Swarm Intelligence in MapReduce

Penuparty Srikant
M.Tech., School Of Computer Engineering
KIIT University
Bhubaneswar, India

Ananta Chandra Das
M.Tech., School Of Computer Engineering
KIIT University
Bhubaneswar, India

Santwana Sagnika
Asst. Professor, School of Computer Engineering
KIIT University
Bhubaneswar, India

***Abstract*— In the last two decades, the continuous increase of computational power has produced an overwhelming flow of data. Big data is not only becoming more available but also more understandable to computers. For example, modern high-energy physics experiments, such as Dzero, typically generate more than one Tera Bytes of data per day. The famous social network Website, Facebook, serves 570 billion page views per month, stores 3 billion new photos every month, and manages 25 billion pieces of content. Google's search and ad business, Facebook, Flickr, YouTube, and LinkedIn use a bundle of artificial-intelligence tricks, require parsing vast quantities of data and making decisions instantaneously. Big data and cloud computing are both the fastest-moving technologies identified in Gartner Inc.'s 2012 Hype Cycle for Emerging Technologies4. Cloud computing is associated with new paradigm for the provision of computing infrastructure and big data processing method for all kinds of resources. Moreover, some new cloud-based technologies have to be adopted because dealing with big data for concurrent processing is difficult. Then what is Big Data? In the publication of the journal of Science 2008, "Big Data" is defined as "Represents the progress of the human cognitive processes, usually includes data sets with sizes beyond the ability of current technology, method and theory to capture, manage, and process the data within a tolerable elapsed time". Recently, the definition of big data as also given by the Gartner: "Big Data are high-volume, high-velocity, and/or high-variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization". According to Wikimedia, "In information technology, big data is a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools".**

Present big data needs cannot be suitably handled with conventional relational databases, desktop analysis and statistical packages. There is need of immense parallel processing of software running on hundreds and thousands of machines concurrently. However, big data pose four unique dimensions of problems described as five Vs, namely, Volume, Value, Velocity, Variety and Veracity.

Keywords- Big data; data analytics; data storage procedure; big data landscape; hadoop; Hadoop Distributed File System(HDFS); Map reduce, Swarm Intelligence.

I. BIG DATA MANAGEMENT SYSTEM

Many researchers have suggested that commercial DBMSs are not suitable for processing extremely large scale data. Classic architecture's potential bottleneck is the database server while faced with peak workloads. One database server has restriction of scalability and cost, which are two important goals of big data processing. In order to adapt various large data processing models, D. Kossmannet al. presented four different architectures based on classic multi-tier database application architecture which are partitioning, replication, distributed control and caching architecture. It is clear that the alternative provider shave different business models and target different kinds of applications: Google seems to be more interested in small applications with light workloads whereas Azure is currently the most affordable service for medium to large services. Most of recent cloud service providers are utilizing hybrid architecture that is capable of satisfying their actual service requirements. In this section, we mainly discuss big data architecture from three key aspects: distributed file system, non-structural and semi-structured data storage and open source cloud platform.[17]

II. DISTRIBUTED FILE SYSTEM

Google File System (GFS)[4] is a chunk-based distributed file system that supports fault-tolerance by data partitioning and replication. As an underlying storage layer of Google's cloud computing platform, it is used to read

input and store output of MapReduce [5] . Similarly, Hadoop also has a distributed file system as its data storage layer called Hadoop Distributed File System (HDFS), which is an open-source counterpart of GFS. GFS and HDFS are user level file systems that do not implement POSIX semantics and heavily optimized for the case of large files (measured in gigabytes). Amazon Simple Storage Service (S3) is an online public storage web service offered by Amazon Web Services. This file system is targeted at clusters hosted on the Amazon Elastic Compute Cloud server-on-demand infrastructure. S3 aims to provide scalability, high availability, and low latency at commodity costs. ES2 is an elastic storage system of epic, which is designed to support both functionalities within the same storage. The system provides efficient data loading from different sources, flexible data partitioning scheme, index and parallel sequential scan. In addition, there are general file systems that have not to be addressed such as Moose File System (MFS), Kosmos Distributed File system (KFS).[17]

III. NON-STRUCTURAL AND SEMI-STRUCTURED DATA STORAGE

With the success of the Web 2.0, more and more IT companies have increasing needs to store and analyze the ever growing data, such as search logs, crawled web content, and click streams, usually in the range of peta bytes, collected from a variety of web services. However, web data sets are usually non-relational or less structured and processing such semi-structured data sets at scale poses another challenge. Moreover, simple distributed file systems mentioned above cannot satisfy service providers like Google, Yahoo!, Microsoft and Amazon. All providers have their purpose to serve potential users and own their relevant state of-the-art of big data management systems in the cloud environments. Big table is a distributed storage system of Google for managing structured data that is designed to scale to a very large size (peta bytes of data) across thousands of commodity servers. Big table does not support a full relational data model.

However, it provides clients with a simple data model that supports dynamic control over data layout and format. PNUTS [11] is a massive scale hosted database system designed to support Yahoo!'s web applications. The main focus of the system is on data serving for web applications, rather than complex queries. Upon PNUTS, new applications can be built very easily and the overhead of creating and maintaining these applications is nothing much. The Dynamo [12] is a highly available and scalable distributed key/value based data store built for supporting internal Amazon's applications. It provides a simple primary-key only interface to meet the requirements of these applications. However, it differs from key-value storage system. Facebook proposed the design of a new cluster-based data warehouse system, Llama, a hybrid data management system which combines the features of row-wise and column-wise database systems. They also describe a new column-wise file format for Hadoop called CFile, which provides better performance than other file formats in data analysis.

IV. OPEN SOURCE CLOUD PLATFORM

The main idea behind data center is to leverage the virtualization technology to maximize the utilization of computing resources. Therefore, it provides the basic ingredients such as storage, CPUs, and network band width as a commodity by specialized service providers at low unit cost. For reaching the goals of big data management, most of the research institutions and enterprises bring virtualization into cloud architectures. Amazon Web Services (AWS), Eucalptus, Opennebula, Cloudstack and Openstack are the most popular cloud management platforms for infrastructure as a service (IaaS). AWS9 is not free but it has huge usage in elastic platform. It is very easy to use and only pay-as-you-go. The Eucalyptus works in IaaS as an open source. It uses virtual machine in controlling and managing resources. Since Eucalyptus is the earliest cloud management platform for IaaS, it signs API compatible agreement with AWS. It has a leading position in the private cloud market for the AWS ecological environment.

OpenNebula has integration with various environments. It can offer the richest features, flexible ways and better interoperability to build private, public or hybrid clouds. OpenNebula is not a Service Oriented Architecture (SOA) design and has weak decoupling for computing, storage and network independent components. CloudStack10 is an open source cloud operating system which delivers public cloud computing similar to Amazon EC2 but using users' own hardware. Cloud Stack users can take full advantage of cloud computing to deliver higher efficiency, limitless scale and faster deployment of new services and systems to the en duser. At present, CloudStack is one of the Apache open source projects. It already has mature functions. However, it needs to further strengthen the loosely coupling and component design. OpenStack11 is a collection of open source software projects aiming to build an open-source community with researchers, developers and enterprises. People in this community share a common goal to create a cloud that is simple to deploy, massively scalable and full of rich features. The architecture and components of OpenStack are straight forward and stable, so it is a good choice to provide specific applications for enterprises. In current situation, OpenStack has good community and ecological environment. However, it still have some shortcomings like incomplete functions and lack of commercial supports.

V. APPLICATIONS AND OPTIMIZATION

A. Application

In this age of data explosion, parallel processing is essential to perform a massive volume of data in a timely manner. The use of parallelization techniques and algorithms is the key to achieve better scalability and performance for processing big data. At present, there are a lot of popular parallel processing models, including MPI, General Purpose GPU (GPGPU), Map Reduce and MapReduce-like. MapReduce proposed by Google, is a very popular bigdata processing model that has rapidly been studied and applied by both industry and academia. MapReduce has two major advantages: the MapReduce model hide details related to the data storage, distribution, replication, load balancing and so on. Furthermore, it is so simple that programmers only specify two functions, which are map function and reduce function, for performing the processing of the big data. We divided existing MapReduce applications into three categories: partitioning sub-space, decomposing sub-processes and approximate overlapping calculations. While MapReduce is referred to as a new approach of processing big data in cloud computing environments, it is also criticized as a "major step backwards" compared with DBMS. We all know that MapReduce is schema-free and index-free. Thus, the MapReduce framework requires parsing each record at reading input.

As the debate continues, the final result shows that neither is good at the other does well, and the two technologies are complementary. Recently, some DBMS vendors also have integrated MapReduce front-ends into their systems including Aster, HadoopDB, Greenplum and Vertuca. Mostly of those are still databases, which simply provide a MapReduce front-end to a DBMS. HadoopDB is a hybrid system which efficiently takes the best features from the scalability of MapReduce and the performance of DBMS. The result shows that HadoopDB improves task processing times of Hadoop by a large factor to match the shared nothing DBMS. Lately, J. Dittrich et al. propose a new type of system named Hadoop++ which indicates that HadoopDB has also severe drawbacks, including forcing user to use DBMS, changing the interface to SQL and soon. There are also certain papers adapting different inverted index, which is a simple but practical index structure and appropriate for MapReduce to process big data, such as etc. We also do intensive study on large-scale spatial data environment and design a distributed inverted grid index by combining inverted index and spatial grid partition with MapReduce model, which is simple, dynamic, scale and fit for processing high dimensional spatial data[11].

B. Optimization

In this section, we present details of approaches to improve the performance of processing big data with MapReduce.

1) ***Data Transfer Bottlenecks****: It is a big challenge that cloud users must consider how to minimize the cost of data transmission. Consequently, researchers have begun to propose variety of approaches. Map-Reduce-Merge is a new model that adds a Merge phase after Reduce phase that combines two reduced outputs from two different MapReduce jobs into one, which can efficiently merge data that is already partitioned and sorted (or hashed) by map and reduce modules. Map-Join-Reduce [13] is a system that extends and improves MapReduce runtime framework by adding Join stage before Reduce stage to perform complex data analysis tasks on large clusters. They present a new data processing strategy which runs filtering-join aggregation tasks with two consecutive MR jobs. It adopts one-to-many shuffling scheme to avoid frequent check pointing and shuffling of intermediate results. Moreover, different jobs often perform similar work, thus sharing similar work reduces overall amount of data transfer between jobs. MR Share is a sharing framework proposed by T. Nykielet al. that transforms a batch of queries into a new batch that can be executed more efficiently by merging jobs into groups and evaluating each group as a single query. Data skew is also an important factor that affects data transfer cost. In order to overcome this deficiency, we propose a method that divides a MapReduce job into two phases sampling MapReduce job and expected MapReduce job. The first phase is to sample the input data, gather the inherent distribution on keys' frequencies and then make a good partition scheme in advance. In the second phase, expected MapReduce job applies this partition scheme to every mapper to group the intermediate keys quickly.*

2) ***Iterative Optimization:*** *MapReduce also is a popular platform in which the dataflow takes the form of a directed acyclic graph of operators. However, it requires lots of I/Os and unnecessary computations while solving the problem of iterations with MapReduce. Twister [3] proposed by J. Ekanayake et al. is an enhanced MapReduce runtime that supports iterative MapReduce computations efficiently, which adds an extra Combine stage after Reduce stage. Thus, data output from combine stage flows to the next iteration's Map stage. It avoids instantiating workers repeatedly during iterations and previously instantiated workers are reused for the next iteration with different inputs. Hadoop is similar to Twister, which is a modified version of the MapReduce framework that supports for iterative applications by adding a Loop control. It also*

allows to cache both stages' input and output to save more I/Os during iterations. There exist lots of iterations during graph data processing. Pregel implements a programming model motivated by the Bulk Synchronous Parallel (BSP) model, in which each node has its own input and transfers only some messages which are required for the next iteration to other nodes.[5]

3) Online: There are some jobs which need to process online while original MapReduce can't do this very well. MapReduce Online is designed to support online aggregation and continuous queries in MapReduce. It raises an issue that frequent check pointing and shuffling of intermediate results limit pipelined processing. They modify MapReduce framework by making Mappers push their data temporarily stored in local storage to Reducers periodically in the same MR Job. In addition, Map-side pre-aggregation is used to reduce communication. Hadoop Online Prototype (HOP) proposed by Tyson Condie is similar to MapReduce Online. HOP is a modified version of MapReduce framework that allows users to early get returns from a job as it is being computed. It also supports for continuous queries which enable MapReduce programs to be written for applications such as event monitoring and stream processing while retaining the fault tolerance properties of Hadoop. D. Jiang et al. found that the merge sort in MapReduce costs lots of I/Os and seriously affects the performance of MapReduce. In the study, the results are hashed and pushed to hash tables held by reducers as soon as each map task outputs its intermediate results. Then, reducers perform aggregation on the values in each bucket. Since each bucket in the hash table holds all values which correspond to a distinct key, no grouping is required. In addition, reducer scan perform aggregation.

VI. BIG DATA STORAGE AND MANAGEMENT

Current technologies of data management systems are not able to satisfy the needs of big data, and the increasing speed of storage capacity is much less than that of data, thus a revolution re-construction of information framework is desperately needed. We need to design a hierarchical storage architecture. Besides, previous computer algorithms are not able to effectively storage data that is directly acquired from the actual world, due to the heterogeneity of the big data. However, they perform excellent in processing homogeneous data. Therefore, how to re-organize data is one big problem in big data management. Virtual server technology can exacerbate the problem, raising the prospect of overcommitted resources, especially if communication is poor between the application, server and storage administrators. We also need to solve the bottleneck problems of the high concurrent I/O and single-named node in the present Master-Slave system model.[15]

VII. MAPREDUCE

MapReduce is a framework using which we can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner. MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job. The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into mappers and reducers is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

Generally MapReduce paradigm is based on sending the computer to where the data resides.

- **Map stage**: The map or mapper's job is to process the input data. Generally the input data is in the form of file or directory and is stored in the Hadoop file system (HDFS). The input file is passed to the mapper function line by line. The mapper processes the data and creates several small chunks of data.

- **Reduce stage**: This stage is the combination of the **Shuffle** stage and the **Reduce** stage. The Reducer's job is to process the data that comes from the mapper. After processing, it produces a new set of output, which will be stored in the HDFS.

- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.

- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.

- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.

After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

	Input	Output
Map	<k1, v1>	list(<k2, v2>)
Reduce	<k2, list(v2)>	list(<k3, v3>)

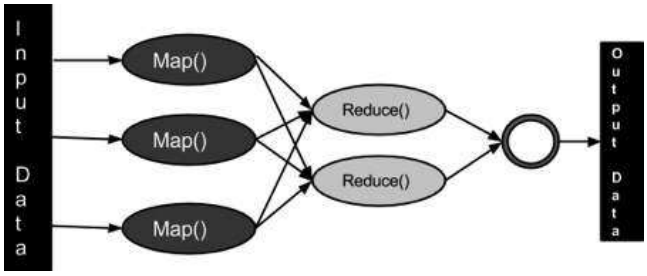

Fig1: MapReduce Procedure

Inputs and Outputs (Java Perspective)

The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.

The key and the value classes should be in serialized manner by the framework and hence, need to implement the Writable interface. Additionally, the key classes have to implement the Writable-Comparable interface to facilitate sorting by the framework. Input and Output types of a MapReduce job: (Input) <k1, v1> -> map -><k2, v2>-> reduce -><k3, v3>(Output).

TABLE I. TABLE SHOWING MAPREDUCE PROCEDURE

VIII. SWARM INTELLIGENCE

Swarm intelligence comes from swarming behaviors of groups of organisms. Group living enables organisms to solve problems that are difficult or impossible for single individuals to resolve (see [1-5]). So, swarm intelligence can be seen as a mechanism which individuals can use to overcome some of their own cognitive limitations. Swarm intelligence claims the ability to manage complex systems of interacting individuals through minimal communication with only local neighbors to produce a global emergent behavior. They typically do not follow commands from a leader, or some global plan. These special features make swarm intelligence play important roles in many engineering applications such as formation control of multi-robot system, massive distributed sensing using mobile sensor networks, combat using cooperative unmanned aerial vehicles, flocking, etc. Also, for several decades, many researchers have been devoted much efforts to such systems, which can be roughly divided into three groups: 1) biological researchers, which are the first groups to make the early efforts to study swarm behavior; 2) Artificial algorithm researchers, which have done important work on swarming topology; 3) engineering application researchers, which have increased much interest on swarming engineering such as multi-robot, air vehicles, sensor network, etc. All three groups of researchers have greatly advanced the swarm intelligence by delivering large number of significant results in the recent decades.

A. Biological Basis

A variety of organisms have the ability to cooperatively forage for food while trying to avoid predators and other risks. This kind of motion can be called "swarm behavior". Naturalists and biologists have found that it provides many more chances for surviving than a single organism. So, they have been working on understanding and modeling of swarm behavior for a long time (see [19-20]). The swarm achieves its objectives via the interactions of the entire group. The organisms use simple local rules to govern their actions. They typically do not follow commands from a leader, or some global plan. For example, in [23] Bonabeau explains how social foragers as a group more successfully perform chemo taxis over noisy gradients than individually. In other words, individuals do much better collectively compared to the case when they forage on their own. Operational principles from such biological systems can be used in engineering for developing distributed cooperative control, coordination, and learning strategies for autonomous multi agent systems, such as autonomous multi-robot applications, unmanned undersea, land, or air vehicles.

B. Artificial Literature

In 1986, Reynolds introduced three heuristic rules that led to creation of the first computer animation of a flock of birds. These rules are also known as cohesion, separation, and alignment rules in the literature (see [24]). The general approach of the first groups of physicists who studied swarming behavior is to model each individual as a particle, which they usually call a self-driven or self propelled particle, and study the collective behavior due to their interaction. Vicsek et al (see [25]) investigated the emergence of coherent collective motion in a self-driven discrete system with some biologically motivated interactions, and presented numerical evidence that this model results in a kinetic continuous phase transiting from no transport to finite net transport through spontaneous symmetry breaking of the rotational symmetry. Toner and Tu constructed a continuum, hydrodynamic description of the flock and develop a quantitative continuum theory of

flocking (see [25]). This approach captures the essential physics of Vicsek's model and represents the existence of an ordered phase of flocks, in which all members of even an arbitrarily large flocks move together.

C. Swarm Engineering

A swarm engineering approach incorporates a large number of relatively simple robots given simple commands whose interactions cause a global emergent behavior. This approach is extracted from nature by analyzing social insects. For example, ants, in addition to being excellent architects, also have the ability to find the shortest path to food yet posses no advanced communication abilities. Success is finally achieved by following a simple swarm-based design approach. Swarm engineering was declared to be a formal research field by Kazadi in 2000 (see [25]). He defined swarm engineering as "two-step" process. The first step is the generation of a swarm condition and the second step is the fabrication of behaviors that can satisfy the swarm condition. The goal of swarm engineering is to produce a general condition or set of conditions which may be used to generate many different swarm designs any of which can complete the global goal (see [25]). Kazadi provides formal definitions and proposes a high level mathematical framework for swarm engineering. Recently, He also used the phase space diagrams to predict the final states and system evolution of swarms (see [23]).

There are many techniques of Swarm Intelligence.

- ABC
- PSO
- Ant Colony

IX. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. The detailed information will be given in following sections.

Artificial Life

The term "Artificial Life" (ALife) is used to describe research into human-made systems that possess some of the essential properties of life. ALife includes two-folded research topic. ALife studies how computational techniques can help when studying biological phenomena. ALife studies how biological techniques can help out with computational problems.

Algorithm

As stated before, PSO simulates the behaviors of bird flocking. Suppose the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration. So what's the best strategy to find the food? The effective one is to follow the bird which is nearest to the food. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called lbest.

After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).

v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) (a)

present[] = persent[] + v[] (b)

v[] is the particle velocity, persent[] is the current particle (solution). pbest[] and gbest[] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. usually c1 = c2 = 2.

The pseudo code of the procedure is as follows

```
For each particle
    Initialize particle
END
Do
   For each particle
      Calculate fitness value
         If the fitness value is better than the best fitness value (pBest) in history
            set current value as the new pBest
 End
```

Choose the particle with the best fitness value of all the particles as the gBest

For each particle

Calculate particle velocity according equation (a)

Update particle position according equation (b)

End

While maximum iterations or minimum error criteria is not attained Particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax.

FUTURE WORK

Hadoop environment and Computational Intelligence using various artificial methods like" Artificial intelligence", "PSO", "Ant colony optimization" are closely related for big data analysis. However, the big data analysis using Hadoop is nature inspired and is an effective method for analyzing and mining tons of data for useful information.

CONCLUSION

Data is being generated for years and is collected by the organizations for proper functioning of the organizations. By the advancement in information technology the rate of data generation and its volume has increased so many times and is termed as Big Data. Although the term Big Data is an umbrella term for a high velocity, high volume, high variety and veracity of data that is difficult to manage by traditional solutions. In this paper we have proposed an economical and effective solution for Big Data.

We have proposed a framework that will provide and economical data store data in cloud on the commodity hardware. Our framework will also extract some metadata information from the data that will be used to provide a schema for Big Data. In future we will extend this work through proper experimentation and results. The big data analysis can be optimized taking advantage of various already discovered algorithms using swarm intelligence, artificial intelligence incorporating efficient machine learning for better understanding. This is used for training the machines and carrying forward the tasks of predictive analysis, collaborative filtering and also building empirical statistical predictive models. Even Map Reduce is a very good technology in Big Data, still there are some complexities. Using Swarm Intelligence we can avoid that complexities. Future work will focus on using PSO technique and algorithm in map reduce to avoid the complexity measures in map reduce.

REFERENCES

[1] A.C. Das, S.N. Mohanty, and S.K. Pani, "A comparative study on data ananlytics and Big Data analytics", IJCSITR, Vol. 4, Issue 1, pp: 67-75, 2016

[2] Douglas and Laney, "The importance of 'big data': A definition", 2008.

[3] D. Kossmann, T. Kraska, and S. Loesing, "An evaluation of alternative architectures for transaction processing in the cloud," in Proceedings of the 2010 international conference on Management of data. ACM, 2010, pp. 579–590.

[4] S. Ghemawat, H. Gobioff, and S. Leung, "The google filesystem," in ACM SIGOPSOperating Systems Review, vol. 37,no. 5. ACM, 2003, pp. 29–43.

[5] J. Dean and S. Ghemawat, "Mapreduce: simplified dataprocessing on large clusters," Communications of the ACM,vol. 51, no. 1, pp. 107–113, 2008.

[6] D. Borthakur, "The hadoop distributed file system: Architectureand design," Hadoop ProjectWebsite, vol. 11, 2007.

[7] A. Rabkin and R. Katz, "Chukwa: A system for reliablelarge-scale log collection," in USENIX Conference on LargeInstallation System Administration, 2010, pp. 1–15.

[8] S. Sakr, A. Liu, D. Batista, and M. Alomari, "A survey oflarge scale data management approaches in cloud environments,"Communications Surveys & Tutorials, IEEE, vol. 13,no. 3, pp. 311–336, 2011.

[9] Y. Cao, C. Chen, F. Guo, D. Jiang, Y. Lin, B. Ooi, H. Vo,S. Wu, and Q. Xu, "Es2: A cloud data storage system forsupporting both oltp and olap," in Data Engineering (ICDE),2011 IEEE 27th International Conference on. IEEE, 2011,pp. 291–302.

[10] Ramamoorthy.S.,Rajalakshmi.S., "Optimized data analysis in cloudusing BigData analytics techniques", Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference Publication Year: July 2013, Page(s):1-5

[11] Patel, A.B, Birla M., Nair U., "Addressing big data problem usingHadoop and Map Reduce", Engineering (NUiCONE), 2012.:NirmaUniversity International Conference on Engineering,6-8 Dec. 2012

[12] JinsonZhang , Mao Lin Huang., "5Ws Model for Big Data Analysis andVisualization", Computational Science and Engineering (CSE), 2013IEEE 16th International Conference, Publication Year: 2013, Page(s):1021-1028

[13] http://iveybusinessjournal.com/topics/strategy/why-big-datais-the-new-competitive-advantage#.U6Ajm5RdX78

[14] http://www.linkedin.com/today/post/article/20140306073407-64875646-big-data-the-5-vs-everyone-must-know

[15] S. Ghemawat, H. Gobioff, and S. Leung, "The google filesystem," in ACM SIGOPS Operating Systems Review, vol.37, no. 5. ACM, 2003, pp. 29–43.

[16] ChangqingJi, Yu Li, WenmingQiu, UchechukwuAwada, Keqiu Li,"Big Data Processing in Cloud Computing Environments",2012 International Symposium on Pervasive Systems, Algorithms and Networks,Page(s):17-22

[17] Muhammad Adnan,MuhammadAfzal,MuhammadA slam3,Roohl Jan,Martinez-Enriquez A.M.," Minimizing Big Data Problems using Cloud Computing Based on HadoopArchitecture",Page(s)-99-103

[18] Imran Khan,S. K. Naqvi,S. N. A Rizvi,MansafAlam," Data Model for Big Data in Cloud Environment", 2015 2nd International Conference on Computing for Sustainable Global Development,Pages-582-585

[19] K. Jens, D. R. Graeme and K. Stefan, "Swarm Intelligence in Animalsand Humans", Trends in Ecology and Evolution, vol. 25, no.1, pp.28-34, 2010

[20] J. K. Parrish and W. M. Hamner. Editors, "Animal Groups in ThreeDimensions", Cambridge University Press, 1997

[21] J. D.Murray, "Mathematical Biology", Springer-Verlag, New York,1989

[22] S. Gueron, S. A. Levin and D. I. Rubenstein, "The dynamics of herds:From Individuals to Aggregations", Journal of Theoretical Biology,vol. 182, pp. 85-89, 1996

[23] E. Bonabeau, M. Dorigo and G. Theraulaz, "Swarm Intelligence:From Natural to Artificial Systems", Oxford Univ. Press, New York,1999

[24] C. W. Reynolds, "Flocks, Herds, and School: A DistributedBehavioral Model", In Comput. Graph.Proc. Of ACMSIGGRAPH'87, vol. 21, pp. 25-34, Jul.1987

[25] T.Vicsek, A. Czirook, E.Ben-Jacob, I. Cohen, and O. Shochet, "NovelType of PhaseTransition in a System of Self-Deriven Particle", Phys.Rev. Lett., vol.75, no.6, pp.1226-1229, 1995