Review Article *ISSN: 2319-507X*
Prashant Bhalge, IJPRET, 2013; Volume 1(8): 674-686 *IJPRET*

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

A REVIEW: RECENT TRENDS IN CLOUD COMPUTING

SHEETAL RAM PADMAWAR, PRASHANT BHALGE

1. Department of Computer Science and Engg. Babasaheb Naik college of Engineering and Technology, Pusad, Yavatmal , Maharashtra 445204

2. Asst. Professor, Department of Computer Science and Engg. Babasaheb Naik college of Engineering and Technology, Pusad.

Accepted Date:

27/02/2013

Publish Date:

01/04/2013

Keywords

Cloud Computing

PaaS

SaaS

IaaS

Piracy

Big Data

Hadoop

MapReduce

Dataset storage

Corresponding Author

Mr. Prashant Bhalge

Abstract

Today, we're surrounded by data like oxygen. The exponential growth of data first presented challenges to cutting-edge businesses such as Google, Yahoo, Amazon, Microsoft, Facebook, Twitter etc. Cloud computing offers distributed and shared computing resources and services that belong to different service providers and websites. Cloud computing model has enabled IT organizations to serve the users globally. Data volumes to be processed by cloud applications are growing much faster than computing power. This growth demands new strategies for processing and analyzing information. Hadoop-Map Reduce has become a powerful Computation Model addresses to these problems. Hadoop HDFS became more popular amongst all the Big Data tools as it is open source with flexible scalability, less total cost of ownership & allows data stores of any form without the need to have data types or schemas defined. Hadoop Map Reduce is a programming model and software framework for writing applications that rapidly process vast amounts of data in parallel on large clusters of compute nodes. In this paper I have provided an review of recent trends in cloud computing which includes, Hadoop, HCFS (Hadoop Cluster File System) and Map Reduce programming model As in File Access System Service the users are not aware of the locations of the files in Clouds. File access seems to them as a single coherent file system. Considering this aspect, this paper is concerned with an effective service and cost model for providing Cloud Computing service for writing and compiling source code remotely from any hand held device using the distributed service model. According to this model, for providing a service, the service provider may subsequently use the services of other service providers in the Cloud without the awareness of the client. The model is expected to be mutually beneficial to the client and its service provider. The client is paying only for the service it receives and the service provider only provides the service that is requested by the client .The cost is evaluated based on the kind of service provided. The model is also expected to reduce software piracy to a considerable extent.

1. INTRODUCTION

Cloud computing involves delivering hosted services over the Internet on demand. Cloud computing is scalable and managed infrastructure. End-users simply consume these services and pay on usage basis or subscription basis. There are three famous service models of cloud computing as described below:

- Software as a Service (SaaS).
- Platform as a Service (PaaS):
- nfrastructure as a Service (IaaS)

Cloud computing has been driven fundamentally by the need to process an exploding quantity of data. One critical trend shines through the cloud is Big Data. When a company needed to store and access more data they had one of two choices. One option would be to buy a bigger machine with more CPU, RAM, disk space, etc. This is known as scaling vertically. Of course, there is a limit to how big of a machine you can actually buy and this does not work, when you start talking about internet scale. The other option would be to scale horizontally. This usually meant contacting some database vendor to buy a bigger solution. These solutions do not come cheap and therefore required a significant investment. Dealing with big datasets in the order of terabytes or even petabytes is a challenging. In Cloud computing environment a popular data processing engine for big data is Hadoop-MapReduce due to ease-of-use, scalability, and failover properties. Hadoop-MapReduce programming model consists of data processing functions: Map and Reduce . HDFS supports reliability and fault tolerance of MapReduce computation by storing and replicating the inputs and outputs of a Hadoop job.

Cloud Computing is a service model where services are providing to the end users with flexible and scalable services available through the internet [1]. The service can be a software, hardware, infrastructure or platform. This is a device independent model because it resources can be accessed not just from any computer on the internet, but also any type of machines, provided that it has an internet connection and web browser.

Several technologies are related to cloud computing, and the cloud has emerged as a convergence of several computing trends.

Cloud computing is a paradigm shift to computing [2] that sees and delivers computing as a service rather than as a resource. Cloud computing encapsulates several layers of computing provisioning that include the hardware resources located at the data centres of cloud providers, the operating system and virtualization software on top of that hardware, and the applications that are delivered as services over the internet. These services are provided as utility to customers who are billed based on usage, similar to the billing scheme of traditional public services such as electricity, telephone and water.

1.1. Theoretical baselines

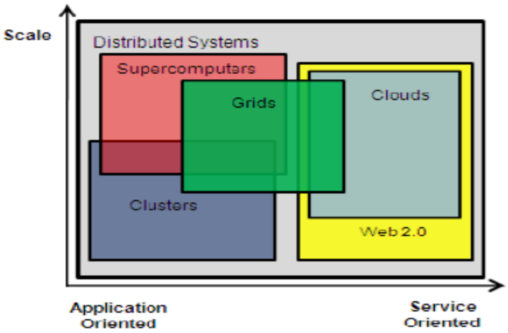
Cloud computing today is the beginning of "network based computing" over Internet in force. It is the technology of the decade and the beginning to the end of the dominance of desktop computing such as that with the Windows. It is also the beginning of a new Internet based service economy: the Internet centric, Web based, on demand, Cloud applications and computing economy [3].

1.2. Distributed Computing

Distributed computing refers to the very idea of using distributed systems that are generally multiple computers connected to each other via computer networks to collaboratively process a common goal. Those computers communicating can be homogeneous or heterogeneous, distributed globally or locally. According to the characteristics of localization or equality, distributed systems have different subsets, such as supercomputers, grids, clusters, web 2.0 and clouds [4].

Before going further into the subsets of distributed computing, an illustration is provided to visualize the interconnection between the concepts that will be explained in the following Figure 1: Distributed Computing and its subsets.

Figure 1: Distributed Computing and its subsets [5].



Every participating machine in a distributed system is able to download the piece of software and then interconnect to a centralized server. The servers provide the input from their sensors with a huge amount of data, making the calculation of this data normally very complicated. While a single computer would not be able to do that in an appropriate amount of time, millions of computers that are interconnected anywhere in the world would be able to achieve that.

1.2.1 Clusters

Characteristics of clusters are that the computers being linked to each other are normally distributed locally, and have the same kind of hardware and operating system. Therefore cluster work stations are connected together and can possibly be used as a super computer [6].

1.2.2 Supercomputers

Supercomputers can be easily compared to clusters, because it follows the same concept, except the fact that it is merged into one box already and is not locally interconnected with other machines [6]. IBM is constructing those machines consisting with a lot of processors that are merged into one machine with high performance capabilities [7]. The only disadvantage is that they are usually expensive and have the necessity of a huge amount of energy.

1.2.3 Grids

When defining grid computing it is necessary to differ it from clusters. Grids involve heterogeneous computers that are connected to each other and distributed globally. The OS and hardware that run on those machines can also be different from each other [6].

The computers that are interconnected over the internet can come from anywhere while there is usually no obligation to pay. For this reason already it is obvious that grids being connected are not nearly as expensive as the supercomputers that are offered from IBM and other technology companies.

1.2.4 Clouds

Together with virtualization, clouds can be defined as computers that are networked anywhere in the world with the availability of paying for the used clouds in a pay-per-use way, meaning that just the resources that are being used will be paid for [8].

1.3 Service models

 Software as a Service (SaaS). In this model, software application is hosted as service and end-users use the application on the web browser.

 Platform as a Service (PaaS): In this model, end-user creates, test and upload applications using tools and libraries hosted by the service provider.

 Infrastructure as a Service (IaaS): This model involves hosting of hardware computing services like storage, hard-drive, servers and network components. Service provider is responsible for maintenance and managing all these resources.

2. CLOUD COMPUTING RELATED TECHNOLOGIES

Several technologies are related to cloud computing, and the cloud has emerged as a convergence of several computing trends. Clouds extend the capabilities of other domains with the specific goal to achieve scalability/ elasticity, availability with optimal resource utilisation, which is as such only partially addressed in other domains. Specifically, clouds belong to the wider areas of Internet of Services (including Web Services, Web3.0, Service Oriented Architecture (SOA) etc.) and Utility Computing (including Grid, Virtual Organisations etc.) and implicitly inherit multiple aspects from these domains, such as virtualisation and outsourcing. Depending on usage, they may extend these characteristics, in particular by adding a new business model.

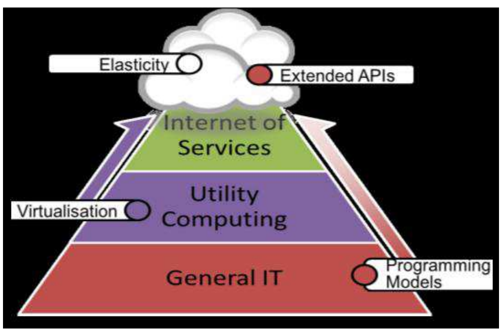


Figure 2: Inheritance and Extension of Characteristics across the Related Domains [9]

We refer Figure 2 here in particular to three related domains, though the list can be easily extended and refined:

 Internet of Service: covering areas such as Web Services, Web 3.0, and SOA etc. In other words, individual service provisioning without specific means for dealing with availability, that is, just network load balancing.

 Utility Computing: including Grids, Virtual Organisations, and also High Performance Computing (HPC) to some degree, as they were originally conceived and realised.

 General IT or more correctly “non-web” IT, including all computer science aspects concerning isolated machines, i.e. without making explicit use of the web. This includes theory of computation, hardware architectures, operating systems etc.

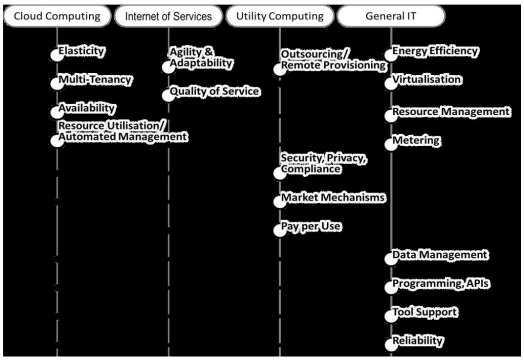


Figure 3: Overview of the Cloud Characteristics and their Relationship to other Domains. Inheritance is from Right to Left. The Further Left is Characteristics Advances, the more it needs to be adapted for the respective domain(s), [9].

More specifically, the characteristics identified above can be classified with respect to the domains they were originally conceived in and to the ones which take up/extend them (see Figure 3). The Columns denote the domain to which the concepts apply, so that any entry within an according column implies that the respective characteristic is specifically adjusted and or extended to meet the domain‟s requirements. In other words, single column entries imply that all domains to the left inherit the characteristics without significant adaptations,domains to the right do not support this characteristic.

If an entry spans multiple columns, it means that the base concept is [10] conceived in the right-most domain and that all domains to the left adapt the concept to their respective needs.

Cloud defines a four-layed architecture

Cloud infrastructure typically rely on Web forms (over SSL) to create and manage account information for end-users, and allows users to reset their passwords and receive new passwords via Emails in an unsafe and unencrypted communication.

Figure 4: Cloud Computing [3]

3. BIG DATA

Big data is a collection of data sets so large and complex which is also exceeds the processing capacity of conventional database systems. The data is too big, moves too fast, or doesn‟t fit the structures of our current database architectures. Big Data is typically large volume of un-structured (or semi structured) and structured data that gets created from various organized and unorganized applications, activities and channels such as emails, tweeter, web logs, Facebook, etc. Figure 5 provides an example of big data. The main difficulties with Big Data include capture, storage, search, sharing, analysis, and visualization.

Figure 5. Example of Different Sources of Big Data

3.1 Big Data Processing Methods

3.1.1 Hadoop

Hadoop is a batch processing system for a cluster of nodes that provides the underpinnings of most BigData analytic activities because it bundle two sets of functionality most needed to deal with large unstructured datasets nsmely, Distributed file system and MapReduce processing. it is a project from the Apache Software Foundation written in Java to support data intensive distributed applications. Hadoop enables applications to work with thousands of nodes and petabytes of data. Hadoop‟s biggest contributor has been the search giant Yahoo, where Hadoop is extensively used across the business platform. Hadoop is an umbrella of sub-projects around distributed computing and although is best known for being a runtime environment forMapReduce programs and its distributed filesystem HDFS, the other sub-projects provide complementary services and higher level abstractions, as shown in Figure 6 below. Some of the current sub-projects are as Core, MapReduce, HBase, Pig, ZooKeeper, Hive, Chukwa,HCatalog.

Figure 6. High Level Architechture of Hadoop

3.1.2 Hadoop distributed file system (HDFS)

An HDFS cluster has two types of node operating in a master-worker pattern: a NameNode (the master) and a number of DataNodes (workers). The namenode manages the filesystem namespace. It maintains the filesystem tree and the metadata for all the files and directories in the tree. The namenode also knows the datanodes on which all the blocks for a given file are located. Datanodes are the workhorses of the filesystem. They store and retrieve blocks when they are told to (by clients or the namenode), and they report back to the namenode periodically with lists of blocks that they are storing. Name Node decides about replication of data blocks. In a typical HDFS, block size is 64MB and replication factor is 3 (second copy on the local rack and third on the remote rack). The Figure 7 shown architecture distributed file system HDFS.

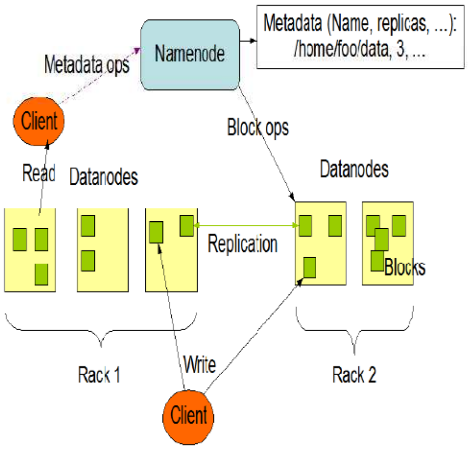


Figure 7. Hadoop Distributed Cluster File System Architechture

Hadoop MapReduce applications use storage in a manner that is different from general-purpose computing. To read an HDFS file, client applications simply use a standard Java file input stream, as if the file was in the native filesystem.

3.1.3 Map-Reduce

MapReduce is a data processing or parallel programming model introduced by Google. In this model, a user specifies the computation by two functions, Map and Reduce. In the mapping phase, MapReduce takes the input data and feeds each data element to the mapper. In the reducing phase, the reducer processes all the outputs from the mapper and arrives at a final result. Map Reduce has gained a great popularity as it gracefully and automatically achieves fault tolerance. It automatically handles the gathering of results across the multiple nodes and returns a single result or set. MapReduce model advantage is the easy scaling of data processing over multiple computing nodes.

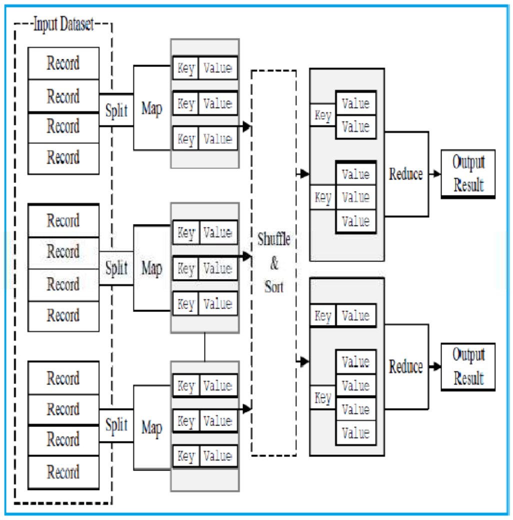


Figure 8. Execution process of MapReduce Programming Model

3.1.3.1 Execution Process in MapReduce Programming Model

In MapReduce programming model and a MapReduce job consists of a map function, a reduce function, and When a function called the below steps of actions take place.

 MapReduce will first divide the data into N partitions with size varies from 16MB to 64MB

 Then it will start many programs on a cluster of different machines. One of program is the master program; the others are workers, which can execute their work assigned by master. Master can distribute a map task or a reduce task to an idle worker.

 If a worker is assigned a Map task, it will parse the input data partition and output the key/value pairs, then pass the pair to a user defined Map function. The map function will buffer the temporary key/value pairs in memory. The pairs will periodically be written to local disk and partitioned into P pieces. After that, the local machine will inform the master of the location of these pairs.

 If a worker is assigned a Reduce task and is informed about the location of these pairs, the Reducer will read the entire buffer by using remote procedure calls. After that, it will sort the temporary data based on the key.

 Then, the reducer will deal with all of the records. For each key and according set of values, the reducer passes key/value pairs to a user defined Reduce function. The output is the final output of this partition.

 After all of the mappers and reducers have finished their work, the master will return the result to users' programs. The output is stored in F individual files.

4. A DISTIBUTED SERVICE AND BUSINESS MODEL FOR PROVIDING CLOUD COMPUTING SERVICE

4.1 Objective

It is the intention to develop an approach to build a versatile and effective service model for providing cloud computing service for compiling source code remotely from any hand held device (having limited resources) using the distributed service model. According to this model, for providing a service, the service provider may subsequently use the services of other service providers in the cloud without the awareness of the client. The cost model effectively evaluates the cost which each client pays to each of its service providers for receiving the services. The approach will provide services to the end user having limited configuration in their hand held devices. The Cloud service provider will provide its services through web and the consumer will pay for using the services. In it, one service provider may subsequently use the services of other service providers without the awareness of consumers. The model is expected to be mutually beneficial to the client and its service provider.

4.2 Proposed Approach

The approach is to develop a suitable Cloud Computing service and cost model for providing cloud computing service (like Compilation service, file service etc.)

SERVICE MODEL

The steps of this model are summarized as follows:-

Step 1: In accordance with the service model the client (CL) requests for a service from a suitable service provider (SP).

Step 2: The service provider on receiving the request ensures whether it is possessing sufficient resource to provide the service requested by its client.

Step 3: If the service provider finds that it possesses sufficient resource to provide the service then it provides the service to its client directly else

{ if the service provider finds that it does not possess all the necessary resource then it finds suitable service providers and requests those service providers to provide it with the required resource. Again the steps from Step 2 onwards are repeated for each service provider requested (for providing service).

5. CONCLUSION

Cloud computing is the on-demand utilization of shared computing resources available from the Internet. When these services are used properly, they can reduce cost and management responsibilities in addition to increasing efficiency and performance of an enterprise.

BigData is still in its early infancy. The size of these datasets suggests that exploitation may well require a new category of data storage and analysis systems with different architectures. Hadoop-MapReduce programming paradigm have a substantial base in the Big Data community due to the cost-effectiveness on commodity Linux clusters, and in the cloud via data upload to cloud vendors who have implemented Hadoop/HBase. HDFS, the Distributed File System, is a distributed file system designed to hold very large amounts of data (terabytes or even petabytes), and provide high-throughput access to these information.

The proposed approach comprising of service and cost model for providing Cloud computing service for writing and compiling source code remotely from any hand held device using the distributed service model is expected to be mutually beneficial to the client and its service provider. This model is also expected to reduce software piracy to a considerable extent thereby helping the vendors in boosting up their business,In this paper, I have conducted a review of recent trends in cloud computing.

REFERENCES

1. Yashpalsing Jadeja, Kirit Modi "Cloud computing concepts, architecture and challenges", *International Conference on Computing, Electronics and Electrical Technologies [ICCEET] 2012, Kumaracoil,* pp.877-880, 2012.

2. Voas, J., & Zhang, J. (2009). Cloud Computing: New Wine or Just a New Bottle? *IT Professional*, 15-17.

3. Srinivasa et al. ().Cloud Computing: An Overview. Journal of Theoretical and Applied Information Technology, vol. pp. 71-79. www.jatit.org.

4. Foster I. et al. (2008). Cloud Computing and Grid Computing 360-Degree Compared. IEEE Grid Computing Environment Workshop (GCE „08), pp 1-10.

5. Foster I. et al. (2008). Cloud Computing and Grid Computing 360-Degree Compared. IEEE Grid Computing Environment Workshop (GCE „08), pp 1-10.

6. Buyya, R., Yeo, C., Venugopal, S., Broberg, J. & Brandic, I. (2009). Cloud Computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems,* 25, 599-616.

7. IBM (2009). IBM continues TOP500 lead. *IBM Systems.* Available at: http://www

03.ibm.com/systems/deepcomputing/top500.html#a1. Accessed on: 17 September, 2012.

8. Armbrust, M., Fox, A. & Griffith, R. (2009). Above the Clouds: A Berkeley View of Cloud Computing. *Electrical Engineering and Computer Sciences University of California at Berkeley, pp.*1-8.

9. Lutz S & Keithe J. (2012): Advances in Clouds: Research in Future Cloud Computing, Expert Group Report. Information Society and Media, European Union, Belgium.

10. Azeem Sarwar*, Muhammad Naeem Ahmed Khan* (2013)

11. A Review of Trust Aspects in Cloud Computing Security. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)* Vol.2, No.2, April 2013, pp. 116~122 ISSN: 2089-3337.

12. Rabi Prasad Padhy (2013). Big Data Processing with Hadoop-MapReduce in Cloud Systems. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)* Vol.2, No.1, February 2013, pp. 16~27 ISSN: 2089-3337.

13. Shreya Bhadra*, Tirthankar Gayen** (2013). A Distributed Service and Business Model for Providing Cloud Computing Service. *International Journal of Cloud Computing and Services Science*

(CLOSER) Vol.2, No.2, April 2013, pp. 148~158 ISSN: 2089-3337.

14. Abah Joshua*, Francisca N. Ogwueleka** (2013). Cloud Computing with Related Enabling Technologies. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)* Vol.2, No.1, February 2013, pp. 40~49 ISSN: 2089-2333.