IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 8, Issue 11, November 2019

Comparing Tradeoff Between Offloading of Data and Offloading of Function

Priyanka Wadagave[1]

M. Tech Student, Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, Pune, India[1]

Abstract: Storage Area Network and Cloud Computing are the domains which deal with various storage related computations that lead to better storage management. Storage management techniques for homogeneous and heterogeneous systems aredifferent. Much heterogeneous system uses offloading of various logical parameters and also provides intermediates such as interface engines in case of luster computing to achieve better system performance. while using such system lead to problems such as offloading of resources, large compute time, synchronization required between server and storage devices, large waiting time required for I/O operations, offloading large amount of data and various computational function etc. there are various approaches have been proposed to solve as system base or resource base application. Active storages etc. however, upgrading the system performance often required in existing solutions. The system performance can be improved by increasing speedup, decreasing latency period, etc. To achieve this we have come with a comparative study of tradeoff between.

1) Offloading of data and
2) Offloading of compute function

This will lead in better decision making in order to achieve increased system performance. The tradeoff between these two deals with throughput, resource complexities, I/O time and bandwidth etc. The further advancements can be done in this field by studying more number of parameters and comparing them with respect to offloading of data and offloading of compute functions.

Keywords: Active storages, compute, homogeneous, heterogeneous, interface engine, luster computing, offloading

I. INTRODUCTION

Applications in storage area network are I/O intensive because of their requirement of vast data to be accessed and processed. There is an ever growing gap between the performance of compute and I/O access rates with the current technology of disk drives[6]. Even though if raw capacity hardware like processors and storage are available, it becomes more vital to understand the complexity of scaling and parallelism in order to provide software which increases the performance of the applications in high performance computing.[6].

Now days a large class of applications has been developed and deployed in almost all the domains. These applications generate enormous amount of data and process it. These kinds of applications generating, storing, retrieving and processing huge data are called data intensive applications. Since these applications have to interact with the I/O subsystems frequently they are sometimes also regarded as I/O intensive applications[6]. There are many emerging technologies in the storage field like software defined storage, storage virtualization etc., but every time they are not sufficient to give solution to existing problems[6].

There are many solutions to address such problems out of which we are focusing only on data offloading and application offloading. One drawback with these solution is based upon application requirement the decision is taken whether to offload data or application. The primal idea of this paper is to provide a solution to make decision of above problem by considering various system and application requirement

II. RELATED WORK

Response time is the total amount of time it takes to respond to a request for service. That service can be anything from a memory fetch, to a disk IO, to a complex database query, or loading a full web page. Ignoring transmission time for a moment, the response time is the sum of the service time and wait time. The service time is the time it takes to do the work you requested. For a given request the service time varies little as the workload increases – to do X amount of work it always takes X amount of time. The wait time is how long the request had to wait in a queue before being serviced and it varies from zero, when no waiting is required, to a large multiple of the service time, as many requests are already in the queue and have to be serviced first.

Response time is given as,

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 11, November 2019

$$t_r = \frac{A_{t_s}}{1 - Utilization}$$

Where, tr = response time
Ats= Array service time

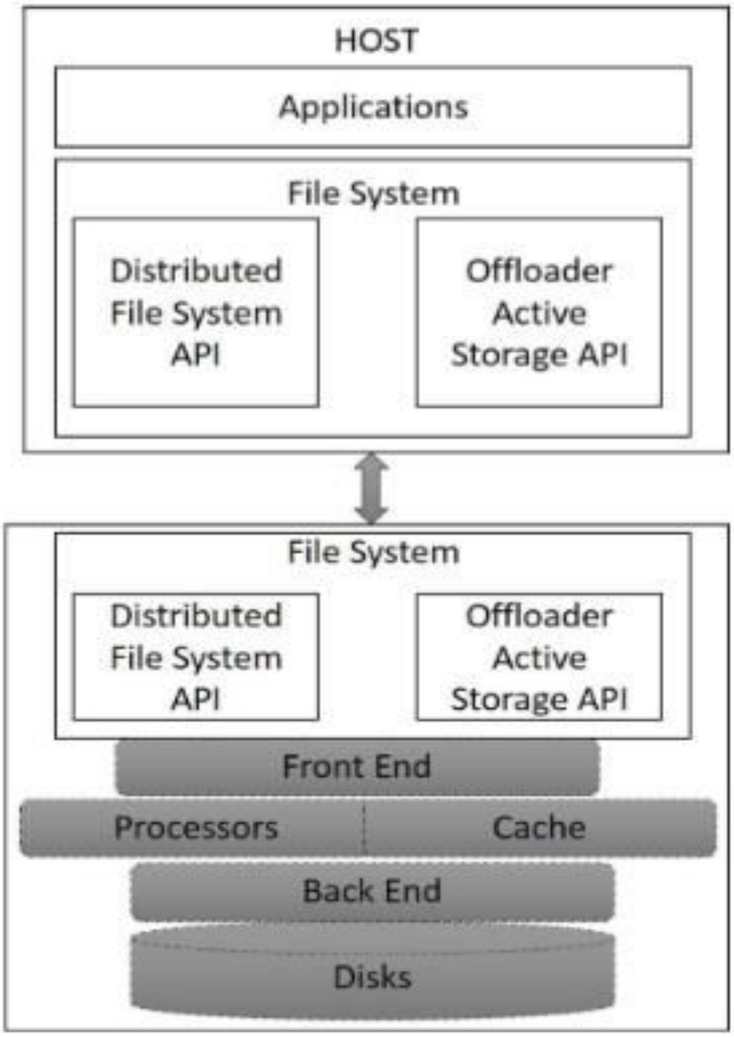

Fig 1: offloading Model

The CPU clock frequency will be constant, thus capacity is the product of number of cores and time interval for they are used. The required capacity is calculated by the product of interval of time CPU's are required and Number of CPU's consumed. The total capacity is the product of total number of CPU's and the total interval time **%CPU utilization = required CPU capacity / Total CPU capacity *100**
Capacity-utilization rate is a measure of what percentage of capacity a business is currently performing at. The formula for capacity-utilization rate is actual output divided by the potential output.

III. CONCLUSION

This paper discusses about the key factors, using which decides when to offload data and when application is offloaded to the storage array. Using these factors or performance metrics, it can be decided that either the functions or tasks if offloaded will improve efficiency and performance of the storage array or not. The metrics include % CPU utilization,execution time and storage space. These metrics elaborated in the above section will guide us in designing and developing the offloading logics that can be deployed in the active storage architectures or frameworks whereby, the resources can be utilized to their efficiency by offloading being permissible or not.

REFERENCES

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 11, November 2019

[1]. Desai, P. and Jayakumar, N., An Extensible Framework Using Mobilityrpc For Possible Deployment Of Active Storage On Traditional Storage Architecture.
[2]. Desai, P.R. and Jayakumar, N.K., A Survey on Mobile Agents.
[3]. Jayakumar, N. and Kulkarni, A., 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), pp.2313-2318.
[4]. 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), pp.188-191.
[5]. Salunkhe, R., Kadam, A.D., Jayakumar, N. and Joshi, S., 2016, March. Luster a scalable architecture file system: A research implementation on active storage array framework with Luster file system. In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) (pp. 1073-1081). IEEE.
[6]. Jayakumar, N., Iyer, M.S., Joshi, S.D. and Patil, S.H., A Mathematical Model in Support of Efficient offloading for Active Storage Architectures.
[7]. Naveenkumar, J. and Joshi, S.D., 2015. Evaluation of Active Storage System Realized through MobilityRPC.
[8]. Jayakumar, N., Bhardwaj, T., Pant, K., Joshi, S.D. and Patil, S.H., A Holistic Approach for Performance Analysis of Embedded Storage Array.
[9]. Naveenkumar, J., Makwana, R., Joshi, S.D. and Thakore, D.M., 2015. Performance Impact Analysis of Application Implemented on Active Storage Framework. International Journal, 5(2).
[10]. Naveenkumar, J. and Joshi, S.D., 2015. Evaluation of Active Storage System Realized Through Hadoop.
[11]. Zaeimfar, S.N.J.F., 2014. Workload Characteristics Impacts on file System Benchmarking. Int. J. Adv, pp.39-44.
[12]. Jayakumar, D.T. and Naveenkumar, R., 2012. SDjoshi,“. International Journal of Advanced Research in Computer Science and Software Engineering,” Int. J, 2(9), pp.62-70.
[13]. Suryawanshi, A.U. and Naveenkumar, J., Privacy-Preserving For A Secure Data Storage On Cloud Using Public Auditing Technique.
[14]. Suryawanshi, A.U. and Kumar, J.N., A Multiuser Model of Privacy-Preserving Auditing for Storing Data Security in Cloud Computing.
[15]. Naveenkumar J and Raval K.S, Clouds Explained Using Use-Case Scenarios.
[16]. Sagar S Lad, S.D Joshi, N.J, Comparison study on Hadoop’s HDFS with Lustre File System. International Journal of Scientific Engineering and Applied Science, 1(8), pp.491–494, 2015.
[17]. Jayakumar, N., Bhardwaj, T., Pant, K., Joshi, S.D. and Patil, S.H., 2015. A Holistic Approach for Performance Analysis of Embedded Storage Array. Int. J. Sci. Technol. Eng, 1(12), pp.247-250.
[18]. Kumar, N., Kumar, J., Salunkhe, R.B. and Kadam, A.D., 2016, March. 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). ACM.

BIOGRAPHIES

Priyanka Ravindra Wadagave
Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, Pune.

 DOI 10.17148/IJARCCE.2019.81121