iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** VI **Month of publication:** June 2018

DOI: http://doi.org/10.22214/ijraset.2018.6083

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VI, June 2018- Available at www.ijraset.com

Active Storage System

Ritesh Mishra[1] Shivam Soni[2,] Prateek Bansal[3]
[1, 2, 3]*Bharati Vidyapeeth Deemed University College of Engineering,Pune INDIA*

Abstract: The active storage system collects and then implements the application code for large amounts of data by using the unused processing power of the storage nodes. For computer applications, the performance might not be that good due to less processing power of the storage devices. It provides a method to reduce the data transfer between storage system and the parallel file system nodes. Various data processing operations can be implemented directly on the parallel file systems storage nodes. This is made possible by the general purpose server and the operating system using memory resources of the underlying processor and storage nodes

I. INTRODUCTION

Improvements in the field of storage technologies such as high storage capacity and cost effectiveness, and have made it possible to build systems of high power by integrating thousands of compute nodes and storage nodes. However, for large-scale simulations that use the nodes, the efficient management of increasing volumes of data still remains a challenging problem because the transfer cost of moving the data between the processing nodes and the storage devices has not improved as compared to the disk storage capacity. The approach to minimize the bandwidth requirements between storage and computing devices that to move the process of computation closer to the storage devices. One way to take the advantage of the underutilized CPU in the storage nodes is that to extend the concept of traditional active disk to the parallel file systems of high performance architectures. This approach in the context of the parallel file systems is called ACTIVE STORAGE. There are two important differences with respect to the old storage device mechanism. They are:

A. The modern storage devices are now full-fledged computers,
B. Linux operating system includes a feature-rich environment

Active Storage process makes it possible to reduce the movement of data between the computing device and storage node hence, it manages the overall network traffic. By performing data processing at the source node, there is no need of transfer of data between file system servers and computer nodes. Active storage often targets the applications at I/O stages that involves independent data sets. For example, it can be used to process on-line or off-line data, output files from scientific simulation runs. Some examples active storage system includes: By implementing the above mentioned operations for the storage nodes, we not only achieve the benefits with respect to the resource usage but it can also have huge impact on the scientific data usage patterns by the storage node and it could be achieved by implementing input/output tasks which are necessary for the application to work. As we know the concept of Active Storage is widely used, a scalable and flexible mechanisms needs to be developed that allows applications to use various algorithms and data processing patterns.

II. ARCHITECTURE

For the modern day computing systems, basically there are two ways to deploy storage and compute the nodes. The first way is that we deploy storage nodes from the compute nodes separately. It provides highly parallel I/O and is used mainly where there is a need high performance from the computing system. The second approach is that to implement storage as well as compute operations on the same node, which is used in case of Map Reduce and Hadoop file system. The processing Kernels are designed in such a way that various components of the storage system can function independently. They are called by the Active Storage (AS) component which is a helping process when it comes to active storage of input/output operations is done. The abstraction of the local input/output API is done from parallel file systems. It provides a method that abstracts local and reads local data for the Kernels to process the data.

A. The Dynamic Active Storage System Works as Follows

1) First the applications pass input/output requests to the Active Storage Client and then the Active Storage Client analyses the transfer cost of the operation by analysing how the requested data and data pattern are distributed within a storage node.
2) If the particular operation requires more bandwidth than expected, the request will be served as normal request instead of an active storage request.

©IJRASET: All Rights Reserved

3) If the request is considered as an active storage request, then it gets transferred to the storage nodes through parallel I/O APIs.
4) In the meantime, the dynamic active storage calculates how the data is being distributed by taking into consideration the data dependence. Then arranges the data in such a way that data movement is minimized among the storage servers. A helper process is then called process kernels to carry out static computations.

III. APPLICATIONS

Active Storage media can be used in many ways. Some of the following applications include:

A. *Redundant Array of Independent disks (Raid)*

Redundant Array of Independent Disks (RAID) is a data storage virtualization technique that combines many levels of physical disk drive components into one or more logical units.

B. *The Main Goal of RAID is*

1) To minimize the data redundancy of the storage system
2) Improve performance of the storage system or both.

Distribution of data across the RAID levels, depends upon the required level of redundancy and performance. The different data distribution levels, are called as “RAID" followed by a level number, for example RAID 0 or RAID 1. Each RAID level offers a different goals such as reliability, availability, performance, and capacity. The data is place on multiple disks in RAID technology and it allows the input/output operations to overlap in a balanced way and thus improves performance of the storage system. Because RAID uses multiple disks there is also an increase in the mean time between failures (MTBF and thus data redundancy increases fault tolerance. In a single-user system where there are large set of records, such as medical or other scientific images, , the RAID stripes are assigned smaller values such as 512 bytes. This is done so that a single record can be easily be accessed by reading data from all the disks at the simultaneously. In a multiuser system, there is a better performance as compare to the single user system and it requires that the RAID stripe should be wide enough to hold maximum record. Thus the concept of allows overlapped disk input/output operations are employed the across drives.

C. *Network-Attached Storage (NAS)*

Network-attached storage (NAS) is a file storage server that is connected to the physical device by providing access to the by a heterogeneous group of clients.NAS systems have one or more storage drives that are arranged in logical storage containers form or RAID.NAS storage system is not responsible for the files that operates on other servers using the same network. Various network file sharing protocols are NFS, SMB/CIFS, or AFP. A NAS computer unit is connected to a network that enables only file-based storage services to other devices on the network. Although it is technically possible to run other software on a NAS unit, but it is not designed to functions as a general-purpose server.

For example, NAS units are controlled and configured over the network, using a browser and do not have a keyboard or display

For personal use, NAS storage is often used for storing multimedia files and for automated backups. Many smart homes technologies employ NAS providing centralized storage for smart devices, surveillance systems and other Internet of Things (IOT) and other components of the home.

D. *Storage Area Network (SAN)*

A Storage Area Network (SAN) is a high-speed network through which access is given to block-level network connected to storage device. SAN systems comprises hosts, switches and storage devices that are interconnected using a various technologies and protocols.

A SAN first establishes local connection between the storage devices and the host storage device. The connection of storage device to a host is accomplished by the use of different types of virtualization such as desktop virtualisation, server virtualisation, and network virtualisation etc.

The main functions of a storage area network (SAN) includes the following:

1) A network of high-speed storage devices.
2) It establishes connections the storage devices and servers.
3) Applications on networked servers can access SAN.
4) It provides data backup and disaster recovery.
5) Management tasks can also be simplified using SAN.

©IJRASET: All Rights are Reserved

6) Flexibility, efficiency, reliability availability and performance are some of the key features of SAN.

IV. MECHANISMS

A. Remote procedure call (rpc)

In RPC program on one computer on a network makes a remote call in order fetch to a program on another computer on the same/different network without having the network's details. The RPC protocol is a protocol that is used for point-to-point communication within or between software applications. An RPC is also called a subroutine call.

1) The client calls a client stub passes parameters to the stub.
2) The client stub packs the parameters in form of a message. Packing includes converting parameters into a predefined format and also copy each parameter in form of a message.
3) Then the message is send to the transport layer by the stub, which is again sent to the remote server of another machine.
4) The transport layer of the server then passes the message to server stub, which unpacks the parameters and then server routine is called and further processing is done using the regular procedure call mechanism.
5) When the server process completes, the parameters are passed to the server stub, which then packs the parameters in form of a message. Then the message is handed to the transport layer by the server stub.
6) The transport layer of the server sends the result back to the transport layer of the client, which then passes the message back to the client stub.
7) The client stub unpacks the return parameters and executes the process.

B. Ibm Netezza

IBM Netezza is a tool that is used to design high-performance data warehouse applications and modern analytics applications which is used for enterprise data warehousing, business intelligence, predictive analysis and continuity of business. designed to handle very large queries from multiple users. It is divided into two tiers, we will discuss them in detail. The first tier is a Linux SMP host and the function of first tier is to compile data and query tasks that are being received from business applications and then it generates query execution plans. Then the query is divided into a series of sub-tasks that are to be executed in parallel and then finally the sub task are distributed second tier for execution. In the second tier there are hundreds of sub tasks processing blades or S-Blades, where all the primary processing work is carried out. The S-Blades makes up massively parallel processing (MPP) engine of the appliance and thus they are called the intelligent processing nodes. Each S-Blade for a device is made up of an independent server that contains multi-core Intel-based CPUs, Netezza's proprietary multi-engine and FPGAs. The S-Blade is combined with a special Netezza Database Accelerator card that snaps alongside the blade. Each S-Blade is, in turn, connected to multiple disk drives processing multiple data streams in parallel in Twin Fin or Skimmer. AMPP employs industry-standard interfaces (SQL, ODBC, JDBC, and OLE DB) and provides load times in excess of 2 TB/hour and backup/restore data rates of more than 4 TB/hour.

C. Orchestration

Cloud orchestration uses the programming technology and is used to manage the interconnections s on both public as well as private cloud infrastructure. Its main function is to connect the automated tasks into a sequence of workflow to in order to achieve a goal and also taking care of the permissions and policy enforcement guidelines. Typically the role of Cloud orchestration is following

1) Deployment or start of new servers.
2) Acquiring and assigning storage capacity for each disks
3) Manage various networking operations/processes, issues.
4) Creation of Virtual Machines (VMs).
5) Gaining access to various software on cloud services.

Cloud orchestration technology should always work with heterogeneous systems globally deploying a cloud network in different geographical locations and also with different providers.

D. Thus the Advantages of Using Cloud Orchestration Are

1) High availability rates
2) Better scalability

©IJRASET: All Rights are Reserved

3) Failure recovery
4) Dependent management system

V. CHALLENGES

Most of the programs that are used in computers are sequential in format. These programs are developed by many developers. They are professional, well trained and experienced and are able to understand the developing program. However this is not the case with parallel program developers as they are difficult to find, self-learned and are also not the well trained professional developers. Thus there is a large gap between parallel program and sequential program developers and we need to bridge this gap. This is possible by giving that the professional training to the parallel program developers. This will enhance their skills and make them expert in the respective domain. The second problem with the active storage systems is of compatibility such as hardware as well as software compatibility. For example the parallel program written for storage system with six core processor will not perform efficiently on different number of core processor. The third problem is a lack of tools and technique for the process of debugging and testing the active storage system program and the tools need to be developed for testing, debugging as per the requirement and for this purpose a lot of the research work is required.

VI. CONCLUSION

In this chapter we have discussed the general characteristics of active storage systems, arrangement ways, applications and mechanisms. Despite the progress of computer science in field of processor, network and storage areas, the management of large amount of data still remains a challenging problem in many areas of computing. Thus taking the advantage of the underutilized CPU time of the storage nodes in modern parallel file systems, the concept of Active Storage comes into role. It aims at performing the simple processing of the data that is stored in parallel file systems. The paper describes on how to implement Active Storage concept based on a user-space approach and also compares it to the existing kernel-space method of implementation. The results obtained show that both the method of implementations are able to scale up the performance of data-based applications and also reduce the overall traffic of the network simultaneously. In addition to the high-performance, the user space method implementation seems to be more flexible, portable, and readily deployable as compared to the kernel-space method. Active storage also provides a methodology for the applications by moving the computations tasks from compute nodes to storage nodes. Thus Active storage system provides an efficient way to improve the overall performance of high-performance computing tasks. With the gain in popularity of new storage technologies, such as solid-state drives (SSD), the use and implementation hybrid active storage systems have become possible and also feasible.

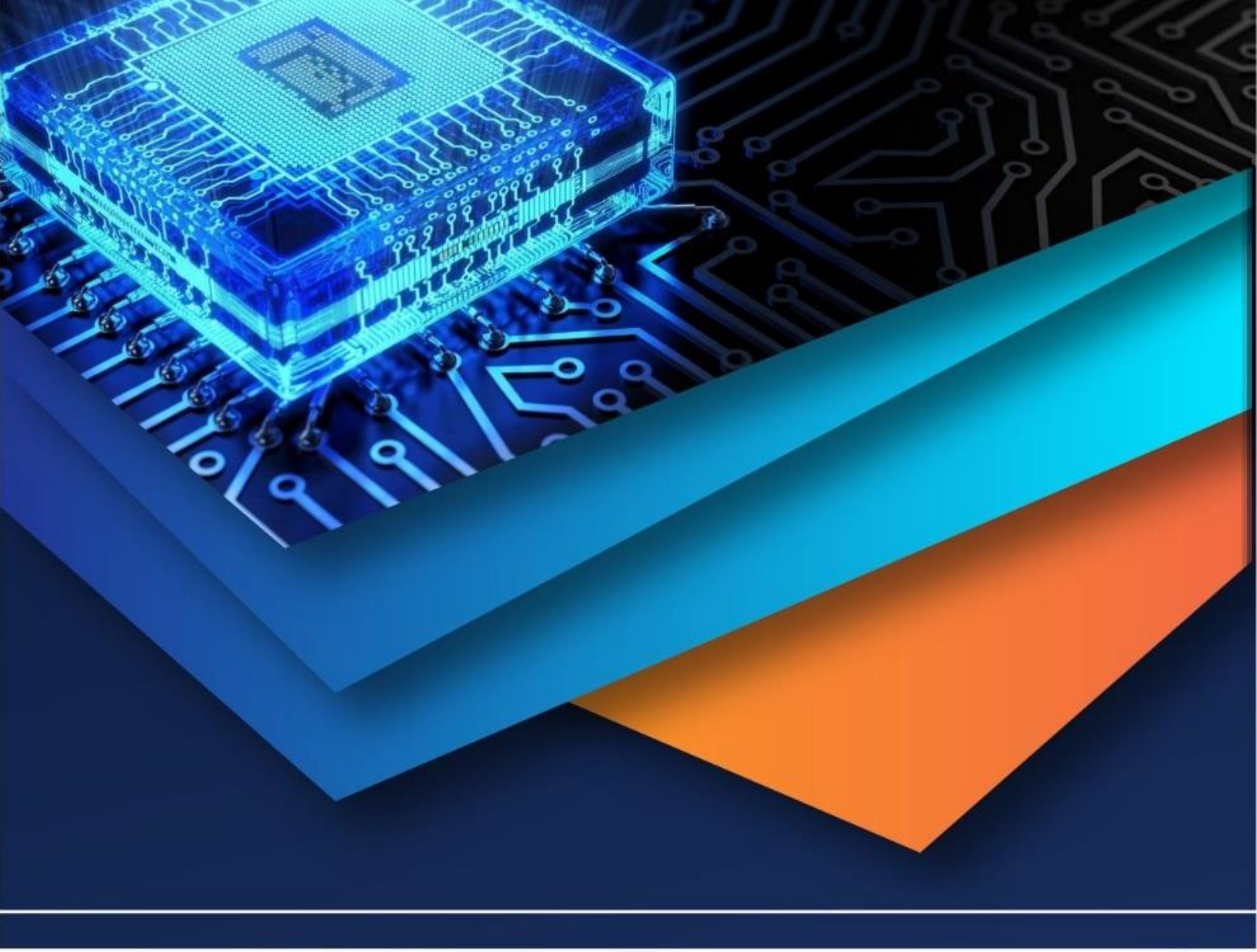
REFERENCES

[1] Archana, R. C., Naveenkumar, J., & Patil, S. H. (2011). Iris Image Pre-Processing And Minutiae Points Extraction. International Journal of Computer Science and Information Security, 9(6), 171.
[2] Bhore, P. R., Joshi, S. D., & Jayakumar, N. (2016). A Survey on the Anomalies in System Design: A Novel Approach. International Journal of Control Theory and Applications, 9(44), 443–455.
[3] Bhore, P. R., Joshi, S. D., & Jayakumar, N. (2017a). A Stochastic Software Development Process Improvement Model To Identify And Resolve The Anomalies In System Design. Institute of Integrative Omics and Applied Biotechnology Journal, 8(2), 154–161.
[4] Bhore, P. R., Joshi, S. D., & Jayakumar, N. (2017b). Handling Anomalies in the System Design: A Unique Methodology and Solution. International Journal of Computer Science Trends and Technology, 5(2), 409–413.
[5] Desai, P. R., & Jayakumar, N. K. (2017). A Survey on Mobile Agents. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 5(XI), 2915–2918.
[6] Gawade, M. S. S., & Kumar, N. (2016). Three Effective Frameworks for semi-supervised feature selection. International Journal of Research in Management & Technology, 6(2), 107–110.
[7] GAWADE, S., & JAYKUMAR, N. (2017). ILLUSTRATION OF SEMI-SUPERVISED FEATURE SELECTION USING EFFECTIVE FRAMEWORKS. Journal of Theoretical & Applied Information Technology, 95(20).
[8] Jaiswal, U., Pandey, R., Rana, R., Thakore, D. M., & JayaKumar, N. (2017). Direct Assessment Automator for Outcome Based System. International Journal of Computer Science Trends and Technology (IJCS T), 5(2), 337–340.
[9] Jayakumar, D. T., & Naveenkumar, R. (2012). SDjoshi, International Journal of Advanced Research in Computer Science and Software Engineering, Int. J, 2(9), 62–70.
[10] Jayakumar, M. N., Zaeimfar, M. F., Joshi, M. M., & Joshi, S. D. (2014). INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY (IJCET). Journal Impact Factor, 5(1), 46–51.
[11] Jayakumar, N. (2014). Reducts and Discretization Concepts, tools for Predicting Student's Performance. International Journal of Engineering Science and Innovative Technology (IJESIT), 3(2), 7–15.
[12] Jayakumar, N. (2015). Active storage framework leveraging processing capabilities of embedded storage array.
[13] Jayakumar, N., Bhardwaj, T., Pant, K., Joshi, S. D., & Patil, S. H. (n.d.). A Holistic Approach for Performance Analysis of Embedded Storage Array.
[14] Jayakumar, N., Iyer, M. S., Joshi, S. D., & Patil, S. H. (2016). A Mathematical Model in Support of Efficient offloading for Active Storage Architectures. In International Conference on Electronics, Electrical Engineering, Computer Science (EEECS) : Innovation and Convergence (Vol. 2, p. 103).

©IJRASET: All Rights are Reserved

[15] Jayakumar, N., & Kulkarni, A. M. (2017). A Simple Measuring Model for Evaluating the Performance of Small Block Size Accesses in Lustre File System. Engineering, Technology & Applied Science Research, 7(6), 2313–2318.
[16] Jayakumar, N., Singh, S., Patil, S. H., & Joshi, S. D. (n.d.). Evaluation Parameters of Infrastructure Resources Required for Integrating Parallel Computing Algorithm and Distributed File System.
[17] KAKAMANSHADI, M. G., NAVEENKUMAR, M. J., & PATIL, S. H. (2011). A METHOD TO FIND SHORTEST RELIABLE PATH BY HARDWARE TESTING AND SOFTWARE IMPLEMENTATION. International Journal of Engineering Science.
[18] Kulkarnia, A., & Jayakumar, N. (2016). A Survey on IN-SITU Metadata Processing in Big Data Environment. International Journal of Control Theory and Applications, 9(44), 325–330.
[19] Kumar, N., Angral, S., & Sharma, R. (2014). Integrating Intrusion Detection System with Network Monitoring. International Journal of Scientific and Research Publications, 4, 1–4.
[20] Kumar, N., Kumar, J., Salunkhe, R. B., & Kadam, A. D. (2016). A Scalable Record Retrieval Methodology Using Relational Keyword Search System. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies (p. 32).
[21] kumar Singha, A., Patilb, S. H., & Jayakumar, N. (2017). A Treatment for I/O Latency i n I/O S tack. Http://Www.Ijcstjournal.Org/Volume-5/Issue-2/IJCST-V5I2P83.Pdf.
[22] Namdeo, J., & Jayakumar, N. (2014). Predicting Students Performance Using Data Mining Technique with Rough Set Theory Concepts. International Journal of Advance Research in Computer Science and Management Studies, 2(2).
[23] Naveenkumar, J. (2012). Keyword Extraction through Applying Rules of Association and Threshold Values. International Journal of Advanced Research in Computer and Communication Engineering, 1(5), 295–297. Retrieved from http://www.ijarcce.com/upload/july/3-Keyword Extraction.pdf
[24] Naveenkumar, J., & Joshi, S. D. (2015). Evaluation of Active Storage System Realized Through Hadoop. International Journal of Computer Science and Mobile Computing, 4(12), 67–73.
[25] Naveenkumar, J., Makwana, R., Joshi, S. D., & Thakore, D. M. (2015a). OFFLOADING COMPRESSION AND DECOMPRESSION LOGIC CLOSER TO VIDEO FILES USING REMOTE PROCEDURE CALL. Journal Impact Factor, 6(3), 37–45.
[26] Naveenkumar, J., Makwana, R., Joshi, S. D., & Thakore, D. M. (2015b). Performance Impact Analysis of Application Implemented on Active Storage Framework. International Journal, 5(2).
[27] Naveenkumar, J., & Raval, K. S. (2011). Clouds Explained Using Use-Case Scenarios. In INDIACom-2011 Computing For Nation Development (pp. 1–5).
[28] Naveenkumar J, P. D. S. D. J. (2015). Evaluation of Active Storage System Realized through MobilityRPC. International Journal of Innovative Research in Computer and Communication Engineering, 3(11), 11329–11335.
[29] NAVEENKUMAR, M. J., Bhor, M. P., & JOSHI, D. R. S. D. (2011). A Self Process Improvement For Achieving High Software Quality. International Journal of Engineering Science, 3.
[30] RAVAL, K. S., SURYAWANSHI, R. S., NAVEENKUMAR, J., & THAKORE, D. M. (2011). The Anatomy of a Small-Scale Document Search Engine Tool: Incorporating a new Ranking Algorithm.
[31] Rishikesh Salunkhe, N. J. (2016). Query Bound Application Offloading: Approach Towards Increase Performance of Big Data Computing. Journal of Emerging Technologies and Innovative Research, 3(6), 188–191.
[32] Salunkhe, R., Kadam, A. D., Jayakumar, N., & Joshi, S. (n.d.). Luster A Scalable Architecture File System: A Research Implementation on Active Storage Array Framework with Luster file System.
[33] Salunkhe, R., Kadam, A. D., Jayakumar, N., & Thakore, D. (n.d.). In Search of a Scalable File System State-of-the-art File Systems Review and Map view of new Scalable File system. In nternational Conference on El ectrical, Electronics, and Optimization Techni ques (ICEEOT) - 2016 (pp. 1–8).
[34] Sawant, Y., Jayakumar, N., & Pawar, S. S. (2016). Scalable Telemonitoring Model in Cloud for Health Care Analysis. In International Conference on Advanced Material Technologies (ICAMT) (Vol. 2016).
[35] Singh, A. K., Pati, S. H., & Jayakumar, N. (2017). A Treatment for I/O Latency in I/O Stack. International Journal of Computer Science Trends and Technology (IJCS T), 5(2), 424–427.
[36] Zaeimfar, S. D. J. N. J. F. (2014). Workload Characteristics Impacts on file System Benchmarking. Int. J. Adv, 39–44.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)