Pig [7] is an open source platform for analyzing the huge dataset that having high level language (HLL) representation of data analysis program. For the evaluation of

program, it is also coupled with infrastructure. The main characteristic of Apache Pig is that its structure is flexible to substantial parallelization that allows the handle the large dataset. Currently, infrastructure layer of pig have a complier the production sequence of map and reduce programs, for that large-scale parallel implantations already exists. Language layer of pig consists of textual language, which is called Pig Latin; it has following characteristics, ease of programming, optimization opportunities and extensibility.

Apache Hive [8] is open source software that enables managing a large data sets and querying that existing in distributed storage. It is provides facilities like adhoc query, analysis of huge dataset and summarization of data store in Hadoop file. It is data warehouse infrastructure that builds on the Hadoop. To overcome the problems of MapReduce they introduce the Hive. It provides a scheme to project structure, onto the data. It queries the data using a SQL like language called HiveQL. When there is difficulty to express the logic in HiveQL, it also enables the map reduce developer to custom the map and reduce. It have **characteristics** like familiarity mostly user are familiar to SQL query so its to use, provide quick response on large data set, and scale and extensible. Data extraction transfer and load (ETL) is easy using its tools. It executes the query using the MapReduce.

Apache storm [9] is a distributed real-time computation system. It is real time stream data processing system. It is am open source software system processing the big data. It enhances the trustworthy real time processing data potentials to enterprise Hadoop. It is very simple; any programming environment can use it. It is very useful for the YARN circumstances requiring machine learning, ETL, continuous monitoring of

operations, distributed RPC and real-time analytics. It has five characteristics that make it ideal for real time processing of data. There **characteristics** are scalable, fault tolerant, reliable, fast and easy to operate.

Spark [10] is an open source tool for processing the large data set. It is also next generation of big data applications and alternate for Hadoop. To overcome the issues like disk I/O and performance improvement of Hadoop they introduced the spark. It has several features like memory computation that make it unique. It provides facility like cashing the data in memory. Spark supports the several programming languages i.e. python, Java and scala for the processing the large volume of data. It performs the processing on the large data set very quickly. It process data 100x faster than Hadoop and map reduce. It has exaction engine like DAG, which care a cyclic data flow, and in memory computing. It has these characteristic speed preform task very fast, ease of use that it supports several platforms, generality it means that there are various liberalities i.e. SQL, GraphX, and Data Frame that can be combined in the single application. It also runs anywhere means we can access the data from HBase, Hive and Tachyon. Limitation of spark includes that memory consumption issues are not easy to hand it user-friendly way it takes more memory. Spark till debugging the errors, will take a time to mature.

Spark Streaming [11] is a component of spark [10]. It is a core API of spark. It makes the streaming processing of data more fault tolerant, scalable and provide the high throughput. It uses the complex algorithm to process the data source like Twitter, Kafka, Kinesis and TCP sockets for the advanced methods like joins windows map and reduce. Lastly, processed data could be presented in dashboard, DB, and file system. Moreover we can also apply the algorithm of spark on the data stream these algorithms

are graph processing and machine learning. It also proved the various other features like fault tolerance, spark integration, ease of use having different programming languages Java, python and scala. It also provides deployment option it can read the data from the ZeroMQ, Twitter, and HDFS etc. In short spark streaming enables the streaming application scalable, reliable and fault tolerant.

Blink DB [12] is used for processing the large scale of data for interactive quires of SQL with extremely parallel, approximate query engine. It enables the clients to adjust the accuracy of query for the response time, allows collaborative queries at large data by executing the queries on the sample data. There two main features of Blink DB, first it have an adaptive optimization framework which constructs and keeps multi dimensional data samples from the original data over time. Secondly it has the strategy for appropriate sample selection, which is dynamic, and on the basis of query's response time and accuracy needs. Main goal of Blink DB is to support the interactive query i.e. aggregate query on large amount of data.

Berkeley data analytics stack (BDAS) [13] is an open source software tool [84] for processing the big data. It developed by the AMPLab, which combine the different component of software for application of big data. It has five layers namely resource virtualization at bottom, storage, processing engine, access and interfaces, at each layer there are different components. At resource virtualization it have two components. Mesos [14] and Hadoop YARN [5]. Basic purpose of mesos, it runs the kernel on each machine and delivers application like kafKa, Spark, and Hadoop. Mesos also provides feature like scalability, fault tolerance, scheduling of multiple resources such as ports, memory, disk, and CPU. Next layer is storage layer, it consist of Tachyon, HDFS,

Succinct, S3 and Ceph. Tachyon [15] is a storage system, which is memory centric that allows the user to share reliable data very high speed through the cluster framework. It has higher performance as compared to the HDFS in terms of memory. It has caching mechanism, which enables it to decrease the memory access.

On the processing layer it contains the spark core, which has great feature like speed, ease of use and generality. At they top layer there is access and interface, it contain several components. These components are BlikDB, SharkSQL, GraphX, MLBase, MLib, SparkR, Sample Clean, Splash, MLpipelines, velox and Spark streaming. On this top layer there are many wrapper of spark like spark streaming (Large scale real time processing system), MLBase (Distributed machine learing liabray for the spark)[16],MLib(Machine learning library)[17], SparkSQL[18](It is a module of spark with wroks on structured data), SparkR (Contains distributed data frame implementation), GraphX(Resilient distributed graph system on spark) [19] BlikDB(queries with bounded errors and bounded response time on large volume of data). Presently, spark and BDAS got popularity due to the higher performance on Hadoop. At the most top layer it contained most popular and very valuable components but still there is enhancement is needed.

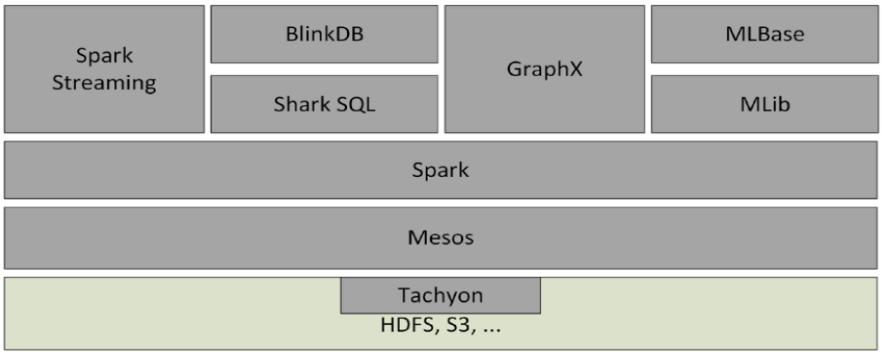


Figure 2.4 Berkeley Data Analysis Stack [4]

MongoDB [20] is an open source tool for processing the large amount of data store in a database. It has key features like automatic scaling, high performance and availability. In automatic scaling it has horizontal scaling, which is the main fragment of core functionality. Data is distributed among cluster using the automatic sharding. The set of replica delivers the low latency and high throughput for consists read. In high availability it contain the replica sets that provides recovery on the failure automatically and data redundancy. For high performance, it has the indexes, which makes the querying process very faster. Moreover, It minimizes the I/O activity on database by embedding the models.

R [21] is an analytical tool for big data application. It is used for the statistical analysis, packages, and graphical visualization. R offers a broad range of statistical analysis i.e. classification, classical data testing, clustering, analysis of time series, graphical methods nonlinear modeling and linear modeling. It handles the data in effective manner, and ability of data storage. It guarantees the smooth procedure for the matrices and vector data so that statistical calculations to be optimized in terms of language. It has R package, which is comprehensive R archive network, so that user can get any required statistics process library. In statistical filed it can be promoted as open source. It is very supportive for the many platforms such as Mac OS, Unix and Windows.

Drayed [22] enable the developer and programmers to utilize the computer resources cluster, data center for running parallel. It also allows the user to use the many machines having multiple cores, with out knowledge program currency. It is framework for parallel processing of data establishing path among the programs in type of graph. Often, programmer writes the functions of map and reduces by using the dryad they

required create a graph that evaluate the conforming data. Dryad processes the data in a direct acyclic graph (DAG) manner. This framework has sufficient function for processing the big data instead of parallel data operations. There are some obstacles for using the system, inexperience programmers, data minors and analyzer of the data. Consequently, there is need to develop new technique for processing of data, through high quality.

Syncfusion [23] platform for the big data designed for window. There are several challenge exits for the big data processing and managing. Thus it provides very useful feature including processing the query on the structured and unstructured data, cost effective storage. It deals with marvelous potential. With linear scalability we can be able store any type of data on the commodity hardware. Currently, Syncfusion has made these dominant technologies available on Windows. Using this platform we can access the Hadoop framework completely. There are many complies using this framework Adobe, Yahoo, Facebook, Hulu, Microsoft and Amazon.

Cloudera [24] is a commercial platform for the processing big data. It provides the Impala that deals with real time massively big data processing to the Hadoop. It provides the very secure, simple, and fastest platform for the Hadoop. Its helps the user to solve the most challenging issues related to business domain. Moreover Cloudera also contains the core components of Hadoop these components are map reduce, YARN, and HDFS. It has several feature flexibility, integration, security, scalability, high availability and compatibility [25]. It has the distributing computing, web based interface having an important enterprise capabilities and scalable storage.

Now days Pivot HD [26] is an important component of Apache Hadoop for the analysis of big data. It has the various useful features that enable the user to adopt the business needs for the enterprise IT. It uses the Hadoop native tools for the processing of big data. Pivot HD uses the native tool of Hadoop that makes more convenient choice for the user to use to big data processing. It supports the flexible models such as various packages for the commodity hardware. Due to this flexibility it provides the various enterprise concerns over, requirements, security, performance, regulatory, control, cost etc. Pivotal HD offers the backup and recovery, data protection, geo redundancy, robust availability in case of failure.

High performance computing cluster (HPCC) [27] is an open source platform for the processing of huge amount of data set. It is huge parallel computing platform for analytics. It enables the user to process the big data efficiently and effectively. It is more reliable than other exiting platforms. This platform provides the scalability, high performance and agility. It has real time data delivery engine for data warehouse and query processing. It has very powerful programming language for processing the large data set. For the big data query its has platform which is based on standard web services. HPCC provides various feature that very necessary to overcome the challenges of big data. It can run the commodity hardware. It also has distributed build in file system, fault tolerant, IDE for development, different modules like machine learning, and operation tools. It has three main components [28], one HPCC data refinery is an ETL engine that enables the user to integrate the data and manipulation the data. Second component is data delivery engine; it provides the low latency, fast query response and high throughput. Third component is an enterprise control language (ECL), which distributes

the workload among the nodes, library of machine learning development, and synchronization of algorithms. In HPCC, it has Thor and Roxie but on other hand Hadoop have MapReduce and HDFS for processing.

EC2 [29] is an interface for web service that enables the user configure and obtained with minimal friction. It allows the user to run the computing environment of amazon and user has complete control on the computing resource. It minimize the required time for getting the instance of server, it is compatible in terms of computing requirement. It allow the user to pay the only capacity that user occupied. It allows the user to fault tolerant application and remove them self in case of failure. It provides the several features flexibility, security, scalability and reliability. On the other hand is very costly, common user cannot afford it.

Neo4j [30] is graph-based database. It is an open source database for processing the large amount of data. It has its own query language call the cypher. Graph has two important graph technologies one is graph-based storage and second is graph based processing engine. Graph based processing engine is provide the native graph processing. Since nodes are physically connected to the graph, so it provides an efficient means of graph processing. On the other hand it provides the native processing and storage for the graph based database. It provides the better scalability as compared to other database. It has following characteristics vertical scaling, higher performance and concurrency. It also supports various platforms like Java, Python, Ruby, .Net and PHP, which make it more useful for the developers. More over it also has limitations, i.e. there is no support for sharding. More over, there are also limitation on nodes properties and relationships.

Pentaho [31] is a data integration and business analytics tool. It is an open source platform. It is very popular for blending, analyzing and visualizing the big data. It has wide capabilities of the data mining and analysis. It is an alternate choice for the developers and also the well-updated user interface. It provides the native support for the Hadoop and any type of data source, having the NoSQL and analytics. Pentaho and its user do not need the writing the code for integration. It also provides the several other features to the user business intelligence (BI), data integration, dash boarding, ETL capabilities, OLAP services and data mining. Limitation of Pentaho include, data visualization and analytics tool required more improvement. It is inconsistent and inconvenient in working manners, the layer of metadata it have is very hard to use and understand. More is there little help from documentations.

Talend [32] is an open source platform for processing the big data. Architecture of Talend has all needs that user required regarding to the integration and governance of data. It has various features like scalability; ease of use and reliability, these features makes it well suitable for the developers. The tools of Talend consist of products for development, data management, deployment, and integration of products. There are many advantages using the Talend, with ETL tool it can also equalize the burden among the server processing on the cluster. It also uses the Jasper soft BI software. The interface of ETL provides supports to import the metadata. Its configuration and linking of various components enables the developers to generate more productivity than any existing programming language. Moreover it also has the limitations such as there is need of JDBC for access of source. There is no product for metadata management and

quality of data. There is bottleneck in jobs due automation of data partitioning and repartitioning, and allocation of resources among the grid.

Tableau [33] platform provides analysis, reporting, and visualization of data using this framework. Main goal of Tableau framework is to provide the fast and interactive reports. It is very admired and important business intelligence tool across the non-technical and technical user for the various intentions. One it is self-service BI in which user business intelligence and business analysis can create their reports with having the dashboard. It also enables the user quick development that it has drag and drop features allow any type of user to build their own dashboard. Last it has data visualization, which permit any type of user to effective analytics and visualize the trends. It supports various platforms such as python, XML, API and JavaScript. Its feature includes excellent user interface, integration, mobile support, customer services, user forum, low cast and easy to upgrade. On other hand, it has limitations like preparation of initial data, statistical features are avoided in this framework and financial reporting.

IBM [34] is vendor of big data. It provides various platforms for the integration and building the big data warehouse. It has various products for the user such Info Sphere warehouse it have its on build in data mining tools and capability of cubing. It also has new component which pure data system, it is package, which have advance analysis tool for the big data. It also has the business intelligence (BI) having the capabilities of big data such as computing of stream, solutions of data warehouse and enterprise class Hadoop. Info Sphere stream strictly integrated with statistical packages social science including the capabilities of dynamically changes the real time data.

SAS [35] provides the various techniques for the analysis of big data providing the infrastructure of high performance analytics and statistical software. It provides the multiple processing of distribution. It provides following features grid commuting, database analytics and in memory computation. It can do deployment in the cloud and on site. It also provides the solution to the complex problems. It is an advance analytical tool of leading industry.

Oracle [36] provides the wide range of tool for big data. It has its own platform for the big data that is Oracle Exadata. It uses the R platform for the advance analytics, in memory computation having the Exalytics of oracle and data warehouse of oracle. It has its Big Data Appliance that combines an Intel server with a number of Oracle software products. They include Oracle NoSQL Database, Apache Hadoop, Oracle Data Integrator with Application Adapter for Hadoop, Oracle Loader for Hadoop, Oracle R Enterprise tool, which uses the R programming language and software environment for statistical computing and publication-quality graphics, Oracle Linux and Oracle Java Hotspot Virtual Machine.

Teradata's Aster [37] platform has a mix of analytics, including the Discovery Platform, a database, a discovery portfolio with pre-built functions for a broad set of Big Data applications, the Aster SQL-GR next-generation graph analytics engine, SNAP Framework for integration and a unified SQL interface across multiple analytic engines and data sources and its own Map Reduce.

Chapter 3

Literature Review

We have find out various challenges and issue for the big data platforms. In these categories there are various work have been proposed by the authors to solve these challenges and issues. These are all given below:

3.1 Literature review: challenges and issues

K.Radha et al. [38] have described that various algorithms have been proposed to solve the issue like data locality and straggler issues. Straggler problem occurs due to the unavoidable runtime clash for the transmission capacity of system, processor and memory. To adjust the single and cluster jobs to adjust the usage, they proposed the speculative execution performance balance. Data locality can be achieved by slot prescheduling. More over they argued that they delay scheduling is feasible for improving the data locality of MapReduce [39][40]. They also proposed the dynamic map reduce for the concentration of Hadoop fair scheduler as compared to FIFO scheduler. It have three main components one is prescheduling of slots, speculative execution balance and slot allocation of Hadoop dynamically. Main objectives of proposed approach, is to improve the job execution time on the cluster and performance of MapReduce.

Zhenhua et al. [41] have described that in the architecture of traditional high performance computing (HPC) did the computation of node and storage separately. For the multiple users it fulfills the needs by providing the high-speed link interconnection. But the issue is capacity of these links are very less than bandwidth of computing node.

They argued that parallel system based on the commodity hardware and every node takes the computation and storage role, which makes the compute the data. Data locality is a main characteristic of parallel system as compared to the traditional HPC systems. Data locality minimizes the network traffic and bottleneck in the intensive applications. In this paper they explored the data locality in detail. Firstly they provided the mathematical model for the MapReduce scheduling find out the impact of data locality theoretically. Secondly they investigated the scheduling of Hadoop and proposed the algorithm. Main objective of this algorithm was to schedule the multiple jobs into concurrently instead of one by one for the optimal data locality. They also did experiments for the proposed schemes in terms of execution time of a job cross and single cluster environment.

Eltabakh et al. [42] have described that Hadoop is very famous platform for the analysis of large amount of data. They find out the major bottleneck of the Hadoop, which is collocate the related data on same set of nodes. To address this issue they introduces the cohadoop which an extension of the Hadoop framework. It also enables the applications to control where data is stored. In cohadoop it have the flexibility of Hadoop as compared to the state of art approaches. There is no need to converts the data into the certain format. Instead there application will helps by giving the hits to cohadoop that some set of files are related and processed jointly. They developed this technique in such a way so that it can achieve the fault tolerance of Hadoop. They argued that colocation can be used for the improvement of efficiency various operations having sessionization, aggregation, columnar storage indexing and joins in context of log processing. For the processing of the log they find out two use cases one is

sessionization and second is joins, so that query processing can be speed up significantly of copartitioning the related files by collocating. In the results they showed that if do the copartitioning and collocating together higher performance can be achieve on the joins and sessionization.

In [43] Wang et al. they have introduced G-Hadoop, a framework of map reduce, whose main objective is provide the large-scale distribution among the various clusters. The need for data intensive analysis among the many distributed clusters for scientific data have rise up significantly now days such as high-energy physics (HEP). MapReduce is very popular programming model it has limitations such as application created for single cluster cannot be work for large-scale data processing for distributed environment. It also has limitation for distributed data processing such as efficiently and reliability. To overcome these issues they designed a new architecture of G-Hadoop which master and slave commutation model. They also stated that existing cluster could be added G-Hadoop with little modification is required for new clusters.

In [44] Hsu et al. they have described that big data is so large that cannot be process by the traditional tool system. There are many challenges in the processing, storage and analysis of big data. However framework like MapReduce that can handle the big data. It handles the data by processing the data by distributing among the various computers. Advantage of MapReduce is it allows the users to perform the task without having the internal knowledge and detail of the system. But the problem occurs when it works on the heterogeneous environment. To solve this they have introduce new approach improving the data locality and load balancing using the virtual machine mapping. Before the mapping it dynamically divides the data and utilize the resources by using the

virtual machine mapping in the reduce phase. Main objective of this technique was to enhance the performance of MapReduce in the heterogeneous environment and cloud hybrid. Advance using this strategy is minimizes the overhead in distributed system and optimizes the shuffling performance. At run time it balance the workload. Experiment shows that proposed technique enhance the performance of MapReduce in terms of data locality and total completion time.

In [45] Yu et al. have described that improving the performance of map reduce Hadoop cluster by avoiding the off switch communication. Presently, main focus of Hadoop is to minimize off switch task of map. Data access across the whole cluster shuffle by reduce task due to the scattered file blocks. They argued that if we group the blocks in racks could significantly minimize the exchange of off switch data. By this it can decrease the execution time of the execution. Therefore they have proposed the grouping block approach for the to minimize the off switch data access and execution time. They also find out that there loss of parallelism due to sticky effects and conflict during the grouping blocks for the enhancing the data access. To minimize these effects they also introduced the task scheduling and data placement to reduce the impact of grouping blocks approach on parallelism. They also conducted the experiments to validate their grouping blocks approach and methodology. It shows that execution time of the job reduced up to fifty six percent and efficiently reduces loss of parallelism.

Lin et al. [5] have described that MapReduce the extensively join operation during the processing of the large amount of data. However, they argue that traditional approaches are not effective due to the partitioning skew and take long to response for the execution. More over, if an appropriate algorithm is not applied for the partitioning of the join it

will lead to degradation of performance of the system. To solve this problem they proposed the skew avoiding and locality aware algorithm. It used the volume aware locality in spite of hash partitioning. It very useful due to the following reasons, it distribute among the reduces equally only when there is skew data, secondly it also transfer the data based on the locality on the network and thirdly, there is no need for change in MapReduce framework. They have also checked the performance of the proposed technique in term of effectiveness of partitioning, degree of skew and response time of the join operation.

Rhine and Bhuvan [6] have described that there two major component of the Hadoop, as these are HDFS and MapReduce. They argued that by improving the performance of the MapReduce overall performance of the system could be increased. They have proposed the new scheme based on the locality aware for MapReduce. It has scheduling and splitting algorithm strategies. Splitting and scheduling based on the locality. In locality aware scheduling approach it checks that slot is available for the local data. On the other hand, in input splitting approach it checks that the cluster data blocks from the same node into single partition so that it can process the single partition. It also considers the nonlocal data on the second preferences. They executed these Splitting and scheduling methods on the separately, shows the better performance without having any changes.

Chen et al. [7] have describe that data intensive applications and cloud environment bring new challenges for the data such as computation, assessment and placement. More over, they also argue that network traffic also plays an important role in the performance data intensive applications, some it reduce the performance of the system. So there are

some problems in the Hadoop of data locality. However, these exiting challenges can be reduce increasing the data locality in the distributed applications. To solve the data locality issues in the Hadoop they have proposed the locality aware scheduling (LaSA). In this approach they have done three improvements for the data locality, first they proposed that mathematical model for weight of the interface of the data, secondly they have use the weight of interface the locality of the data based on the resource assignment in scheduling of the Hadoop. Main objective of the proposed scheme was to avoid required data missing to arrange assignment of the resources.

Tan et al. [8] have described that Large-scale Hadoop clusters improving the data locality of the MapReduce is very important by applying the principle of the Hadoop, moving the computation instead of data. Scheduling the task can be minimizing the overhead of network traffic that is an essential for the efficiency and satiability of the system. They argue that most of the available schedulers do not consider the data locality during the Reduce Tasks scheduling. It also affects the performance of the system. To improve the data locality they have applied the threshold based optimal placement for the reduce task. It reduces the fetching cost of the data. They have also proposed the horizon base control policy optimal solutions under constrained states. They have also done the experiments on the proposed approach and find out the performance is enhanced.

Gu et al. [46] have described that MapReduce is famous programming model for processing of large data set. It has various characteristics such as fault tolerance, programming interface and scalability. However, in various situations performance of MapReduce suffers. To overcome these problems they have optimize the job and

execution of tasks and introduced the optimized Hadoop named as SHadoop to improve of the jobs of MapReduce. It has two main phases of optimization of the jobs, one job initialization and second job termination. It provides the instant messages for the communication for efficient and important event notifications instead of heartbeat mechanism. It also minimizes the startup and cleanup of the jobs from optimization. It is very efficient for the job with small running time. They also showed the broad benchmarks to estimate the performance for the improvements of jobs for the MapReduce and 25% improvement as compared to the default approach.

Ye Chen et al. [47] tried to solve the skew data problem, which cause the imbalance in data. Normally map and reduce stages are used to parallelize the data, so that it can use the large data processing efficiently. But it is not perfect due above-mentioned problem. This issue is not only appearing on map phase but also in reduce phase. They argued that many other people [48] did intensive study on this issue but can't able to solve this issue. Some of them tried to solve it by join operation, due the speciation of particular application it does not fulfill the requirements[49][50].

To overcome this issue they have introduced a new parting algorithm cluster locality algorithm. It consists of three parts. First, part is preprocessed, in this part they select the sample from the main collection of data, do that they can understand the distribution of data. Second part is clustering, in this part data is formed from the various cluster data. Data that have same key kept with the same cluster, so that it can reduce the size of data. Third and final part is partition a locality, it assign the data to the cluster suitable according to data locality. They also did analysis according to the execution time, data locality and skew degree. They showed that proposed technique is better than default

technique. But according our opinion this technique is very costly as they are using the extra MapReduce and it will also create a new overhead.

In [51] Kamal et al. they have described that Hadoop applications run as containers, so problem like concurrency affect the job completion time and resource usage of the system. When there are too many concurrent containers, resources bottleneck occur and there are too few, system resources are underutilized. As there are main issues to increase in execution time. To overcome these issues they have introduced new approach concurrent container slot (CCS) and tried to enhance the performance of Hadoop applications. This approach is dynamic and it uses the controllers that takes instantaneous score or score to CSS ratio as input and generate the new CSS as output. This score is combination of user CPU, blocked processes and context switches values. They also assessed the water level, PD, and PD+pruning controller. On the other hand dynamic controller have the better performance. In order to make sure that there work is applicable able to the all of the map reduce applications they selected the six applications in this work have diverse usage of IO and CPU usage profile. Finally they find out that using the dynamic tuning offer better performance instead of using the exiting default best settings. It also increases the performance and response time by avoiding the resource bottleneck.

Indranil et al. [52] have described that there was a problem in tradition shard memory architecture and single machine was not suitable for the distributed cloud environment. To overcome this problem they proposed parallel-boosting algorithms one is ADABOOST.PL and second is LOGITBOOST.PL. These two algorithms provide the participation simultaneously in the several computing nodes to build the increased

classifier. The presented algorithms in terms of generalization performance are very competitive to the conforming serial versions. They also define framework for the MapReduce to increase its performance. They applied these proposed algorithms to the architecture of MapReduce. They argued that parallelism is very essential for space due to the limiting aspect of memory. There is large amount of data it does not fit into main memory so they need to move to the disk. So they showed that presented algorithm needs less communication among nodes for parallelism in both case in terms of memory and space. They did analysis by using the amazon EC2 for the performance metrics such as scale up, accuracy and speedup.

In [53] Yin et al. have described that Hadoop distributed file system is an open source data processing systems. It is popular system for the processing of large dataset including parallel program processing based on the MPI, graph processing, Java/Scala based framework to enable the user to do the iterative and interactive in memory data analysis. They investigated the problems in parallel system as they find out the due to lack of intension in data distribution and access in HDFS, request for the parallel data sometime serves as the imbalance and remote fashion. Due to this problem it become I/O bottleneck for the storage nodes. To solve this problem, a new technique was introduced which I/O middleware and match based algorithm for optimization data access in distributed file systems. To attain this they get the data from the storage based on the distribution. It also preforms the mapping relation among the process and chunk file where these files are associated with data processing task and operators. They also showed the relation in form of Bitrate matching graph. With this technique parallel data access is used in such manner so that it will be useful for the balanced and locality

access so that it can attain the higher I/O performance. At last they have presented the content aware method for fast access of data without scanning the regardless data from the large dataset.

Sui et al. [54] have described surveyed various technique and frameworks for the load balancing for data intensive computation. In the past mostly processed of was performed on the distributed collection of machines to ensure the task will complete in appropriate time. Load balancing is the most important challenge in the data intensive applications. Load balancing is very essential only implacability for the minimizing of time of execution but also cost, overhead of network, cost and energy of usage. In this paper they also explored the various data intensive frameworks such as MapReduce framework of Google, Hadoop framework, Dryad framework of Microsoft, processing the data streams these all platforms for the processing of big data at various levels. They also discovered the application of the cloud scale such as azure of micro soft and app engine of Google. These applications manage the data of the user without worry of the user about the server. They also find out various techniques of the load balancing, mainly there are two categories, one is static load balancing and second is dynamic load balancing. Static load balancing techniques usually based on the greedy algorithms, search algorithm and machine learning algorithms. Usually overheads occur at the start of the static load balancing. In dynamic load balancing there are also various techniques such stream based, cloud computing, discrete event simulation. If we compare the static and dynamic load balancing, dynamic load balancing have more advantage as compared to the static load balancing. In dynamic load balancing it have less complexity, and takes less execution time.

Ajitha et al. [55] have described that web based application have large amount of data and handing the million of users. So traditional approaches to process the data are inefficient due to computation cost. Most of the frameworks have been introduced to work parallel on the homogenous data, but performance of these frameworks suffers when it comes under heterogeneity of data. Therefore, allocation and de allocation of the nodes at run time at the time run increase the cost and execution time. To solve this issue they have introduces the dynamically allocation if the resources based on the direct acyclic graph and preforms its processing parallel. Proposed architecture consists of Job manager which responsible for managing the jobs and divided the jobs into tasks also coordinates its executions. Second component is task manger, which is run by the virtual machines, when it receives the new task it execute it and inform about its completions of task. Third component is load balancer, it uses the join idle queue algorithm, and whenever task manger becomes ide it joins the queue. Main objective of proposed approach was to enhance the throughput, load balancing and decrease the latency in the heterogeneous environment.

Nishanth et al. [56] have described that Hadoop is very commonly used framer for the processing of large data. But there is problem in this framework that related data is not colocated by default. Due to this issue its problem suffers a lot, but this issue can be handle by grouping and partitioning the log processing operations i.e. grouping joins, sessionization and indexing. To support this, partitioning should be place the similar set of node in the cluster. Therefore, they proposed new approach that attains the load balancing and colocation of HDFS data blocks. It was developed on the Cohadoop. It divides the input files on the bases of grouping that is an attribute of the similar data.

Same set of data is colocated so that similar data can be partitioned. So the problem load balancing can be occur due to the skew data. In cohadoop this was the issue that was not covered. So they enhanced the Cohadoop and introduced the new algorithm to solve the problem of load imbalancing. Proposed scheme ensure the load balancing, fault tolerance and minimize the execution time.

Xu et al. [57] have described that MapReduce is very popular for the processing of distributed large amount of data. But there is problem of load imbalance due to the skew of data. Data skew occur in various applications like mining of data, operations of joins, and graph based. There a lot of consequences of data skew such as it decrease the performance and increase the execution time of the system. There are some approaches that handle the skew of the data but it leads to additional overhead and computation cost. To solve this problem they have proposed a new technique that have two phases one is sampling and second phase is job execution of MapReduce. In sampling phase, it computes the key frequency approximated distribution to make the good partitioning scheme. In MapReduce job execution phase, it applies the partitioning scheme on the each phase for grouping the keys quickly. To achieve the load balancing they further proposed the cluster split and cluster combination. First it assigns the cluster having largest pair of key to reducer having the small number of key. They also did experiments on the above proposed schemes in terms of execution time and load balancing. It shows that it achieves the desire goal but still many questions arise due expensive computations and overhead of proposed schemed.

Chen et al. [58] have described that MapReduce is very famous for the parallel processing of data. It can process large amount of data. But in this framework problem

arise of data skew due the data assigned to the task. This is the effect of some task, takes long time to finish and degrade the performance. So overcome this issue they have proposed the lightweight data skew mitigation to solve the load imbalance problem. As compared to the existing in this work they do not need sampling at run time. It is an innovative method in which to balance the load among the reduces task that encourage to split the large keys when there is semantics of applications. It also modifies the load of reduce when there is heterogeneous environment. To attain, goal of load balancing they have they tried achieving the transparency, parallelism, accuracy, large cluster splitting, heterogeneity issue and enhancement of the performance through proposed approach. They have also evaluated their technique in terms of execution time, and data skew degree. It shows that proposed technique enhance the performance of the system.

Zhou and Wen [59] have proposed the new scheme to improve the load balancing in Hadoop. They argue that previous approaches have not considered the data node but in their approach they also are considering the both DataNode and storage factor for the load balancing. So ignoring the data storage can easily lead to imbalance the load for log time of the system. Proposed scheme is based on the user history for specific access of data and mixture of analytical process hierarchy (APH) for the load balancing. It checks the history of file access, capacity of the machine and CPU, utilization of memory. It also automatically balances the load of system when user request second time.

Gao et al. [60] have described that MapReduce is an important framework for the processing of distributed parallel processing of large amount of data. Hadoop uses the MapReduce for the processing of data. Often some times MapReduce is applied on the small jobs, so there response time is critical. MapReduce have good performance in

homogenous environment but in the heterogeneous environment its performance is suffered and takes long time to execute the job. There are also many devices that great computation capacity, memories, power and architecture. But when various nodes work together on the same amount of data problem of load balancing arise. To solve this problem they have proposed the new scheme load balancing based on the performance of the node (LBNP). It is based the history that evaluated the performance of the node and assign the task according the performance of nodes. They tried to enhance the efficiency of the MapReduce by shortening the tasks of the reduce phase.

Fadika et al. [61] have described that MapReduce is very popular for the processing of big data. For the processing of large cluster data, MapReduce efficiently perform processing. However its performance suffers in heterogeneous environment and also causes the load imbalance. There also some limitations in the existing technique such as static load balancing and less effectiveness in large size of cluster. To solve this problem they have proposed a new technique MapReduce with adaptive load balancing for heterogeneous and Load imbalanced cluster for the MapReduce. In this technique it balance the load not only homogenous environment but also in heterogeneous environment. It also equally split the in out for its nodes as in existing MapReduce. It also a large number of tasks created from the given input splits, many times it is bigger than the sum of its nodes. This technique also admits take part the nodes to demand tasks at there own pace. They also compared the their own technique with exiting technique, proposed technique is perform better.

Shi et al.[62] have described that shuffling phase in the MapReduce is very critical phase, it a lot time to complete the job. It is also an important for the scheduling the job.

In shuffling phase, job scheduler assigns the reduce tasks to the set of reduce nodes. So it need multiple data items that relocate the key to set the reduce nodes. In turn also cause the large data relocation, it also cause the load imbalance of the workload various nodes. To overcome this relocation of the shuffling phase in MapReduce, they aim to reduce the network traffic, balance the workload, and remove the network hotspot for the enhancing the overall performance. To achieve these goals as mention above they have introduced a new technique call the smart shuffling scheduler. They showed that proposed scheduler performs well as compered to CoGRS scheduler and random scheduler.

Xie et al. [63] have described that present Hadoop implementation assumes that computing nodes in a cluster are homogenous in nature. Data locality has not been taken into account for launching speculative map tasks, because it is assumed that most maps are data-local. Unfortunately, both the homogeneity and data locality assumptions are not satisfied in virtualized data centers. They also described that ignoring the data locality issue in heterogeneous environments can noticeably reduce the MapReduce performance. To overcome these issues they have introduced their own technique that how to place the data across the node in a way that each node has a balanced data processing load. They also identified performance problem in HDFS (Hadoop Distributed File System) on heterogeneous clusters. Motivated by the performance degradation caused by heterogeneity, they have designed and implemented a data placement mechanism in HDFS. The proposed scheme distributes fragments of an input file to heterogeneous nodes based on the computing capabilities. From the results it shows that it improves performance of Hadoop heterogeneous clusters.

In [64] Rajashekhar et al. they described that MapReduce introduced to solve the data intensive application problem on distributed clusters [6]. On the other hand MapReduce consist of various nodes that are different according to the computing capacity of various nodes in cluster. They made an assumption that it is necessary for data placement algorithm to divide the input data based on the computing capability of node in clusters. So they proposed a mathematical model, based on the specification of hardware it, it estimates the computation ratio. The model calculates the computational ratio on the basis of resources available at execution time. Secondly they provided a method, which is based on the history that calculates the computation ratio of a machine when a job is completed. And also the data distribution tool which is based on computation ratio. At the end they evaluated the performance of map reduce based on their technique. We have seen that proposed technique have a lot of computation overhead. This technique is less efficient so enhancement is needed.

In [65] Lee et al. they have described that most frequently used computer application Internet, and network services with rapid development. Social media, search engine, and webmail are the presently data intensive applications. Due the increase in using the web services by the various people, there is problem in processing the large amount of data efficiently. So presently there is a limitation in Hadoop, it assumes each node in cluster have same capability of processing the data and task are local. But on the other hand, it decreases the performance and increases the overhead MapReduce. To over come these issues they proposed the dynamic data placement (DDP) policy for mapping task of data locality to allocate data blocks. DDP consist of two different phase, in first phase input data is written into the HDFS, in second phase given job is processed. By default

Hadoop policy of data placement can be applied to the homogenous environment but in heterogeneous environment, it create problems like load imbalance and unnecessary overhead. They introduce the DDP algorithm solve the problem like load unbalancing, data locality, and reduce the overhead. For the analysis they did experiments on the two types of job wordcount and grep to evaluate the performance of proposed scheme in Hadoop heterogeneous cluster. They find out DDP can improve the wordcount up to 24.7% with an average of 14.5%. But in grep the DDP can improve up to 32% with average enhancement up to 23.5%.

In [66] Sujitha et al. they described that growth of data is increasing every day exponentially. Many data center and scalable computer are formed to fulfill the demand of many applications. System like may consist of various commodity hardware connect the local are network. It is very huge amount scale as compared to the traditional cluster. Hadoop is the well know framework to deal with large data of distributed applications among the many cluster of commodity servers. Main advantage of the Hadoop is it handles failures. Hadoop has by default homogenous scheduling approach for the processing of various jobs. But in the cluster performance of Hadoop suffer due heterogeneous environment. To overcome this issue they proposed the new technique for heterogeneity and data locality for Hadoop. In this approach they define a new architecture, which consist of various Levels. First level consist of operating system, which controls the hardware, in second level there is java virtual machine. Third level consists of three components i) processing of data ii) data storage iii) coordination, basic purpose of these components was to make processing and storage of fault tolerant, fast, and scalable. Forth and fifth level consists of network and application layer respectively.

They also evaluated their technique terms of response and execution time, fairness and velocity, mean time for completion.

In [67] Fan et al. have described that is very costly to fetch data from the remote server in the large processing of data.so it necessary that data should be co-located in terms of computation. They have proposed a new technique dependency aware data locality for the map reduce (DALM). It is a replication-based approach for the general input real world data that is dependent and skewed. DALM adjust the dependency of the data in locality framework based on such vital factor storage budget and popularity. DALM has two main parts one is computing the factor of each file and second is replica placement. In replication factor, it requires the information about the popularity file.

They applied the monitor daemon for the NameNode, which get information about the dependency of data and access of files. After this they calculate the factor of replication using the SMO solver. To set the factor of replication for the every file they did write the program. Using the improved k-medoids algorithm determine where to keep the each replica and it also the NameNode replica placement output. For placement of replica approach they improve the system of file placement in Hadoop. As the output of this the module of replica placement, it will transfer the replica to conforming server on the bases of improved k-medoids algorithm. Advantage of this technique is compatible with traditional infrastructure and can be extended to the other distributed paradigm of computation on bases of strong dependencies.

In [68] Khan et al. have described that MapReduce is very widely used for the various application such as data intensive, distributed and parallel processing. Hadoop is very famous open source tool for processing of large amount of data. In Hadoop the input

data of file is divided into various data blocks. These blocks are distributed into many nodes in cluster. One of important concept of the Hadoop is move the computation instead of moving the data when we are dealing with large amount of data. It helps to improve the performance and network congestion of the system where the currently running application have large amount of data. In this research paper they have described there are three level of locality according to the map tasks.

First level locality is node locality, it is most efficient locality where processing map tasks. Second Level locality is rack level locality if a task is failed to achieve the first level locality then scheduler launch the task where computation node and data node on the same rack. If still it cannot able get the second level locality it again lunch the task on the node having the different racks, it also called the rack off level locality. More over in this research paper they described the various scheduling algorithm how these are improving the locality of data. DARE scheduling, Delay scheduling, Matchmaking scheduling algorithm, prefetching, pre shuffling algorithms and next-k-node algorithms. Basic purpose of this algorithm was to improve the data locality. They also evaluated these algorithms and find out that these algorithms solve the data locality problem but raise many other issues like cost of replication, overhead of computation.

In [69] Li et al. have presented that scheduling is very important problem in big data cloud computation. To deal with the large amount data locality is an important feature in Hadoop. Data locality is an important problem in Hadoop based applications. The main cause is the bandwidth in the Hadoop cluster is network is much less than the total bandwidth of one node of hard disk. In research paper they analyzed the scheduling algorithm and improve the LATE algorithm. Basic principle o LATE is it analyzes the

how much time is need for the specific task to complete it. Main fault of LATE algorithm are it has not ability deal the data locality in cloud computation. It execute the reduce task with out considering the whether data is local or not. On the other hand it execute the map task locally. To improve this algorithm they have proposed following steps. One check the speed of the node it is slow than the threshold, it better to ignore it otherwise continue the processing. According to the request of task node in the specific rack, make sure that task is slow, if it is then find out how much time is remaining to complete that task. Third step is to make sure that task is being executed in other rack is slow; it is slow then keeps it to another queue. Fourth step is find out the task has the data locality it don't have then make sure the time reaming of task whose time is longer than threshold in queue. Their experiments shows that the proposed enhancement reduces the response time and improve the throughput of the system.

Ubarhande et al. [70] have described that main purpose of the Hadoop and MapReduce was develop to process the data intensive application. Both frameworks handle the intensive applications by divide the task in to smaller task, then it operate on the smaller sub tasks. Hadoop basically support the homogenous task now days Hadoop facing a big challenge in the heterogeneous environment especially in cloud environment. They argued that existing techniques have several issues i.e. consumes the unnecessary bandwidth, not useful when history is not accessible and when certain cluster size increase the complexity of the system also affects. To solve this challenge they have proposed the new data distribution technique for the heterogeneous environment. In this approach, it places the data and distributes the data in distributed environment of Hadoop by exploiting data in cluster Hadoop heterogeneous

environment for slave node. Distribution of data includes the calculation of capability of processing of every node specified in Hadoop cluster. Algorithm begins with response of the log file of the every slave node, which is generated by the speed analyzer execution. Using the proposed technique they tried to improve the response time of the Hadoop system. More they also verify the proposed technique whether the response time is increased or not, they tested their technique using the two applications of MapReduce. They find out the performance of the Hadoop is improved in heterogeneous environment when data size is increased.

In [71] Huang et al. they have described that MapReduce framework is very famous platform for the parallel processing of large amount of data. Hadoop map reduce allow user to flexible customization and the convenient usage. Besides the feature of Hadoop MapReduce there are some issues in the processing of the large amount of data, one of the issue is straggler task, which delay the processing of the job due to the slow running tasks. To solve this issue they have proposed the two techniques one is estimate remaining time using linear relationship model (ERUL) and second is extensional maximum cost performance. ERUL strategy for heterogeneous environment on which remaining time of the task is determined by the system load by using the Hadoop ERUL. It is dynamic load aware approach. It also performs well and overcome the issues of the stat-of-art approach such as LATE and MCP. In exMCP they also tried allocated the slots value that are ignore in the traditional MCP. They also conducted the experiment and find out that their proposed approach estimate the time of remaining time of task effectively and also detect the stragglers accurately.

According to Prasad et al. [72] they did the analysis on the performance of scheduler of multi jobs in the Hadoop cluster. As there are many challenges exists in Hadoop ranges from the job scheduling, locality of data processing, Usage of resources efficiently, and fault tolerance. But in this research paper they have focus on the to job scheduling to achieve the efficiency. Hadoop uses by default first in first out (FIFO) scheduler. Therefore task assignment is the responsibility of scheduler, a job will be moved to the queue when submit this job. From the job queue the job is divided into tasks and distributed into different nodes. By using the proper assignment of tasks, time of the job completion can be reducing. The performance of Hadoop framework, it depends on the hardware configuration in each node, cluster size. But in this research paper they have focused on the problem that how to utilizes the hardware in case of different workload (homogeneous and heterogeneous) run on cluster on MapReduce. On the other hand Hadoop uses the two different types of scheduling polices one job level and second is task level. But the main goal of the research paper to compare the job execution time with the default Hadoop scheduler. For the experiment they used the homogeneous and heterogeneous workload. They have compared different scheduler, which default scheduler (FIFO), fair scheduler and capacity scheduler. They compared these schedulers in terms of job execution time and waiting time of job. They find out that default scheduler takes less time when job is small but takes the more time when job is increases; fair scheduler takes the less time for job execution and capacity scheduler more time for the job execution as compared to the fair scheduler.

In [73] Sethi et al. they described that scheduling a job in MapReduce is an important and critical issue that affects the performance of Hadoop framework. To

optimize the data locality, a delay scheduling has the little delay during the job scheduling. After a job is scheduled a delay scheduler can introduces scan a job more than one time. This is the main problem for the scheduler and overhead on scheduler. To overcome this issue they have introduced a new algorithm that runs the high priority jobs on the free nodes. After this node will execute the job locally or scheduler will some other node based on the availability of free node. The main objective of the proposed technique are to process the jobs locally to improve the throughput and response time, because the non locally data takes more network bandwidth and time. More over identify the problem with delay scheduling regarding the overhead on job tracker. The proposed algorithm reduces the overhead on the job tracker by distributing the load on task tracker.

Zaharia et al. [39] have presented a new technique for the cluster scheduling to achieve the locality and fairness. They argued that there is a need to share the cluster between users due to the use of data intensive cluster systems.i.e Dryad, Hadoop and Hive etc. But there are issues between fairness and data locality. They also mentioned that in cluster system if there is fairness then data locality will be not good. On the other hand if data locality is great then fairness will be poor. So to solve the fair sharing problem there are two things, one is to kill the task or wait for the job to be completed and then submit the new job. Second thing is to how to achieve the data locality. To solve this issue an algorithm was proposed called delay scheduling, which solves the fairness and data locality issue to wait the task until it completes. In cluster environment there are two main aspects one is task are short and second is multiple locations. For the

multiple locations to read the block a task can be run, Hadoop also supports the various task as per slot.

They also argued that delay scheduling not is performing well for the environment like Hadoop. But this technique is not suitable where the fraction of task is longer that job or few nodes per slots. However, they believe that it can improve by two things first by short tasks and second multiple cores that will support more clusters. They also discussed various ways for the generalization of delay scheduling i.e. scheduling preferences, load management mechanism, scheduling polices and distributed scheduling. They also implemented the delay scheduling in Hadoop fair scheduler (HFS). It improves the throughput for heavy workload, and improves the execution time for shorter jobs.

In [74] Sun et al. they describes that data locality is important factor for the performance of MapReduce on the cluster environment. To attain the locality of memory task of map will be decrees the completion time and enhance the throughput. For this purpose there is key issue to enhance the performance of the map reduce on the cluster. If map task are allocated to the node with having the input data, it can produce the delay for data access.

To solve this problem they have proposed the high performance scheduling optimizer (HPSO). Main objective of HPSO is to decrees the execution time of MapReduce and improve it performance. HPSO consist of three modules prefetching, scheduling optimizer and prediction. The main job of scheduling optimizer is predict the suitable nodes of task tracker in which upcoming task will be allocated. Once the decision of scheduling the map task schedule then HPSO loads the estimated data using

the prefetching module. HPSO improves the data locality for the job of MapReduce by prefetching. HPSO works by predicting the most suitable node on the bases of present pending task by determining when to prefect the task. For launching new task it preload the data require to the memory with out delay. It also minimizes the time of map task and decreases the time of MapReduce job. Authors also claim that this method enhance the efficiency of MapReduce.

In [75] Sadasivam et al. have presented the new technique to solve the problem of task assignment in the heterogeneous and homogenous environment by applying the hybrid particle swarm optimization genetic algorithm (HPSO-GA). The operation of GA such as mutation and crossover it utilizes the resource of data intensive application effectively and completes the task with the given time. The main of proposed algorithm was to dived the workload of the according to the processing capacity of node in the heterogonous environment.

Proposed algorithm is the combination of both particle swarm optimization and genetic algorithm. The operations of PSO and GA are applied specific particles. Allocations of the tasks are based on the fitness value to balance the load. There are various steps in this algorithm, initially it initialize the count of the generation, size of the population and maximum generation. It generates the initial population of particles randomly then calculates the fitness value of the population also updates the generation of population and velocity. After they perform the crossover operation. They also evaluate the fitness function of the particles. They continue this process until they get the global best particles. HPSO-GA they applied on the MapReduce framework to improve the utilization of the resource and load balance in grid environment. In Hadoop cluster it

find out which node exactly need to assign the task. They also tried improving the reliability, efficiency, and scalability of MapReduce.

Jun et al. [76] have described that in data intensive application, there is a great need of high performance computation and massive resources. Many scientific and engineering applications many people and researchers are interested on the subset of the whole dataset, so they can access it frequently than others. Another example in bioinformatics in which X and Y chromosomes are related to the gender's offspring often researchers analyzed in the research rather than whole chromosomes. There is another of climate weather forecasting, in which scientist are only interested in the time periods. These groups of data can process by the specific domain applications. Assumption, if two pieces of data have been processed at the same time it is highly possibility that data will be process in the future as group. Problem with approach of the random strategies of the Hadoop workload will not be evenly distributed, it may be balanced the overall data. Hadoop has the random placement method for the load balance and simplicity. In MapReduce and Hadoop there is a by default random placement which does not count the semantics of data grouping. Cloud cluster grouped into many small no of groups, which limits the degree of parallelism for the data and outcome of this performance bottleneck. They also argue that there are other method was proposed for data locality was not effective due to various issues overhead and cost.

Solution, ideal data placement is evenly distributing the grouping data. Their observation was that random data placement is affected by the three factors. One each replica should have the data block on each rack. Second, there should be maximum no of map task on every node and last one is data grouping access patterns. However, default

random data placement can achieve the optimal data placement by holding the each one copy of data on the same node there maximized the parallelism can be achieved in case of rack (NR) is extremely large. Secondly, NS extremely large then node (NS) equals to the number of affinitive data.

To overcome this problem they presented a new technique "a new data group aware data placement scheme" (DRAW). It takes the run time data group patterns data and equally distributes the data. It consists of three phases: Firstly, there is data group information learning from the log. They have used the history data access graph (HDAG) to approach the files; it is based on the history. Hadoop cluster rack, name node maintains the log history of the each operation and including the accessed file. However, it also have two issues such as log is huge which also have traversal latency problem, and second issue is that frequently accessed files are not similar. Therefore to solve these issue there is need of checkpoint to traverse the log of name ode. Secondly, there is clustering the data group matrix (DGM), which shows the relationship between data blocks. It can be generated on the bases of the HDAG. Thirdly, data group reorganization based on the optimal data placement algorithm (ODPA). ODPA is based on the sub matrix for the cluster data-grouping matrix. They also prove that Hadoop random placement of data is not efficient. They did experiment on the real world applications, it reduce the completion time of map phase and execution time of MapReduce.

Lee et al. [77] have described that graph is important for discovering the knowledge from the given data set. They described that there are two dimension of graph analysis. One is graph mining (GM); it has main focus on automatically knowledge discovery.

Second is online graph analytic processing (OLGAP), its main concentration is on pattern matching on sub graph. Both dimensions are complementary and concentrate on the solving the complex problems. Both dimension covers the analysis of holistic in-situ in a sole system. There is difficult in processing the multiple graphs and sending the results to the system. Mostly state of art graph analysis based on only on dimension. To overcome this issue they took new technique enabling graph-mining abilities in RDF triple store. For achieving the desire goal they implemented the six different graph mining algorithms using SPARQL. It enables wide variety of RDF datasets which applicable to graph analysis for holistic. For evolution of proposed technique they used the various computing environment, nine different data sets. Evolution shows the scale able performance for in real world graph analysis.

Yinglong et al. [78] have described that big data analytics are very important to discover for such entities that can easily represent in the form graph. For the analytics of large-scale graph there is main problem for the delivery of efficient solutions that irregular data access. It is a main challenge for the processing of computation of graph-based patterns. Major challenges [79] that big data is facing for the graph processing are performance, large volume of data, and irregular data access. Big data tool are not suitable for the processing the graph due to the following reasons such as scalability, architecture awareness and systematic optimization. To overcome these issues they proposed a system called G. It enables the user to organize the data for the architecture of parallel computing. It also consists of visualization, graph storage and analytics. They also analyze the data locality in terms of graph processing, and its effects of the performance of cache memory on processor. They also measured the traversal

performance of the graph by both serial and parallel execution. For the experiment they did analysis on the parallel system and G system that is proposed system. They showed that G system perform much better then the traditional systems.

Fang et al. [80] have proposed the new technique called bipartite request dependency graph (BRDG) for the investigation of the relationship between objects of the web. They also leverage the MapReduce programming model for the construction of the BRDG from the large network data. Nodes of BRDG have two types of objects primary objects such as URL in web browser and secondary objects are those who trigger the primary objects. It is very difficult to derive the structural features of BRDG. To overcome this difficult they also proposed the co-clustering algorithm, form the BRDG it extracts the co-clustering coherent. In co-clustering of BRDG, each signifies the strongly connection to the bipartite sub graph. A parallel tri-nonnegative matrix factor (tNMF) algorithm they designed and implemented, basic purpose of this algorithm is provide efficient large-scale graphics decomposition. For the experiment they also divide the sub graph into four structural patterns such as click star embed star, single layer mesh and multiple layer mesh. They find out the multi layer of mesh are very famous structural pattern.

Xue et al. [81] have proposed the new technique based on the bipartite graph oriented locality scheduling (BLOS) for the MapReduce frameworks. Scheduling of the task is an important make span for the job of MapReduce. Improving the locality of the scheduling approach effectively, it is necessary to have the tasks and it related on the same node. They also argue that data locality refers to the task that have obtain from the same local machine. A higher degree of local localization can enhance the performance

of the system. Due to the localization it minimize the execution time of the job, reduces the communication time. Moreover, they described that there are three different types of the priority according to the proximity principles. First the priority will be given to the local task within the node. Second, priority will be given to the task within the rack and at last priority will be given to the off rack tasks. However, there is problem in this issue regarding the locality optimization, through this approach there will increase another problem called global optimization instead of local optimization. However they find out that the recent studies are not enough to tackle the data locality issues.

To overcome these issue they have proposed the new scheduling algorithm called the BOLAS for the MapReduce tasks. BOLAS can operate on any environment either it is homogenous or heterogeneous. Main aim of the BOLAS was to enhance the data locality without affecting the efficiency of execution. Bolas start solving the problem of scheduling of data locality by matching the bipartite graph in it try to allocate the data, which is close to the task. BOLAS have global data placement method through this they can achieve the better performance without affecting the nodes. It also avoids the network congestions with out generating the local nodes. Design of the algorithm consists of the two parts one is resource modeling and second is computing node performance estimation. In resource modeling, MapReduce have two major resources such as data blocks and name nodes. On the nodes blocks and tasks run, on the other hand task scheduling actually solutions between mapping of nodes and blocks. In performance estimation of computing node, BOLAS dynamically dispatches the blocks according the performance of the particular node. BOLAS also build bipartite graph in various cases by adding the virtual blocks and nodes if there needed. Then also apply the

KM algorithm for the optimal matching result with minimum weight. The benefit of the BOLAS is that it has high data locality. On the other hand it also increase the throughput of the cluster system and minimize the congestion of network. In the experiment they evaluated the ratio of data locality, execution time and network bandwidth. Results show that they improve the data locality of proposed technique by nearly 100% and minimize the execution time up to 67%.

Orozco et al.[82] have described that there are various challenges in the stencil application for the computations. One of the major issues the read modify writes operation the array data. The main limitation of these applications is off chip memory access, in terms of the latency and the data. To overcome theses issues they have proposed the locality optimization based on the data dependency graphs for the stencil applications. For this purpose first they have presented the formal descriptions that data that is not generated from the source code. They proved it my using the mathematical method for the commutation power and the bandwidth of the multi core architecture. For experiment they have used the various approaches such as Naïve, overlapped, split, and diamond tiling. They find out the diamond tiling technique is better than other approaches.

Hassanzadeh-Nazarabadi et al. [83] have presented the locality aware skip graph to overcome the issues of the name id assignment method of the skip graph's node. Skip graph locality is assigning the name id to the nodes such that more two nodes are close to each other, the more common prefix they would have in their name ids. They also argued that states of art method have not considered the skip graph's node locations. Main objective of proposed technique was to minimize the latency in the search query

from end-to-end. They proposed the dynamic and fully decentralized algorithm (DPAD). This method assigns the locality aware identifier will assign to the node instead of assigning the identity randomly. From the results they have shown the 82% improvement the data locality and 40% search query end-to-end latency.

Kandemir et al. [84] have provided the solution to the optimal memory layout detection. The Proposes approach has two basic elements such construction of the problem in a special graph structure and second is the use of linear programming (ILP) solver to define memory layout. It is the first approach that allows to dynamically changing layout cache locality. They have also shown that proposed framework for the locality is powerful. But there is dynamic layout modification for the large applications still remaining to explore.

Chernov et al.[85] have described the problem thread partitioning of the sequential programs. They proposed a new algorithm to overcome this issue. They also argued that performance could improve by considering the locality of the program. Many program have by nature locality characteristics. So they combined two optimizations for new algorithm. One, non-loop region can be parallelized. Second, perform the partitioning in such way that data locality can be improve of new thread. For the evolution of their approach they have generated the data dependency graph (DDG) and showed that their proposed approach is feasible.

Zhang et al. [86] have described that k nearest neighbor(*k*NN) have is popular approach for the various application specially machine learning and graph based applications. Besides its various characteristics but it has the computation complexity. To overcome this problem they have proposed the new algorithm that divides the whole

dataset into the small groups, and it make sure the each item of kNN belongs to the group. It leads to the accuracy and fast speed. To make the proposed algorithm more accurate, they have also divide the groups in such as way that similarity between the items remain in the same group. Secondly, they kept the group size small as possible. For the groups they also propose the locality sensitive hashing (LSH), which guarantees the rigorous performance even for the worst-case performance. They evaluated their approach that method can efficiently generate the graph with good accuracy.

Zhang et al.[87] have described that there is widely explored area of graph databases is similarity in graph processing. They also argue that graph have an important role in various applications like pattern recognition, information retrieval etc. They find out two categories of the graph search, one is sub graph search, second is similarity search. In this paper, they have explored the k-NN similarity search problem using the locality sensitivity hashing. They proposed the algorithm for the fast graph search that it transforms the complex graphs into vectorial representations based on the prototype in the database. After this it also accelerate the query efficiency in the Euclidean space by employing locality sensitive hashing. They evaluated their approach against the real datasets, which achieves the high performance in accuracy and efficiency.

Yuan et al. [88] have described that there are two main challenges in processing of the large scale graph processing: one is lack of the efficient storage, second is lack of the locality access. They also described that there are various popular approaches for processing of graph and storing the data such as storage centric, vertex centric and edge centric. They listed the common graph partitioning approach such as vertex cuts and edge cuts. But commonly used partitioning scheme for the processing of the graph is

vertex centric hashing. However, these approach having the issues of the very poor locality and communication overhead. To solve these issues they have proposed new path-centric approach for the fast iterative computation of the graph computation for the extreme large graphs. It is a path centric at both storage and processing tier. It is an efficient approach for storage and design structure for storage tier. It also allows the fast loading in edge and out edge for the gathering and scattering the parallel computation of the graph. But for the computation tier, its processing is used in such a way that optimized the locality of the large-scale graph. They iterate the processing or computation chunk level or partition level computation. Work stealing approach has been introduced in the work to balance the load among parallel threads. For the evaluation of their approach they have chosen the real dataset in which they demonstrated that proposed approach performance in terms of better balance and speedup.

Shao et al. [89] have described that partitioning schemes have great importance in parallel processing of the graph computation. In current graph system, they find out unaware partitioning issue for the computation of the graph and it also has the various drawbacks in the processing of parallel large-scale graphs processing. State-of-art approaches for the partitioning schemes are preferred in case of small edge cut ratio. The advantage of this was less communication among the working nodes. Sometimes, partitioning approaches of graph may have worst performance as compared to the simple partitioning. It causes the local messaging passing to super pass the cost of communication in many cases. They also argue that these systems are having parallel graph partitioning approaches. However, sometimes it can happen when users try to

integrate the graph partitioning methods into parallel graph computations system is not a trivial task. But there are some system, which they have good balanced, partitioned approach leads to the overall computation performance. They also analyzed the graph partitioning schemes on various systems using the web graph dataset. They find out the current parallel graph systems cannot be benefited from the high quality graph partitioning.

To overcome these issue they have proposed a new approach partitioning aware graph computation engine (PAGE). The benefit of this approach is that it controls the online graph partitioning statistics of under lying results of graph. Second, it monitors the parallel processing resources and enhances the computation resources. Third, it was also designed to support the various graph partitioning qualities. In this work they have analyzed the cost of graph partitioning and the cost of the parallel processing of graph. Page also consists of master worker paradigm is responsible for the aggregating the global statistics and coordinating the global synchronization. But the page worker takes the graph computation tasks aware partitioning informations. Page has two modules such as communication module and partitioning aware module. In communication module, it has concurrent dual messages processor. Partitioning aware module monitors the status of the system. Moreover, they also designed the Dynamic Concurrency Control Module (DCCM). DCCM has various heuristics rules for the message processor to optimize the concurrency. It also processes the concurrent local and remote messages. For evaluation of chosen schemes they showed that it preforms well under various partitioning approaches with various qualities.

Qin et al. [90] have described that mining of the dense sub graphs from the large graphs is fundamental mining task that can be apply on the various applications domains such as network science biology, graph database, web mining etc. They also argue that exiting schemes have concentrated on the just dense sub graph or identifying an optimal clique like dense graphs. In these schemes they have implemented greedy approaches to find out the top k dense graph. But on the other hand, their identified sub graph cannot be represent as dense region. So identified sub graph should be the highest density region in the graph. In this work they introduced a local dense subgraph (LDS) in the graph. It can used to find out the dense sub region from the sub graph and can be apply to the various applications. LDS also have some useful properties such as pairwise disjoint, locally dense and compact/cohesive. They also define how local dense sub graph in terms of parameter free with useful characteristics. They also have presented an elegant dense sub graph model. They also showed that LDS problem can be solve in polynomial time. They presented the three optimization approaches to enhance the algorithm. To evaluate the LDS, they have analyzed their approach using the four different qualities of measures such as density, relative density, edge density and diameter. They also did experiment with real web scale graph having 118.14 million nodes with 1.02 billion edges for the demonstration of the efficiency and effective ness of proposed algorithm.

Zamanian et al. [91] have described that horizontal partitioning approach has been extensively is used for the processing of large amount of structured data. But there is issue for using the horizontal partitioning approach that cost of the network must be minimize for given workload and schema of database. However, there is popular

approach to minimize the cost of network for parallel processing of database is co-partitioning the given table on their join key to avoid expensive remote join operations. On the other hand these approaches have issue of the replications and co-partitioning with sharing in the same join key. To solve these issues they have introduced a new approach predicate-based reference partitioning (PREF). It enables the to co-partitioning sets of tables based on the given join predicates. Main goal of PREF was to enhance the data locality and minimize the data redundancy. For this purpose, they also designed two new algorithms. In these two algorithms, first algorithm required the schema as input whereas second algorithm takes the workload as input. In experiments they showed that workload driven design algorithm is more efficient for the complex schema with large number of tables.

Chen et al. [92] have described that various existing techniques supports the vertex-oriented execution model. It also allows the user to create their own logics on their vertices. But there is an issue in terms of network traffic overhead for the vertex-oriented computation. However, graph partitioning is very useful for the minimizing the network traffic in the processing of graph. They argue that there is very little attention has been made for the partitioning of graph effectively integrated into the large graph processing in the cloud environment. Furthermore, the motivation they got that surfer is a master-slave system, consisting of one master server and many slave servers. The slave servers store graph partitions and perform graph computation. They studied the factors affecting the network performance of graph processing. They assume that the amount of network traffic sent along each cross-partition edge is the same. So they have proposed a new framework of graph partitioning to enhance the performance of the network for

graph partitioning itself, storage of partitioned graph and vertex oriented processing of graph. They have developed all these optimizations for the cloud network environments. They also have used the two models such as partition sketch and machine graph. Basic purpose of using these two graphs was to capture the features of graph partitioning process and network performance.

In experiments, they have developed a new prototype for the Pregel and enhanced it with their own framework of graph partitioning. They also showed the efficiency and effectiveness of the their proposed technique for the processing large graphs.

Zeng et al. [93] have described that processing of distributed graphs is very costly for the computation of large amount of the data in case moving the data among various computers. However, they argue that state-of-art approaches have high computation overhead and costly communication when apply on the distributed environment. To overcome these issues they have proposed new parallel multi level stepwise partitioning algorithm. They divided this algorithm into two phases; one is aggregate phase and second is partition phase. In the phase one, it uses the multilevel weighted label propagation for aggregation of the large graph into the small graph. But in second phase: It has the K-way balance-partitioning preforms on the weighted based on the stepwise mining RatioCut method. It reduces the RatioCut step by step. In each step, sets of vertices are extracted by reducing the part of RatioCut and these vertices are removed from the graph. In this they have obtained the k-way balance partitioning by this algorithm. In experiments they have made the comparison with various other existing partitioning approaches using the dataset of graph. It preforms well as compare

to the state of art approaches in terms of scalability, performance and can enhance the efficiency of graph mining on the real distributed computing system

Lee et al. [94] have described that scientific and engineering domains are growing various type of information and size these days. Due these needs there is another demand arise in cluster computing of cloud for efficient processing of the large heterogeneous graphs. They described that heterogeneous large graph have various characteristics in terms of big data processing. Firstly, graph data is highly correlated and topological structure of big graph can be viewed as a correlation of the vertices and edges. Graph that are heterogeneous they add the extra overhead as compared to homogenous graphs in terms of processing and storage. Secondly, queries are over the graphs is typically subgraph matching operations. But, they argue that it is ineffective of modeling the heterogeneous graphs as big table entity vertices. They described that HDFS is very popular for the processing of the partitioning big data among the large cluster of computing nodes in clouds. But, the HDFS is not optimized for the processing of the highly correlated data for large dataset such graphs. Therefore data generated by such random partition methods cause an extra overhead. Moreover, theses types of random partition methods also cause overhead for the graph patterns query. There is another problem arise with it that how to partition the graphs that it preforms efficiently. To solve this problem they have introduced vertex block partitioner. It is a distributed model for the data partitioner for the large-scale graphs in the cloud. It has three features. First, It has the vertex bock and it also extends the vertex block as building blocks for the semantic large-scale graphs. Second, vertex block partitioner uses vertex block grouping algorithm to place the high correlation in graph into same partition.

Third, the VB partitioner speedup the parallel processing of graph pattern queries by minimizing the inter-partition query processing. In results they showed that proposed approach have higher query latency and scalability over large-scale graphs.

LeBeane et al. [95] have described that large scale graph analytics are very essential issue in present data center. Large multi node processing is a critical computational problem due the popularity of the big data and cloud computing. There are many state-of-art frameworks available for the processing of large graphs such frameworks are: PowerGraph, Pregel, and Giraph. However, there is a problem in these frameworks that any change occur due to the cloud and big data, these frameworks cannot handle it. These state-of-art frameworks cannot be scalable. Data is coming from the various sources for the processing. So they argue that data center are trending towards the large number of heterogeneous processing nodes. Moreover, they also described that virtualized environment can also create the heterogeneity by partitioning the cluster of homogenous machines into the variety of configurations. However, frameworks of the graph analytics are still working under assumptions. This assumption lead to the imbalance of the load and cause the faster node finish the processing their chunk of data earlier than slower nodes. As in the heterogeneous in the data due to the visualization, load balance and heterogeneous nodes becomes more critical for the graph processing. To solve this issue they have used the popular framework for the heterogeneity aware data strategies. For the heterogeneous partitioning of the data, they divided the input data into shards that every node will receive the data according to the to metrics. They also define the data-partitioning ratio between to be the skew factors of the clusters. In this work they also described three different types of the skew factors

such as Thread based, Memory based and profiling based. Although there are many complex method are available for the calculation of the skew factors. However, these can provide the benefits to the performance for the heterogeneity aware partitioning algorithms. They also illustrated the simple estimate of intranode throughput that skew factor can drive the huge performance using the heterogeneity aware partitioning strategies. They also showed that their proposed strategies minimize the 64% execution time.

Chen et al. [96] have described that there is unique challenge in the graph partitioning and computation especially in the natural graph with skewed distribution. They argue that exiting graph processing system have the problem of "one size fit all" that impact on the performance, load imbalance and contention for the high degree vertices. Even though exiting graph-processing system also have the high communication cost and high memory consumption. To overcome these issue they have introduced a new graphic engine called PowerLyra. It is hybrid and adaptive design that has the dynamic partition and computation approaches for the various vertices. It also combines the edge cut and vertices cut with heuristic using the hybrid algorithm of graph partitioning. It also provides the data locality aware layout optimization to enhance the cache locality during the communication. They also did the experiment using the two different cluster using the graph analytics and for the machine learning and data mining algorithm. It performs much faster and consumes less memory.

Xu et al. [97] have described that graph partitioning now days is popular strategies to balance the work load due the scale of graph data and increasing availability. Due the cost of partitioning of the graph, recently researcher are more

focusing on the stream graph partitioning that is very fast, incremental update and easily parallelize. But, it has also the challenges such as the imbalance workload due to access pattens during the supersets. To solve these issues they have proposed a log based dynamic graph partitioning method. This method uses the recodes and reuses the historical statistical information to refine the partitioning result. It can be used as middleware and deployed to many existing parallel graph-processing systems. It also uses the historical partitioning results for creation of the hyper graph and it also uses a new hyper graph streaming strategies to generate the better stream graph partitioning result. Moreover it also dynamically partitions the huge graph and also uses the system to optimize the graph partitioning to enhance the performance.

Yang et al. [98] have described that searching and mining the large graphs now days is very critical in various application domains. So, for the processing of the large scalable graphs need careful partitioning and distribution of graphs across the clusters. In the paper, they have explored the issue of the managing the large-scale clusters and various proprieties of local queries such as random walk, SPARQL queries and breadth first search. They have also proposed a new approach called self-evolving distributed graph management environment (sedge). It reduces the communication during the processing of the graph query on the multiple machines. It also has two level of partitioning such as primary partition and dynamic secondary partitioning. These two types of partitions can adapt any kind of real environment. Results show that it enhances the distributed graph processing on the commodity clusters.

3.2 Applications of Big Data

Nuaimi et al. [99] have surveyed the various applications of big data for the support to a smart city. They have compared the various definitions of big data to the smart city. They also find out different challenges, and issues in big data for smart cities these are security and privacy, smart city population, cost, data quality, data and information sharing, data sources and characteristics. They explained that many governments are implementing the smart city ideas to improve the living standards of their people. The smart city uses the various technologies to enhance the performance of transportation, health, water services, education and energy that leads to higher levels of relief. Big data analytics is latest technology due to huge number data is digitized as this relevant to healthcare, education, etc.

These are two new terms smart city and big data that can be integrated together for better reliance, quality of life and effective management systems. They are also explored the concepts and common features for both terms that mention above. Based on these concepts and characteristic they have defended the advantage of big data for the smart city. As these advantages are efficiently utilization of resources, better quality of life, openness and transparency of higher level. They also described many applications of big data, where it is very effective, as these are smart education, healthcare, public safety, natural resource and energy smart traffic lights and smart grid. On the basis of review they suggested the requirements of for the big data for smart city and its applications. These requirements are management of big data, platform for processing the big data, infrastructure of smart network, advance algorithms, and security and

privacy. They argue that if we focus on these requirements we can overcome the challenges of big data for the smart city. Finally they have described the open issues that needed to be investigating for more enhancements for smart city. They conclude that there are several opportunities are available for smart city by employment of big data.

Raja et al. [100] have described that primary goal of a health care is to diagnose, treatment to aliments and avoidance disease by proper medication. Presently, health care reaches at the higher level of its reign, where we can apply the information technology get effects to the health informatics. Moreover in past there are lot issue to manage the large amount of data, now a days big data tools offers the health informatics. The arrival of health informatics has been big change in the area of health care. Traditionally, relational data base management system (RDBMS) was used for the find out the hidden value. But it has several limitations due to lack of processing of unstructured data, fault tolerance and linear scalability. On the other hand Hadoop have the better results for the large amount data as compared to the RDBMS. In many situation RDBMS solution failed as volume is increases. To over come these issues they have proposed a graph based database toget more information about the health care data. They proposed the graph-based database to store the information about different properties and relation between entities. In traditionally database store the information only about the entity. The primary objective of the graph database is to manage and traverse the connected data. Based on the proposed framework first they have did the analysis on the RDBMS and Hadoop by uploading the records ranges from 5000 to 50 millions and they did perform the query. They have found out the Hadoop perfume better than RDBMS and Hadoop have better fault tolerance. Secondly they did analysis on the graph database

and RDBMS. They analyzed that graph database have the better performance as compared to the RDBMS.

Raghupathi et al. [101] have described that information and communications technology (ICT) have basic components cloud computing, Internet of Thing (IoT) and big data, if we combine the features of these components, it can be possible for the shaping the next generation of e-health system. They attempted to analyze the current components and techniques securely integrating the big data processing with cloud machine-to-machine system based on Remote Telemetry. They also described various issues in the state of art methods for developing the health application. To over come these issues, they have proposed the machine-to-machine system based decentralized cloud architecture, general system and remote telemetry units , for e-health applications. They developed this system for the big data. They collect the information from the sensor theses information will be huge volume. The processes system consists of following principles data will be processed and stored closely. Multiple sources can be implemented seamlessly using the real-time data from the cross-domain applications. Using the inferred content scalability and energy efficiency can be achieved.

Collins [102] has described that upcoming year the big data have huge effect on the field of health. From the health point of view it may be possible the more data will be gathered and merged from the desperate information source and with automated evaluation. Moreover having the bio monitoring and lifestyle data this will enable health intermediations to be tailored more to individuals. They also argued that big data have the potential to advance the health economics as discipline. For this purpose they have did the SWOT (strength weakness opportunities and threads) analysis on the big data

approaches. They also described that the big data have the more opportunities and strengths for the health care. Consumer behavior can be monitor from the larger data sets through accurate analysis. They also described different strengths of health economics that with availability of lager data we can easily identifies which medicine is effective for particular individuals, hence it will increase the more reliable decision making for health improvement. People having the different background can use the open data for purpose of analysis. Weaknesses are that it is very costly to store and manipulation huge amount of data. In big data there're a lot of opportunities that the having the larger data set, it can communicate with every one and can be generate the better and complex analysis. One of the major threads is the privacy to data that people feel that their data may be misused. On the other hand big data offers a lot benefits for the individual, better health monitoring, smart health solutions and fewer mistakes. They concluded the there are more need for analytic skills in big data. There should be more chances in making the health system more effective in tailoring the medicine and more knowledge to persons.

Raghupathi et al. [103] have presented the comprehensive overview of big data analysis for the health care specialists and scientists. They described that the various possibilities and abilities of a big data for the healthcare analytics. They stated that big data in health care refers to electronic data set and complex that is difficult to mange with state of art software and hardware. From the big data researchers can able to more patterns and trends within the data. More over they are also stated the benefits of big data in health care due this disease at the earlier stage can be detected, can apply the more affective treatment, can also detect the health care fraud more quickly. Big data

can be helpful in sector in health care i.e. clinical operations, research & development; public health, evidence-based medicine, genomic analytics and patients profile analytics. They have also purposed architectural framework for the health care, it consists of four components names as big data source, big data transformation, big data platform & tools and big data analytics. Big data source it consists of internal & external, multiple locations, applications and formats. Big data transformation consists of extract transform and load (ETL), in this phase data will be transform into unique formats. Big data platform & tools are Hadoop, MapReduce, HBase and Hive where these are can be used to process the data. Big data analytics consist of quires, reports, OLAP and data mining. They also presented the different challenges big data analytics platform in health care must perform the processing for the given data. There should be a criteria for the evaluation of platform, it have ease of use availability, security and privacy, continuity, and quality assurance. They concluded that applications of big data in health care at the earlier stage of development, but they believe that with more advancement in big data tools and platform can quicken their development process.

Mehmood and Graham [104] have presented a model for the health care transport capacity sharing. Now days in 21st century health care industry is concentrate on investigating the reason of failure in the basic quality control process, customer needs and services and duplication logistics. They also explained the logistics; as it is analysis and modeling of transport and distribution systems through large data sets created by GPS, combine with human produced activity. They also explained that logistic firms need more support in term of technically to the there V's of big data. Main objective of this research was to provide more awareness about the capability sharing and

optimization in a smart city perspective. There is a need to develop the new tool and method to overcome the challenges of big data to smart city. Main goal was to contribute in improvements in transport capacity sharing through big data. They are explained that there many problems are exists in transportation as these are poor coordination of transports, lack if vehicles, poor maintenance and repair. To over come these issue there is huge need for new approach for the transport provision. They have introduced a new framework mixing the literature on smart city, big data logistics and capacity sharing. They also developed the Markov model by matching the needs of transport of patients with health care transport service provision. Basic purpose of this model was sharing of transport capacity in a smart city enhance the efficiencies in meeting patient needs for the city health care service. For analysis they have consider the thirteen different scenarios they find out the likelihood for system letdown and performance alteration tends to be highest in scenario highest needs sharing.

Huang et al. [105] have reviewed the different application big data in the health sciences containing the various systems having the source of big data these are recommendation systems, epidemic surveillance based on internet, food based monitoring, sensor based health condition monitoring, inferring air quality using big data, GWAS and eQTL. Main purpose of these systems to collect the data from the various people and make the analysis much faster than official channels. They also suggested that to start any big data project we have to follow the some steps in which first one is to choose the right problem. There are also three types of problems one, which can be, solve easily through state of art methods; there is no need for the big data. Second, these problems cannot be easily solvable, using the present technology. Third,

there are some difficulties that can be solve using the present technology i.e. big data. Second step they defined that find the source of data, this data we can collect the internet, smart devices, hospitals, social media and omics profiling. Third step, in this step data that collected from the previous step will be store in the NoSQL, GEO and bdGap. Step five, create the report of analysis using the vivid visualization. Data also be analyze using the three different systems as these are recommended systems i.e. collaborative filtering, content based filtering and hybrid filtering. Data can be through the deep learning and network analysis. For the visualization of big data results there are various tools can be used R, circus, Gephi and Tableau etc. Each of these have its own pros and cons. They also described that before the big data people get the information through the TV, newspaper and internet, it takes the years to create the awareness the about the health care. But in big data age people directly pushed to the smart devices.

Barkhordari et al. [106] have described that now a days there are huge amount of information is generated, traditional management systems can support the analysis on various field including the social networks, medical networks, scientific instruments and meteorology. When we retrieve, store and capture the data or information is fundamental problem in traditional systems. To overcome these issues there is need of scale able and distributed solution of these kinds of problems. So they argued that heath care is field where there is a need of distributed and scalable solution, due to present solution that cannot provide the this area problems. To overcome all these problems they proposed a ScaDiPaSi, a scalable distributable technique for the exploring the patient similarity. In this method they have used the various data sources unlike other they did not concentered on the structured and semi structured data source. For two patient same data

sources have different items. All these formats can retrieve by data integration. It is dynamic method store information about the patience and easily distribute on the hardware node. For the analysis of proposed technique they did the evolution based on the execution time and accuracy of ScaDiPaSi method. By using the ScaDiPaSi method they find out that they can easily achieve the desire results on distributed and scalable structure. They have measured the accuracy of their result obtained by the proposed method. They also find out the accuracy up to 63% of their proposed scheme.

Toga et al. [107] presented the a framework establishing the reasonable and practical data sharing policies that incorporate the sociological, financial, technical and scientific needs of maintainable big data community. They also argued that big data already stresses having the challenging needs of sharing the big data. Main features of big data comprises of size of data, incompleteness of data, incompatibility of data incongruent of sampling and heterogeneity of data. Sharing the big data needs innovative policies and clear guidelines that can enhance the cooperation instead of other problems and complexities. To overcome the complexities they have introduced a big data policy framework, which consists of various suggestions for sharing the big data depending upon the domain of application. As these suggestions are: i) Policies for storing and securing data and ensuring human subjects protections. ii) Process and policies for data sharing. iii) Protection of data from unauthorized access. iv) Agreement for data usage. v) Data value, sharing the data that is incomplete will have less value.vi) Big Data sharing policies for achieving cost efficiencies. Therefore there various health care and biomedical study have the important impact having the large, incongruent and heterogeneous datasets. So there are many important barriers like technical, social and

regulatory needed to overcome to ensure the power of big data. They concluded that there is need of implementation of above-mentioned policies for sharing of data, security for personal information; it will have the long-term impact on the big data analytics.

Jun et al. [76] have described that in data intensive application, there is a great need of high performance computation and massive resources. Many scientific and engineering applications many people and researchers are interested on the subset of the whole dataset, so they can access it frequently than others. Another example in bioinformatics in which X and Y chromosomes are related to the gender's offspring often researchers analyzed in the research rather than whole chromosomes. There is another of climate weather forecasting, in which scientist are only interested in the time periods. These groups of data can process by the specific domain applications. Assumption, if two pieces of data have been processed at the same time it is highly possibility that data will be process in the future as group. Problem with approach of the random strategies of the Hadoop workload will not be evenly distributed, it may be balanced the overall data. Hadoop has the random placement method for the load balance and simplicity. In MapReduce and Hadoop there is a by default random placement which does not count the semantics of data grouping. Cloud cluster grouped into many small no of groups, which limits the degree of parallelism for the data and outcome of this performance bottleneck. They also argue that there are other method was proposed for data locality was not effective due to various issues overhead and cost.

Solution, ideal data placement is evenly distributing the grouping data. Their observation was that random data placement is affected by the three factors. One each replica should have the data block on each rack. Second, there should be maximum no of

map task on every node and last one is data grouping access patterns. However, default random data placement can achieve the optimal data placement by holding the each one copy of data on the same node there maximized the parallelism can be achieved in case of rack (NR) is extremely large. Secondly, NS extremely large then node (NS) equals to the number of affinitive data.

To overcome this problem they presented a new technique “a new data group aware data placement scheme” (DRAW). It takes the run time data group patterns data and equally distributes the data. It consists of three phases: Firstly, there is data group information learning from the log. They have used the history data access graph (HDAG) to approach the files; it is based on the history. Hadoop cluster rack, name node maintains the log history of the each operation and including the accessed file. However, it also have two issues such as log is huge which also have traversal latency problem, and second issue is that frequently accessed files are not similar. Therefore to solve these issue there is need of checkpoint to traverse the log of name ode. Secondly, there is clustering the data group matrix (DGM), which shows the relationship between data blocks. It can be generated on the bases of the HDAG. Thirdly, data group reorganization based on the optimal data placement algorithm (ODPA). ODPA is based on the sub matrix for the cluster data-grouping matrix. They also prove that Hadoop random placement of data is not efficient. They did experiment on the real world applications, it reduce the completion time of map phase and execution time of MapReduce.

Lee et al. [77] have described that graph is important for discovering the knowledge from the given data set. They described that there are two dimension of graph analysis.

One is graph mining (GM); it has main focus on automatically knowledge discovery. Second is online graph analytic processing (OLGAP), its main concentration is on pattern matching on sub graph. Both dimensions are complementary and concentrate on the solving the complex problems. Both dimension covers the analysis of holistic in-situ in a sole system. There is difficult in processing the multiple graphs and sending the results to the system. Mostly state of art graph analysis based on only on dimension. To overcome this issue they took new technique enabling graph-mining abilities in RDF triple store. For achieving the desire goal they implemented the six different graph mining algorithms using SPARQL. It enables wide variety of RDF datasets which applicable to graph analysis for holistic. For evolution of proposed technique they used the various computing environment, nine different data sets. Evolution shows the scale able performance for in real world graph analysis.

Yinglong et al. [78] have described that big data analytics are very important to discover for such entities that can easily represent in the form graph. For the analytics of large-scale graph there is main problem for the delivery of efficient solutions that irregular data access. It is a main challenge for the processing of computation of graph-based patterns. Major challenges [79] that big data is facing for the graph processing are performance, large volume of data, and irregular data access. Big data tool are not suitable for the processing the graph due to the following reasons such as scalability, architecture awareness and systematic optimization. To overcome these issues they proposed a system called G. It enables the user to organize the data for the architecture of parallel computing. It also consists of visualization, graph storage and analytics. They also analyze the data locality in terms of graph processing, and its effects of the

performance of cache memory on processor. They also measured the traversal performance of the graph by both serial and parallel execution. For the experiment they did analysis on the parallel system and G system that is proposed system. They showed that G system perform much better then the traditional systems.

Fang et al. [80] have proposed the new technique called bipartite request dependency graph (BRDG) for the investigation of the relationship between objects of the web. They also leverage the MapReduce programming model for the construction of the BRDG from the large network data. Nodes of BRDG have two types of objects primary objects such as URL in web browser and secondary objects are those who trigger the primary objects. It is very difficult to derive the structural features of BRDG. To overcome this difficult they also proposed the co-clustering algorithm, form the BRDG it extracts the co-clustering coherent. In co-clustering of BRDG, each signifies the strongly connection to the bipartite sub graph. A parallel tri-nonnegative matrix factor (tNMF) algorithm they designed and implemented, basic purpose of this algorithm is provide efficient large-scale graphics decomposition. For the experiment they also divide the sub graph into four structural patterns such as click star embed star, single layer mesh and multiple layer mesh. They find out the multi layer of mesh are very famous structural pattern.

Xue et al. [81] have proposed the new technique based on the bipartite graph oriented locality scheduling (BLOS) for the MapReduce frameworks. Scheduling of the task is an important make span for the job of MapReduce. Improving the locality of the scheduling approach effectively, it is necessary to have the tasks and it related on the same node. They also argue that data locality refers to the task that have obtain from the same local

machine. A higher degree of local localization can enhance the performance of the system. Due to the localization it minimize the execution time of the job, reduces the communication time. Moreover, they described that there are three different types of the priority according to the proximity principles. First the priority will be given to the local task within the node. Second, priority will be given to the task within the rack and at last priority will be given to the off rack tasks. However, there is problem in this issue regarding the locality optimization, through this approach there will increase another problem called global optimization instead of local optimization. However they find out that the recent studies are not enough to tackle the data locality issues.

To overcome these issue they have proposed the new scheduling algorithm called the BOLAS for the MapReduce tasks. BOLAS can operate on any environment either it is homogenous or heterogeneous. Main aim of the BOLAS was to enhance the data locality without affecting the efficiency of execution. Bolas start solving the problem of scheduling of data locality by matching the bipartite graph in it try to allocate the data, which is close to the task. BOLAS have global data placement method through this they can achieve the better performance without affecting the nodes. It also avoids the network congestions with out generating the local nodes. Design of the algorithm consists of the two parts one is resource modeling and second is computing node performance estimation. In resource modeling, MapReduce have two major resources such as data blocks and name nodes. On the nodes blocks and tasks run, on the other hand task scheduling actually solutions between mapping of nodes and blocks. In performance estimation of computing node, BOLAS dynamically dispatches the blocks according the performance of the particular node. BOLAS also build bipartite graph in

various cases by adding the virtual blocks and nodes if there needed. Then also apply the KM algorithm for the optimal matching result with minimum weight. The benefit of the BOLAS is that it has high data locality. On the other hand it also increase the throughput of the cluster system and minimize the congestion of network. In the experiment they evaluated the ratio of data locality, execution time and network bandwidth. Results show that they improve the data locality of proposed technique by nearly 100% and minimize the execution time up to 67%.

Orozco et al.[82] have described that there are various challenges in the stencil application for the computations. One of the major issues the read modify writes operation the array data. The main limitation of these applications is off chip memory access, in terms of the latency and the data. To overcome theses issues they have proposed the locality optimization based on the data dependency graphs for the stencil applications. For this purpose first they have presented the formal descriptions that data that is not generated from the source code. They proved it my using the mathematical method for the commutation power and the bandwidth of the multi core architecture. For experiment they have used the various approaches such as Naïve, overlapped, split, and diamond tiling. They find out the diamond tiling technique is better than other approaches.

Hassanzadeh-Nazarabadi et al. [83] have presented the locality aware skip graph to overcome the issues of the name id assignment method of the skip graph's node. Skip graph locality is assigning the name id to the nodes such that more two nodes are close to each other, the more common prefix they would have in their name ids. They also argued that states of art method have not considered the skip graph's node locations.

Main objective of proposed technique was to minimize the latency in the search query from end-to-end. They proposed the dynamic and fully decentralized algorithm (DPAD). This method assigns the locality aware identifier will assign to the node instead of assigning the identity randomly. From the results they have shown the 82% improvement the data locality and 40% search query end-to-end latency.

Kandemir et al. [84] have provided the solution to the optimal memory layout detection. The Proposes approach has two basic elements such construction of the problem in a special graph structure and second is the use of linear programming (ILP) solver to define memory layout. It is the first approach that allows to dynamically changing layout cache locality. They have also shown that proposed framework for the locality is powerful. But there is dynamic layout modification for the large applications still remaining to explore.

Chernov et al.[85] have described the problem thread partitioning of the sequential programs. They proposed a new algorithm to overcome this issue. They also argued that performance could improve by considering the locality of the program. Many program have by nature locality characteristics. So they combined two optimizations for new algorithm. One, non-loop region can be parallelized. Second, perform the partitioning in such way that data locality can be improve of new thread. For the evolution of their approach they have generated the data dependency graph (DDG) and showed that their proposed approach is feasible.

Zhang et al. [86] have described that k nearest neighbor(*k*NN) have is popular approach for the various application specially machine learning and graph based applications. Besides its various characteristics but it has the computation complexity.

To overcome this problem they have proposed the new algorithm that divides the whole dataset into the small groups, and it make sure the each item of kNN belongs to the group. It leads to the accuracy and fast speed. To make the proposed algorithm more accurate, they have also divide the groups in such as way that similarity between the items remain in the same group. Secondly, they kept the group size small as possible. For the groups they also propose the locality sensitive hashing (LSH), which guarantees the rigorous performance even for the worst-case performance. They evaluated their approach that method can efficiently generate the graph with good accuracy.

Zhang et al.[87] have described that there is widely explored area of graph databases is similarity in graph processing. They also argue that graph have an important role in various applications like pattern recognition, information retrieval etc. They find out two categories of the graph search, one is sub graph search, second is similarity search. In this paper, they have explored the k-NN similarity search problem using the locality sensitivity hashing. They proposed the algorithm for the fast graph search that it transforms the complex graphs into vectorial representations based on the prototype in the database. After this it also accelerate the query efficiency in the Euclidean space by employing locality sensitive hashing. They evaluated their approach against the real datasets, which achieves the high performance in accuracy and efficiency.

Yuan et al. [88] have described that there are two main challenges in processing of the large scale graph processing: one is lack of the efficient storage, second is lack of the locality access. They also described that there are various popular approaches for processing of graph and storing the data such as storage centric, vertex centric and edge centric. They listed the common graph partitioning approach such as vertex cuts and

edge cuts. But commonly used partitioning scheme for the processing of the graph is vertex centric hashing. However, these approach having the issues of the very poor locality and communication overhead. To solve these issues they have proposed new path-centric approach for the fast iterative computation of the graph computation for the extreme large graphs. It is a path centric at both storage and processing tier. It is an efficient approach for storage and design structure for storage tier. It also allows the fast loading in edge and out edge for the gathering and scattering the parallel computation of the graph. But for the computation tier, its processing is used in such a way that optimized the locality of the large-scale graph. They iterate the processing or computation chunk level or partition level computation. Work stealing approach has been introduced in the work to balance the load among parallel threads. For the evaluation of their approach they have chosen the real dataset in which they demonstrated that proposed approach performance in terms of better balance and speedup.

Shao et al. [89] have described that partitioning schemes have great importance in parallel processing of the graph computation. In current graph system, they find out unaware partitioning issue for the computation of the graph and it also has the various drawbacks in the processing of parallel large-scale graphs processing. State-of-art approaches for the partitioning schemes are preferred in case of small edge cut ratio. The advantage of this was less communication among the working nodes. Sometimes, partitioning approaches of graph may have worst performance as compared to the simple partitioning. It causes the local messaging passing to super pass the cost of

communication in many cases. They also argue that these systems are having parallel graph partitioning approaches. However, sometimes it can happen when users try to integrate the graph partitioning methods into parallel graph computations system is not a trivial task. But there are some system, which they have good balanced, partitioned approach leads to the overall computation performance. They also analyzed the graph partitioning schemes on various systems using the web graph dataset. They find out the current parallel graph systems cannot be benefited from the high quality graph partitioning.

To overcome these issue they have proposed a new approach partitioning aware graph computation engine (PAGE). The benefit of this approach is that it controls the online graph partitioning statistics of under lying results of graph. Second, it monitors the parallel processing resources and enhances the computation resources. Third, it was also designed to support the various graph partitioning qualities. In this work they have analyzed the cost of graph partitioning and the cost of the parallel processing of graph. Page also consists of master worker paradigm is responsible for the aggregating the global statistics and coordinating the global synchronization. But the page worker takes the graph computation tasks aware partitioning informations. Page has two modules such as communication module and partitioning aware module. In communication module, it has concurrent dual messages processor. Partitioning aware module monitors the status of the system. Moreover, they also designed the Dynamic Concurrency Control Module (DCCM). DCCM has various heuristics rules for the message processor to optimize the concurrency. It also processes the concurrent local and remote messages.

For evaluation of chosen schemes they showed that it preforms well under various partitioning approaches with various qualities.

Qin et al. [90] have described that mining of the dense sub graphs from the large graphs is fundamental mining task that can be apply on the various applications domains such as network science biology, graph database, web mining etc. They also argue that exiting schemes have concentrated on the just dense sub graph or identifying an optimal clique like dense graphs. In these schemes they have implemented greedy approaches to find out the top k dense graph. But on the other hand, their identified sub graph cannot be represent as dense region. So identified sub graph should be the highest density region in the graph. In this work they introduced a local dense subgraph (LDS) in the graph. It can used to find out the dense sub region from the sub graph and can be apply to the various applications. LDS also have some useful properties such as pairwise disjoint, locally dense and compact/cohesive. They also define how local dense sub graph in terms of parameter free with useful characteristics. They also have presented an elegant dense sub graph model. They also showed that LDS problem can be solve in polynomial time. They presented the three optimization approaches to enhance the algorithm. To evaluate the LDS, they have analyzed their approach using the four different qualities of measures such as density, relative density, edge density and diameter. They also did experiment with real web scale graph having 118.14 million nodes with 1.02 billion edges for the demonstration of the efficiency and effective ness of proposed algorithm.

Zamanian et al. [91] have described that horizontal partitioning approach has been extensively is used for the processing of large amount of structured data. But there is

issue for using the horizontal partitioning approach that cost of the network must be minimize for given workload and schema of database. However, there is popular approach to minimize the cost of network for parallel processing of database is co-partitioning the given table on their join key to avoid expensive remote join operations. On the other hand these approaches have issue of the replications and co-partitioning with sharing in the same join key. To solve these issues they have introduced a new approach predicate-based reference partitioning (PREF). It enables the to co-partitioning sets of tables based on the given join predicates. Main goal of PREF was to enhance the data locality and minimize the data redundancy. For this purpose, they also designed two new algorithms. In these two algorithms, first algorithm required the schema as input whereas second algorithm takes the workload as input. In experiments they showed that workload driven design algorithm is more efficient for the complex schema with large number of tables.

Chen et al. [92] have described that various existing techniques supports the vertex-oriented execution model. It also allows the user to create their own logics on their vertices. But there is an issue in terms of network traffic overhead for the vertex-oriented computation. However, graph partitioning is very useful for the minimizing the network traffic in the processing of graph. They argue that there is very little attention has been made for the partitioning of graph effectively integrated into the large graph processing in the cloud environment. Furthermore, the motivation they got that surfer is a master-slave system, consisting of one master server and many slave servers. The slave servers store graph partitions and perform graph computation. They studied the factors affecting the network performance of graph processing. They assume that the amount of network

traffic sent along each cross-partition edge is the same. So they have proposed a new framework of graph partitioning to enhance the performance of the network for graph partitioning itself, storage of partitioned graph and vertex oriented processing of graph. They have developed all these optimizations for the cloud network environments. They also have used the two models such as partition sketch and machine graph. Basic purpose of using these two graphs was to capture the features of graph partitioning process and network performance.

In experiments, they have developed a new prototype for the Pregel and enhanced it with their own framework of graph partitioning. They also showed the efficiency and effectiveness of the their proposed technique for the processing large graphs.

Zeng et al. [93] have described that processing of distributed graphs is very costly for the computation of large amount of the data in case moving the data among various computers. However, they argue that state-of-art approaches have high computation overhead and costly communication when apply on the distributed environment. To overcome these issues they have proposed new parallel multi level stepwise partitioning algorithm. They divided this algorithm into two phases; one is aggregate phase and second is partition phase. In the phase one, it uses the multilevel weighted label propagation for aggregation of the large graph into the small graph. But in second phase: It has the K-way balance-partitioning preforms on the weighted based on the stepwise mining RatioCut method. It reduces the RatioCut step by step. In each step, sets of vertices are extracted by reducing the part of RatioCut and these vertices are removed from the graph. In this they have obtained the k-way balance partitioning by this algorithm. In experiments they have made the comparison with various other existing

partitioning approaches using the dataset of graph. It preforms well as compare to the state of art approaches in terms of scalability, performance and can enhance the efficiency of graph mining on the real distributed computing system

Lee et al. [94] have described that scientific and engineering domains are growing various type of information and size these days. Due these needs there is another demand arise in cluster computing of cloud for efficient processing of the large heterogeneous graphs. They described that heterogeneous large graph have various characteristics in terms of big data processing. Firstly, graph data is highly correlated and topological structure of big graph can be viewed as a correlation of the vertices and edges. Graph that are heterogeneous they add the extra overhead as compared to homogenous graphs in terms of processing and storage. Secondly, queries are over the graphs is typically subgraph matching operations. But, they argue that it is ineffective of modeling the heterogeneous graphs as big table entity vertices. They described that HDFS is very popular for the processing of the partitioning big data among the large cluster of computing nodes in clouds. But, the HDFS is not optimized for the processing of the highly correlated data for large dataset such graphs. Therefore data generated by such random partition methods cause an extra overhead. Moreover, theses types of random partition methods also cause overhead for the graph patterns query. There is another problem arise with it that how to partition the graphs that it preforms efficiently. To solve this problem they have introduced vertex block (VBs) partitioner. It is a distributed model for the data partitioner for the large-scale graphs in the cloud. It has three features. First, It has the vertex bock (VBs) and it also extends the vertex block (EVBs) as building blocks for the semantic large-scale graphs. Second, vertex block

partitioner uses vertex block grouping algorithm to place the high correlation in graph into same partition. Third, the VB partitioner speedup the parallel processing of graph pattern queries by minimizing the inter-partition query processing. In results they showed that proposed approach have higher query latency and scalability over large-scale graphs.

LeBeane et al. [95] have described that large scale graph analytics are very essential issue in present data center. Large multi node processing is a critical computational problem due the popularity of the big data and cloud computing. There are many state-of-art frameworks available for the processing of large graphs such frameworks are: PowerGraph, Pregel, and Giraph. However, there is a problem in these frameworks that any change occur due to the cloud and big data, these frameworks cannot handle it. These state-of-art frameworks cannot be scalable. Data is coming from the various sources for the processing. So they argue that data center are trending towards the large number of heterogeneous processing nodes. Moreover, they also described that virtualized environment can also create the heterogeneity by partitioning the cluster of homogenous machines into the variety of configurations. However, frameworks of the graph analytics are still working under assumptions. This assumption lead to the imbalance of the load and cause the faster node finish the processing their chunk of data earlier than slower nodes. As in the heterogeneous in the data due to the visualization, load balance and heterogeneous nodes becomes more critical for the graph processing. To solve this issue they have used the popular framework for the heterogeneity aware data strategies. For the heterogeneous partitioning of the data, they divided the input data into shards that every node will receive the data according to the to metrics. They also

define the data-partitioning ratio between to be the skew factors of the clusters. In this work they also described three different types of the skew factors such as Thread based, Memory based and profiling based. Although there are many complex method are available for the calculation of the skew factors. However, these can provide the benefits to the performance for the heterogeneity aware partitioning algorithms. They also illustrated the simple estimate of intranode throughput that skew factor can drive the huge performance using the heterogeneity aware partitioning strategies. They also showed that their proposed strategies minimize the 64% execution time.

Chen et al. [96] have described that there is unique challenge in the graph partitioning and computation especially in the natural graph with skewed distribution. They argue that exiting graph processing system have the problem of "one size fit all" that impact on the performance, load imbalance and contention for the high degree vertices. Even though exiting graph-processing system also have the high communication cost and high memory consumption. To overcome these issue they have introduced a new graphic engine called PowerLyra. It is hybrid and adaptive design that has the dynamic partition and computation approaches for the various vertices. It also combines the edge cut and vertices cut with heuristic using the hybrid algorithm of graph partitioning. It also provides the data locality aware layout optimization to enhance the cache locality during the communication. They also did the experiment using the two different cluster using the graph analytics and for the machine learning and data mining algorithm. It performs much faster and consumes less memory.

Xu et al. [97] have described that graph partitioning now days is popular strategies to balance the work load due the scale of graph data and increasing availability. Due the

cost of partitioning of the graph, recently researcher are more focusing on the stream graph partitioning that is very fast, incremental update and easily parallelize. But, it has also the challenges such as the imbalance workload due to access pattens during the supersets. To solve these issues they have proposed a log based dynamic graph partitioning method. This method uses the recodes and reuses the historical statistical information to refine the partitioning result. It can be used as middleware and deployed to many existing parallel graph-processing systems. It also uses the historical partitioning results for creation of the hyper graph and it also uses a new hyper graph streaming strategies to generate the better stream graph partitioning result. Moreover it also dynamically partitions the huge graph and also uses the system to optimize the graph partitioning to enhance the performance.

Yang et al. [98] have described that searching and mining the large graphs now days is very critical in various application domains. So, for the processing of the large scalable graphs need careful partitioning and distribution of graphs across the clusters. In the paper, they have explored the issue of the managing the large-scale clusters and various proprieties of local queries such as random walk, SPARQL queries and breadth first search. They have also proposed a new approach called self-evolving distributed graph management environment (sedge). It reduces the communication during the processing of the graph query on the multiple machines. It also has two level of partitioning such as primary partition and dynamic secondary partitioning. These two types of partitions can adapt any kind of real environment. Results show that it enhances the distributed graph processing on the commodity clusters.

Wang et al. [108] have described that Genome sequence is an emerging filed for research on various domains such as big data, biomedical and biological. So next generation sequencing (NGS) has the short read problems. On the other hand data is increasing with the scientific research but technologies depends on the acceleration techniques. However, there are many open source software that supports the short read mapping by running cluster on the processor. But the enhancement in the genome sequencing leading the problem of bottleneck from the sequencing to the short read mapping problem. Short read problem have been proved that it work well under the map reduce framework. But it can lead it to the degradation of the performance of the response time. To solve these problem they have proposed a new architecture for the acceleration of read and map reduce processing when it faces the many request from the filed programmable gate array (FPGA). A map reduce framework, they introduced to mange the specific task into the various FPGA. Thus the MapReduce algorithm is based on the RMP sequencing algorithm. They also described that there are three types state of art of acceleration techniques for the short read mapping as these are GPU based, FPGA based and MapReduce based. They also did the analysis on MapReduce for every chip FPGA. They did experiments with respect to utilization of hardware, sensitivity, rate of error, quality and speed up of hardware.

Jaiswal et al. [109] have introduces the enhanced framework for Genomics using the big data computing. In this framework, they described that Genomics is all about the study of genetic organism. Genomics includes the mapping, sequencing and analyzing of the broad range of codes of DNA, RNA and many other things across the human life. They argued that that arrival of genome sequence, large quantity of nucleotide and

amino acid data sequence has been formed. However, It is an essential to be verify and analyze these data effectively and efficiently. Because this data is type of data is used in the biological invention. On the other hand they argued that it depends upon how accurately we interprets the huge volume of high dimension data. But there is another problem in this that it increases suddenly at the exponential growth.

Due to the scalability of data tools that currently available do not provide such analysis that required. They described that there are two types of genomics computing, one is cloud and second is big data computing. So presently the main challenge is processing of data to maintain the infrastructure of growth of the data. Ho ever people are applying the map educe based technique in cloud having advantages like reduce the memory and execution time. But the cloud itself also has problems like data protection, availability, and recovery of the data. They also describes that Hadoop is very popular overcome all these challenges that described above.so they proposed a new technique for the cloud and big data that enhance the genomic computing. They designed the basic local alignment search tool (BLAST) on the basis of pig that very reliable and efficient technique. Mainly the proposed framework work accurately manages the load of reducers.

Qin et al. [110] have described that there are three major types of biological macromolecules as RNA, protein and DNA having huge basic elements in the human biological organism. They are all functions at the particular level such as cellular and organismal level, interactively and individually. In this research paper they have described various high throughput sate of art techniques and data collaborative technique for the biomedical and biological methods. They also described that RNA, which is

ribonucleic acid, is also the product of DNA transcription. On the other hand protein also the very important functional macromolecules. They described some traditional techniques such as macromolecules that based experiments, which consumes the time. Last few years many people are working on the new technologies having the high throughput such as next generation sequencing (NGS) and microarray. In this paper they have demonstrated the NGS, which is used to discover the variation of genetics linked with diseases.

However transcriptomic data formed the landscape of all transcripts in various cell types. On the other hand where as proteomics data is helpful in which measure the quantity presence of proteins and monitor the PTMs of proteins. So big data is produced at the various levels of molecular interactions, it is very helpful for the understanding of the biological working of the organism.so there are many project are introduced for the consortiums regarding to the big data. As in big data there are still problems is some area like data storage, processing, security, visualization and transferring however these issues and challenges also in the filed of biomedical fields. At present high performance computing cluster handle and stores the sequencing data. It needs huge amount of storage and high requirements for computation speed. Thus processing of the data is the challenging issue even different tools have been developed for the various data types interpretation and integration. So these are not up to the mark as required due to the technical flaws and noise. Main objective of the biomedicine is to minimize the noise and enhance the efficiency and accuracy of the computation of biological process, mathematics, statistics and IT science.

Davis et al. [111] have described that in metabolic modeling genome maintaining the genome consistency is necessary for various computational tasks. A process is implemented to enhance the consistency of annotation among all microbial genomes for the protein having the conserved context of genomic and sequence. In this research paper they have presented that how the enhancement can be resultant through the efforts for the genome annotation in the seed. More over solid cluster is fully automated due to the generation; it also provides opposite methodologies to the state of art approaches of genome annotations that structures the hierarchical annotations like seed subsystems. They have also compared their seed genome annotation with other regularly used resources such as IMG, RefSeq etc. They also assessed the consistency of present sets of annotations from the number of resources.

Yeo et al. [112] have described that big data management is typically require in the healthcare and intensive applications. It very important for the modern advances next generation sequencing (NGS) tools which growing very fast with large amount of information in timely manner. In this paper they have concentrated in the application of health care for the testing as well scale of commercial big data platform with apply of map reduce tool. Moreover they also choose the Bowtie, Contrail-bio and Blast workload of genomics for the platform of big data, they have tested it whether the cluster can scale on the Hadoop HDFS as file system. Their results shows that the cluster can handle the can handle extensive variety of applications of bioinformatics while providing suitable computational scalability and efficiency. By compressing the intermediate data they speed up the process up to twenty percent.

Heinzlreiter et al. [113] have described that due the increase in the data various needs for the frameworks and tools for the analysis of these data. There are also challenges like handing these types of data. In this research paper they focused mainly on the genome sequence implementations and comparison with the bioinformatics domain relying the HBase tool that are running on the top of the Hadoop for the computation jobs of MapReduce. They also described there are many challenges in bioinformatics such as in single genome consists of hundred of megabyte in single genome. Due to the next generation sequence (NGS) genome sequence rate is increasing exponentially. To handle this type of data there is need for framework that manages this type of data efficiently. For this purpose they have developed the application of the genome in Hadoop and also made comparison with HBase table and also executed on the MapReduce. The proposed strategy supports the reuse of intermediate data that have been generated before the processing step preformed on the single genome. Input data for the preprocessing steps is performed by the map recue jobs to compare the gene. After this they stored the results in the HBase tables, then generated the graph of the scalable genome comparison.

Liang et al. [114] have described that frequent item set mining (FIM) is an essential topic of the research sue its various application such DNA sequence discovery, real world applications and pattern mining behaviors of the human. The process of the FIM is computational and memory intensive. Now days data is increasing exponentially, the challenges and issues like scalability and efficiency become more severe. To solve these issue they proposed the distributed new FIM algorithm names as sequence growth, which was executed on the MapReduce framework. The proposed algorithm constructs

the tree in the lexicographical tree sequence, which permits to discovery the all-frequent item, sets with out exhaustive search over transaction database. They find out that proposed algorithm effective remove the generation of large amount of intermediary data and also executes the algorithm in memory fitted in the better way. They also did experiments and checked the effectiveness and efficiency of their algorithm. They find out that their sequence growth can be modified easily according to the association rules mining algorithm can be adapted on map reduce framework easily.

Dodson et al. [115] have described that there are many advance in the resent technologies such as Next Generation sequences (NGS) tool to enhance the speed of the DNS collection of samples, sequencing, preparation and collection. On the single run one genetic sequence can generate the over 60 GB sequence of genetic. Presently technology can rapidly recognize the DNA sequence that which system it likely to belong. But problem for the recognizing the system of the human sample can take up to 45 days which major bottleneck for the analysis of sequence and shipment of sample sequence center. It also makes the opportunity of the grip efficient growing workload. Therefore to solve these issue they proposed the new fast and efficient method for the genetic sequencing and analysis called the dynamic distributed dimensional data model (D4M). D4M is a novelty in computer programming that associates the characteristics of five processing technologies such as associative array, sparse linear algebra, triple store database, linear algebra and distributed array. On the base triple store database, they candle the large amount of data. D4M provides the parallelism by using the linear algebra on the interfaces of the triple store.

Phinney et al. [116] have described that there is huge challenge to identifying the gene sequence repetitively that occurs within the DNA sequence. It is very simple conceptually but computationally it is huge challenges. Moreover there is also a big challenge for the biological research such that many regions within the sequence of genomic have not change for so many years. Identifying these uncovered regions is big challenge for the researchers. From the perceptive of the evolutionary, in DNA initial step is to find out the diverters and similarities. Main problem arise due to the analysis of large amount of data. It rises with each genome, which is sequenced. They argued that state of art approaches needs the pair wise sequence for the comparison of the chromosomes that consumes the time in terms of execution. To reduce this overhead they designed the new algorithm that partitioned the sequence of genome based on the value and repetitive sequence. It also consists of single aggregation of global step that recognize the all matches among the sequence concurrently. Scalability of this approach is the main characteristic. Additionally, using the more node for the computation for the can increase the performance of the system until the map and reduce nodes increases than the computation nodes. By managing the storage and memory their method can employ on the HBase.

Toh et al**.** [117] have described that DNA sequence is growing every day, for the each database it takes the longer time to find the specific sequence. Using the supercomputer can be possible but it is very difficult for very one get it, there due to this limitation, there is need of an efficient algorithm to search the sequence from the amount of DNA. For this, they proposed new technique, which is sequence search and alignment by hash algorithm (SSAHA) for the search of the sequence from the database. In this

approach they divide the each database into the k-tuples of k contiguous bases and then process the query base by base. They find out that if query do not match on the first hit it is high possibility, it will not match later on hits.

They argue that it is possible to apply BLAST on the first stage of the SSAHA, so it can create the overlapping words having the consistent length with query can increase the sensitivity. Therefore, they also proposed the new technique called the Basic sequence Search by Hash algorithm (BSSHA). It divide the algorithms into two parts first part is time hash for creating the hash table in the database, second is part is time search for the processing of the query. Moreover database table is create in the main memory, it will reside the memory so it will save the time of disk access. They also showed that time complexity and processing time for the BSSAH is less than exiting techniques such as SSAHA.

Meng et al. [118] have described that to process the bioinformatics information, sequence alignment is the basic method and information evolutions. They described that we can divide into two categories, in category one, there are two types one is sequence alignment of pairwise, second is multiple sequence alignment. In second categories there are also two types one is local alignment that is based on the sequence range second is global alignment, which is based on the whole sequence alignment. Local sequence is more is very useful than the global alignment. They described problems related to the local alignment related to the blast algorithm. Currently, serial Blast is very complex and not useful for the large amount of data. It did not meet the needs of the current large amount of genetic data; implantation on GPU and heterogonous computing is very complicated.

Therefore, in this paper they tired to improve the blast algorithms in terms of distributed parallel processing. They achieved the parallel blast algorithm on the of Hadoop blast algorithm. Hadoop blast algorithms select the words that have strong correlation as an item of the list using the Hadoop parallelization. The proposed scheme performs well when there is large amount of data. The execution efficiency of the blast algorithm is enhanced on the bases of the Hadoop.

O'Driscoll et al. [119] have described that there is exponential growth in the database of the genome it is very hard to process these biological computations. As they argue that parallel approach of divide and conquer can be apply to the query sequence of input and data set. So there is a still issue of the scalability in terms of memory or huge amount of data lead to suffer the performance. Therefore, they have introduce the new approach the Hadoop Blast (HBlast), in this scheme partitioned the sequence of the query by using the virtual partitioning. Using this approach it increases the performance of scalability over available solutions. It also balances the computation workload by keeping the replication and segmentation database to least. It also has little over head as in the communication and mapper.

Sait et al. [120] have described that Basic local alignment search tool (BLAST) is very commonly used for the programs of bioinformatics to search the available sequence of the database similarities among the DNA and protean using the technique of sequence alignment.

Currently, there many search algorithms have been proposed for the publically available clusters of gnome database. So in this research paper they have evaluated the performance of serial and parallel blast using the traditional state of art diskless cluster

of HPC. Important inspiration behind the proposed is the source of failure of HPC clusters. It also enhances the reliability and minimizes the cost of the communications as compared to the traditional disk full clusters.

Borantyn et al. [121] have described that BLAST is very commonly used tool for the search of the sequence from the database. In this work they have focused on the sequence of the protein. Blast based on the position specific iterations search the sequence of pattern from database iteratively. It also uses the position specific matrix (PSM) for the round matching. Another tool was developed called the context sensitive blast (CS-BLAST). It combines the information of query that is recently search by the specific query, derived from the protein profile for the achieving the better homology detection as compared to the PSI-BLAST, that construct the position specific matric from the scratch. They also proposed the new method call the domain enhancement lookup time accelerated blast (DELTA-BLAST). It began with query by conserved domain database then it makes the multiple alignment of the conserved domains after this it compute the PSSM at last it apply sequence search query on the database. It searches the database using the query by pre construct PSSMs before began to search the sequence of protein from the data base and have the better similarities.

Chapter 4

Application

In this chapter we shall discuss about our proposed application, which will developed on the bases of the proposed technique. So we have proposed the "A Healthcare Transport Application" for which we shall apply the chosen technique.

4.1 Datasets

4.1.1 Geographical Data

Geographical data is a fundamental data type in urban visual analytics, which provides basic structure as well as semantic information for urban computing scenarios. In the field of visualization, road network data, transportation network data and POI data (point of interest) are frequently used data of this type.

Road network data: is usually in the structure of a graph that is comprised of a set of edges and nodes, representing road segments and intersections respectively. Each node is described by a unique set of geographical coordinates, while each edge is associated with other related properties, such as length, speed limit, type of road and number of lanes.

Transportation network data: includes transit routes and stop facilities of the bus and metro network, which is modeled as a directed graph. Each stop facility is described with ID, geographical coordinates and related edge connection in the network. In addition, schedule information is often included with a timetable showing when each bus/metro leaves its starting terminal, and reaches each stop along its transit route.

Tiger Data: In TIGER [122] dataset there are various types of data information are available such as hospitals, educations systems, transportations and roads network data. This data is xml format we can convert this data into the graph format using the specific tools but there will require lot efforts to get this data to required format that we can process. This data set is updated every year and each year data is separately available.

Open Street Map (OSP): Metro Extractor of OSM [123] automatically creates snapshots of Open Street Map data into manageable, metro-area files in a variety of formats for you to use. To get the data about related to open street map (OSM), we can generate the metro extractor. This metro will have the all information about road, city, location and POIs. It also creates the three types of files such pbf, xml and raw text file. To get the extract data from these files we need special library like **this** or we write own code to read this file. But due to shortage of time we have used above-mentioned library.

4.2 DIMACS

The DIMACS [124] is collection of various datasets. It also has road network dataset of having more than 50 states of United States (US) and various districts. It is undirected weighted graphs consist of billion of edges and nodes. Each node has node id, latitude and longitude. Every edge also has source node id, target node id, travel time, distance and category of road. We have choose the Rhode Island data it has 53 thousands of vertices and 69 thousand of edges. As this Data is visualize it in the next coming sections. The format of the uncompressed file is very simple. It is a whitespace-separated list of numbers, **Number of nodes:** for each node there is id, longitude, and latitude.

There also **Number of edges**, for each edge there is id of the source node, id of the target node, travel time, spatial, distance in meters, road category. Moreover, there is the *id* is a number between 0 and "*number of nodes*"-1, *Longitude/latitude* are given in the format used in the TIGER/Line files (TIGER).

Excerpt: Coordinates are decimal degrees expressed in Federal Information Processing Standard (FIPS) notation, where positive latitude represents the Northern Hemisphere and a negative longitude represents the Western Hemisphere. All coordinates are expressed as a signed integer with six decimal places of precision implied.

Table 4.1 Latitude and Longitude Selection

	Actual	**TIGER/Line file**
Latitude	15 Deg. S to 72 Deg. N	-15000000 to +72000000
Longitude	64 Deg. W to 131 Deg. E	-64000000 to -180000000 +179999999 to +131000000

Road category is adopted directly from the TIGER/Line files (refer to the documentation). Spatial distance in meters is the great circle distance from source to target node travel time is the spatial distance divided by some average speed that depends on the road category:

Table 4.2 Roads Categories

Category Code	**Category Name**	**Average Speed**
A1	Primary Highway With Limited Access (e.g. interstates)	1.0
A2	Primary Road Without Limited Access (e.g. US highways)	0.8
A3	Secondary and Connecting Road (e.g. state highways)	0.6
A4	Local, Neighborhood, and Rural Road	0.4

4.3 Getting Point of Interest (POI) from Open Street Map (OSM)

In section, we shall present that how to obtain the POIs from OSM files using the metro extract of any city, state, country of the world. But in out work we have chosen the Rhode Island an American state, following steps will show you the whole detail.

Step 1: Start
Step 2: To the Open Street Map (OSM) "website" extract the point of interest (POI) using the metro extractor Mapzen (POI can be from any of these categories such as Amenity, Geological, Highway, Historic, Man made, Military, Place, Public transport, Sport,Tourism, Building, Using the Mapzen metro extractor, we get the OSM protocol buffer Binary format (PBF) file that has all POI of particular city, state and country.
Step 3: After getting the PBF file , we shall follow the all steps of "Tool for extracting
POIs from OSM binary files" (available here) and Download the latest **osmpois.jar** file from the releases page.
Step 4: In next Step we shall get the planet file or any other such as ABC.osm.pbf file.
Step 5: Next we shall put the OpenStreetMap binary file (.pbf) in same directory as the jar file.
Step 6: Now, we shall run Java in a terminal (make sure you have at least 9GB of free RAM for the whole planet).
Step 7: We shall write following Command in terminal to get all POI in CSV file format "java -Xmx9g -jar osmpois.jar ABC.osm.pbf".
Step 8: Program creates a CSV file (by default a pipe-seperated-file | as these are rarely used in location names). The first four columns are always the same. The following columns depend on which outputTags have been defined, in order they appear in the parameter list (see "Parameters" below). By default "name" is the only tag that's exported.
Step 9: Stop

4.4 Getting POIs Neighbourhood (NB)

In this section, we presented the neighbourhood (NB) from osm file. In OSM there are various types of POIs are available as there are range from amenity, geological, highway historic, man made, place etc. For our work we have chosen places because we

interested only neighborhoods we can get it easily from this category. Inside of places, there're also many other thing like village, hamlet etc.

In this work we taken the OSM metro extract of Rhode Island (RI), an American state data. This data we have in form of pbf file and it has all information of RI. So we applied the all steps as mentions in previous section "*POIs from OSM*" and got following: village 98, hamlet, 508, isolated_dwelling 0, farm 02, suburb 02, neighbourhood 07. As these are total no 617 NBs.

4.5 Getting POIs Healthcare Centre (HC)

In this section, we presented the Healthcare Centre (HC) from osm file. In OSM there are various types of POIs but we have chosen amenity. Inside of amenity, there are also many other amenity types like café, fast food, food court, ice cream, hopitals etc.
So in this work we taken the OSM metro extract of Rhode Island (RI) is an American state. This data we have in form of pbf file and it has all information of RI. So we applied the all steps as mentions in previous section "POIs from OSM" and got following: clinic 08, dentist 01, doctors 11, hospital, 47, pharmacy 66, veterinary 04, social center 01, nursing home 08, social facility 08. As these are total no 154 HCs.

4.6 Shortest Path between POI (Direct Displacement)

In this section we have calculated the shortest distance between two places based on location latitude and longitude. Second, we did this on the RI data for roads and POIs of RI data, chose the nodes from RI data that are near to or on the POIs, for this we chose proposed following algorithm.

An Algorithm to get the POIs and Nodes using direct displacement

1: Start
2: Add the input files POIs (NB&HC) file and Nodes files
3: Getting the Size of from the POIs (NB&HC) file
4: Declaration of Matrix to store the data from POIs (NB&HC)
file and Nodes files
5: Declaration of Matrix Size to store the Storing Data into
Matrix for POI
6: Storing Data into matrix for POIs
7: Storing Data into matrix for Nodes
8: Printed the store data to check that it stored in correct form
9: Initialize variables such var dist :Double = 0;
var min = 0.0; var max =0.0; var POI_ID=0.0 var Node_ID=0.0
10: Finding the closest point using shortest distance formula
for two pints on the map based on latitude and longitude
10.1: Repeat the steps until Outer_loop < size of POIs
10.2: Assigned the longest Integer to variable min
10.3: Repeat the steps until Inner_loop < size of Nodes.
10.4: Compute the direct displacement formula
dlon ← lon2 - lon1
dlat ←lat2 - lat1
a ← (sin(dlat/2))^2 + cos(lat1) * cos(lat2) * (sin(dlon/2))^2
c ← 2 * atan2(sqrt(a), sqrt(1-a))
d ← R * c (where R is the radius of the Earth)
d1 ← d*3.1415
dist ← d1/180
10.5: If dist<min
min ← dist
POI_ID ← MatrixPOIs(ii)(0)
Node_ID ← MatrixNodes(jj)(0)
10.6: End of Inner_loop
10.7: Storing the POIs, nodes and distance
MatrixMIN(ii)(0) ← POI_ID
MatrixMIN(ii)(1) ← Node_ID
MatrixMIN(ii)(2) ← min
10.8: End of Outer_loop
11: Write Data into Text File from MatrixMIN
12: Stop

4.7 Shortest path between HC and NB: distance by hops/roads

In this section we calcuted the distance between HC and NB by roads such as total number road involve in the source to destination. For this purpose we computed the shortest distance by no of road this using the BFS in Graph frames and GraphX.

An Algorithm for SP distance by hops/roads

1: Start
2: Input the text files
Reading the Nodes file
Reading the Edges file
Reading the HC file
Reading the NB file
3:Array Creation:
Created the array list from HC file
Created the array list from NB file
4: Generation of Graph
Converting the Edges List into Data Frames
Converting the Vertices List into Data Frames
Generated the Graph from above Edges and Vertices
5: Variable Declaration Source Variable, Destination Variable
6: Computer Path from NB to HC
For loop until the Size of NB
For loop until the Size of HC
Assigned the value at index i of NB loop to Source
Assigned the value at index j of HC loop to Destination
If both source and destination is same then distance will be 0 and write into the File.
Else Calculate the BFS from given source to destination vertices and Assigned to Path variable Again If path count equal 0 set the infinity write into file
Else Calculate the total distance store into the variable and write to file.
End of Else
End of Else
End of Inner For Loop
End of Outer For Loop
7: Write the output in text file
8: Stop

4.8 Shortest Path and Distance between POI (NB and HC) using Dijkistrah

In this section we have computed the shortest distance between NB (source) to HC (destination) using Dijkistrah Algorithm using RDDs in Apache Spark. We got total distance between particular NB to HC and full path.

An algorithm to calculate the shortest path:

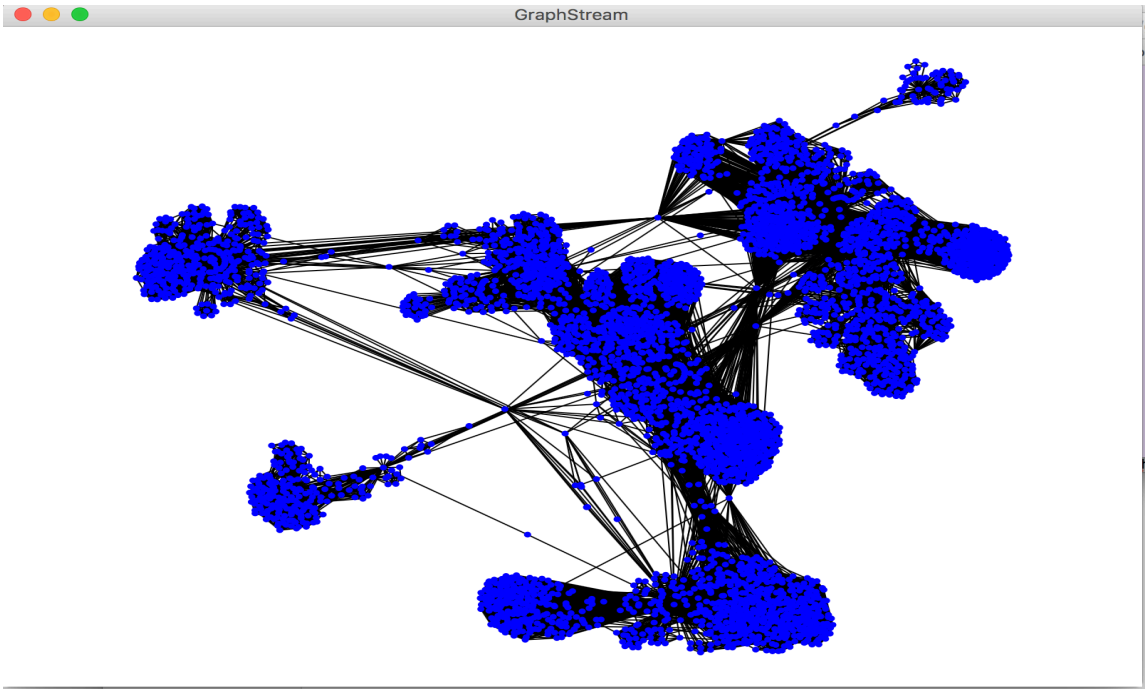
1: Start
2: Add the input the vertices text file
3: Add the input the edges text file
4: Map the vertices from the text file based on the fields in text file and store in vertices RDD.
5: Map the Edges from the text file based on the fields in text store in edges RDD.
6: Generate the Graph from edges and vertices RDDs.
7: Add the input files POIs NB file and HC files
8: Getting the Size of from the NB & HC file
9: Declaration of Matrixes to store the data from NB & HC file.
10: Storing Data into matrix for NB
11: Storing Data into matrix for HC
12: Display the store data
10: Finding the shortest distance formula from NB to HC using Dijkistarh
- **10.1**: Input the sourceId:NB_ID and destinationId:HC_ID and
- **10.2**: Repeat the steps until Inner_loop < size of HC.
- **10.3**: Repeat the steps until Outer_loop < size of POIs
- **10.4**: Initialize the variable Initial Graph if source and destination is same
  return dist
  0 else compute the next step
- **10.5:** Initialize the variable sssp that contains the shortest path list destination and total distance for shortest path.
  var sssp ← initialGraph.pregel(Double.PositiveInfinity)((id, dist,
  newDist) => math.min(dist, newDist),
  triplet => if triplet.srcAttr + triplet.attr < triplet.dstAttr)
  Iterator triplet.dstId, triplet.srcAttr + triplet.attr
  else Iterator.empty,(a, b) => math.min(a, b))
- **10.6:** Output stored in sssp variable
- **10.7**: Store the total distance from NB to HC in dist variable.
- **10.8:** Write the NB, HC and dist into Text File.

11: Stop Inner_loop
12: Stop Outer_loop
13: Stop

4.9 Graph Visualization using GraphX and Graph Stream

An algorithm to visualize the graph

1: Start
2: Import the file that are required for Graph Visualization for graph stream
import org.graphstream.ui.j2dviewer.J2DGraphRenderer
import org.graphstream.graph
import org.graphstream.graph.implementations._
import org.graphstream.graph.EdgeRejectedException
3: Add the input the edges and vertices text file
4: Generate the graph using the vertices and edges file.
5: Set the Path for the CSS file and visual Graph Attributes for the visualization
6: Loading the Vertices from the GraphX to Graph Stream using the method
asInstanceOf[SingleNode]
7: Loading the Edges from the GraphX to Graph Stream using the the method
asInstanceOf [AbstractEdge]
8: Visualize the Graph using function graph.display()
9: Stop



4.10 Graph Visualization of NB and HC using Gephi

In this section we have visualize the NB, HC and other vertices using different colours in Gephi. Generated graph is the obtained by processing "direct displacement algorithms" on the RI data, remember it is not direct generated from mentioned algorithm. After applying this algorithm on the data. Our data look like given below figure 6 and figure 7.

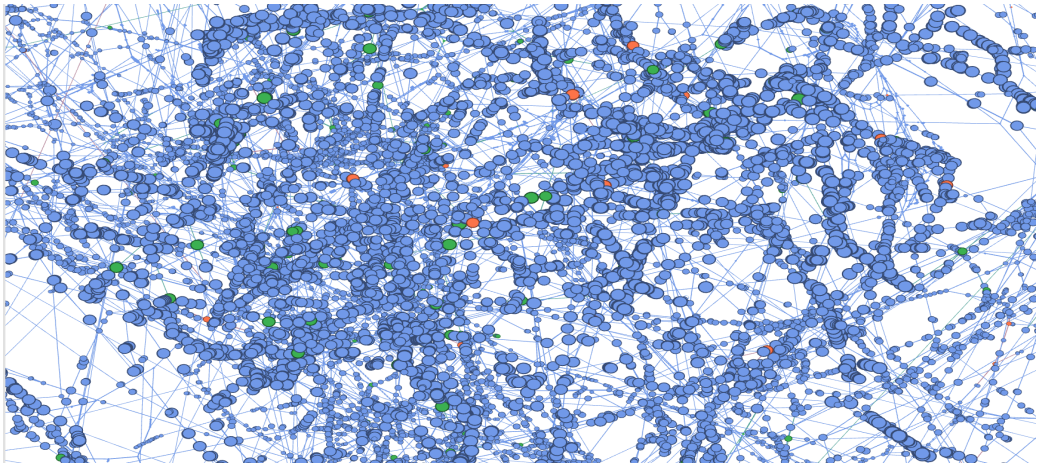


Figure 4.1: Visualization of HC and NB

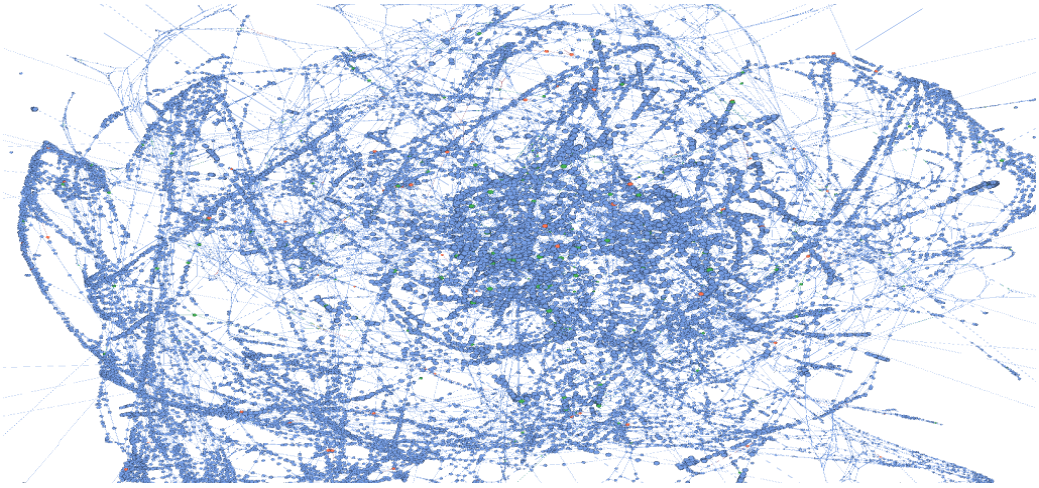


Figure 4.2 Visualization HC and NB

Chapter 5

Load balancing aware Data Locality

5.1 Load Balancing

Load balancing is very important in big data computing due to the large amount of data need to be processed at by many nodes in the cluster, if any node whose computing capacity not less than other nodes in the cluster, it will lead to consumes the large amount of time of other nodes in the cluster. In case of homogenous performance is alike, but in the heterogeneous environment node perforce is quite different because node very the capacity of Computation, Communication, architecture, memories and Power. In case of Power performance Node Will Spend more time of the Whole MapReduce Job is prolonged. Data Skewness takes longer time to finish the task and significant impact on the performance. MapReduce Depends on the slowest running of task in the job, if task takes longer time to finish then other (straggler) it can delay the process of entire job. MapReduce uses the Static Hash function to partition the Data. Straggling cause by several factors such as Fault in hardware, slow machine, and Speculative execution is solution for the above statement but it decrees the job execution time.

Recent studies on this category suggest considering other important issues such as network and data locality. In appropriate partition algorithm may result in poor network quality, overloading some reducer and extension of the execution time of job. Using

appropriate algorithms to process the skew data will lead negative impact on the system performance.

- All Reduce do have same performance.
- Hash partitioning lead to partitioning skew (it brings another problems).
- Join Algorithm takes long time to balance the data cause by the partitioning skew.
- Moving mapping results to Reduce over network cause another extra overhead

5.2 Data Locality

Data locality is another important element in big data computing. Data locality decreases the network traffic and increases the performance. Locality aware resource allocations in data intensive computing application challenges such as data placement, access and computation can be minimized by data locality. Due to distribute file system data locality problem occurs data intensive applications and cloud environment bring new challenges for the data such as computation, assessment and placement. These exiting challenges can be reduce.

Moreover, MapReduce contains the various nodes which heterogeneous in their computing capacity for the various. It is an important for the partitioning algorithm to place the based the capacity of the nodes in clusters.

Data locality is an important factor on the heterogeneous environment. Heterogeneous environment the capacity of the Hard disk is not same. It uses the data locality aware partitioning scheme. Current Hadoop implementation assumes the computing nodes in the cluster are homogenous in nature. Data locality has not been taken into account for launching the speculative map task; it assumes that most maps are

local data. Unfortunately, both homogeneity and data locality assumption is not satisfied in the virtual environment. Ignoring the data locality is decrease the performance of the map reduces.

During literature we seen that some researchers only focusing on the data locality they are ignoring the load balancing, if they are focusing on the load balancing then they are ingoing the data locality. However data locality and load balancing have close relationship both can improve the processing performance of MapReduce in terms of execution and job processing time. If one of them is not there we cannot get the desire improvement in the job execution time. So we are going to propose the new technique that will consider not only data locality but also the load balancing.

5.3 Load Balancing in Spark

Load balancing has great effect on the job processing for big data computing. Spark has adopted the various strategies to balancing the load; these are having various types such input file partitioning, second task parallelization, and third partition strategy.

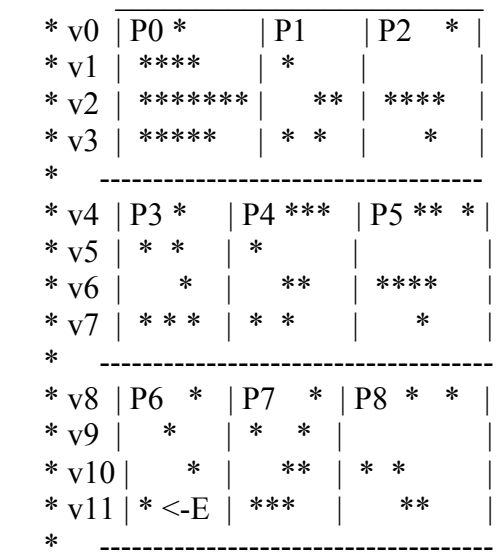
Input file partitioning: In this approach we partitioning the input data file into various partitioning based nodes and cores available on the cluster then we parallelize it for load balancing.

Tasks parallelize: To balance the load on the spark we can also parallelize the task. This work can be achieved by using the scheduler. There are different types scheduler available in the spark we can use them as per our need.

Partition strategy: There are various partitioning strategies in the spark such hash partitioning, graph partitioning etc. For our work we are working with spark, graphx so we are working with different graph partitioning approaches.

Graph partitioning*:* For the processing the graph on the distributed way, graph required the distributed approaches. Normally there are two types of graph partitioning vertex-cut and edge-cut strategies. In vertex cut approach there is randomVertexCut that calculates the hash value of source and destination vertex IDs, using the modulo (by numberOfParts) as the edge's partition ID. The edges partitioned into the same partition two vertices have the same direction. Vertex cut partitioning strategy of the graph also contains various strategies such as "EdgePartitoning2D, EdgePartitoning1D, RandomVertexCut, and CanonicalRandomVertexCut ". In CanonicalRandomVertexCut partitions the edges regardless of the direction another two partitioning schemes are EdgePartition1D and EdgePartition2D. EdgePartition1D, edges are assigned to the partitions only according to their source vertices. A very large prime (mixingPrime) is used in order to balance the partitions. But such operation can't eliminate the problem totally. EdgePartition2D is a bit more complex. It uses both the source vertex and the destination vertex to calculate the partition. It's based on the sparse edge adjacency matrix. Here's an example extracted from the source code of Given Apche Spark GraphX. Suppose we have a graph with 12 vertices that we want to partition over 9 machines.

We can use the following matrix representation:



In above figure we can see that E<v11, v1> is partitioned into P6. But it's also clear that P1 contains too many edges (far more than other partitions) which results in unbalance of partitioning. So mixing Prime is also used in EdgePartition2D. More Over, to load the edge file would actually be to use GraphLoader.edgeListFile (sc, path, minEdgePartitions =16).partitionBy (PartitionStrategy.RandomVertexCut. The exact size of the graph (vertices + edges) in memory depends on the graph's structure, the partition function, and the average vertex degree, because each vertex must be replicated to all partitions where it is referenced. Once we call cache() on the graph, all 16 partitions will be stored in memory.

5.4 Data Locality in Spark

To processing of the Spark jobs data locality have great importance. Both computing the jobs and data together have great effects on processing of the jobs. In another case, if data and code are not together, to improve the job performance we have movie one of them together. Usually data size is greater than code, so it's easily to move

the code in serialize form than data in from chunks. To maintain the data locality spark have its own scheduler for the data locality. Data locality is all about how to place the data close to process it. There are various types of data locality in the spark as these are given in below.

- **Process Local:** It means that running code and JVM are in same location.
- **Node Local:** Node local mean the data same node and computation is also in the same node. It is little bit slower as compare to the process local due data traveling among the process.
- **No Pref:** In this level of the data locality data can be accessed from anywhere with out data locality preference.
- **Rack local:** In rack level data locality is inside the same server's rack. So data can be transfer is on various server can be send through network. On this level of data locality network is overhead.
- **Any:** It mean that data is not on the rack; data is transfer from anywhere else.

Spark prefers to schedule all tasks at the best locality level, but this is not always possible. To achieve the data locality spark schedule the all tasks to achieve the higher level of the data locality, but in reach case is not happen always. So in this case, if there is processed data on the idle executor, spark switches the lower data locality levels. For the CPU there are two choices one wait until CPU frees and began the task on the data, second straightaway start the task and move data there.

5.5 Proposed Algorithm

An algorithm: Load balancing and data locality

```
Input: List of vertices, List of edges, source vertex
Output: list of shortest paths
1:Function main (vertices, edges)
 2:        Locality (local execution, true)
 3:        Locality (locality wait for process, 3S)
 4:        Locality (locality wait for node, 3S)
 5:        locality (locality wait for rack, 3S)
 6:        Nodes List ← path for the input vertices file.
 7:        Edges List ← path for the input vertices file.
 8:        Vertices ← map nodes List
 9:        Edges ← map Edges List
10:        Graph ← create graph from vertices and edges
11:        Graph Partition ← partition graph by partitioning strategy edge partition two
            dimension
12:        Finding the shortest distance formula from using shortest path algorithm
13:        Source Vertex: ← Vertex
14:        initial Graph ← Graph (Double, List(VertexId)) Double] and graph map
Vertices
15:        if (id  sourceId)(0.0,List[VertexId](sourceId))
16:        else (Double.PositiveInfinity, List[VertexId]()))
17:        sssp ← initial graph pregel (double positive Infinity, List(VertexId))
edgedirection)
18:        defined id, dist and newDist
19:        if (dist < newDist) dist else newDist,
20:        triplet ← if (triplet.srcAttr < triplet.dstAttr - triplet.attr )
21:                    Iterator(triplet.dstId, (triplet.srcAttr + triplet.attr ,
                       triplet.srcAttr:+triplet.dstId))
20:        else
21:                    Iterator.empty
22:        (a, b) ← if (a._1 < b._1) a else b)
23:        println(sssp.vertices.collect.mkString("\n"))
```

Proposed Algorithm: partitioning the graph into equal partitions

```
Input: numParts, src, desti
Output: Equally partitioned the data.
1: Function main (numParts, src, desti)
2:        ceilSqrtNumParts: Partition ID ← math.cei l(math.sqrt(numParts)).toInt
3:        mixingPrime: VertexId ← 1125899906842597L
4:        Col: PartitionID ← (math.abs(src * mixingPrime) % ceilSqrtNumParts).toInt
```

5: *Row: PartitionID ← (math.abs(dst * mixingPrime) % ceilSqrtNumParts).toInt*
6: *(Col * ceilSqrtNumParts + row) % numParts*

In above algorithm we have developed new approach for the big data load balancing and data locality. In this algorithm first we are setting for the data locality either it is process, node and rack. In data locality, if any node needs the task or data from other node, it will check first for that either this data is it own process then check the node. Before requesting it to the other node it will wait for 3 seconds, either it available on the node or not. If data or task is not available on the node then it will check with in the rack, similarity again it will wait for 3 seconds. For Load balancing, we have applied graph based partitioning scheme called the two dimension edges partitioning which equally partitioning the graph and distribute among all the nodes in the cluster. After we also applied the shortest path algorithm that was used in our application.

Chapter 6

Results

In this section we shall present the implantation methodology and result that we got from our prosed technique. We are setup the experimental setup will be discussed in detail.

6.1 Descriptions of Dataset for Experiment

We have the DIMACS data, as it describes in chapter no 4, section 4.2, here in this section we are going to describe size of the dataset and nature of the dataset. We took Graph data of USA road network, the Whole USA and Five states of USA, District of Columbia (DC), Rhode Island (RI), Colorado (CO), Florida (FL), California (CAL). The graph data that we have chosen is undirected it has the billion of edges and vertices. Following table 6.1 shows the no of edges and vertices in different states and whole USA.

Table 6.1 USA road network dataset

Name of Road Network	Vertices	Edges	Type
District of Columbia (DC)	9559	14909	Undirected
Rhode Island (RI)	53658	69213	Undirected
Colorado (CO)	435,666	1,057,066	Undirected
Florida (FL)	1,070,376	2,712,798	Undirected
California (CAL)	1,890,815	4,657,742	Undirected
Full USA	23,947,347	58,333,344	Undirected

6.2 Degree distribution and the histogram of vertex degrees of chosen Dataset

In this we have plot the degree distribution and the histogram of vertex degrees of different states and full USA graph based road network dataset. The main purpose of this plot was to see the nature of dataset that we are using in the experiment.

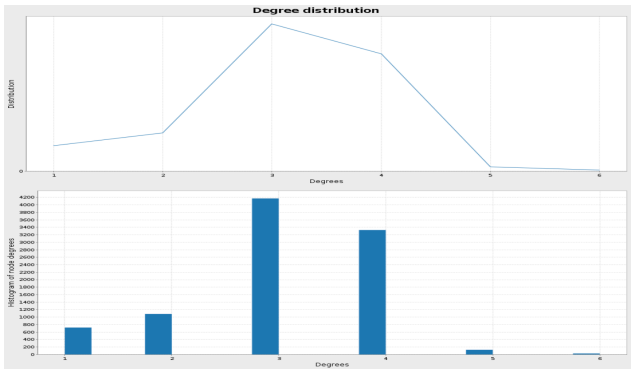


Figure 6.1: DC

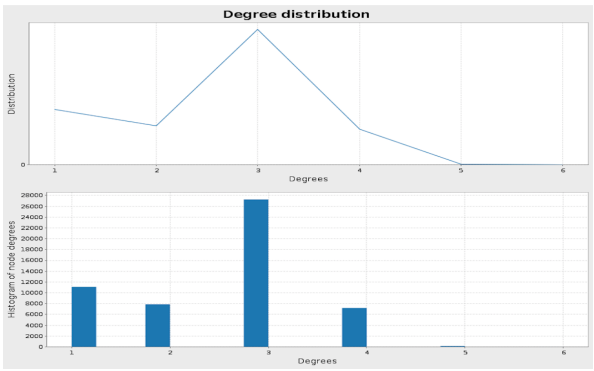


Figure 6.2: RI

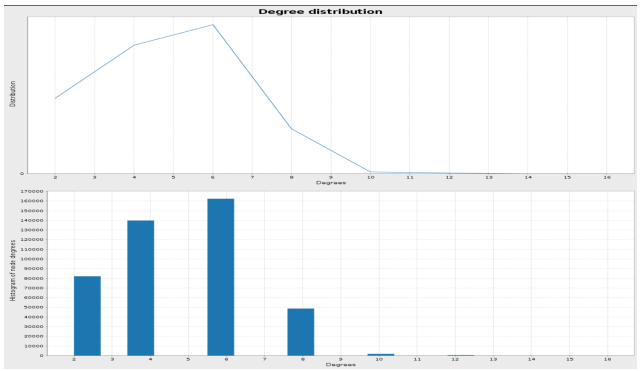


Figure 6.3: CO

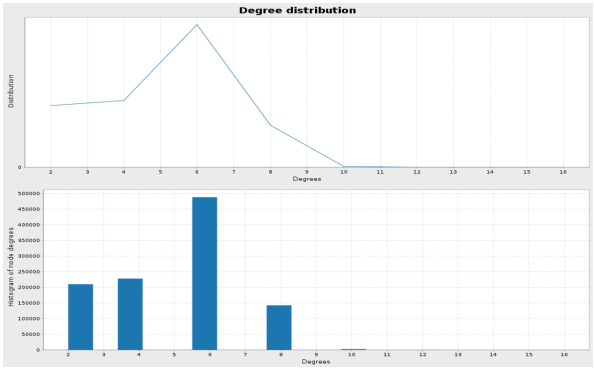


Figure 6.4: FL

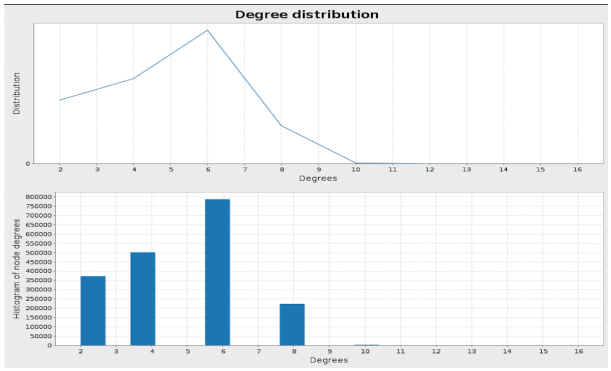


Figure 6.5: CAL

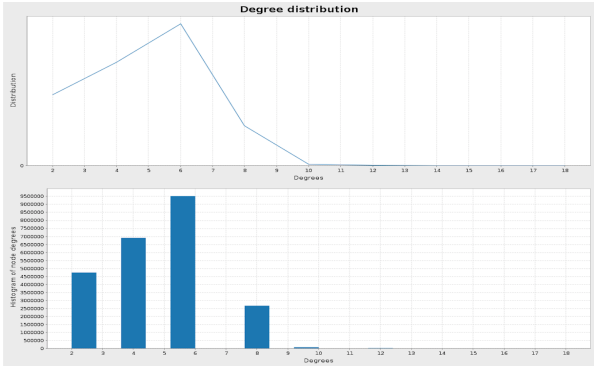


Figure 6.6: Full USA

6.3 Visualization of the graph using the Graph stream and Gephi

In this we have visualization only RI and DC state. In figure 6.7 and figure 6.8 we have visualize the dataset of the DC. In figure 6.9 and figure 6.10 we have visualize the dataset of the RI road network. We cannot able to visualize the other sates data because it's a so huge cannot able to handle on the single standalone PC. We have just visualized the only two sates to see how our dataset look likes.

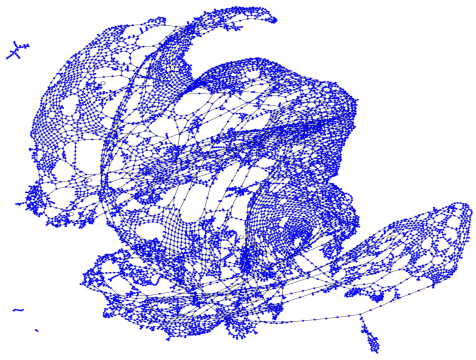


Figure 6.7: DC road network using graph stream

Figure 6.8: DC road network using Gephi

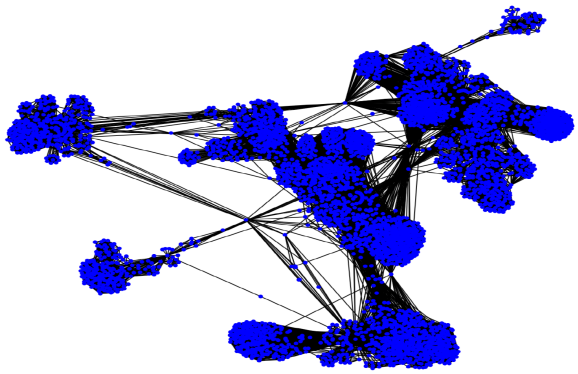


Figure 6.9: RI road network using graph stream

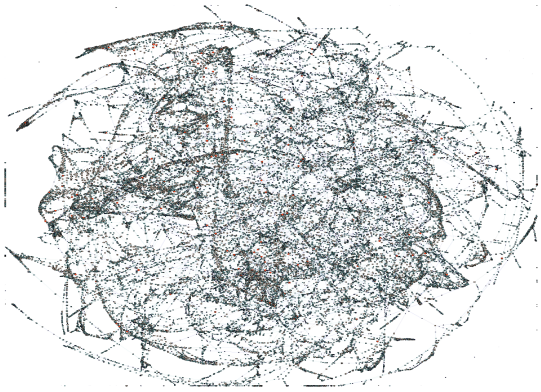


Figure 6.10: RI road network using the Gephi

6.4 Implementation Methodology and Design Tools

To implement our proposed technique we have build the spark cluster setup using the Aziz super computer. In this setup we have used the different Aziz nodes, as it varies for the same dataset for 1, 2,4,8 and 16 Aziz nodes.

6.5 Experimental Setup

For experimental setup we have use the Aziz Super computer. We have used Apache Hadoop HDFS as input and processed the data in apache spark cluster. Rest of software and hardware configuration is given below in the table.

Table 6.2 Configuration environment

The node type	**Master**	**Slave**
Software and Hardware environment	Linux centOS, JDK 1.7, Processor 2.4 GHz, Apache Spark 2.0.2, 24 cores, Apache Hadoop HDFS.	Linux centOS, JDK 1.7, Processor 2.4 GHz, 24 cores, Apache Spark 2.0.1, Apache Hadoop HDFS.
Memory	94G	94G per slave
Quantity	1	Different slave as 1,2,4,8 and 16.

6.6 Results

We implemented the parallel and nonparallel code on the spark cluster using the Aziz super computer. We write down the timing for number nodes processed in the particular time. We have run locally on the Aziz for all different sates RI, DC, CO, CAL, FL and whole USA. Similarly, we run the data for the 1, 2, 4,8 and 16 Aziz node using the spark cluster and note down the timing of each. As shown on the figure 6.11 comparison of different USA states dataset, from this figure 6.11 we have seen, as data is small we can get less parallelism and it will also take more time. But we will small dataset to higher will get more parallelism and it will also take the less time. Another, thing we also noted the as move for the higher size of data we with more node we will get more parallelism and take less time to process the data.

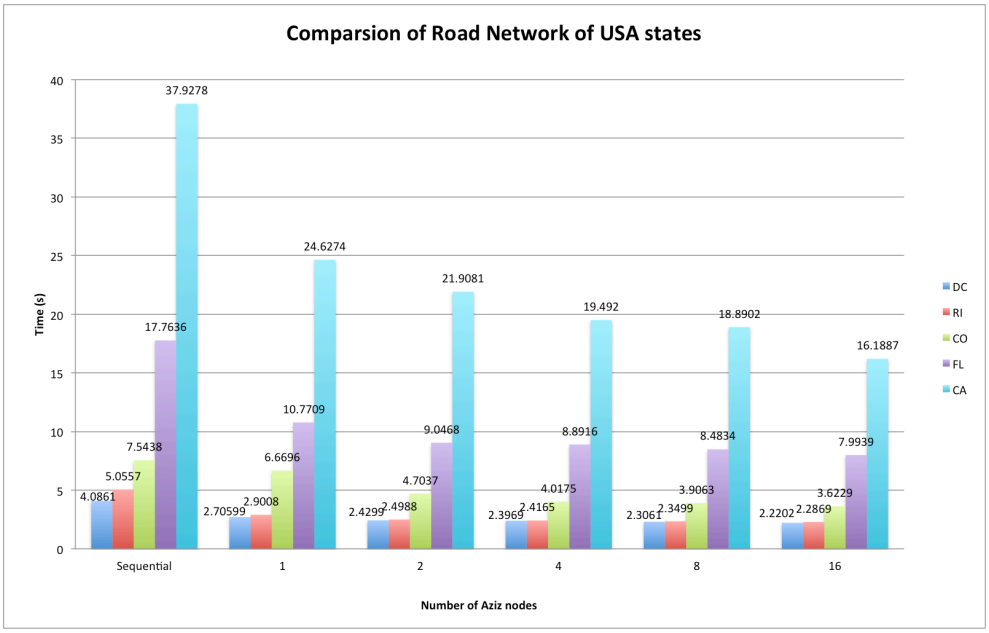


Figure 6.11: Performance comparisons of five different states on the different Aziz nodes in Spark Cluster

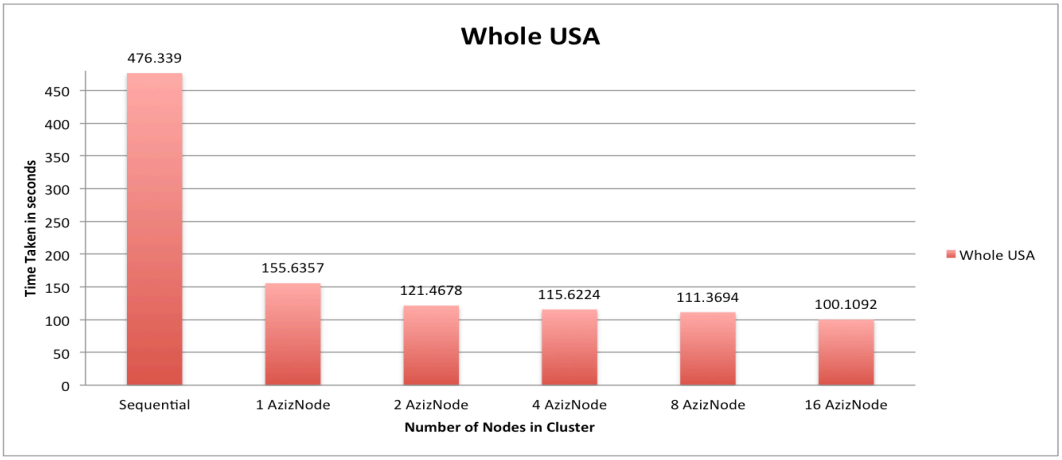


Figure 6.12: Performance comparisons different Aziz nodes in Spark Cluster of whole USA road network

Chapter 7

Conclusion and Future Work

7.1 Discussion

Big data is the data that cannot process by the traditional tools and platforms. This data is ranges from the terabyte to yottabyte. There are different characteristics of big data such as volume, velocity and variety. It also have various sources of such data social media, Facebook twitter etc. There are various tools has been introduced to process such data but these tool have various challenges and issues. In our thesis, we explored these challenges and issues for processing of big data. In this review, our contribution is five fold. First of all we have find out the challenges and issues on the big data platforms. Secondly, we have made the taxonomy of the proposed techniques against the challenges and issues. Thirdly, we have also listed the applications of big data. Fourthly, we have listed the all platforms for processing of big data with features and limitations. Fifthly, we have also classified these platforms according to the capability of the processing in various layers. At last we have proposed the new technique for the loadbalcing and data locality for the graph applications.

7.2 Conclusion

In this thesis, a detailed review of the literature is presented to identify major challenges in big data research. These challenges include, among others, load balancing and data locality. A load balancing technique that is also aware of data locality has been developed and is applied to a graph-based road transportation problem. We have modeled the entire US road network data that contains approximately 24 million vertices

and 58 million arcs; the specific aim of this research is to identify various points of interests (PoIs) in the regions including living places and healthcare centres. These PoIs are subsequently used to find the shortest paths among them for planning and operations purposes. The relevant algorithms that we have developed are presented in the thesis. The algorithms have been implemented using the Spark big data platform tools on the Aziz supercomputer, a Top500 supercomputer in the world according to the June and November 2015 rankings. The results are collected in terms of the shortest paths and the road networks are visualised as graphs. The performance of the load balancing and data locality awareness techniques is analysed against a varying number of nodes on the Aziz supercomputer and a good speedup has been reported.

7.3 Future Work

For future application work, there are various areas of big data that needs more attentions such as bioinformatics, big data analytics, big data management and machine learning, as these all areas has different challenges for the big data computing. Future work will focus on improving the data locality aware load balancing algorithm and extension of our work to complex planning and operations problems. We will also concentrate the on big data analysts with machine learning.

LIST OF REFERENCES

[1] "Welcome to Apache™ Hadoop®! - index.pdf." [Online]. Available: https://hadoop.apache.org/index.pdf. [Accessed: 29-Mar-2016].

[2] "HDFS Architecture Guide." [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html#Introduction. [Accessed: 29-Mar-2016].

[3] "Index of /docs/r2.6.1/api/org/apache/hadoop/mapreduce." [Online]. Available: https://hadoop.apache.org/docs/r2.6.1/api/org/apache/hadoop/mapreduce/. [Accessed: 29-Mar-2016].

[4] D. Singh and C. K. Reddy, "A survey on platforms for big data analytics," *J. Big Data*, vol. 2, no. 1, p. 8, 2014.

[5] A. C. Murthy, "Apache Hadoop YARN," 2014. [Online]. Available: http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html. [Accessed: 29-Mar-2016].

[6] "Posts about HADOOP on Business Intelligence Technology Trends." [Online]. Available: https://businessintelligencetechnologytrend.wordpress.com/tag/hadoop/. [Accessed: 30-May-2016].

[7] "Welcome to Apache Pig!," *Apache Software Foundation*, 2012. [Online]. Available: https://pig.apache.org/. [Accessed: 30-Mar-2016].

[8] Hive, "Apache Hive TM," 2015. [Online]. Available: http://hive.apache.org/. [Accessed: 30-Mar-2016].

[9] J. Manning, "Apache Storm," 2004. [Online]. Available: http://storm.apache.org/. [Accessed: 30-Mar-2016].

[10] Apache Spark, "Apache Spark™ - Lightning-Fast Cluster Computing," *Spark.Apache.Org*, 2015. [Online]. Available: https://spark.apache.org/. [Accessed: 30-Mar-2016].

[11] Apache, "Spark Streaming," 2015. [Online]. Available: http://spark.apache.org/streaming/. [Accessed: 30-Mar-2016].

[12] S. Agarwal, A. Panda, B. Mozafari, S. Madden, I. Stoica, A. Panda, H. Milner, S. Madden, I. Stoica, B. Mozafari, S. Madden, I. Stoica, and U. C. Berkeley, "BlinkDB : Queries with Bounded Errors and Bounded Response Times on Very Large Data," *Proceedings of the 8th ACM European Conference on Computer Systems - EuroSys '13*, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2465351.2465355\nhttp://arxiv.org/abs/1203.5485. [Accessed: 30-Mar-2016].

[13] "Software | AMPLab – UC Berkeley." [Online]. Available: https://amplab.cs.berkeley.edu/software/. [Accessed: 30-Mar-2016].

[14] B. Hindman, "Apache Mesos." [Online]. Available: http://mesos.apache.org/. [Accessed: 30-Mar-2016].

[15] I. Li, Haoyuan and Ghodsi, Ali and Zaharia, Matei and Baldeschwieler, Eric and Shenker, Scott and Stoica, "Tachyon: Memory Throughput I/O for Cluster Computing Frameworks," *Memory*, vol. 18, p. 1, 2013.
[16] Ampl. at U. Berkeley, "ML Base." [Online]. Available: http://www.mlbase.org.
[17] "MLlib | Apache Spark." [Online]. Available: http://spark.apache.org/mllib/. [Accessed: 30-Mar-2016].
[18] Apache Software Foundation, "Spark SQL & DataFrames | Apache Spark." [Online]. Available: https://spark.apache.org/sql/. [Accessed: 30-Mar-2016].
[19] "GraphX | Apache Spark." [Online]. Available: http://spark.apache.org/graphx/. [Accessed: 30-Mar-2016].
[20] MongoDB Inc., "MongoDB for GIANT Ideas | MongoDB." [Online]. Available: https://www.mongodb.com/. [Accessed: 30-Mar-2016].
[21] J. Chambers and Bell Laboratories, "What is R?," 2000. [Online]. Available: http://www.r-project.org/. [Accessed: 29-Mar-2016].
[22] "Data Dryad," 2016. [Online]. Available: http://datadryad.org/.
[23] "Syncfusion Big Data Platform | Big Data Platform simplifies working with Hadoop on Windows." [Online]. Available: https://www.syncfusion.com/products/big-data. [Accessed: 30-Mar-2016].
[24] "Cloudera." [Online]. Available: http://www.cloudera.com/. [Accessed: 30-Mar-2016].
[25] "active-active-deployments-with-cloudera-enterprise-whitepaper.pdf." [Online]. Available: http://www.cloudera.com/content/dam/www/static/documents/whitepapers/active-active-deployments-with-cloudera-enterprise-whitepaper.pdf. [Accessed: 30-Mar-2016].
[26] P. Software, "Pivotal HDB | Big Data." Pivotal Software, Inc., 13-Feb-2015.
[27] NexisNexis Risk Solutions, "HPCC Systems," 2012. [Online]. Available: http://hpccsystems.com/. [Accessed: 30-Mar-2016].
[28] S. Sagiroglu and D. Sinanc, "Big data: A review," *Int. Conf. Collab. Technol. Syst.*, pp. 42–47, 2013.
[29] Amazon Web Services, "Amazon Web Services (AWS) - Cloud Computing Services," 2015. [Online]. Available: http://aws.amazon.com/. [Accessed: 30-Mar-2016].
[30] "Neo4j: The World's Leading Graph Database." [Online]. Available: http://neo4j.com/. [Accessed: 29-Mar-2016].
[31] "Pentaho | Data Integration, Business Analytics and Big Data Leaders." [Online]. Available: http://www.pentaho.com/. [Accessed: 29-Mar-2016].
[32] "Talend Open Source Data Integration Software." [Online]. Available: https://www.talend.com/. [Accessed: 29-Mar-2016].
[33] "Business Intelligence and Analytics | Tableau Software." [Online]. Available: http://www.tableau.com/. [Accessed: 29-Mar-2016].
[34] "IBM big data platform - Bringing big data to the Enterprise." IBM Corporation, 09-Feb-2016.
[35] "Big Data | SAS." [Online]. Available: http://www.sas.com/en_us/insights/big-data.html. [Accessed: 29-Mar-2016].
[36] "Big Data | What is Big Data? | Oracle." [Online]. Available:

https://www.oracle.com/big-data/index.html. [Accessed: 29-Mar-2016].
[37] "Big Data, Big Data Beyond the Hype and Big Data Successes | Teradata." [Online]. Available: http://bigdata.teradata.com/. [Accessed: 29-Mar-2016].
[38] K. R. and B. T. Rao, "Information Systems Design and Intelligent Applications," *Adv. Intell. Syst. Comput.*, vol. 435, pp. 19–26, 2016.
[39] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, "Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling," *Proc. 5th Eur. Conf. Comput. Syst.*, pp. 265–278, 2010.
[40] M. Hammoud and M. F. Sakr, "Locality-aware reduce task scheduling for mapreduce," *Proc. - 2011 3rd IEEE Int. Conf. Cloud Comput. Technol. Sci. CloudCom 2011*, pp. 570–576, 2011.
[41] Z. Guo, G. Fox, and M. Zhou, "Investigation of data locality in MapReduce," *Proc. - 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGrid 2012*, pp. 419–426, 2012.
[42] M. Y. Eltabakh, Y. Tian, F. Özcan, R. Gemulla, A. Krettek, and J. McPherson, "CoHadoop: flexible data placement and its exploitation in Hadoop," *Proc. VLDB Endow.*, vol. 4, no. 9, pp. 575–585, 2011.
[43] L. Wang, J. Tao, R. Ranjan, H. Marten, A. Streit, J. Chen, and D. Chen, "G-Hadoop : MapReduce across distributed data centers for data-intensive computing," *Futur. Gener. Comput. Syst.*, vol. 29, no. 3, pp. 739–750, 2013.
[44] C.-H. Hsu, K. D. Slagter, and Y.-C. Chung, "Locality and loading aware virtual machine mapping techniques for optimizing communications in MapReduce applications," *Futur. Gener. Comput. Syst.*, vol. 53, pp. 43–54, 2015.
[45] X. Yu and B. Hong, "Grouping Blocks for MapReduce Co-Locality," *2015 IEEE Int. Parallel Distrib. Process. Symp.*, pp. 271–280, 2015.
[46] R. Gu, X. Yang, J. Yan, Y. Sun, B. Wang, C. Yuan, and Y. Huang, "SHadoop: Improving MapReduce performance by optimizing job execution mechanism in Hadoop clusters," *J. Parallel Distrib. Comput.*, vol. 74, no. 3, pp. 2166–2179, 2014.
[47] Y. Chen, Z. Liu, T. Wang, and L. Wang, "Load Balancing in MapReduce Based on Data Locality," *Algorithms Archit. Parallel Process. Ica3pp 2014, Pt I*, vol. 8630, pp. 229–241, 2014.
[48] Y. Kwon, M. Balazinska, B. Howe, and J. Rolia, "A Study of Skew in MapReduce Applications."
[49] Y. Xu, P. Kostamaa, X. Zhou, and L. Chen, "Handling data skew in parallel joins in shared-nothing systems," *Proc. 2008 ACM SIGMOD Int. Conf. Manag. data - SIGMOD '08*, p. 1043, 2008.
[50] Y. Xu and P. Kostamaa, "Efficient outer join data skew handling in parallel DBMS," *Proc. VLDB Endow.*, vol. 2, no. 2, pp. 1390–1396, 2009.
[51] K. Kc and V. W. Freeh, "Dynamically Controlling Node-Level Parallelism in Hadoop," *2015 IEEE 8th Int. Conf. Cloud Comput.*, pp. 309–316, 2015.
[52] I. Palit and C. K. Reddy, "Scalable and parallel boosting with mapReduce," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 10, pp. 1904–1916, 2012.
[53] J. Yin and J. Wang, "Optimize Parallel Data Access in Big Data Processing," *2015 15th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput.*, pp. 721–724, 2015.
[54] L. S. Perkins, P. Andrews, D. Panda, D. Morton, R. Bonica, N. Werstiuk, and R.

Kreiser, “A Survey of Load Balancing Techniques for Data Intensive Computing,” *2009 Int. Symp. Collab. Technol. Syst. CTS 2009*, vol. 41, no. 4, pp. c1–c1, 2011.

[55] A. Ajitha and D. Ramesh, “Improved task graph-based parallel data processing for dynamic resource allocation in cloud,” *Procedia Eng.*, vol. 38, pp. 2172–2178, 2012.

[56] S. Nishanth, B. Radhikaa, T. J. Ragavendar, C. Babu, and B. Prabavathy, “CoHadoop ++ : A Load Balanced Data Co - location in Radoop Distributed File System,” *Proc. - 2013 5th Int. Conf. Adv. Comput.*, pp. 100–105, 2013.

[57] Y. Xu, W. Qu, Z. Li, Z. Liu, C. Ji, Y. Li, and H. Li, “Balancing reducer workload for skewed data using sampling-,” *Comput. Electr. Eng.*, vol. 40, no. 2, pp. 675–687, 2014.

[58] Q. Chen, J. Yao, and Z. Xiao, “LIBRA: Lightweight Data Skew Mitigation in MapReduce,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 9219, no. c, pp. 1–1, 2014.

[59] I. Workshop, S. Technology, B. Beijing, S. Technology, and B. Beijing, “Load Balancing Solution Based on AHP for Hadoop Huixiang Zhou,” pp. 633–636, 2014.

[60] Z. Gao, D. Liu, Y. Yang, J. Zheng, and Y. Hao, “A load balance algorithm based on nodes performance in Hadoop cluster,” *APNOMS 2014 - 16th Asia-Pacific Netw. Oper. Manag. Symp.*, pp. 1–4, 2014.

[61] Z. Fadika, E. Dede, J. Hartog, and M. Govindaraju, “MARLA: MapReduce for heterog[1] Z. Fadika, E. Dede, J. Hartog, and M. Govindaraju, ‘MARLA: MapReduce for heterogeneous clusters,’ Proc. - 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGrid 2012, pp. 49–56, 2012.eneous clusters,” *Proc. - 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. CCGrid 2012*, pp. 49–56, 2012.

[62] Y. Wang and W. L. Croft, “Smart Shuffling in MapReduce : a solution to Balance Network Traffic and Workloads .,” 2015.

[63] J. X. J. Xie, S. Y. S. Yin, X. R. X. Ruan, Z. D. Z. Ding, Y. T. Y. Tian, J. Majors, A. Manzanares, and X. Q. X. Qin, “Improving MapReduce performance through data placement in heterogeneous Hadoop clusters,” *Parallel & Distrib. Process. Work. Phd Forum (IPDPSW), 2010 IEEE Int. Symp.*, vol. 9, pp. 29–42, 2010.

[64] R. M. Arasanal and D. U. Rumani, “Improving mapreduce performance through complexity and performance based data placement in heterogeneous hadoop clusters,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7753 LNCS, pp. 115–125, 2013.

[65] C. W. Lee, K. Y. Hsieh, S. Y. Hsieh, and H. C. Hsiao, “A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments,” *Big Data Res.*, vol. 1, pp. 14–22, 2014.

[66] S. Sujitha and S. Jaganathan, “Aggrandizing Hadoop in terms of node Heterogeneity & Data Locality,” *2013 IEEE Int. Conf. "Smart Struct. Syst. ICSSS 2013*, pp. 145–151, 2013.

[67] X. Fan, X. Ma, J. Liu, and D. Li, “Dependency-aware data locality for MapReduce,” *IEEE Int. Conf. Cloud Comput. CLOUD*, pp. 408–415, 2014.

[68] M. Khan, Y. Liu, and M. Li, “Data locality in Hadoop cluster systems,” *2014 11th*

Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2014, pp. 720–724, 2014.

[69] L. Li, Z. Tang, R. Li, and L. Yang, "New improvement of the Hadoop relevant data locality scheduling algorithm based on LATE," *Proc. 2011 Int. Conf. Mechatron. Sci. Electr. Eng. Comput. MEC 2011*, pp. 1419–1422, 2011.

[70] V. Ubarhande, A.-M. Popescu, and H. Gonzalez-Velez, "Novel Data-Distribution Technique for Hadoop in Heterogeneous Cloud Environments," *2015 Ninth Int. Conf. Complex, Intelligent, Softw. Intensive Syst.*, pp. 217–224, 2015.

[71] X. Huang, L. Zhang, R. Li, L. Wan, and K. Li, "Novel heuristic speculative execution strategies in heterogeneous distributed environments," *Comput. Electr. Eng.*, vol. 0, pp. 1–14, 2015.

[72] G. P. M S, N. H R, and S. Prabhu, "Performance Analysis of Schedulers to Handle Multi Jobs in Hadoop Cluster," *Int. J. Mod. Educ. Comput. Sci.*, vol. 7, no. 12, pp. 51–56, 2015.

[73] and D. R. Sethi, Krishan Kumar, "Delay Scheduling with Reduced Workload on JobTracker in Hadoop," *Innov. Bio-Inspired Comput. Appl.*, vol. 424, 2016.

[74] M. Sun, H. Zhuang, C. Li, K. Lu, and X. Zhou, "Scheduling algorithm based on prefetching in MapReduce clusters," *Appl. Soft Comput.*, vol. 38, pp. 1–10, 2015.

[75] G. S. Sadasivam and D. Selvaraj, "A novel parallel hybrid PSO-GA using MapReduce to schedule jobs in Hadoop data grids," *Proc. - 2010 2nd World Congr. Nat. Biol. Inspired Comput. NaBIC 2010*, pp. 377–382, 2010.

[76] J. Wang, Q. Xiao, J. Yin, and P. Shang, "DRAW: A new data-grouping-aware data placement scheme for data intensive applications with interest locality," *IEEE Trans. Magn.*, vol. 49, no. 6, pp. 2514–2520, 2013.

[77] S. Lee, S. R. Sukumar, S. Hong, and S.-H. Lim, "Enabling graph mining in RDF triplestores using SPARQL for holistic in-situ graph analysis," *Expert Syst. Appl.*, vol. 48, pp. 9–25, 2016.

[78] Y. Xia, I. G. Tanase, L. Nai, W. Tan, Y. Liu, J. Crawford, and C. Lin, "Explore Efficient Data Organization for Large Scale Graph Analytics and Storage," *Proc. 2014 IEEE BigData Conf.*, pp. 942–951, 2014.

[79] G. Malewicz, M. H. Austern, A. J. . Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: a system for large-scale graph processing," *Proc. 2010 Int. Conf. Manag. data - SIGMOD '10*, pp. 135–146, 2010.

[80] C. Fang, C. A. Secondary, C. Author, C. Fang, J. Liu, N. Ansari, and C. Fang, "Wireless Networks Revealing Connectivity Structural Patterns among Web Objects based on Co- clustering of Bipartite Request Dependency Graph Revealing Connectivity Structural Patterns among Web Objects based on Co-clustering of Bipartite Request Dependenc," *Under Rev.*

[81] R. Xue, S. Gao, L. Ao, and Z. Guan, "BOLAS: Bipartite-graph oriented locality-aware scheduling for mapreduce tasks," *Proc. - IEEE 14th Int. Symp. Parallel Distrib. Comput. ISPDC 2015*, pp. 37–45, 2015.

[82] D. Orozco, E. Garcia, and G. Gao, "Locality optimization of stencil applications using data dependency graphs," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6548 LNCS, pp. 77–91, 2011.

[83] Y. Hassanzadeh-Nazarabadi, A. Küpçü, and Ö. Özkasap, "Locality aware skip graph," *Proc. - 2015 IEEE 35th Int. Conf. Distrib. Comput. Syst. Work. ICDCSW 2015*, pp. 105–111, 2015.

[84] M. Kandemir, A. Choudhary, J. Ramanujam, and P. Banerjee, “A graph based framework to detect optimal memory layouts for improving data locality,” *Ipps*, p. 738, 1999.
[85] A. Chernov, A. Belevantsev, and O. Malikov, “A thread partitioning algorithm for data locality improvement,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3019, pp. 278–285, 2004.
[86] Y. M. Zhang, K. Huang, G. Geng, and C. L. Liu, “Fast kNN graph construction with locality sensitive hashing,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8189 LNAI, no. PART 2, pp. 660–674, 2013.
[87] M. Zhang, F. Shen, H. Zhang, N. Xie, and W. Yang, “Fast Graph Similarity Search via Locality Sensitive Hashing,” *Adv. Multimed. Inf. Process. -- PCM 2015*, vol. 9315, pp. 447–455, 2015.
[88] P. Yuan, C. Xie, L. Liu, H. Jin, and S. Member, “PathGraph : A Path Centric Graph Processing System,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 9219, no. c, pp. 1–15, 2016.
[89] Y. Shao, B. Cui, and L. Ma, “PAGE: A partition aware engine for parallel graph computation,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 518–530, 2015.
[90] L. Qin, R.-H. Li, L. Chang, and C. Zhang, “Locally Densest Subgraph Discovery,” *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD ’15*, pp. 965–974, 2015.
[91] E. Zamanian, C. Binnig, and A. Salama, “Locality-aware Partitioning in Parallel Database Systems,” *SIGMOD*, pp. 17–30, 2015.
[92] R. Chen, M. Yang, X. Weng, B. Choi, B. He, and X. Li, “Improving Large Graph Processing on Partitioned Graphs in the Cloud,” *Proc. Third ACM Symp. Cloud Comput. - SoCC ’12*, pp. 1–13, 2012.
[93] Z. Zeng, B. Wu, and H. Wang, “A parallel graph partitioning algorithm to speed up the large-scale distributed graph mining,” *Proc. 1st Int. Work. Big Data, Streams Heterog. Source Min. Algorithms, Syst. Program. Model. Appl. - BigMine ’12*, pp. 61–68, 2012.
[94] K. Lee and L. Liu, “Efficient data partitioning model for heterogeneous graphs in the cloud,” *Proc. Int. Conf. High Perform. Comput. Networking, Storage Anal.*, pp. 1–12, 2013.
[95] M. LeBeane, S. Song, R. Panda, J. H. Ryoo, and L. K. John, “Data partitioning strategies for graph workloads on heterogeneous clusters,” *Proc. Int. Conf. High Perform. Comput. Networking, Storage Anal. - SC ’15*, pp. 1–12, 2015.
[96] R. Chen, J. Shi, Y. Chen, and H. Chen, “PowerLyra:Differentiated Graph Computation and Partitioning on Skewed Graphs,” *Proc. Tenth Eur. Conf. Comput. Syst. - EuroSys ’15*, pp. 1–15, 2015.
[97] N. Xu, L. Chen, and B. Cui, “LogGP:A Log-based Dynamic Graph Partitioning Method,” *Proc. VLDB Endow.*, vol. 7, no. 14, pp. 1917–1928, 2014.
[98] S. Yang, X. Yan, B. Zong, and A. Khan, “Towards effective partition management for large graphs,” *Proc. 2012 Int. Conf. Manag. Data - SIGMOD ’12*, pp. 517–528, 2012.
[99] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, and J. Al-jaroodi, “Applications of big data to smart cities,” *J. Internet Serv. Appl.*, 2015.

[100] C. Selvi, C. Ahuja, and E. Sivasankar, “Intelligent Computing and Applications,” *Adv. Intell. Syst. Comput.*, vol. 343, pp. 367–379, 2015.
[101] G. Suciu, V. Suciu, A. Martian, R. Craciunescu, A. Vulpe, I. Marcu, S. Halunga, and O. Fratu, “Big Data, Internet of Things and Cloud Convergence ??? An Architecture for Secure E-Health Applications,” *J. Med. Syst.*, vol. 39, no. 11, 2015.
[102] B. Collins, “Big Data and Health Economics: Strengths, Weaknesses, Opportunities and Threats,” *Pharmacoeconomics*, vol. 34, no. 2, pp. 101–106, 2015.
[103] W. Raghupathi and V. Raghupathi, “Big data analytics in healthcare: promise and potential,” *Heal. Inf. Sci. Syst.*, vol. 2, p. 3, 2014.
[104] R. Mehmood and G. Graham, “Big Data Logistics: A health-care Transport Capacity Sharing Model,” *Procedia Comput. Sci.*, vol. 64, pp. 1107–1114, 2015.
[105] T. Huang, L. Lan, X. Fang, P. An, J. Min, and F. Wang, “Promises and Challenges of Big Data Computing in Health Sciences,” *Big Data Res.*, vol. 2, no. 1, pp. 2–11, 2015.
[106] M. Barkhordari and M. Niamanesh, “ScaDiPaSi: An Effective Scalable and Distributable MapReduce-Based Method to Find Patient Similarity on Huge Healthcare Networks,” *Big Data Res.*, vol. 2, no. 1, pp. 19–27, 2015.
[107] A. W. Toga and I. D. Dinov, “Sharing big biomedical data,” *J. Big Data*, vol. 2, no. 1, p. 7, 2015.
[108] C. Wang, X. Li, P. Chen, A. Wang, X. Zhou, and H. Yu, “Heterogeneous cloud framework for big data genome sequencing,” *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 12, no. 1, pp. 166–178, 2015.
[109] A. Jaiswal and A. Upadhyay, “An Enhanced Framework of Genomics Using Big Data Computing,” 2015.
[110] Y. Qin, H. K. Yalamanchili, J. Qin, B. Yan, and J. Wang, “The current status and challenges in computational analysis of genomic big data,” *Big Data Res.*, vol. 2, no. 1, pp. 12–18, 2015.
[111] J. Davis, G. Olsen, R. Overbeek, V. Vonstein, and F. Xia, “In search of genome annotation consistency: solid gene clusters and how to use them,” *3 Biotech*, pp. 1–5, 2013.
[112] H. Yeo and C. H. Crawford, “Big Data : Cloud Computing in Genomics Applications,” pp. 2904–2906, 2015.
[113] P. Heinzlreiter, M. T. Krieger, and I. Leitner, “Hadoop-based genome comparisons,” *Proc. - 2nd Int. Conf. Cloud Green Comput. 2nd Int. Conf. Soc. Comput. Its Appl. CGC/SCA 2012*, pp. 695–701, 2012.
[114] Y.-H. Liang and S.-Y. Wu, “Sequence-Growth: A Scalable and Effective Frequent Itemset Mining Algorithm for Big Data Based on MapReduce Framework,” *2015 IEEE Int. Congr. Big Data*, pp. 393–400, 2015.
[115] S. Dodson, D. O. Ricke, and J. Kepner, “Genetic sequence matching using D4M big data approaches,” *2014 IEEE High Perform. Extrem. Comput. Conf. HPEC 2014*, pp. 0–5, 2014.
[116] M. Phinney, H. Cao, A. Dhroso, and C. Shyu, “Ecosystem,” pp. 4–6.
[117] S.-H. T. S.-H. Toh, H.-J. L. H.-J. Lee, and K.-H. D. K.-H. Do, “Basic Sequence Search by Hashing Algorithm in DNA Sequence Databases,” *2009 11th Int. Conf.*

Adv. Commun. Technol., vol. 3, pp. 2317–2320, 2009.
[118] M. Meng, J. Gao, and J. J. Chen, "Blast-Parallel: The parallelizing implementation of sequence alignment algorithms based on Hadoop platform," *Proc. 2013 6th Int. Conf. Biomed. Eng. Informatics, BMEI 2013*, no. Bmei, pp. 465–470, 2013.
[119] A. O'Driscoll, V. Belogrudov, J. Carroll, K. Kropp, P. Walsh, P. Ghazal, and R. D. Sleator, "HBLAST: Parallelised sequence similarity - A Hadoop MapReducable basic local alignment search tool," *J. Biomed. Inform.*, vol. 54, pp. 58–64, 2015.
[120] S. M. Sait, M. Al-Mulhem, and R. Al-Shaikh, "Evaluating BLAST runtime using NAS-based high performance clusters," *Proc. - CIMSim 2011 3rd Int. Conf. Comput. Intell. Model. Simul.*, pp. 51–56, 2011.
[121] G. M. Boratyn, A. a Schäffer, R. Agarwala, S. F. Altschul, D. J. Lipman, and T. L. Madden, "Domain enhanced lookup time accelerated BLAST," *Biol. Direct*, vol. 7, no. 1, p. 12, 2012.
[122] "TIGER/Line® Shapefiles (United States Census Bureau)," 2016. [Online]. Available: http://www.census.gov/cgi-bin/geo/shapefiles/index.php. [Accessed: 04-Aug-2016].
[123] "Metro Extracts · Open Street Map," 2016. [Online]. Available: https://mapzen.com/data/metro-extracts/. [Accessed: 09-Aug-2016].
[124] "DIMACS Implementation Challenge," *May 3, 2002*. [Online]. Available: http://www.dis.uniroma1.it/challenge9/data/tiger/. [Accessed: 22-Aug-2016].