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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

An Efficient Secure Cache Management with Dynamic Pricing Scheme for Cloud Environment

S.Karpagavidhya[1], T.Kalai Selvi[2]

M.E (CSE) Second Year, Erode Sengunthar Engineering College, Erode [1]

Assistant Professor (SG), CSE Department, Erode Sengunthar Engineering College, Erode [2]

ABSTRACT: In this article, we propose the basic concepts about efficient secure cache management with dynamic pricing scheme for cloud environment and survey the list of existing cloud computing techniques. Pricing schemes are used in commercial clouds. Infrastructure as a Service (IaaS) provides resources based on demand model. Data centers are used to share storage spaces and data values. Caching technology improves the performance of the cloud. Cache as a service (CaaS) model is an additional service to IaaS. The proposed system is designed to provide data and services with cache support. Dynamic pricing scheme is used to estimate the price for cache. Data security is provided using Advanced Encryption algorithm. The data verification process is carried out with the secure hashing algorithm. Remote Direct Memory Access is used to access the data storages in the remote cloud server.

Keywords: Cloud Environment, Cache as a service model, Virtual Machine, Remote Memory, Pricing Scheme.

I. INTRODUCTION

Cloud Computing plays an important role in business organization in order to make the business in effective. One of the major motivations for the study of cloud computing is much more interesting than Philosophical problems. Most organization's large databases that contain a wealth of potentially accessible information. Access the information in large databases is very difficult. The Cloud computing, which was coined in late of 2007, currently emerges as a hot topic due to its abilities to offer flexible dynamic IT infrastructures, QoS guaranteed computing environments and configurable software services. Cloud computing is becoming one of the next IT industry buzz words: users move out their data and applications to the remote "Cloud" and then access them in a simple and pervasive way. Until 20 years ago when personal computers came to us, data and programs were mostly located in local resources. Nowadays the Cloud computing comes into fashion due to the need to build complex IT infrastructures. Users have to manage various software installations, configuration and updates. Computing resources and other hardware are prone to be outdated very soon. At the current stage, the Cloud computing is still evolving and there exists no widely accepted definition. Based on our experience, we propose an early definition of Cloud computing as follows:

A Cloud computing is a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing infrastructures on demand, which could be accessed in a simple and pervasive way.

A. Motivation for Cache Management

Over the past decades, caching has become the key technology in bridging the performance gap across memory hierarchies via temporal or spatial localities; in particular, the effect is prominent in disk storage systems. Currently, the effective use of cache for I/O-intensive applications in the cloud is limited for both architectural and practical reasons. Due to essentially the shared nature of some resources like disks, the virtualization overhead with these resources is not negligible and it further worsens the disk I/O performance. Thus, low disk I/O performance is one of the major challenges encountered by most infrastructure services as in Amazon's relational database service, which provisions virtual servers with database servers. At present, the performance issue of I/O intensive applications is mainly dealt with by using high performance (HP) servers with large amount of memory, leaving it as the user's responsibility.

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

B. Cloud Services

Conceptually, users acquire computing platforms or IT infrastructures from computing Clouds and then run their applications inside. Therefore, computing Clouds render users with services to access hardware, software and data resources. The fig.1. Shows the functionality of the cloud.

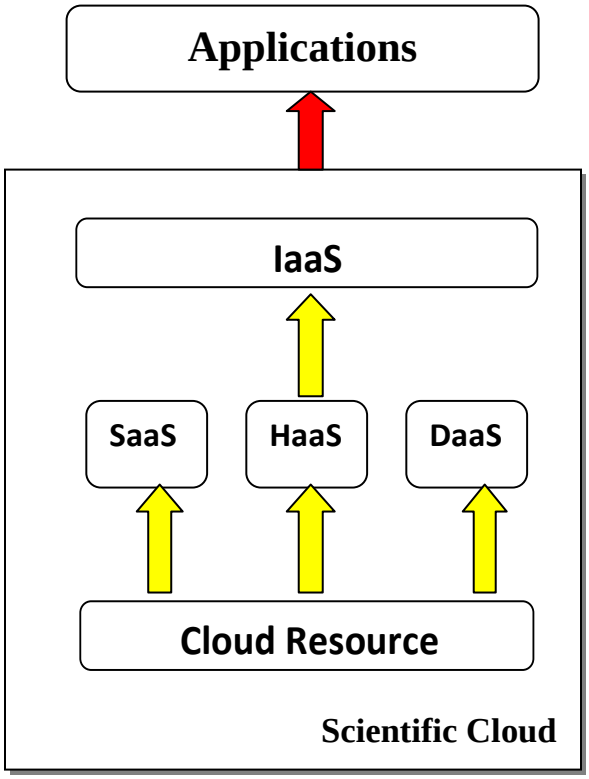

Fig.1 Cloud Functionalities

i. Hardware as a Service (HaaS)

Hardware as a Service was coined possibly in 2006. As the result of rapid advances in hardware virtualization, IT automation and usage metering & pricing, users could buy IT hardware, or even an entire data center, as a pay-as-you-go subscription service. The HaaS is flexible, scalable and manageable to meet your needs. Examples could be found at Amazon EC_2, IBM's Blue Cloud project, Nimbus, Eucalyptus and Anomalism.

ii. Software as a Service (SaaS)

Software or an application is hosted as a service and provided to customers across the Internet. This mode eliminates the need to install and run the application on the customer's local computers. SaaS therefore alleviates the customer's burden of software maintenance, and reduces the expense of software purchases by on-demand pricing. An early example of the SaaS is the Application Service Provider (ASP). The ASP approach provides subscriptions to software that is hosted or delivered over the Internet. Microsoft's "Software + Service" shows another example: a combination of local software and Internet services interacting with one another.

iii. Data as a Service (DaaS)

Data in various formats and from multiple sources could be accessed via services by users on the network. Users could, for example, manipulate the remote data just like operate on a local disk or access the data in a semantic way in the Internet. Amazon Simple Storage Service (S_3) provides a simple Web services interface that can be used to store and retrieve, declared by Amazon, any amount of data, at any time, from anywhere on the Web. The DaaS could also be found at some popular IT services, e.g., Google Docs and Adobe Buzzword.

C. Cloud Technologies

The Cloud computing distinguishes itself from other computing paradigms, like Grid computing, Global computing, Internet Computing in the following aspects:

i. User-centric interfaces

Cloud services should be accessed with simple and pervasive methods. In fact, the Cloud computing adopts the concept of Utility computing. In detail, the Cloud services enjoy the following features:

The Cloud interfaces do not force users to change their working habits and environments, e.g., programming language, compiler and operating system. This feature differs Cloud computing from Grid computing as Grid users have to learn new Grid commands & APIs to access Grid resources & services. The Cloud client software which is required to be

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

installed locally is lightweight. For example, the Nimbus Cloud kit client size is around 15MB.

ii. On-demand service provisioning

Computing Clouds provide resources and services for users on demand. Users can customize and personalize their computing environments later on, for example, software installation, network configuration, as users usually own administrative privileges.

iii. QoS guaranteed offer

The computing environments provided by computing Clouds can guarantee QoS for users, e.g., hardware performance like CPU speed, I/O bandwidth and memory size. The computing Cloud renders QoS in general by processing Service Level Agreement (SLA) with users – a negotiation on the levels of availability, serviceability, performance, operation, or other attributes of the service like billing and even penalties in the case of violation of the SLA.

iv. Autonomous System

The computing Cloud is an autonomous system and it is managed transparently to users. Hardware, software and data inside clouds can be automatically reconfigured, orchestrated and consolidated to present a single platform image, finally rendered to users.

v. Virtualization technology

Virtualization technologies partition hardware and thus provide flexible and scalable computing platforms. Virtual machine techniques, such as VMware and Xen, offer virtualized IT-infrastructures on demand. Virtual network advances, such as VPN, support users with a customized network environment to access Cloud resources.

II. CURRENT RESEARCH IN CACHE MANAGEMENT FOR CLOUD ENVIRONMENT

There have been a number of studies conducted to investigate the issue of I/O performance in virtualized systems. The focus of these investigations includes I/O virtualization, cache alternatives and caching mechanisms. In this section, we describe and discuss notable work related to our study. What primarily distinguishes ours from previous studies is the practicality with the virtualization support of remote memory access and the incorporation of service model; hence, cache as a service.

A. XHive: Efficient Cooperative Caching for Virtual Machines

XHive cooperative caching model is proposed on Xen VMM. Xen is an open-source Virtual Machine Monitor (VMM) environment. It enable multiple VMs to share storage for reducing disk usage and burdens of virtual disk .This model reduces disk I/O operations for shared working sets. [4] XHive provides block-level cooperative caching and VMM-level cache consistency on a read only or copy-on-write disk. Finally, XHive is well tailored to the driver VM model. This model has been prevalent due to safety and reuse of existing device drivers, but induces the performance degradation arising from scheduling overheads: domain switches and scheduling latency. To avoid such overheads, XHive serves a cached block in the VMM without driver VM intervention, while placing metadata in a driver VM.

We demonstrate that our scheme enables us to aggressively consolidate read-intensive shared workloads on a physical machine while improving read performance. In addition,, we show that the contribution to singlet caching with under provisioned VM memory is more efficient for consolidated VMs that have a shared working set. Finally, we compare our scheme to sharing-based memory over commitment techniques with various configurations.

B. Optimizing Xen VMM Based on Intel Virtualization Technology

Xen [2] is a very famous open source hypervisor, originally developed at the University of Cambridge, which targets to support 100 par virtualized guest OSs on one physical hardware machine, and brings forth outstanding performance through par virtualized approach. However, unlike full system virtualization, the par virtualized approach has its intrinsic shortcomings, because it has to modify the OS kernel to shut down the processor's virtualization holes. To implement full virtualization on x86 platform, Intel, as the leading processor manufacturer, enhances x86 architecture to support full virtualization, as described in the Intel Virtualization Technology Specification.

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

The Intel Open source Technology Center (OTC) extended the Xen project with this novel technology and finally achieved full hardware virtualization and successfully ran unmodified guest OSs with high efficiency. We named this project, Extending Xen with Intel® VT and also called full hardware virtualization with hardware-assisted virtualization. To fully use the new hardware virtualization feature, it is always critical to perform performance tuning work for a mature project. In the system demonstrated such performance tuning exercises, and how to improve system performance with it in the Xen/VT project.

C. Efficient Remote Block-Level IO over an RDMA-Capable NIC

The performance of the I/O path in commodity initiator and target architectures, and show limitations in achieving throughput similar to directly attached storage. We use a custom-build system area network that allows us to tune the support in the network interface. Our network is capable of about 500 MB throughputs, using support for RDMA [3] operations but no direct, user-level access. We believe that the features used in our network interface can be provided in most network interfaces at minimal cost.

We examine the effectiveness of three techniques in alleviating these bottlenecks and improving system throughput: interrupt silencing, cooperative batching of requests, and elimination of small messages. We find that each technique is able to improve throughput by up to 50% compared to the base system. When combined, the throughput is improved by up to 100% over a simpler configuration. However, we observe high CPU utilization levels, especially at the I/O target node. Moreover, we identify a further limiting factor, due to the high aggregate interrupt count.

D. Using Transparent Compression to Improve SSD-Based I.O Caches

Performance of storage I/O is an important problem in modern systems. The emergence of flash-based solid state drives (SSDs) has the potential to mitigate I/O penalties: [6] SSDs have low read response times and are not affected by seeks. Additionally, recent SSDs provide peak throughput that is significantly superior to magnetic hard disk drives (HDD).

In this work we design FlaZ, a system that uses SSDs as compressed caches in the I/O path. FlaZ internally consists of two layers, one that achieves transparent compression and one that uses SSDs as an I/O cache. Although these layers are to a large extent independent, in our work we tune their parameters in a combined manner.

The caching layer of FlaZ is a direct-mapped, write through cache with one block per cache line, while the compression layer of FlaZ is more complex and requires addressing. Although our goal in this work is not to examine alternative compression algorithms and possible optimizations, it is important to quantify their performance impact on the I/O path. Finally, compressed caching has a negative impact on response-time bound workloads that use only small I/Os as well as on workloads that fit in the uncompressed cache almost entirely.

E. Profiling Applications for Virtual Machine Placement in Clouds

The system investigates the relationship between application performance characteristics and resource utilization, identify the correlation between them, and present a performance prediction model. To address this generalization issue, we develop an application profiling technique using Canonical Correlation Analysis (CCA) method. For a given application, we vary its workload and run it together with different background loads.

The resulting performance metrics enable us to establish the links between the application's performance and the utilization of underlying system resources through the correlation analysis. The output of the correlation analysis is then used as the profile of the application. We further devise a performance prediction model based on application profiles [8] generated using CCA.

Clearly, the effective profiling of applications is of great practical importance in various respects: (1) the quality of consolidation decisions can be significantly improved in terms of resource utilization and profit, (2) application performance can be guaranteed and (3) the entire system and its power budget are predictable. Our experimental results demonstrate the capability of our profiling technique and the accuracy of our prediction model.

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

III. CACHE AS A SERVICE – A CONCEPT

The CaaS model consists of two main components: an elastic cache system as the architectural foundation and a service model with a pricing scheme as the economic foundation. The basic system architecture for the elastic cache aims to use RM, which is exported from dedicated memory servers. It is not a new caching algorithm. The elastic cache system can use any of the existing cache replacement algorithms. Near uniform access time to RM-based cache is guaranteed by a modern high speed network interface that supports RDMA as primitive operations. Each VM in the cloud accesses the RM servers via the access interface that is implemented and recognized as a normal block device driver. Based on this access layer, VMs utilize RM to provision a necessary amount of cache memory on demand.

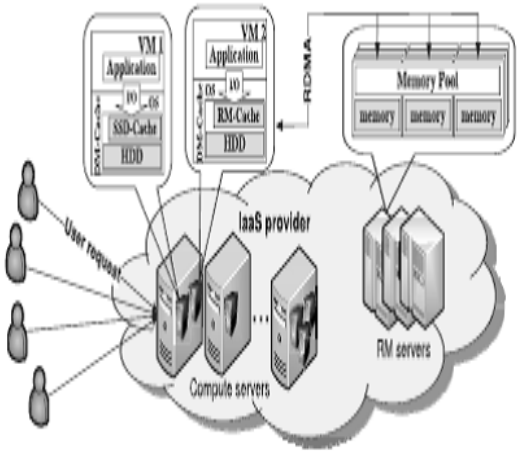

Fig. 2. Overview of CaaS

As shown in Fig. 2, a group of dedicated memory servers exports their local memory to VMs, and exported memory space can be viewed as an available memory pool. This memory pool is used as an elastic cache for VMs in the cloud. For billing purposes, cloud service providers could employ a lease mechanism to manage the RM pool.

To employ the elastic cache system for the cloud, service components are essential. The CaaS model consists of two cache service types based on whether LM or RM is allocated with. Since these types are different in their performance and costs a pricing scheme that incorporates these characteristics is devised as part of CaaS.

Together, we consider the following scenario. The service provider sets up a dedicated cache system with a large pool of memory and provides cache services as an additional service to IaaS. Now, users have an option to choose a cache service specifying their cache requirement and that cache service is charged per unit cache size per time. Specifically, the user first selects an IaaS type as a base service. The user then estimates the performance benefit of additional cache to her application taking into account the extra cost, and determines an appropriate cache size based on that estimation. We assume that the user is at least aware whether her application is I/O intensive and aware roughly how much data it deals with. The additional cache in our study can be provided either from the local memory of the physical machine on which the base service resides or from the remote memory of dedicated cache servers. The former LM case can be handled simply by configuring the memory of the base service to be the default memory size plus the additional cache size.

The cost benefit of our CaaS model is twofold: profit maximization and performance improvement. Clearly, the former is the main objective of service provider. The latter also contributes to achieving such an objective by reducing the number of active physical machines. From the user's perspective, the performance improvement of application can be obtained with CaaS in a much more cost efficient manner since caching capacity is more important than processing power for those applications.

IV. ELASTIC CACHE SYSTEMS

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. A VM demands RM for use as a disk cache. We build an RM-based cache as a block device and implement a new block device driver. In the RM-Cache device, RM regions are viewed as byte-addressable space. The block address of each block I/O request is translated into an offset of each region, and all read/write requests are also transformed into RDMA read/write operations. We use the device-mapper module of the Linux operating system to integrate both the RM-Cache device and a general block device (HDD) into a single block

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

device. This forms a new virtual block device, which makes our cache pluggable and file-system independent.

In order to deal with resource allocation for remote memory requested from each VM, a memory server offers a memory pool as a cache pool. When a VM needs cache from the memory pool, the memory pool provides available memory. To this end, a memory server in the pool exports a portion of its physical memory4 to VMs, and a server can have several chunks. A normal server process creates 512 MB memory space via the malloc function, and it exports a newly created chunk to all VMs, along with Chunk Lock and Owner regions to guarantee exclusive access to the chunk. After a memory server process exchanges RDMA specific information with a VM that demands RM, the exported memory of each machine in the pool can be viewed as actual cache. When a VM wants to use RM, a VM should first mark its ownership on assigned chunks, and then it can make use of the chunk as cache. An example of layered architecture of a VM and a memory pool, both of which are connected via the RDMA interface, is concretely described in Fig.3.

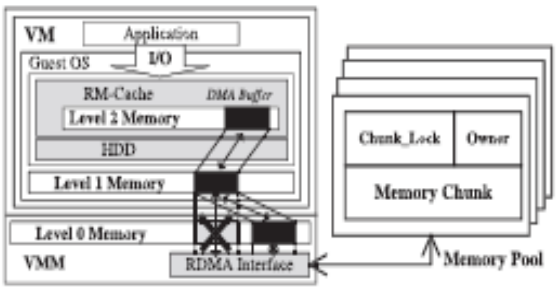

Fig.3. Elastic cache structure and double paging problem

When multiple VMs try to mark their ownership on the same chunk simultaneously, the access conflict can be resolved by a safe and atomic chunk allocation method, which is based on the Compare and Swap operation supported by Infiniband. It atomically compares the 64-bit value stored at the remote memory to a given value and replaces the value at the remote memory to a new value only if they are the same. By the Compare and Swap operation, only one node can acquire the Chunk Lock and it can safely mark its ownership to the chunk by setting the Owner variable to consumer's id.

A. Double paging in RDMA

The double paging problem was first addressed and techniques such as ballooning are proposed to avoid the problem. Since the problem is a bit technical but very critical in realizing CaaS in the clouds platform, we describe what implementation difficulty it causes and how we overcome the obstacle. Goldberg and Hassinger define levels of memory as follows:

- Level 0 memory: memory of real machine
- Level 1 memory: memory of VM
- Level 2 memory: virtual memory of VM.

In VM environments, the level 2 memory is mapped into the level 1 memory, and this is called double paging. For RDMA communication, a memory region should be registered to the RDMA device. Generally, kernel-level functions mapping virtual to physical addresses are used for memory registration to the RDMA device. Since the RDMA device cannot understand the context of level 1 memory addresses, direct registration of level 1 memory space to RDMA leads to malfunction of RDMA communication.

To avoid this type of double paging anomaly in RDMA communication, we exploit hardware IOMMUs to get DMA-able memory. Thus, we use kernel functions related with IOMMUs to get level 0 memory addresses. The RM-Cache device allocates level 2 memory space through kernel level memory allocation functions in the VM. Then, it remaps the allocated memory to DMA-able memory space through IOMMU. The mapped address of the DMA-able memory becomes level 0 memory that can now be registered correctly by RDMA devices.

V. SERVICE MODEL

In this section, we first describe performance characteristics of different cache alternatives and design. Then, we present a pricing model that effectively captures the trade off between performance and cost.

A. Modeling Cache Services

The use of LM as cache delivers incomparably better I/O performance than other cache alternatives; such a use is limited by several issues including capacity and the utilization of host machines. With the consideration of these facts, we have designed two CaaS types as the following:

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

- High performance—makes use of LM as cache, and thus, its service capacity is bounded by the maximum amount of LM.
- Best value (BV)—exploits RM as cache practically without a limit.

In our CaaS model, it is assumed that a user, who sends a request with a CaaS option, also accompanies an application profile including data volume, data access pattern, and data access type. In this paper, we primarily target the scenario in which users repeatedly and/or regularly run their applications in clouds, and they are aware of their application characteristics either by analysing business logic of their applications or by obtaining such information using system tools and/or application profiling. When a user is unable to identify/determine he/she simply rents default IaaS instances without any cache service option since CaaS is an optional service to IaaS. The service granularity in our CaaS model is set to a certain size. In this study, we adopt three default IaaS types: small, medium, and large with flat rates of fs, fm, and fl, respectively.

B. Dynamic Pricing Scheme

Cloud resources are provided on the basis of reservation and on-demand factors. Resource utilization cost for reservation plan is cheaper than on-demand plan. Cloud consumer can successfully minimize total cost of resource provisioning in cloud environments. The Optimal Cloud Resource Provisioning (OCRP) algorithm is used to manage resource allocation and pricing process. OCRP algorithm can effectively save the total cost. Efficient term assignment mechanism is proposed in the system. Resource usage distribution is analysed for term assignment process. Market based price assignment model is applied for price assignment under resource provider. The system minimizes the resource cost for users.

VI. SECURED CACHE MANAGEMENT WITH VIRTUALIZATION

Virtualization leads to Cloud Computing, but cloud computing is more than just virtualization Cloud computing (Public or Private) produce virtual servers or applications, virtualization does not create clouds. Virtualization is a mechanism used in a data center to consolidate and fully use physical devices. For example, a single server for each processing application performed then would be make the proposition very expensive. The CPU of each machine would usually not be fully utilized; resulting in wasted resources. In addition the size of data centers and the growth would be difficult, at best to manage. Virtualization can place several virtual servers on a single physical device, minimizing the need for physical devices; thus lowering cost in processing resources, space in the data center, environmental conditioners and overall power usage. Virtualization allows for consolidation and cost savings. The fig.4. shows this concept.

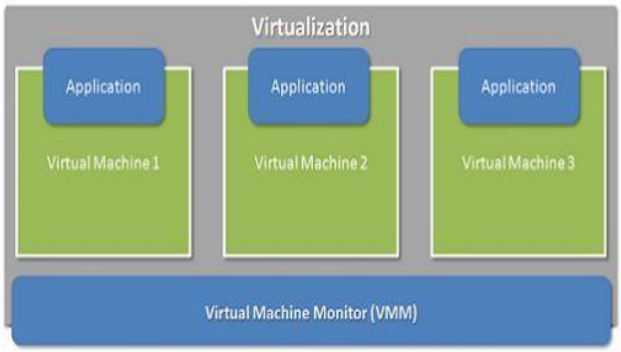

Fig.4. Virtualization mechanism

The Cache management methods are used to maintain the shared data in transmission process. Data values are maintained under RAM and disk cache environments. Dynamic pricing scheme is used for cache storages. Cache assignment is performed with location details. Data distribution is performed with redundant data verification methods. Security is provided with Advanced Encryption Standard algorithm. Data integrity is verified with Secure Hashing Algorithm.

VII. CONCLUSION

Caching technology improves the performance of the cloud. Here, caching plays a crucial role in improving their performance. Our CaaS model and its components are thoroughly validated and evaluated through extensive experiments in both a real system and a simulated

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 1, Issue 10, December 2012

environment. Our RMbased elastic cache system is tested in terms of its performance and reliability to verify its technical feasibility and practicality. Remote memory is used with Remote Direct memory Access. The cache system is improved with privacy and security features. Cache loads are distributed to the nodes. Disk based information systems with a memory-based cache produces better performance in task execution. Cost efficiency and elasticity of the cloud is very useful for both users and providers. Capital and operating cost are reduced by the system.

ACKNOWLEDGMENT

I am grateful to my college which gave me opportunity and lots of resources to accomplish my project in successful way. Without the guidance of my faculties I would not have reached this far, I thank them all for their timely suggestions and motivation. Finally, yet importantly, I express my heartfelt thanks to my beloved parents for their blessings, my friends for their help and wishes for the successful completion of this work.

REFERENCES

[1.] 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.] X. Zhang and Y. Dong, "*Optimizing Xen VMM Based on Intel Virtualization Technology*," Proc. IEEE Int'l Conf. Internet Computing in Science and Eng. (ICICSE '08), 2008

[3.] 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

[4.] H. Kim, H. Jo, and J. Lee, "*XHive: Efficient Cooperative Caching for Virtual Machines,"* IEEE Tran s. Computers, vol. 60, no. 1, Jan. 2011.

[5.] S.-W. Lee and B. Moon, "*Design of Flash-Based DBMS: An In-Page Logging Approach*," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '07), 2007

[6.] T. Makatos, Y. Klonatos, M. Marazakis, M.D. Flouris, and A. Bilas, "*Using Transparent Compression to Improve SSD-Based I/O Caches,*" Proc. Fifth European Conf. Computer Systems (EuroSys '10), 2010

[7.] 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

[8.] A.V. Do, J. Chen, C. Wang, Y.C. Lee, A.Y. Zomaya, and B.B. Zhou, "*Profiling Applications for Virtual Machine Placement in Clouds,*" Proc. IEEE Int'l Conf. Cloud Computing, 2011.

[9.] 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, no. 8, August 2012.

[10.] G. Jung, M. A. Hiltunen, K. R. Joshi, "*Amazon Elastic Compute Cloud"*, Amazon Web Services, 2010.

BIOGRAPHY

Miss. S. Karpagavidhya has completed Bachelor's degree in Computer Applications from Maharaja College for Women, Erode in the year 2008, affiliated to Bharathiar University and Master's Degree in Computer Applications from Sri Vasavi College, Erode in the year 2011, affiliated to Bharathiar University. Presently she is pursuing her M.E (Computer Science and Engineering) from Erode Sengunthar Engineering College, Thudupathi, Erode. Her research interests include computer networks and cloud computing.

Mrs.T.Kalai Selvi obtained her B.E (Computer Science and Engineering) degree and M.E (Computer Science and Engineering) degree from Mahendra Engineering College, Mahendrapuri in the year 2004 and 2009 respectively. She is presently working as Assistant Professor in the Department of Computer Science and Engineering at Erode Sengunthar Engineering College, Thudupathi, Erode. She has presented eight papers in various national and international conferences so far her research areas include Cloud Computing and Mobile Computing.