Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2015, 2(2):99-109**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

A Comparison of Cloud Computing and Big Data Applications with Software Engineering Prospective

Shabir Ahmad[1]*, Bilal Ehsan[1], Muhammad Perbat Baloch[2], Shahida Siddiq[2], Abu Buker Siddique[2]

[1]Department of Computer Sciences, Government College of Commerce, Multan, Pakistan
[2]Department of Computer Sciences, The Islamia University of Bahwalpur, Pakistan

Abstract Software engineering in Cloud computing is an essential aspect for obtaining a systematic, disciplined and quantifiable approach to the development, operation, and maintenance of software and services. Incorporating security and privacy during the engineering process is of vital importance for assuring the development of reliable, correct, robust and trustful systems as well as adaptive and evolving software services that satisfy users' requirements. To this extend the need to investigate methods and tools that will assist developers in constructing more reliable privacy-oriented information systems and services in cloud environments, is fully justified. Big Data delivers game-changing benefits using a new approach to data acquisition, management, and visualization for the emerging Big Data platforms. Organizations have always produced significant amounts of unstructured data from sources such as medical images, blogs, radio-frequency identification (RFID) tags, and locality sensors. Historically, organizations threw away most of the data they could collect to avoid what were once considered excessive costs of managing such a data deluge. In this paper, different aspects of cloud computing application and big data applications are compared. In addition the researcher also provides the possible solutions of these applications with respect to software engineering. This paper will help the new researchers to compare the cloud computing and big data applications and will lead to develop new solutions for the existing inadequacies.

Keywords Cloud Computing, Big Data, Requirement Engineering, Software Process, Modeling, Implementation

1. Introduction

Today, the most popular applications are Internet services with millions of users. Websites like Google, Yahoo! and Face book receive millions of clicks daily. This generates terabytes of invaluable data which can be used to improve online advertising strategies and user satisfaction. Real time capturing, storage, and analysis of this data are common needs of all high-end online applications. To address these problems, a number of cloud computing technologies have emerged in last few years.

Developers with innovative ideas for Internet services no longer need large capital outlays in hardware to deploy their services; this paradigm shift is transforming the IT industry. The operation of large scale, commodity computer datacenters was the key enabler of cloud computing, as these datacenters take advantage of economies of scale, allowing for decreases in the cost of electricity, bandwidth, operations, and hardware [1].

This paper provides a comprehensive background study for scalable data management and analysis. It further focuses on a set of systems which are designed to handle update heavy workloads for supporting internet facing applications. This paper identifies some of the design challenges which application and system designers face in developing and deploying new applications and systems, and expand on some of the major challenges that need to be addressed to ensure the smooth transition of applications from traditional enterprise infrastructure to the next generation cloud infrastructure.

2. Literature and Comparison

This paper, first explores the various frameworks which provide the resources to make computing at an immense scale and then discuss the confidentiality, security and integrity challenges presented by this infrastructure.

2.1. Software Process and Life Cycle Models

In this section the cloud and big data software process and life cycle models are discussed with respect to software engineering.

2.1.1. Cloud Software Process and Life Cycle Models

A cloud security independence employment framework is proposed [2] (Cloud SSDLC), especially from government and industry perspectives. The cloud SSDLC integrates the secure system development life cycle (SSDLC), critical domain guideline for cloud security, and risk concerns. There are five main phases in Cloud SSDLC, initiation, development, implementation, operation, and destruction. Furthermore, cloud security critical domains and corresponding risks are integrated into each phase. From the industry and government perspective, different cases are used to demonstrate practical usage and legal issues in the proposed Cloud SSDLC.

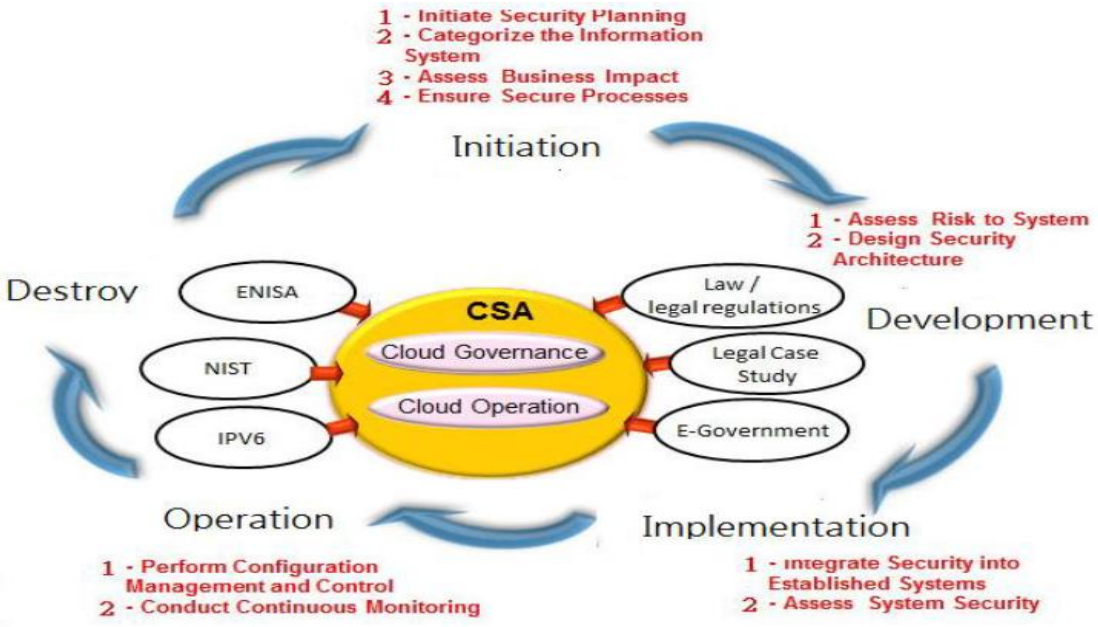

Figure 1: *SSDLC Cloud Computing Model*

2.1.2. Big Data Software Process and Life Cycle Models for Big Data

With the digital technologies proliferation into all facets of business activities, industry and business are entering a fresh playground where they should use scientific solutions to take advantage of the newest opportunities to get and mine data for desirable information, such as for instance market prediction, customer behavior predictions, social groups activity predictions, etc.

Refer to varied blog articles [3] suggesting that the Big Data technologies have to adopt scientific discovery methods including iterative model improvement and number of improved data, reuse of collected data with improved model.

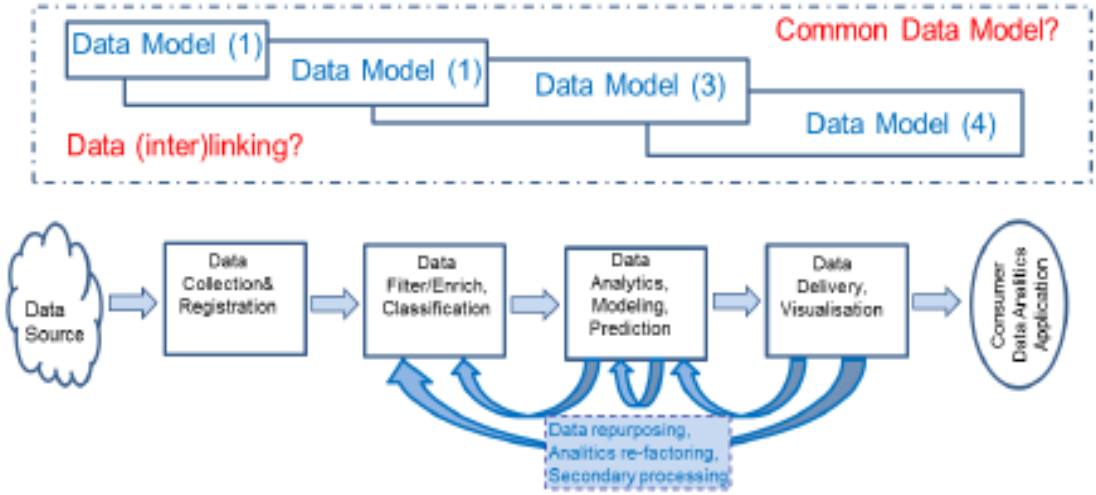

Figure 2: *Big Data Lifecycle in Big Data Ecosystem*

Paper makes reference to the Scientific Data Lifecycle Management model described inside our earlier paper (http://www.forrester.com/pimages/rws/reprints/document/85601/oid/1-LTEQDI) and was an interest for detailed research in another work that reflects complex and iterative procedure for the scientific research that features several consequent stages: research project or experiment planning; data collection; data processing; publishing research results; discussion, feedback; archiving.

2.2. Software Requirements Engineering

This section presents the cloud computing and big data software requirement engineering.

2.2.1. Requirements Engineering for Cloud Computing

Data-intensive systems like cloud computing encompass huge amount of data in terabytes to petabytes (online http://wgiss.ceos.org/dsig/whitepapers/Data%20Lifecycle%20Models%20 and%20Concepts%20v8.docx). So systems need require very big storage and exhaustive computational power to execute queries quickly. To analyzes the extensive requirements the state-of-art paper [4-5], describes various challenges associated with extensive requirements and suggest numerous solutions in meeting these requirements. Figure 3 illustrates the 2 architectural models for this type of system.

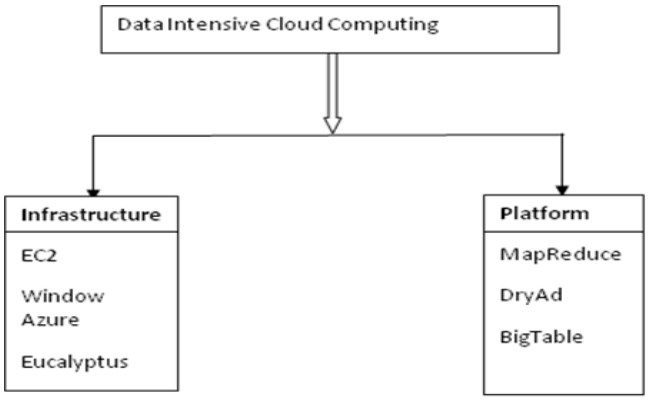

Figure 3: *Data Intensive Cloud Computing*

In the former case, an individual is needed to select tools and a platform for computing, and the cloud provider is accountable for storage and computing power. The provider can also be liable for replication, fault tolerance, and consistency.

Table 1: The following table, enlist the cloud computing requirements, challenges and their possible solutions.

S. No.	Requirements	Challenges	Solutions
1.	Scalability: Cloud should be able to support large no. of users with efficiency.	A cloud must have effective management, proper utilization of resources and mechanism for task-mapping.	Distributed file systems such as (GFS), Hadoop Distributed File Systems (HDFS) [6] Programming Platforms like **MapReduce,** Distributed Storage and Database Systems BigTable are good options
2.	Availability, Fault Detection, and Fault Tolerance	In big data clouds, faults may lead to failure or crashes. There is need of mechanism to deal with these in timely manner.	Kahuna [7] is a fault detection tool. HiTune [8] is used for data flow performance
3.	Flexible and Efficient User Access	In MapReduce strict any task can be converted to map and reduce tasks.	Dryad [9] supports thousands of nodes for large operations. Sawzall [10], is a high performance computing System suitable for MapReduce.
4.	Elasticity	Adjustments are required in clouds for low and high load which is supported by VMs. During physical migration an elastic load balancing Challenges arise.	ElasTras [11] provides elasticity to allocate/de-allocate resources on demand. Zephyr [12] adds to the capabilities of Elastree for live migration.
5.	Sharing of a Clusters for Multiple Platforms	For Hadoop and Dryad, resource usage information is required.	Otus [13] is a tool which monitors the data-intensive applications behavior in a cluster. Mesos [14], provides sharing capability for multiple frameworks on the same cluster.
6.	Disk Head Scheduling	In a shared environment multiple workloads may reduce the speed of the disk I/O. Challenges may arise for multiple clusters.	A disk-scheduling scheme [15], which co-schedules data across all servers in the cluster.
7.	Heterogeneous	The execution speed of tasks varies in In heterogeneous	LATE [16], a scheduling algorithm

	Environment	systems	which identifies slow tasks and prioritizes them according to their expected completion time
8.	Data Handling, Locality, and Placement	Data-analysis servers For data-intensive applications is also critical which may affect the application performance.	Volley [17] is an automatic data placement tool.
9.	Effective Storage Mechanism	For Data-intensive systems leads to high disk usage. This leads to 200% extra utilization of disk space.	DiskReduce [18] is focused to reduce this overhead. It reduces the disk usage between 10 and 25 %..
10.	Privacy and Access Control	Large storage systems require interactions between multiple users. This requirement introduces additional challenges of procurement and management of access controls between the users.	In [19], the authors proposed a model which provides dynamic delegation of rights with capabilities for accounting and access confinement.
11.	Billing	Incorporating accurate billing mechanism is significant and challenging for data intensive cloud systems. With massive requirements to access and compute huge amount of data, appropriate and methods are needed to compute billing.	A fair billing system for data intensive computing entails three components [20]. These include: • Cost of Data Storage • Cost of Data Access • Cost of Computation Of these three components, cost related to computation is normally billed in CPU hours.
12.	Power Efficiency	For scalable systems, power requirements are likely to be increased due to addition of resources. Multicore systems are also being utilized for clouds. Reducing power usage for such systems is also desirable.	FAWN [21] is a flash-memory based system, which is designed to promote low power for data intensive applications requiring random access. Advancements in multi-core technology have lead to their utilization in cloud systems. Shang and Wang [22] have proposed power saving strategies for multi-core data intensive systems.
13.	Network Problems	With low cost switches and top of the rack setup, the buffer of the switch may become full and	To solve the TCP in cast problem, [23] proposed that TCP Retransmission Time Out (RTO) be

		packet loss may result. Barrier synchronized scenarios could encounter TCP Incast problem due to which long delays might occur.	reduced. Through real experiments, the authors observed that microsecond timeouts allowed servers to scale up to 47 in barrier synchronized communication environment.

2.2.2. Requirements Engineering for Big Data

The key elements those are considered in requirement engineering for big data are as follows:

Scalability: To accommodate very large and growing data stores, including the ability to easily add additional storage resources as needed.
High performance: To keep response times and data ingest times low, thus keeping up with the required pace of the business.
High efficiency: To reduce storage and related data center costs.
Operational simplicity: To streamline the management of a massive data environment without additional IT staff.
Enterprise data protection: To ensure high availability for business users and business continuance in the event of a disaster.
Interoperability: To integrate very complex environments and to provide an agile infrastructure that supports a wide range of business applications and analytics platforms such as Hadoop.

2.3. Software System Modeling and Architectural Design

This section presents the cloud computing and big data software system modeling and architectural design.

2.3.1. Cloud Computing Open Architectural Design

For having a reusable and customizable Cloud Computing Open Architecture [24, 25] for enabling application development environment is just a key to success of application sharing within the Internet. This paper tries to limit the scope of the open system to a specialized domain, and leverage service-oriented thinking to simply help modularize open architecture for Cloud Computing. The portion of paper presents a Cloud Computing Open Architecture predicated on seven principles.

The presented Cloud Computing Open Architecture covers cloud ecosystem enablement, cloud infrastructure and its management, service-orientation, cloud core on provisioning and subscription, compostable cloud offerings, cloud information architecture and management, and cloud quality analytics. This can be a logical and modularized separation, which supports isolate concerns of details of every module during the look process. Because the connections involving the identified key architectural principles for Cloud Computing can be complex, the data exchanges are getting through the Cloud Information Architecture and Cloud Ecosystem Management.

2.3.2. Big Data Software System Modeling and Architectural design

Big data are normally distributed both on the collection side and on the processing/access side: data have to be collected (sometimes in an occasion sensitive way or with other environmental attributes), distributed and/or replicated. Linking distributed data is one of many problems to be addressed by Big Data structures and underlying infrastructure.

We are able to mention as the key motivation The European Commission's initiative to guide Open Usage of scientific data from publicly funded projects suggests introduction of the next mechanisms allowing linking publications and data [26, 27] PID - persistent data ID ORCID – Open Researcher and Contributor Identifier (online http://www.openaire.eu/).

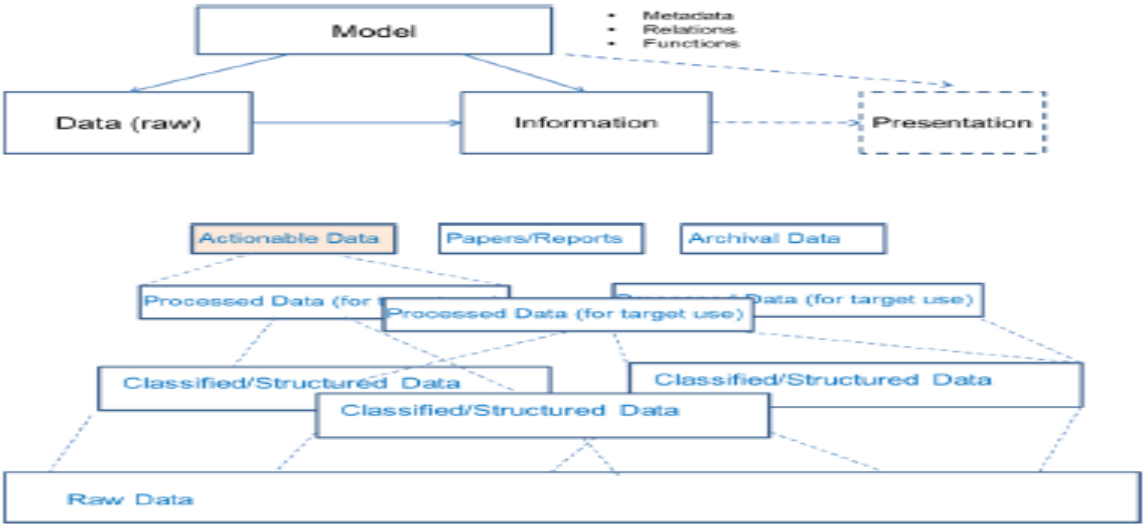

Figure 4: *Big Data Models and Architecture*

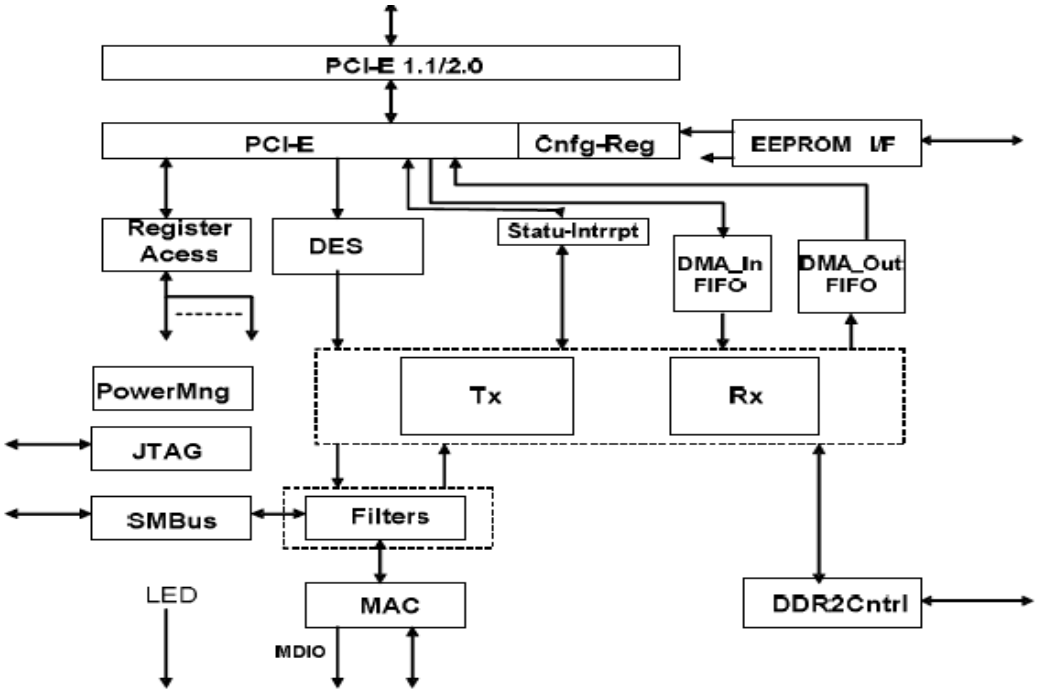

Figure 5: *Cloud Computing-Oriented virtual 10-Gigabit NIC hardware architecture*

2.4 Software Design and implementation

This section presents the cloud computing and big data software design and implementation.

2.4.1 Design and Implementation of a Cloud Computing-Oriented Virtual 10-Gigabit NIC

Cloud platform and services have become the conventional services for supercomputing era [28]. The changes in cloud technologies will greatly affect the industrial technological and industrial competition, and will strong promote the regional science and technology innovation and speed the pace of regional economic restructuring. Virtualization of resources is the key technology of cloud computing. From the point of view the evolution of

virtualization, the stronger performance of the server, the number of virtual machines can run more, and we need more number of NIC ports. The promotion of multi-core server platform and the popularity of cloud computing applications, making the number of applications for the Ethernet port is also rapid growth in demand. Such as five years ago, most servers had a single-core processor, could carry a very limited number of virtual machines, and using virtual machine to integration services application is not realistic, therefore, the demand for the NIC ports is not very high. With the popularity of cloud computing, the application of quad-core processors and even eight-core processor, even on the two-way server can also provide 64 cores, the number of virtual machines can be run on the server enhanced greatly, therefore, we need more NIC ports [29, 30].

2.4.2 Big Data Software Design and implementation

GrayWulf [31] cluster, which is high level system is shown in Figure. In this computing farm is separated from computer resources and shared query able data stores.

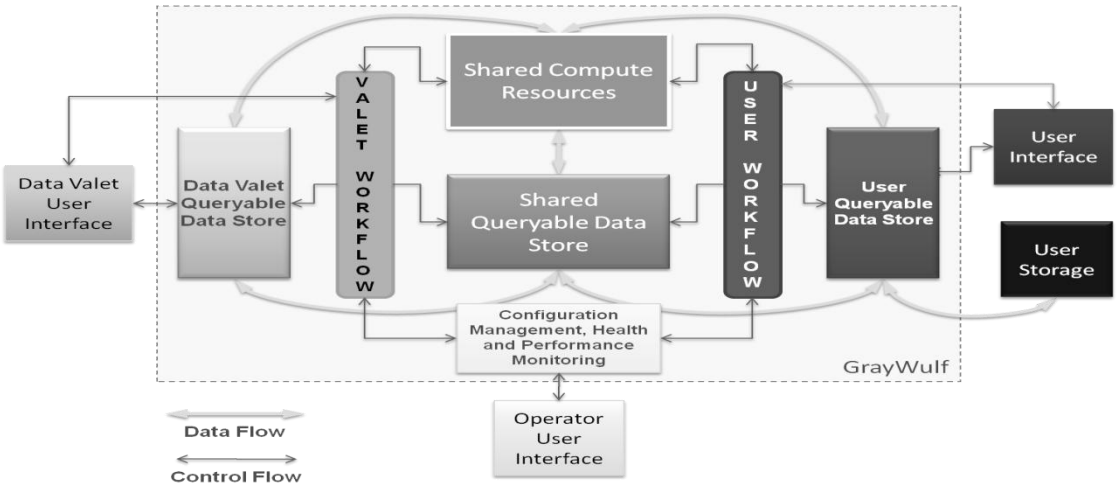

Figure 6: *Big Data Software Design*

The data resources are accessed by three different groups:

Users who perform analyses on the shared database.

Data valets who maintain the contents of the shared databases on behalf of the users.

Operators who maintain the compute and storage resources on behalf of both the users and data valets.

2.5 Software Testing

This section presents the cloud computing and big data software testing.

2.5.1 Software Testing Based on Cloud Computing

Cloud testing is the method of software testing based on cloud computing technology [32, 33]. The paper describe, the definition of cloud testing was derived from the concept of cloud computing. Cloud testing is another type of programming testing in which web applications that utilization distributed computing situations try to mimic true client movement as a method for complex testing and anxiety testing sites. With cloud-testing you have unlimited assets available to you, paying just for what you expend, just when and on the off chance that you consume it.

2.5.2 Software Testing of Big Data

Testing of Big Data is among the biggest challenges faced by organizations as a result of insufficient knowledge about what to try and simply how much data to test. Organizations have already been facing challenges in beginning the test strategies for structured and unstructured data validation, establishing a maximum test environment, dealing with non-relational databases and performing non-functional testing. These challenges are causing in low quality of data in production and delayed implementation and upsurge in cost. Robust testing

approach need to be denied for validating structured and unstructured data and start testing early to identify possible defects early in the implementation life cycle and to reduce the overall cost and time to market [34, 35].

Different testing types like functional and non-functional testing are required along with strong test data and test environment management to ensure that the data from varied sources is processed error free and is of good quality to perform analysis. Functional testing activities like validation of map reduce process, structured and unstructured data validation, data storage validation are important to ensure that the data is correct and is of good quality.

As we're coping with huge data and executing on multiple nodes you will find high chances of getting bad data and data quality issues at each stage of the process. Data functional testing is conducted to spot these data issues as a result of coding errors or node configuration errors. Testing ought to be performed at all the three phases of Big data processing to ensure data gets processed without the errors. Functional Testing includes (i) validation of pre-Hadoop processing; (ii), validation of Hadoop Map Reduce process data output; and (iii) validation of data extract, and load into EDW. Besides these functional validations non-functional testing including performance testing and failover testing must be performed. Figure 7 shows a normal Big data architecture diagram and highlights the areas where testing ought to be focused.

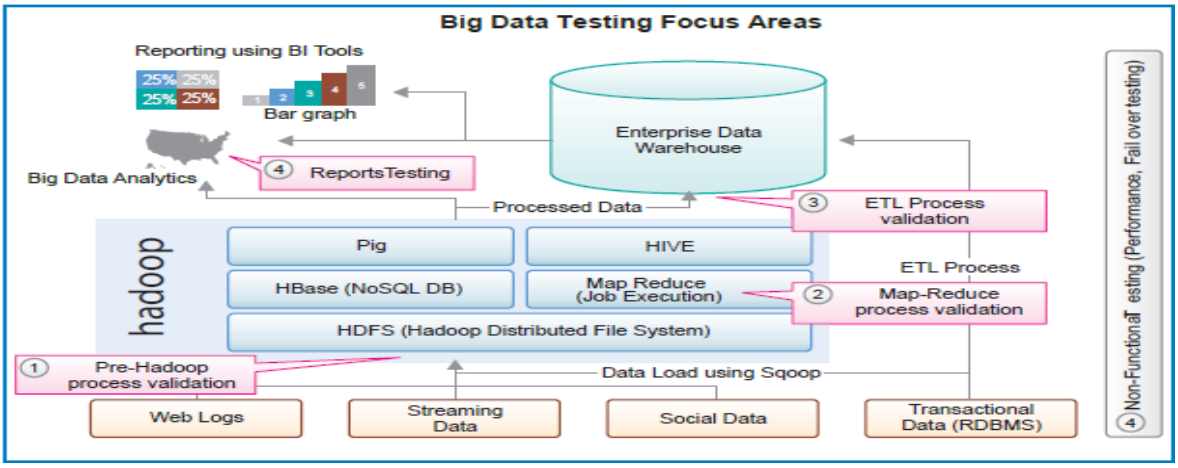

Figure 7: *Big Data Testing Focus Area*

3. Conclusion

The cloud provides the infrastructure necessary to provide services directly to customers over the Internet and the way the data is utilized come under big data. This comparison studied various stages/frameworks in detail, the different software engineering aspects for cloud computing and big data applications.

The limitation of this study is that the researchers studied only the stages from architecture, design, implementation and security point of view. Several challenges remain there, even though most of these frameworks come with solutions, they are naive and not very effective. Security remains the biggest barrier preventing companies from entering into the cloud. Moreover, dynamic scheduling in heterogeneous environments is still an issue in these frameworks.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*,*53*(4), 50-58.
2. J. B. Rothnie Jr., P. A. Bernstein, S. Fox, N. Goodman, M. Hammer, T. A. Landers, C. L. Reeve, D. W. Shipman, and E. Wong. Introduction to a System for Distributed Databases (SDD-1). *ACM Trans. Database Syst.*, 5(1):1–17, 1980.

3. D. J. Dewitt, S. Ghandeharizadeh, D. A. Schneider, A. Bricker, H. I. Hsiao, and R. Rasmussen. The Gamma Database Machine Project. *IEEE Trans. on Knowl. and Data Eng.*, 2(1):44–62, 1990.
4. Jawwad Shamsi, Muhammad Ali Khojaye, Mohammad Ali Qasmi; Data-Intensive Cloud Computing: Requirements, Expectations, Challenges, and Solutions. In: Springer (2013).
5. SIDDIQ, Shahida, et al. "Implementation Issues of Agile Methodologies in Pakistan Software Industry." *International Journal of Natural and Engineering Sciences* 8.3 (2014): 43-47.
6. Grossman, R., Gu, Y.: On the varieties of clouds for data intensive computing. In: IEEE Data Engineering (2009)
7. Bu, Y., Howe B., Balazinska, M., Ernst, M.: Hadoop: efficient iterative data processing on large clusters. J. Proceedings VLDB Endowment **3**(1–2), 285–296 (2010).
8. Tan, J., Pan, X., Kavulya, S., E. Marinelli, E., Kavulya, S., Gandhi, R., Narasimhan, P.: Kahuna: Problem diagnosis for MapReduce-based cloud computing environments. In: 12th IEEE/IFIP NOMS (2010)
9. Dai, J., Huang, J., Huang, S., Bo Huang, B., Liu, Y.: HiTune: dataflow-based performance analysis for big data cloud. In: Usenix HotCloud (2011)
10. Isard, M., Budiu, M., Yu, Y., Birrell, A., Fetterly, D.: Dryad: distributed data-parallel programs from sequential building blocks. In: ACM SIGOPS/Eurosys (2007)
11. Pike, R., Dorward, S., Griesemer, R., Quinla, S.: Interpreting the data: parallel analysis with Sawzall. Sci. Program. J. (Special Issue on Grids and Worldwide Computing Programming Models and Infrastructure) **13**(4), 227–298
12. Das, S., Agrawal, D., Abbadi, A.: ElasTras: an elastic transactional data store in the cloud. In: Usenix Hotclud (2009)
13. Elmore,A., Das, S., Agrawal, D.,Abbadi,A.: Zephyr: live migration in shared nothing databases for elastic cloud platforms. In: ACM SIGMOD (2011)
14. Ren, K., López, J., Gibson, G.: Otus: resource attribution in data-intensive clusters. In: Mapreduce (2011)
15. Hindman, B., Konwinski, A., Zaharia,M., AliGhodsi, A., Joseph, A., Katz, R., Scott Shenker, S., Stoica, I.: Mesos: a platform for fine-grained resource sharing in the data center. In: Usenix NSDI (2011)
16. Wachs, M., Ganager, G.: Co-Scheduling of disk head time in cluster-based storage. In: IEEE SRDS (2009)
17. Zaharia, M., Konwinski, A., Joseph, A., Katz, R., Stoica, I.: ImprovingMapReduce performance in heterogeneous environments. In: Usnix OSDI (2008)
18. Agrawal, S., Dunagan, J., Jain,N., Saroiu, S., Wolman, A., Bhogan, H.: Volley: Automated data placement for geo-distributed cloud services. In: Usenix NSDI (2010)
19. Fan, B., Tantisiriroj, W., Xiao, L., Gibson, G.: DiskReduce: RAID for data-intensive scalable computing. In: PDSW Super Computing (2009)
20. Harnik, D., Kolodner, E., Ronen, S., Satran, J. Shulman-Peleg, A., Tal, S.: Secure access mechanisms for cloud storage. In: 2nd Workshop on Software Services: Cloud Computing and Services: Cloud Computing and Applications based on Software Services (2011)
21. Banker, K.: MongoDB in Action. Manning Publications (2012)

22. Andersen, D., Franklin, J., Kaminsky, M., Phanishayee, A., Tan, L., Vasudevan, V.: FAWN: a fast array of wimpy nodes. In: Communications of the ACM (2011)
23. Shang, P., Wang, J.: A novel power management for CMP systems in data-intensive environment. In: Parallel & Distributed Processing Symposium (IPDPS) (2011)
24. Vasudevan, V., Amar Phanishayee, A., Shah, H., Krevat, E., Andersen, D., Ganger, G., Gibson, G., Mueller, B.: Safe and effective fine-grained TCP retransmissions for datacenter communication. In: ACM SIGCOMM (2009).
25. Khan Kamran et al. "Evaluation of PMI's Risk Management Framework and Major Causes of Software Development Failure in Software Industry". *IJSTR* 3 (11): 120-124, 2014.
26. Ali Arsanjani, Liang-Jie Zhang, Michael Ellis, Abdul Allam, Kishore Channabasavaiah, "S3: A Service-Oriented Reference Architecture," IT Professional, vol. 9, no. 3, pp. 10-17, May/Jun, 2007.
27. Azam, Farooq, et al. "Framework Of Software Cost Estimation By Using Object Orientated Design Approach." *IJSTR* **3**(11): 97-100, 2014.
28. Liang-Jie Zhang and Qun Zhou, *IBM T.J. Watson Research Center, New York, USA*, IEEE International Conference on Web Services (2009)
29. B. Hayes. Cloud Computing. Commun. ACM,51(7):9{11, 2008. Available at http://doi.acm.org/10.1145/1364782.1364786.
30. Baloch, Muhammad Perbat, et al. "Comparative Study Of Risk Management In Centralized And Distributed Software Development Environment." Sci.Int.(Lahore),26(4),1523-1528, 2014.
31. Wang Jun and Fanpeng Meng, 2011 International Conference on Internet Computing and Information Services (IEEE 2011).
32. Afaq Salman et al. "Software Risk Management In Virtual Team Environment**"**. *IJSTR* 3 (12): 270-274, 2014.
33. Hussain, Shafiq, et al. "Threat Modelling Methodologies: A Survey." *Sci.Int.(Lahore)*,26(4),1607-1609, 2014.
34. Ahmad, Shabir, and Bilal Ehsan. "The Cloud Computing Security Secure User Authentication Technique (Multi Level Authentication)." *IJSER* 4(12): 2166-2171 (2013).
35. Siddique, Abu Buker, et al. "Integration of Requirement Engineering with UML in Software Engineering Practices" *Sci.Int.(Lahore)*, 26(5), 2157-2162, 2014.