ISSN (Online): 2456-5717

International Journal of Advanced Research in Basic Engineering Sciences and Technology (IJARBEST)
Vol.3 Issue.6 June 2017

Framework for Efficient Big Data Broadcasting using B+Tree Algorithm

Priyanka.S[1], T.Sheik Yousuf[2]
[1]PG Student, Department of Computer Science Engineering, Mohamed Sathak Engineering College, kilakarai.
[2]Associate Professor, Department of Computer Science Engineering, Mohamed Sathak Engineering College, kilakarai.

Abstract– **Over the last few years' big data broad casting is the new major challenging issue for ICT industry, big data computing faces the various challenging issues for broad casting such as computational complexity. Large amount of data process in ICT (Information and Communication Technology) industryand science field requires the huge storage space, maximum completion time problem and data traffic problem. Our proposed method overcomes the above mentioned problems in big data broadcasting by using the novel pipeline approach. Our proposed concept broadcast the data in parallel manner which splits the data in a form of chunks. Here we broadcast the data from source node to destination node with the help of Lock Step Broadcast Tree [LSBT]. Our proposed method constructs the LSBT based on B+ tree algorithm for minimizing the total completion time and computational complexity during the big data broadcasting. LSBT constructed by the selecting of optimum uplink rate based on this rate we broadcast the data efficiently and minimize the completion time. Here we identify the data traffic problem in big data broadcasting by using the construction of LSBT. Here we evaluate the performance of completion time and computation time with the parameter of completion time, number of chunks, and size of delivery data.**

Index Terms–**Big data computing, data delivery algorithm, cloud computing, distributed computing, big data management.**

I. INTRODUCTION

Big data is a term for data sets that are so large or else complex that traditional data handling applications are insufficient to deal with them. Challenges comprise investigation, capture, data duration, storing, search, distribution, transmission, conception, querying, and informing and information privacy. The term big data often refers simply to the use of predictive analytics, user conduct analytics, or certain other advanced data analytics methods that extract value from data, and infrequently to a particular size of data set There is little doubt that the quantities of data now available are indeed large, but that's not the most applicable characteristic of this new data ecosystem. Analysis of data sets can find new correlations to spot business trends, prevent diseases, and combat crime and so on. Business executives, Scientists, business, practitioners of medicine, governments, and publicity alike commonly meet difficulties with large data-sets in areas including Internet search, urban informatics, finance, and business informatics. Scientists encounter limitations in E-Science work, including genomics, meteorology, connectomics, complex physics simulations, biology and environmental research.

Data sets develop quickly in part because they are increasingly collected by cheap and numerous information-sensing mobile devices, software logs aerial, remote sensing, micro phones cameras, Radio Frequency Identification [RFID] readers and wireless sensor networks The world's technological per capita capacity to store information has unevenly doubled every 40 months since the as of everyday 2.5 Exabyte's [2.5×1018] of data is generated. One question for large creativities is defining who should own big-data creativities that affect the entire association. Relational database management systems and desktop figures and visualization-packages often have trouble handling big data. The work may need massively parallel software running on tens, hundreds, or else even thousands of servers Pardon counts as big data varies depending on the capabilities of the users and their tools, and increasing capabilities make big data a affecting target. For some administrations, facing hundreds of gigabytes of data for the first time may activate a need to reconsider data management options. For others, it may take tens or else hundreds of terabytes before data size becomes a significant thought. Big data can be defined by the following features: Volume-The quantity of generated and stored data. The size of the data determines the value and potential insight- and whether it can actually be considered big data or not Variety-The type and nature of the data. This assistances people who analyze it to effectively use the subsequent insight. Velocity-In this setting, the speed at which the data is produced and processed to meet the difficulties and challenges that lie in the path of expansion and development. Variability-Inconsistency of the data set can basket processes to handle and manage it. Veracity-The quality of captured data can vary importantly, affecting precise analysis.

Factory work and Cyber-physical systems may have a 6C system: Connection [network and sensor] Cloud [data on demanding and computing] Cyber [memory and model] Content [meaning and correlation] Community [sharing and collaboration] Customization [personalization and value] Data must be processed with advanced tools [analytics and algorithms] to expose meaningful information. For sample, to manage a factory one must consider both noticeable and imperceptible issues with several components. Information generation algorithms must detect and address unseen issues such as machine degradation, component wear, etc. on the workshop ground.

Big data includes the data produced by dissimilar devices and applications. Given following are some of the areas that come below the umbrella of Big Data. Black Box Data: It is a component of helicopter, jets and airplanes, etc. It captures voices of the flight crew, recordings of microphones and earphones, and the performance information of the aircraft Social Media Data: Social media such as twitter and Facebook hold information and the views sent by millions of people crossways the world Stock Conversation Data: The stock conversation data grips information about the purchase and sell judgments made on a share of different corporations made by the clients. Power Grid Data: The power grid data holds information spent by a particular node with respect to a base station. Transport Data: Transport data includes model, capacity, and distance and obtain ability of a vehicle. Search Engine Data: Search engines retrieve lots of data from dissimilar databases.

Big data technologies are significant in providing more precise investigation, which may lead to more real choice making subsequent in greater working efficiencies, cost reductions, and concentrated risks for the business. To attach the power of big data, you would need an association that can achieve and develop huge capacities of structured and unstructured data in real time and can defend data confidentiality and safety. There are various technologies in the market from different vendors including Amazon, Microsoft, IBM, etc., to handle big data. While observing into the skills that handle big data, we inspect the subsequent two programs of technology:

Operational Big Data This includes systems like Mongo DB that deliver operational competences for real-time, interactive assignments where data is mainly taken and stored. NoSQL Big Data systems are intended to take benefit of new cloud computing architectures that have emerged over the past decade to allow massive calculations to be run reasonably and proficiently. This makes operational big data workloads much easier to achieve, inexpensive, and faster to implement. Some NoSQL systems can provide visions into designs and trends based on real time data with negligible coding and deprived of the need for data researchers and additional infrastructure.

Logical Big Data This comprises systems like Massively Parallel Processing (MPP) database systems and Map Reduce that deliver analytical capabilities for reflective and composite analysis that may touch most or else all of the data. Map Reduce delivers a new method of investigating data that is balancing to the capabilities provided by SQL, and a system based on Map Reduce that can be mounted up from single servers to thousands of high and low end machines. These two classes of machinery are complementary and frequently positioned together.

Big Data Encounters the major challenges associated with big data are as given as: Catching data portion, Storage, transfer, searching. Distribution, Investigation, Performance, these are the most important challenging problems in big data. Here our proposed work overcome the fewer problems such as completion time, computational complexity and traffic in data broadcasting by using the parallel pipe line manner.

II. System Study

1) Big Data

Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them. Challenges include analysis, capture, data duration, search, sharing, storage, transfer, visualization, querying, and updating and information privacy. The term "big data" often refers simply to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. There is little doubt that the quantities of data now available are indeed large, but that's not the most relevant characteristic of this new data ecosystem.

1.1) Big Data Broadcasting

Big data broadcasting is an important appliance to interconnect nodes in distributed systems. There are several applications that broadcast data nevertheless the data size differs pet bytes. Meanwhile data size develops huge and massive; there is an impact on distribution operations. Here we focus on big data broadcasting problem in heterogeneous networks. In these types of networks, numerous nodes have different uploading capabilities. The problem is how nodes accept data in minimum total broadcast time and complexity.In order to reduce the complexity of broadcast difficult, we present lockstep broadcast tree. By this we attain minimum completion time by enhancing bandwidth r. Here data is separated into chunks and sent in a pipeline manner. This LSBT is very valuable for host applications some of them are topology control, data broadcasting in cloud, energy preservation in peer to peer data delivery services. In preceding studies the LSBT was employed for homogeneous network systems. In these systems all nodes are having same uploading capabilities(c).Each node costs one unit of time, then maximum completion time of optimal solution is s+(log2n), where s is amount of chunks and n the number of nodes.

Assume that there are n nodes (n1, n2...nn) in heterogeneous network and their upload capabilities are dissimilar which means (c1, c2,....cn) restrained in kilobytes per second. The source data is divided into chunks of equal size. Here we introduce lockstep broadcast tree to typical big data broadcasting issue. LSBT is a tree somewhere data chunks transferred in a pipelined manner with as lesser amount of maximum transmission time. The aim of this tree model is to define upload bandwidth r, and uplink of each node is divided into various connections. After that broadcast data separated into s chunks and transmission down complete nodes in pipeline manner. In homogeneous networks all nodes have same uplink capacity c, and uplink rate rather the LSBT is c/2. In heterogeneous networks we use O [nlog2n] algorithm for selecting uplink rate r, there after we construct an optimal LSBT, here we implement LSBT in heterogeneous networks, which having different upload capabilities

1.2) Big Data Broadcasting Problem in Heterogeneous Network

The data broadcasting problem is about disseminating these m chunks to a population of n nodes in as less time as possible, subject to the uploading link capacity constraints of nodes. This problem has been studied in the context of many different network scenarios, such as homogenous and heterogeneous networks. For interested readers, a comprehensive survey can be found in the article

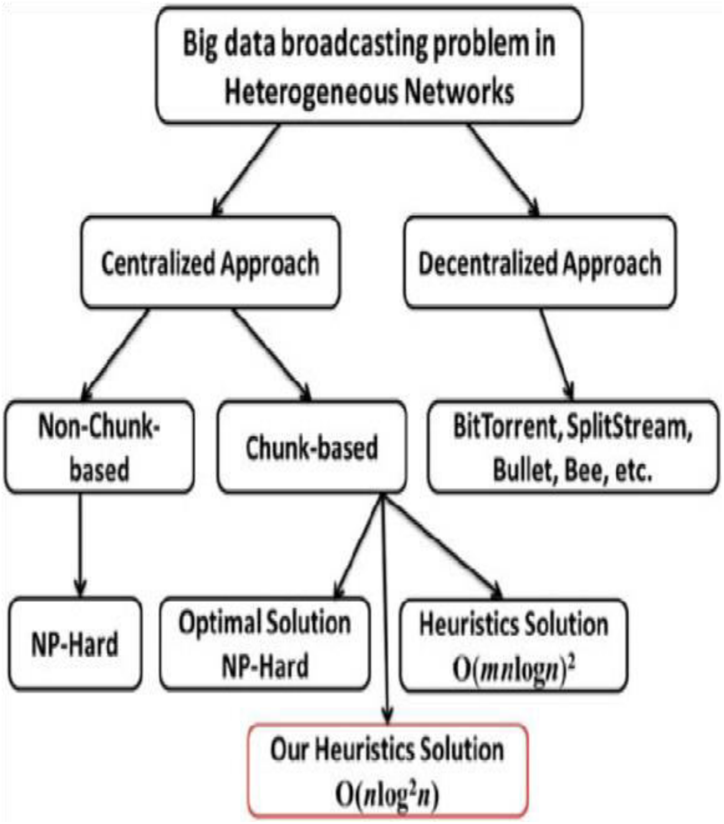
Big data broadcasting problem in Heterogeneous Networks
Centralized Approach
Decentralized Approach
Non-Chunk-based
Chunk-based
BitTorrent, SplitStream, Bullet, Bee, etc.
NP-Hard
Optimal Solution NP-Hard
Heuristics Solution $O(mn\log n)^2$
Our Heuristics Solution $O(n\log^2 n)$

Figure.1 Big Data Broadcasting Problem in Heterogeneous Network

We focus on big data broadcasting problem in heterogeneous networks. Illustrates these solutions to the big data broadcasting problem in heterogeneous networks along multiple dimensions. For the centralized approach, we first look at the results of the Non-Chunk based approach. Khuller and Kim showed that the problem of minimizing the completion time for broadcasting a single chunk (a message) in heterogeneous networks in a NP-hard problem. The authors also showed the Fastest-Node-First (FNF) heuristic method gets a performance ratio of at most 1.5 and the FNF results in optimal solutions in many cases for single chunk broadcast. In additional, Liu showed that the FNF heuristic method is optimal in only two classes of nodes. However, the data broadcasting problem is more complicated when the data consists of multiple chunks and it is still an open problem: Can data broadcasting problem with multiple chunks be solved by a polynomial time algorithm. Within the Chunk-based methods, the optimal solution has been shown in the article. The authors presented an uplink-sharing model for the well-known data broadcasting problem and formulated data broadcasting problem as a mixed integer linear programming (MILP)

However, as the numbers of variables in the linear programming grows exponentially n and m, this method is not practical for large n and m. Goetzmannet. al. show that if peer capacities are heterogeneous WU et al., IIS TECHNICAL REPORT-12-006 3 and symmetric, this problem becomes strongly NP-hard. A recent result presented two heuristic algorithms to schedule data chunks transfer between nodes. The time complexity of both two centralized algorithms is O(m × nlogn) 2. For the decentralized approach, many decentralized systems have been proposed to disseminate chunks via an overlay topology. With overlay-based approaches, nodes maintain a set of overlay links to other nodes and exchange chunks among neighboring nodes. Bit Torrent, Split Steam, Bullet and Bee are some examples of the overlay based approach. In, the authors showed Bee can approach lower bound of the maximum completion time in heterogeneous networks by simulations. In this paper, we retain the interest in the centralized approaches, thus interested readers can find a comprehensive survey of these decentralized systems.

2) Lock Step Broadcast Tree

To reduce the complexity of the original data broadcasting problem. we model it as the Lock Step Broadcast Tree (LSBT) problem. By this we define a performance goal for a single LSBT that is achieving minimum completion time by optimizing the basic bandwidth allocation, r, among LSBT nodes. Different from original problem, we allow data be divided into chunks and sent in a pipeline fashion. Formally, given a set of n nodes N = {n1, . . . ,nn}, each node ni is connected to the network via an access link of upload capacities ci and a size of chunks B. The LSBT problem is to determine the upload bandwidth r ∗ of each uplink to build the LSBT t, in which node ni should allocate upload bandwidth r ∗ to each connection to its child nodes in order to minimize the maximum completion time D for propagating a data chunk. Note that it is possible to handle simultaneously several connections and to fix the bandwidth allocated to each connection. In the following definition, we define the number of edges k in each node for LSBT.

2.1) B+ Tree

A B+ tree is a data structure often used in the implementation of database indexes. Each node of the tree contains an ordered list of keys and pointers to lower level nodes in the tree. These pointers can be thought of as being between each of the keys. To search for or insert an element into the tree, one loads up the root node, finds the adjacent keys that the searched-for value is between, and follows the corresponding pointer to the next node in the tree. Recursing eventually leads to the desired value or the conclusion that the value is not present.

B+ trees use some clever balancing techniques to make sure that all of the leaves are always on the same level of the tree, that each node is always at least half full (rounded) of keys, and (therefore) that the height of the tree is always at most ceiling (log(n)/log(k/2)) (plus or minus a constant, I don't remember exactly) where n is the number of values in the tree and k is the maximum number of keys in each block. This means that only a small number of pointer traversals is necessary to search for a value if the number of keys in a node is large. This is crucial in a database because the B+ tree is on disk. Reading a single block takes just as much time as reading a partial block, and a block can hold a large number of keys/pointers (128-1024 or so).

Finally, the leaves of the B+ tree all contain a "next sibling" pointer for fast iteration through a contiguous block of values. This allows for extremely efficient range queries (find the block containing the first value and blast through siblings until the last value is found). It's especially efficient if the implementation guarantees that leaf blocks are contiguous on disk, though this is a challenge as far as I know.

B+ trees can also be used outside of the disk, but generally a balanced binary search tree or a skip list or something should provide better performance in memory, where pointer following is no more expensive than finding the right pointer to follow.

3) Optimal Lockstep Broadcasting Tree

We present our LSBT algorithm that is also a heuristic for the data broadcasting problem. Given a set of node upload capacities c, we aim at finding an optimal LSBT that is a data broadcast tree where data chunks can be sent in a pipelined manner. We provide a thorough analysis of LSBT in both homogenous and heterogeneous net- work systems. We first clarify LSBT in homogenous networks cases and describe the LSBT algorithm in heterogeneous network cases later.

3.1) Homogenous Network Systems

We present the optimal solution of LSBT when the upload capacities of nodes are identical. In general, we assume that all nodes have upload capacity of c. Mundinger et al. have presented the optimal scheduling solution for broad- casting multiple messages on the uplink-sharing model. The following Theorem 1 (Mundinger's theorem) is proved in the article. If each round costs one unit of time, then the maximum completion time of the optimal solution is m þblog 2nc, where m is the number of chunks and n the number of nodes. Note that each node can only upload one data chunk to another node in each round. By contrast, each node can send a data chunk to k other nodes simultaneously in the LSBT model.

3.2) Heterogeneous Network Systems

A heterogeneous network is a network connecting computers and other devices with different operating systems and/or protocols. For example, local area networks (LANs) that connect Microsoft Windows and Linux based personal computers with Apple Macintosh computers are heterogeneous. The word heterogeneous network is also used in wireless networks using different access technologies. For example, a wireless network which provides a service through a wireless LAN and is able to maintain the service when switching to a cellular network is called a wireless heterogeneous network.

We now consider the general LSTB model in which nodes' upload capacities may be different. First, we present an algorithm to construct an optimal LSBT for a given rate r. We then give both the upper and lower bounds of the value of r*. Finally we present an O(n log2n) algorithm to select the optimal upload bandwidth r* of each uplink and to construct the optimal LSBT.

4) Hadoop Software Environment

Hadoop is an Apache open source outline written in java that agree to distributed dispensation of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and addition across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage. Hadoop framework includes following four modules: Hadoop Common. These are Java public library and utilities required by other Hadoop modules. These libraries provide file system and OS level concepts and contains the essential Java files and scripts required to start Hadoop. Hadoop YARN: This is a framework for job scheduling and cluster resource management**.**

4.1) Hadoop Distributed File System

A distributed file system that provides high-throughput access to application data. Hadoop Map Reduce: This is YARN-based system for parallel processing of large data sets. We can use following diagram to depict these four components available in Hadoop framework.

III. MODULE DESCRIPTIONS

1) Lock Step Broadcast Tree Formation:

Our proposed method broadcast the data by using the LSBT. Here we construct the LSBT tree which constructed by using a B+ tree algorithm. In this algorithm we select the optimum uplink rate and capacity of nodes. Here height of tree determines by the set of upload bandwidth and uploads capacity and completion time of tree depends on the amount of data broadcast and optimum uplink rate. By this tree formation we minimize the completion time of data broad casting.

2) Data Broadcasting

Here we broadcast the data from source node to destination node in form a chunks by splits the data in equal size of chunks Our proposed method we broadcast the data in pipeline manner which means broadcast the data in parallel which. Then we broadcast the data by using lock step broadcast tree, which select the optimum uplink rate by using this method of broadcasting we achieve minimum completion time.

3) Monitor Delivery Service Deadline

In this module we monitor the deadline of service which means completion time of data broadcasting. By this module we analyses the total completion time of data broadcasting and monitor the traffic occurrence in big data broadcasting. We can monitor the service deadline by using the lock step broadcast tree which identifies the network can meet the service of deadline in data broadcasting.

The given data delivery deadline, one can estimate whether a delivery job through a specific network could meet its deadline based on the LSBT model. Developers may take advantage of this property to can maximize the performance of collaborative applications in datacenter networks

4) Performance Evaluation

Our proposed method achieves the lower computational complexity and minimum completion time. Our proposed work evaluate the performance as completion time and computational complexity of LSBT big data broadcasting by using the following parameter such as number of chunks, data size and computing time

Algorithm Description

Our proposed work broadcast data in efficient manner by using B+ tree algorithm in which selects the optimum uplink rate in network by the construction of LSBT tree. B+ tree is n-ary tree with a variable which have more than one children's per node. A B+ tree consists of a root, internal nodes and leaves. The root may be a source node which have a leaf or a node with two or more children.. By using B+ tree we calculate the optimum uplink rate based on node capacity and upload band width. Here we determine the node capacity based on amount of data broadcasting in network. Our proposed work selects the optimum uplink rate in order to improve the efficiency of big data broadcasting.

IV. RESULT ANALYSIS

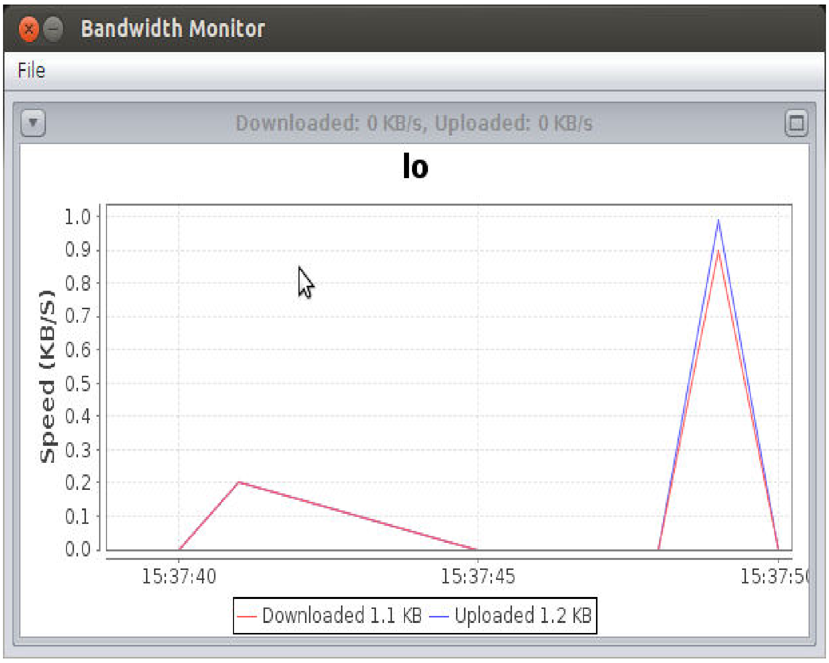

Figure.2 Bandwidth Monitor

Bandwidth monitor shows the download and upload data speeds. Mostly files are divided into numbers of chunks and delivered to the numbers of nodes.

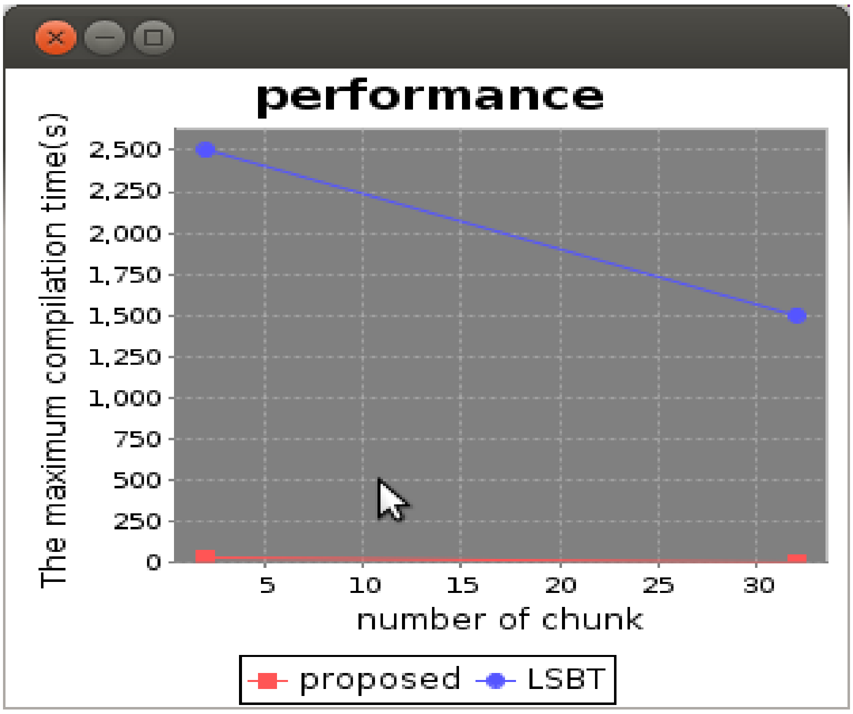

Figure.3 Performance of Bandwidth

In our project we are using chunks broadcasting that is single file divided into number of chunks and delivered to user. The above figure shows the performance of bandwidth.

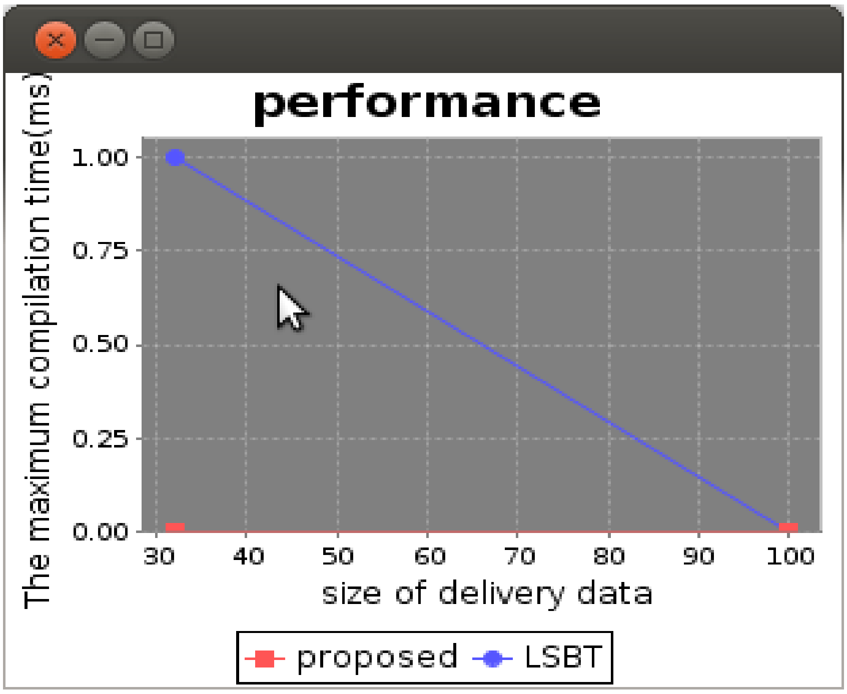

Figure.4 Data Delivery Process

The above figure shows the data delivery process performance in the receiver side.

V. CONCLUSION

Our proposed method presents the efficient pipeline technique for big data broadcasting by using the Lock Step Broadcast Tree [LSBT] which is constructed by using the B+ tree algorithm. Our proposed work minimize the total completion time in data broadcasting by selecting the optimum uplink rate. Here we select the optimum uplink rate by using B+ tree algorithm which is selected by the capacity of nodes and upload bandwidth. Our method broad cast the data in form a chunks in parallel [pipe line] manner. By this method we achieve the efficient big data broadcasting. LSBT tree provides to monitor the deadline of service by this feature we can monitor the traffic occurrence in data broadcasting. Our proposed work analysis the performance in big data broadcasting as completion time and complexity with the parameters of number of chunks and data size.

Our proposed method provides considers only completion time of big data broadcasting; in future we have to plan for data broadcasting in secure manner, we have to planned for designing the security in broadcasting by the way of authentication system, cryptographic technique. In addition we also investigate the attacks such as replicas attack and collusion attack in data broadcasting.

REFERENCES

[1] D. Nukarapu, B. Tang, L. Wang, and S. Lu, “Data replication in data intensive scientific applications with performance guarantee,” IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 8, pp. 1299–1306, Aug. 2011.

[2] C. Peng, M. Kim, Z. Zhang, and H. Lei, “VDN: Virtual machine image distribution network for cloud data centers,” Proc. IEEE Int. Conf. Comput. Commun., 2012, pp. 181–189

[3] L. Massoulie, A. Twigg, C. Gkantsidis, and P. Rodriguez, “P2P streaming capacity under node degree bound,” IEEE 30th Int. Conf. Dist. Comput. Syst., pp. 587–598, 2010.

[4] O. Beaumont, L. Eyraud-Dubois, and S. K. Agrawal, “Broadcasting on large scale heterogeneous platforms under the bounded multi-port model,” in Proc. IEEE Int. Symp. Parallel Distrib. Process. Symp. 2010, pp. 1–11.

[5] C.-J. Wu, C.-Y. Li, K.-H. Yang, J.-M. Ho, and M.-S. Chen, “Timecritical data dissemination in cooperative peer-to-peer systems,” Proc. IEEE Global Telecommun., 2009, pp. 1–6.

[6] M. Banikazemi, V. Moorthy, and D. Panda, “Efficient collective communication on heterogeneous networks of workstations,” in Proc. IEEE Int. Conf. Parallel Process., 1998, pp. 460–467

[7] G. M. Ezovski, A. Tang, and L. L. H. Andrew, “Minimizing average finish time in P2P networks,” in Proc. IEEE Int. Conf. Comput. Commun., 2009, pp. 594–602

[8] C. Chang, T. Ho, M. Effros, M. Medard, and B. Leong, “Issues in peer-to-peer networking: A coding optimization approach,” in Proc. IEEE Int. Symp. Netw. Coding, 2010, pp. 1–6.

[9] S. Liu, M. Chen, S. Sengupta, M. Chiang, J. Li, and P. A. Chou, “P2P streaming capacity under node degree bound,” in Proc. IEEE Int. Conf. Distrib. Comput. Syst., 2010, pp. 587–598.

[10] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang, “Performance bounds for peer-assisted live streaming,” in Proc. ACM SIGMETRICS Conf. Meas. Model. Computer Syst., 2008,pp. 313–324.