Improving I/O Performance using Cache as a Service on Cloud

Apurva A. Khandekar
Department of Computer Engineering and IT
College of Engineering, Pune (COEP)
Pune, India.
e-mail: khandekaraa13.comp@coep.ac.in

Prof. S. B. Mane
Department of Computer Engineering and IT
College Of Engineering, Pune (COEP)
Pune, India
e-mail: sunilbmane.comp@coep.ac.in

Abstract— Caching is gaining popularity in Cloud world. It is one of the key technologies which plays a major role in bridging the performance gap between memory hierarchies through spatial or temporal localities. In cloud systems, heavy I/O activities are associated with different applications. Due to heavy I/O activities, performance is degrading. If caching is implemented, these applications would be benefited the most. The use of a Cache as a Service (CaaS) model as a cost efficient cache solution to the disk I/O problem. We have built the remote-memory based cache that is pluggable and file system independent to support various configurations. The cloud Server process introduce, pricing model together with the elastic cache system. This will increase the disk I/O performance of the IaaS, and it will reduce the usage of the physical machines.

Keywords- *I/O activities, Cloud, Caching, Cache Replacement Algorithms, CaaS.*

*****

I. INTRODUCTION

A. Cloud Computing

Cloud refers to term Internet, and cloud computing comes from how Internet network look alike. Cloud computing can be described as a service provided by the network to deliver the computer resources. The Cloud computing is a pay and use service for network storage or computer resources which have been provided by another provider. In this era of high speed, Internet cloud computing is gaining more popularity. Cloud computing provides large storage space and provides various facilities to the users with minimal cost of infrastructure and development. Important features of cloud computing includes reducing the capital expenditure, improvement in the utilization of resources, scalability on demand, implementation is quick and easy and the most important is it can be accessed from anywhere. Main services provided by cloud computing are SaaS (Software as a service), PaaS (Platform as a service) and IaaS (Infrastructure as a service). These are considered as the backbone of the cloud.

B. Caching

Caching is considered to be the key technology which bridges the performance gap between memory hierarchies through spatial or temporal localities. Particularly, in disk storage system, its effect is considered to be prominent. Caching is used in storage systems to provide fast access to recently or frequently accessed data, with non-volatile devices used for data safety and long-term storage. Much research has focused on increasing the performance of caches as a means of improving system performance. In many storage system configurations, client and server caches form a two-layer (or more) hierarchy, introducing new challenges and opportunities over traditional single-level cache management. These include determining which level to cache data in and how to achieve exclusivity of data storage among the cache levels given the scant information available in all but the highest-level cache. Addressing these challenges can significantly improve overall system performance.

C. Cache Replacement

A cache replacement policy is used to decide which block is the best candidate for eviction when the cache is full. The hit rate is the fraction of page requests served from the cache, out of all requests issued by the application. Numerous studies have demonstrated the correlation between an increase in hit rate and application speedup. This correlation motivates the ongoing search for better replacement policies. The most commonly used online replacement policy is LRU. Pure LRU has no notion of frequency, which makes the cache susceptible to pollution from looping or sequential access patterns.

In the proposed system, Cache Effective Replacement Algorithm (CERA) is used for cache replacements. CERA is proposed and designed by Rassul AYANI et al.[4]In case of a cache miss, cost is incurred to fetch the cache from the server. The cost can be measured in two different ways: (i) the amount of traffic generated to retrieve the missing objects, and (ii) the download latency experienced by the end users to retrieve the missing objects. Cost can be defined as cost incurred due to the download latency and the traffic latency. In CERA, a benefit value (BV) is assigned to each of the object according to the priority. Object having the lowest benefit value is replaced when the cache is full. The BV consists of three parts: normalized cost, re-accessing probability and dynamic aging. CERA algorithm considers miss cost, object (or file) size, and access frequency.

CERA when compared with other algorithms (LRU, LFU) for different metrics like hit-ratio, byte- hit-ratio and cost reduction rate, it outperforms among all. In web traffic, it reduces the overall retrieval cost. CERAs performance depends on the workload characteristics. For larger cache sizes, it has the best byte hit ratio, but not for small caches.[12]

II. RELATED WORK

The following section describes various researches which have been done in past few years on cloud, caching, data availability, issues related to I/O performance of the systems etc. Specifically it focuses on I/O virtualization, caching

 ISSN: 2321-8169
3289 - 3292

mechanism and other alternatives. And how this research is distinguished from previous research work.

Saad Liaquat Kiani and et al [4] have researched about the context caches in the cloud. Context data is collected from different digital sources and services which are then made available via context brokers and servers to similar distributed context consuming applications. They investigated and proposed a mechanism in a cloud based context broker. This proposed mechanism gives context provisioning services provided by context providers and applications which are being consumed and the result give away different caching benefits under the context provisioning system.

Thepparit Banditwattanawong and et al [6] As cloud computing has become heavily networked its rapid growth accelerates public cloud data-out overspends and downstream network-bandwidth saturation. Replication of cloud data initiated by consumers for them is known as client side cloud caching. In this paper researchers have presented a mechanism called as Cloud cache eviction strategy.

Its simulations depicts that, for almost ten cases cloud outperformed three well known web cache eviction strategies across all standard performance metrics. Even in several cases minima of performance of cloud overcame other strategies. Another thing is Cloud can attain optimal hit and byte hit ratios without any loss one to other.

Marazakis et al. [8] researched on improving I/O performance of local disk. For improving the I/O performance in storage area network he tried utilizing Remote Direct Memory Access technology. Disk devices in the local block devices are extracted from the disk devices of disk devices. For write operation memory regions which are RDMA enabled are used in remote machines. This proposal is different from our research where local disk I/O performance is improved using RM as a cache instead of Remote disk.

Ousterhout et al. [9] has recently researched on a new approach to solve the I/O problem in cloud computing. His approach focuses on data processing, and he has proposed architecture, called RAMCloud. In distributed systems, DRAM is used to store the data in the RAMCloud. RAMCloud has showed extremely low latency along with the performance benefits. However, high (operational) cost and high energy usage are incurred by RAMCloud.

III. PROPOSED SYSTEM

In this section, we propose a system which mainly consists of following components: Elastic Cache System and Service Model. The Elastic Cache System uses Remote Memory which is exported from dedicated memory servers or from Solid State Drivers. This system is designed such that it is pluggable and file system independent. For caching it uses Cost Effective Replacement Algorithm (CERA). Service model deals with different cache alternatives and its performance characteristics. It effectively balances conflicting objectives between the user and the provider, i.e., performance vs. profit.

IV. SYSTEM ARCHITECTURE

In this section, we discuss the important components of the elastic cache. The elastic cache system is conceptually composed of two components: a VM and a cache server.

Remote memory is used as a disk cache which is demanded by virtual machine. A RM-based cache is built as a device driver which is then implemented. RM regions of RM cache device are viewed as byte addressable space. I/O requests at each block address are translated into an offset of each region, and read/write requests are translated into RDMA read/write operations.

Memory pool is offered by the memory servers as a cache pool, whenever a remote memory requests for VM for resource allocation this server provide memory from the cache pool. Some portion of the physical memory is exported to VMs and several chunks are formed. These chunks are then sent to all VMs, along with chunk lock and owner regions for exclusive access to the chunk. VM has to mark its ownership on assigned chunks whenever it want to use RM, then it can make use of the chunk as cache.

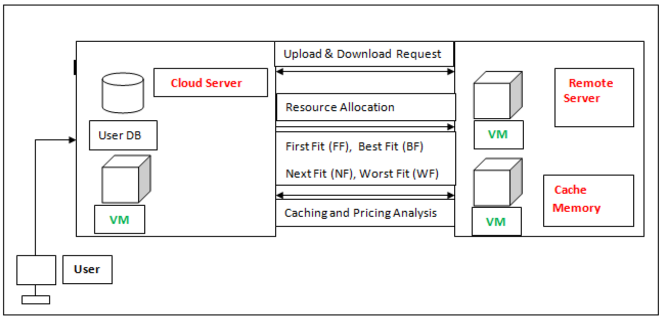


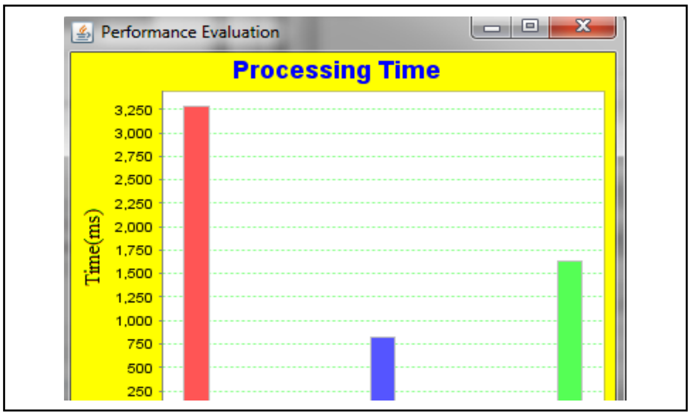
Figure. 1. Architecture of the Proposed System

Initially, user sends request to the cloud server. Cloud server connects with Remote Memory (RM) server and Virtual Machine (VM). As a disk cache, RM is required by VM. We build an RM-based cache as a block device and implement

a new block device driver (RM-Cache device) where RM regions are viewed as byte-addressable space. Now, cloud server sends the data to the idle server through resource allocation process for addressing the user request. At last, RM performance is measured.

V. EVALUATION

In this section, CaaS model is evaluated from the viewpoint of both providers and users. We measure the performance of the system in terms of transactions per minute, hit ratio etc. The actual system level modification is not possible. Hence we could not test our system on real cloud services but we build an RDMA enabled cloud infrastructure to evaluate our system.



ISSN: 2321-8169
3289 - 3292

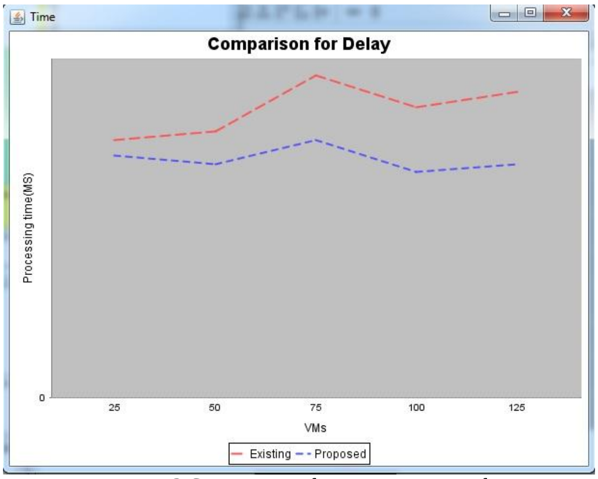


Figure 3 Comparison of Processing Time of Existing System and Proposed System

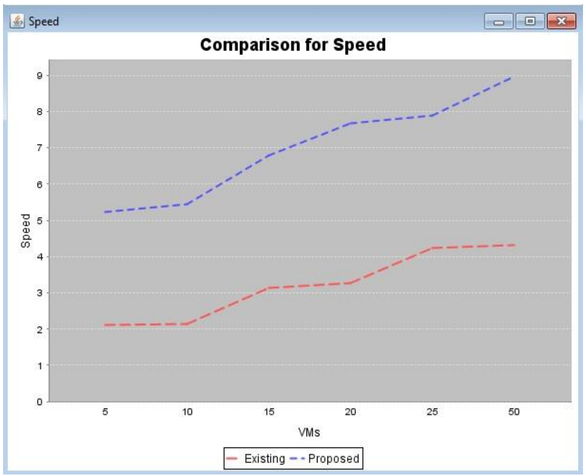


Figure 4 Comparison of Speed of Existing System and Proposed System

Figure 2 shows the graph of measured statistics of processing time when file is retrieved from main server, cache server and remote server. When a file is requested to a server, more processing time is consumed due to file retrieval from the main server. If the file is present in the remote server, it takes relatively less time than the file retrieved from cache that is already present in the main or cache serve. Figure 3 and figure 4 shows a graph of processing time and speed of existing systems which uses traditional cache replacement policies and the new proposed system which uses Cost Effective Replacement Algorithm (CERA) respectively. CERA takes less amount of processing time on VMs as compared to Existing Algorithms. CERA is faster than Existing algorithm.

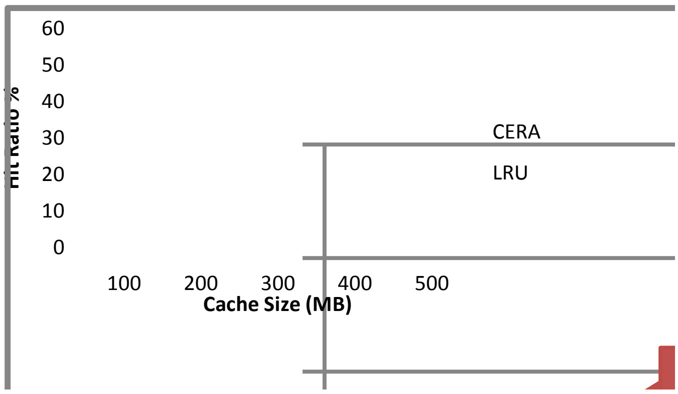


Figure 6. Comparison of Hit Ratio

Figure 6 shows the comparison of performance in terms of Hit Ratio. With cache size of 100MB and 200MB CERA gives 50% improvement when compared to LRU.

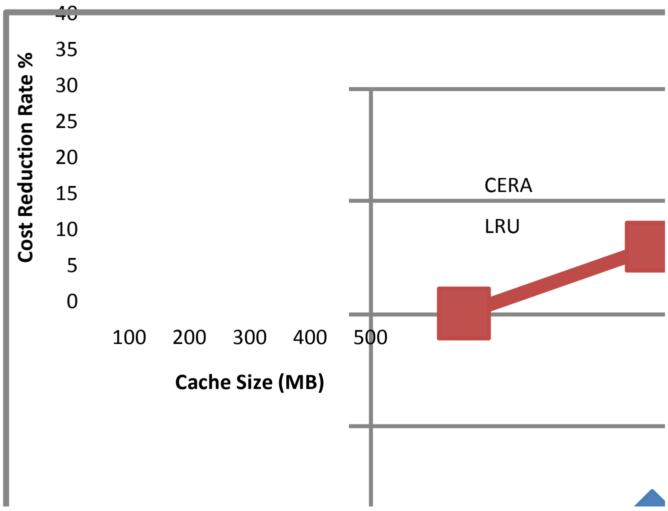


Figure 7. Comparison of Cost Reduction Ratio

Figure 7 shows the comparison of Performance in terms of cost reduction ratio(CRR) of LRU and CERA.

CERA outperforms LRU for all cache sizes. The CRR obtained by CERA is 140% better than LRU for 10MB cache. And for 500MB cache it shows 23% better than LRU. CERA is also better at reducing cost than LRU, LFU. More concretely, performance of CERA is 67% higher than LRU for a 50MB cache and 50% higher for 100 MB cache.

VI. CONCLUSION

Low disk I/O performance is one of the most significant problems. In this paper, we have presented a model which consists of the elastic cache system and a CaaS model. Elastic cache system uses a remote memory based cache which is pluggable and files system independent. CaaS model which presents a cost efficient solution for low disk I/O problem. For cache replacement we have used a Cache Effective Replacement Algorithm (CERA) which shows better performance as compared to other replacement algorithms like

LRU, LFU etc. We have a seen significant performance gain, reduced use of physical machines, increased throughput and hence increase in the profit of user and provider.

REFERENCES

[1] L. Cherkasova and R. Gardner, "Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor," Proc. Ann. Conf. USENIX Ann. Technical Conf. (ATC '05), 2005.

[2] A. Menon, J.R. Santos, Y. Turner, G.J. Janakiraman, and W. Zwaenepoel, "Diagnosing Performance Overheads in the Xen Virtual Machine Environment," Proc. First ACM/USENIX Int'l Conf. Virtual Execution Environments (VEE '05), 2005.

[3] R. Yanggratoke, F. Wuhib, and R. Stadler, "Gossip-based resource allocation for green computing in large clouds," International Conference on Network and Service Management in 2011.

[4] Saad Liaquat Kiani∗ , "Towards Context Caches in the Clouds," in Fourth IEEE International Conference on Utility and Cloud Computing, 2011.

[5] Salvatore J. Stolfo, "Fog Computing: Mitigating Insider Data Theft," IEEE Symposium on Security and Privacy Workshop in 2012.

[6] Uthayopas, Thepparit Banditwattanawong and Putchong, "Improving Cloud Scalability, Economy and Responsiveness with Client-Side Cloud Cache," in IEEE 2013.

[7] Ibikunle F., Kuyoro S. O., Awodele O. "Cloud Computing Security Issues and Challenges," International Journal of Computer Networks (IJCN), vol. 3, no. 5, 2011.

[8] M. Marazakis, K. Xinidis, V. Papaefstathiou, and A. Bilas, "Efficient Remote Block-Level I/O over an RDMA-Capable NIC," Proc. 20th Ann. Int'l Conf. Supercomputing (ICS '06), 2006.

[9] J. Ousterhout, P. Agrawal, D. Erickson, C. Kozyrakis, J. Leverich,D. Mazie`res, S. Mitra, A. Narayanan, G. Parulkar, M. Rosenblum, S.M. Rumble, E. Stratmann, and R. Stutsman, "The Case for RAMClouds: Scalable High-Performance Storage Entirely in DRAM," ACM SIGOPS Operating Systems Rev., vol. 43, pp. 92-105, Jan. 2010.

[10] K. Lim, J. Chang, T. Mudge, P. Ranganathan, S.K. Reinhardt, and T.F. Wenisch, "Disaggregated Memory for Expansion and Sharing in Blade Servers," Proc. 36th Ann. Int'l Symp. Computer Architecture (ISCA '09), 2009.

[11] Hyuck Han, Young Choon Lee, Woong Shin, Hyungsoo Jung, Heon Y. Yeom and Albert Y. Zomaya "Cashing in on the Cache in the Cloud," IEEE Transactions On Parallel And Distributed Systems, Vol. 23, 2012.

[12] Rassul AYANI, Yong Meng TEO and Peng CHEN, "Cost-based Proxy Caching", Proceedings of the International Symposium on Distributed Computing and Applications to Business, Engineering and Science, December 2002.