ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 442 – 446 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

Concept of Cashing in on the Cache in the Cloud

1 Mr. P. Vijender , **2** Mr. A. Vijendera goud 3.Munawerliet **4** Mr. Mazeed
1 M Tech Student, LIET, India, vijenderpuppala@gmail.com
2 M Tech Student, LIET, India, vijay.30689@yahoo.com
3. Munawerliet@gmail.com
4. Asst.Professor, LIET, India

Abstract

Over the past decades, caching has become the key technology used for bridging the performance gap across memory hierarchies via temporal or spatial localities; in particular, the effect is prominent in disk storage systems. Applications that involve heavy I/O activities, which are common in the cloud, probably benefit the most from caching. The use of local volatile memory as cache might be a natural alternative, but many well known restrictions, such as capacity and the utilization of host machines, hinder its effective use.

In addition to technical challenges, providing cache services in clouds encounters a major practical issue (quality of service or service level agreement issue) of pricing. Currently, (public) cloud users are limited to a small set of uniform and coarse-grained service offerings, such as High-Memory and High-CPU in Amazon EC2. In this paper, we present the cache as a service (CaaS) model as an optional service to typical infrastructure service offerings. Specifically, the cloud provider sets aside a large pool of memory that can be dynamically partitioned and allocated to standard infrastructure services as disk cache. We first investigate the feasibility of providing CaaS with the proof of concept elastic cache system (using dedicated remote memory servers) built and validated on the actual system; and practical benefits of CaaS for both users and providers (i.e., performance and profit, respectively) are thoroughly studied with a novel pricing scheme.

Our CaaS model helps to leverage the cloud economy greatly in that (1) the extra user cost for I/O performance gain is minimal if ever exists, and (2) the provider's profit increases due to improvements in server consolidation resulting from that performance gain. Through extensive experiments with eight resource allocation strategies we demonstrate that our CaaS model can be a promising cost-efficient solution for both users and providers.

Key words: Cloud Computing, Cache as a Service, Remote Memory, Cost Efficiency

Introduction

The resource abundance (redundancy) in many large data centers is increasingly engineered to offer the spare capacity as a service like electricity, water, and gas. For example, public cloud service providers like Amazon Web Services virtualize resources, such as processors, storage, and network devices, and offer them as services on demand, i.e., infrastructure as a service (IaaS) which is the main focus of this paper. A virtual machine (VM) is a typical instance of IaaS. Although a VM acts as an isolated computing platform which is capable of running multiple applications, it is assumed in this study to be solely dedicated to a single application, and thus, we use the expressions VM and application interchangeably hereafter. Cloud services as virtualized entities are essentially elastic making an illusion of "unlimited" resource capacity. This elasticity with utility computing (i.e., pay-as-you-go pricing) inherently brings cost effectiveness that is the primary driving force behind the cloud.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 442 – 446 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

However, putting a higher priority on cost efficiency than cost effectiveness might be more beneficial to both the user and the provider. Cost efficiency can be characterized by having the temporal aspect as priority, which can translate to the cost to performance ratio from the user's perspective and improvement in resource utilization from the provider's perspective. This characteristic is reflected in the present economics of the cloud to a certain degree . However, the conflicting nature of these perspectives (or objectives) and their resolution remain an open issue for the cloud.

In this paper, we investigate how cost efficiency in the cloud can be further improved, particularly with applications that involve heavy I/O activities; hence, I/O-intensive applications. They account for the majority of applications deployed on today's cloud platforms. Clearly, their performance is significantly impacted on by how fast their I/O activities are processed. Here, caching plays a crucial role in improving their performance.

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 (not performance isolatable), 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 amounts of memory, leaving it as the user's responsibility.

To overcome low disk I/O performance, there have been extensive studies on memory-based cache systems . The main advantage of memory is that its access time is several orders of magnitude faster than that of disk storage. Clearly, disk-based information systems with a memory-based cache can greatly outperform those without cache. A natural design choice in building a disk-based information system with ample cache capacity is to exploit a single, expensive, large memory computer system. This simple design—using local volatile memory as cache (LM cache)—costs a great deal, and may not be practically feasible in the existing cloud services due to various factors including capacity and the utilization of host machines.

In this paper, we address the issue of disk I/O performance in the context of caching in the cloud and present a cache as a service (CaaS) model as an additional service to IaaS. For example, a user is able to simply specify more cache memory as an additional requirement to an IaaS instance with the minimum computational capacity (e.g., micro/small instance in Amazon EC2) instead of an instance with large amount of memory (high-memory instance in Amazon EC2). The key contribution in this work is that our cache service model much augments cost efficiency and elasticity of the cloud from the perspective of both users and providers. CaaS as an additional service (provided mostly in separate cache servers) gives the provider an opportunity to reduce both capital and operating costs using a fewer number of active

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 442 – 446 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

physical machines for IaaS; and this can justify the cost of cache servers in our model. The user also benefits from CaaS in terms of application performance with minimal extra cost; besides, caching is enabled in a user transparent manner and cache capacity is not limited to local memory. The specific contributions of this paper are listed as follows: first, we design and implement an elastic cache system, as the architectural foundation of CaaS, with remote memory (RM) servers or solid state drives (SSDs); this system is designed to be pluggable and file system independent. By incorporating our software component in existing operating systems, we can configure various settings of storage hierarchies without any modification of operating systems and user applications. Currently, many users exploit memory of distributed machines (e.g., memcached) by integration of cache system and users' applications in an application level or a file-system level. In such cases, users or administrators should prepare cache-enabled versions for users' application or file system to deliver a cache benefit. Hence, file system transparency and application transparency are some of the key issues since there is a great diversity of applications or file systems in the cloud computing era.

Second, we devise a service model with a pricing scheme, as the economic foundation of CaaS, which effectively balances conflicting objectives between the user and the provider, i.e., performance versus profit. The rationale behind our pricing scheme in CaaS is that the scheme ensures that the user gains I/O performance improvement with little or no extra cost and at the same time it enables the provider to get profit increases by improving resource utilization, i.e., better service (VM) consolidation. Specifically, the user cost for a particular application increases proportionally to the performance gain and thus, the user's cost eventually remains similar to that without CaaS. Besides, performance gains that the user get with CaaS has further cost efficiency implications if the user is a business service provider who rents IaaS instances and offers value-added services to other users (end users).

Finally, we apply four well-known resource allocation algorithms (first-fit (FF), next-fit (NF), best-fit (BF), and worst-fit (WF)) and develop their variants with live VM migration to demonstrate the efficacy of CaaS.

Our CaaS model and its components are thoroughly validated and evaluated through extensive experiments in both a real system and a simulated environment. Our RM-based elastic cache system is tested in terms of its performance and reliability to verify its technical feasibility and practicality. The complete CaaS model is evaluated through extensive simulations; and their parameters are modeled based on preliminary experimental results obtained using the actual system.

Existing System:

Public cloud service providers like Amazon Web Services virtualize resources, such as processors, storage and network devices, and offer them as services on demand, i.e., infrastructure as a service (IaaS).

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 442 – 446 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

Disadvantages:

1. A virtual machine (VM) is a typical instance of IaaS. Although a VM as an isolated computing platform which is capable of running multiple applications, but in this application it is dedicated to a single application.
2. Over the past decades, caching has become the key technology in bridging the performance gap across memory hierarchies via temporal or spatial localities, but it is not implemented in the existing system.

Proposed System:

We address the issue of disk I/O performance in the context of caching in the cloud and present a cache as a service (CaaS) model as an additional service to IaaS. For example, a user is able to simply specify more cache memory as an additional requirement to an IaaS instance with the minimum computational capacity (e.g., micro/small instance in Amazon EC2) instead of an instance with large amount of memory (high-memory instance in Amazon EC2). The key contribution in this work is that our cache service model much augments cost efficiency and elasticity of the cloud from the perspective of both users and providers.

Advantages:

1. CaaS as an additional service (provided mostly in separate cache servers) gives the provider an opportunity to reduce both capital and operating costs using a fewer number of active physical machines for IaaS .

2. Caas can justify the cost of cache servers in our model.

3. The user also benefits from CaaS in terms of application performance with minimal extra cost.

4. Besides, caching is enabled in a user transparent manner and cache capacity is not limited to local memory.

Software Requirements Specification:

Software Requirements:

Language	:	JDK (1.7.0)
Frontend	:	JSP, Servlets
Backend	:	Oracle10g
IDE	:	my eclipse 8.6
Operating System	:	windows XP

Hardware Requirements

Processor	:	Pentium IV
Hard Disk	:	80GB
RAM	:	2GB

Modules Description:

1. Design Rationale:

Among many important factors in designing an elastic cache system, we particularly focus on the type of cache medium, the implementation level of our cache system, the communication Medium between a cache server and a VM, and reliability.

2. Modeling Cache Services

I/O-intensive applications can be characterized primarily by data volume, access pattern and access type; i.e., file size, random/sequential and read/write, respectively. The identification

of these characteristics is critical in choosing the most appropriate cache medium and proper size since the performance of different storage media (e.g., DRAMs, SSDs and HDDs) varies depending on one or more of those characteristics.

3. Pricing:

A pricing model that explicitly takes into account various elastic cache options is essential for effectively capturing the trade-off between (I/O) performance and (operational) cost.

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.1, Pages : 442 – 446 (2013)
Special Issue of ICACSE 2013 - Held on 7-8 January, 2013 in Lords Institute of Engineering and Technology, Hyderabad

Conclusion:

With the increasing popularity of infrastructure services such as Amazon EC2 and Amazon RDS, low disk I/O performance is one of the most significant problems. In this paper, we have presented a CaaS model as a cost efficient cache solution to mitigate the disk I/O problem in IaaS. To this end, we have built a prototype elastic cache system using a remote-memory-based cache, which is pluggable and file-system independent to support various configurations. This elastic cache system together with the pricing model devised in this study has validated the feasibility and practicality of our CaaS model. Through extensive experiments, we have confirmed that CaaS helps IaaS improve disk I/O performance greatly. The performance improvement gained using cache services clearly leads to reducing the number of (active) physical machines the provider uses, increases throughput, and in turn results in profit increase. This profitability improvement enables the provider to adjust its pricing to attract more users.

References

[1] L. Wang, J. Zhan, and W. Shi, "In Cloud, Can Scientific Communities Benefit from the Economies of Scale?," *IEEE Trans. Parallel and Distributed Systems,* vol. 23, no. 2, pp. 296-303, Feb. 2012.

[2] S. Jiang, K. Davis, and X. Zhang, "Coordinated Multilevel Buffer Cache Management with Consistent Access Locality Quantification," *IEEE Trans. Computers,* vol. 56, no. 1, pp. 95-108, Jan. 2007.

[5] H. Kim, H. Jo, and J. Lee, "XHive: Efficient Cooperative Caching for Virtual Machines," *IEEE Trans. Computers,* vol. 60, no. 1, pp. 106-119, Jan. 2011.

[3] Y. Dong, J. Dai, Z. Huang, H. Guan, K. Tian, and Y. Jiang, "Towards High-Quality I/O Virtualization," *SYSTOR '09: Proc. Israeli Experimental Systems Conf.,* 2009.

[4] 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.

[5] E.R. Reid, "Drupal Performance Improvement via SSD Technology," technical report, Sun Microsystems, Inc., 2009.