ScienceDirect

Future Generation Computer Systems

Volume 55, February 2016, Pages 116-128

Multi-provider cloud computing network infrastructure optimization

Thepparit Banditwattanawong [a 1], Masawee Masdisornchote [a 1], Putchong Uthayopas [b]

Show more

Share Cite

https://doi.org/10.1016/j.future.2015.09.002 Get rights and content View accessibility information

Highlights

- Byte-hit rate and hit rate could be optimal simultaneously in a nonuniform cost model.
- i-Cloud outperformed popular LRU, GDSF and LFUDA schemes in a nonuniform cost environment.
- i-Cloud's performances were stable and close to those of infinite cache size.
- Window size had small performance effect when relative cache sizes were big.
- Accounting data-out charge rates improved all performance aspects at small cache sizes.

Abstract

Cloud-adopting enterprises have been increasingly employing multiple cloud providers concurrently, for example, to consume unique services and to mitigate data lock-in risk. As a consequence, the enterprises must be able to address contrasting quality-of-service degrees offered by the different providers. This paper presents an intelligent cloud cache eviction approach, namely i-Cloud, as the core component of a client-side cloud cache. i-Cloud is capable of reducing public cloud data-out expenses, improving cloud network scalability and lowering cloud service access latencies specifically in multi-provider cloud environments. Trace-driven simulations have shown that i-Cloud outperformed well-known approaches in all performance metrics. In addition, i-Cloud is not only able to achieve optimal performances in all metrics simultaneously but also delivered relatively stable performances across all performance metrics. The results have also indicated that taking the nonuniformity of data-out charge rates into cache eviction decisions improved caching performances in all metrics.

Introduction

Big data has been increasingly hosted in clouds as in, for example, Facebook, Youtube, Street View Google Maps, iTune Store. This is because cloud enables not only fast, economical and greener big data analytics [1], [2], [3] but also highly-scalable and distributed sharing of the data. This is the evolution of data volume, data placement and data consumption behavior that altogether has led both practitioners and researchers to address several new cloud-computing problems including the downstream bandwidth saturation of network connections between external cloud and consumer premises, increase in external private-cloud data-out charge imposed by public cloud providers (e.g., [4], [5], [6]) and long-delayed cloud service responsiveness. These difficulties have been recognized in [7], [8], [9], [10] as the forms of data transfer bottlenecks, data transfer costs or cloud computing economics and SaaS SLA responsiveness. A straightforward way to handle these problems is network bandwidth upgrade, which unfortunately impedes cloud economy. A wiser means is disk shipping by overnight delivery services [7], [8] that is only applicable for delay-tolerant cloud services. Another solution to meet these scalability, economy and responsiveness requirements of cloud computing services at the same time is the consumer-initiated partial replication of cloud-hosted data to consumer (nearby) locus as in Amazon CloudFront [11], which offers a caching service through content delivery network. We refer to this solution as client-side cloud caching.

Client-side cloud caches are located in or nearby consumer premises where HTTP requests to external private clouds are proxied by the cloud caches, which in turn reply with the valid copies of the requested data objects either from their local storages (i.e.,cache hits) or by downloading updated copies from the clouds (i.e.,cache misses). The cloud caches are basically evolved from traditional forward web caching proxies since cloud data is also delivered by using the same set of HTTP/TCP/IP protocol stacks as in WWW. Unavoidably, the same limitation as in web caching is

also shared in the cloud caches that is caching entire remote data in local cache storage is not economically sensible, thus a cache replacement policy is also mandatory for the cloud caches. In fact, client-side cloud caches can also be deployed in a hierarchical structure as portrayed in Fig. 1: the web-browser level cloud caches are nonshared by individuals whereas the enterprise- and ISP-level cloud caches are shared. This paper focuses on a client-side shared cloud cache at the enterprise level. Another main characteristic of the client-side shared cloud cache is multi-tenancy support whereby workloads initiated by different enterprises are shared for hardware but isolated for user data. The other properties of the client-side shared cloud cache lie in its various Cache-as-a-Service (CaaS) models [12] (e.g., RAM Multitenancy Isolated database, and SSD Multitenancy Isolated database), which are not offered by the traditional web caching proxies.

As illustrated in Fig. 1, an enterprise-level cloud cache can be deployed in a multi-provider cloud environment where a consumer enterprise employs multiple concurrent cloud providers to host its external private clouds that are either independent from each other [13], [14] or interconnected as a hybrid cloud [15]; both are the forms of federated cloud [16], [17]. There are several benefits of multi-provider cloud deployment such as load balancing [18], [19], enabling of planned downtime for system maintenance [17], confidentiality protection [20], risk mitigation (mandated by hospitals, stock markets, air transportation controls, etc.) [20], [21] and the utilization of unique capabilities offered by different cloud providers [13]. Nevertheless, the deployment of multiple cloud providers usually imposes different levels of QoS perceived by consumers in the forms of unbalancing downstream network throughputs and nonuniform cloud data-out charge rates. This complicates cloud caching optimization in that cloud cache replacement must be aware of QoS heterogeneity to achieve both efficiency and economy. To accomplish such optimization, our extensive investigation of related work in Section 6 gives several reasons that we cannot simply exploit existing web cache eviction techniques. The other reason for that client-side cloud cache replacement policies must be designed differently from the traditional WWW ones lies in the intrinsic characteristic of cloud computing data itself: Cloud data objects (e.g., big data) have larger sizes than traditional WWW objects [22]. To our current experiences, requesting for big objects made available via cloud computing services (e.g., cloud storages) still keep users waiting for long whereas downloading small objects from clouds nowadays is fast as if the objects were fetched from user locality. This means that priority must be given to the loading optimization of big objects whereas existing web cache replacement policies had been totally optimized for small objects [23], [24] (because, in the past, fetching small data object was so delayed due to slow Internet connection that it was unacceptable for users to experience such delays on every request).

As an early attempt in the field of enterprise-level client-side shared cloud caching towards multi-provider clouds, this paper presents an intelligent cloud cache replacement policy, i-Cloud

(named so for its intended application domain), along with its technical and economical performances and important findings.

The organization of this paper is as follows. Section 2 explains monetary cost models as a basis for multi-provider cloud employment. Section 3 explains our experimental data sets and how the raw traces were preprocessed to obtain training and evaluation data sets that realistically represent the traffics of shared cloud networks. i-Cloud is described in detail in Section 4. Section 5 discusses comparative performance results and presents algorithmic success factor analysis to formulate an open question for future research in this new field of study. We extensively contrast and compare this work with relevant ones in Section 6. Section 7 draws the conclusion of crucial findings.

Access through your organization

Check access to the full text by signing in through your organization.

Access through **your organization**

Section snippets

Monetary cost models

When an organization employs cloud computing, there are two possible monetary cost models [25] depending on the number of contracted cloud providers. If a single provider is contracted, a uniform cost model is employed in which object requests made by the organization entirely go to the same cloud provider, thus a single data-out charge rate is applied to all of the requested objects. Otherwise, the other nonuniform cost model is applied in which object requests separately go to multiple cloud …

Experimental data sets

To understand the following sections, it is necessary to clarify our original data sets and their descendant ones used throughout this study.

Since a cloud consumer enterprise usually deploys various cloud computing services and locate them by specific domain names. This means that an enterprise-level shared cloud cache (the focus of this paper) must serve mixed HTTP request streams to multiple cloud services. Hence, raw HTTP traces, each of which was composed of mixed request streams generated …

A proposed cloud cache replacement policy

To design a cloud cache eviction approach requires the new perspective of problem understanding as justified in Section 1. A set of the design goals of i-Cloud that truly accommodates the efficient decisions of cached object evictions in cloud computing environments are the optimization of cost-saving ratio, byte-hit ratio and delay-saving ratio. These three performance metrics including a standard hit rate metric are revisited below. The detailed statements of the design goals have been …

Results and discussions

We have evaluated the i-Cloud by HTTP trace-driven simulations. In particular, the one-month data sets listed in Table2 have been used as separate workloads to supply the values of i-Cloud parameters denoted in Eq. (5) except c_i, which was set to either 0.1535 USD/GB or 0.0829 USD/GB depending on the domain names of object references as explained in Section 3. Besides using the multiple independent workloads for cross checking, we have also varied the configurations of i-Cloud's window size …

Related works

We have characterized a number of client-side web cache eviction schemes based on their key parameters that distinguish them from our scheme as follows. …

Conclusions

This paper presents i-Cloud cache replacement policy that is capable of improving multi-provider cloud economy, scalability and responsiveness. i-Cloud has access recency as a priority factor for object replacement decision. i-Cloud parameterizes an MLP-based self-tuning window size to generalize the recencies of objects within a formulated object cluster. The lowest profitable clustered objects are purged from cloud cache. By trace-driven simulations, we discovered the following facts. i-Cloud …

Acknowledgments

This research is financially supported by Thailand's Office of the Higher Education Commission in conjunction with Thailand Research Fund, and Sripatum university (grant MRG5580114). The authors also thank Duane Wessels, National Science Foundation (grants NCR-9616602 and NCR-9521745) and the National Laboratory for Applied Network Research for the trace data used in this study. …

Thepparit Banditwattanawong received B.E. from King Mongkuts Institute of Technology Ladkrabang, Thailand and M.E. from Asian Institute of Technology, Thailand. He obtained Ph.D. in Informatics from The Graduate University for Advanced Studies, Japan. He was a founder and leader of a national project Thailand's IPv6 testbed at National Electronics and Computer Technology Center. He is currently an assistant professor and an acting director of Ph.D.IT. program at Sripatum university. His main …

…

…

Recommended articles

References (65)

S. Marston *et al.*
Cloud computing—the business perspective
Decis. Support Syst. (2011)

M. Arlitt *et al.*
Performance evaluation of web proxy cache replacement policies
Perform. Eval. (2000)

M.I. Jordan *et al.*
Forward models: Supervised learning with a distal teacher
Cogn. Sci. (1992)

J.-C. Bolot *et al.*
Performance engineering of the world wide web: application to dimensioning and cache design
Comput. Netw. ISDN Syst. (1996)

J. Cobb *et al.*
Web proxy cache replacement scheme based on back-propagation neural network
J. Syst. Softw. (2008)

W. Ali *et al.*
Intelligent web proxy caching approaches based on machine learning techniques
Decis. Support Syst. (2012)

T. Koskela *et al.*

Web cache optimization with nonlinear model using object features
Comput. Netw. (2003)

Amazon.com, Inc., Amazon Web Services: Big Data, 1 February 2014. URL:...

Google Inc., Google App Engine: BigQuery, 1 February 2014. URL:...

Microsoft, Windows Azure: Big Data, 1 February 2014. URL:...

View more references

Cited by (10)

On formulation of online algorithm and framework of near-optimally tractable eviction for nonuniform caches

2020, Computer Networks

Citation Excerpt :

...The distributed data sharing incurs nonuniform costs, for instance, origin server loads, monetary server-data-out charges, downloading latencies, and consumed downstream bandwidth. To optimize such costs, the client initiated replication of data, aka caching, is widely adopted in various forms such as traditionally single caching proxies [1], cooperative caches [2], content-centric caches [3], and cloud caches [4]. The focus of this paper is noncooperative caching in host centric networks as opposite to content centric networks....

Show abstract

Analytics in/for cloud-an interdependence: A review

2018, Journal of Network and Computer Applications

Citation Excerpt :

...Keeping above-mentioned issues in mind, we reviewed a number of papers under this section. The reviewed literature are grouped under two category viz. resource allocation based study (Banditwattanawonga et al., 2016; Bennani and Menasce, 2005; Calheiros et al., 2015; Caron et al., 2010; Díaz et al., 2017; Grande et al., 2011; Guiyi et al., 2010; Imam et al., 2011; Kundu et al., 2012; Pandey et al., 2010; Rahman et al., 2011; Rankothge et al., 2017; Tsai et al., 2013; Zhang et al., 2017; Zhao et al., 2015, 2017) and scaling based study (Ashraf et al., 2016; Chou et al., 2016; da Rosa Righi et al., 2017; Gong et al., 2010; Jiang et al., 2013a, 2013b; Mao and Humphrey, 2011; Roy et al., 2011; Shahin, 2016). In a cloud environment, if resource allocation policy is effective and efficient then it saves considerable cost for the service provider....

Show abstract

Ontology-based broker system for interoperability of federated cloud computing platforms ↗

2021, International Journal of Computer Applications in Technology

The empirical discovery of near-optimal offline cache replacement policies for nonuniform objects ↗

2019, Advances in Intelligent Systems and Computing

A decentralized replica placement algorithm for edge computing ↗

2018, IEEE Transactions on Network and Service Management

Research on the prediction of the e-commerce profit based on the improved parallel PSO-LSSVM algorithm in cloud computing environment ↗

2016, International Journal of Grid and Distributed Computing

View all citing articles on Scopus ↗



Thepparit Banditwattanawong received B.E. from King Mongkuts Institute of Technology Ladkrabang, Thailand and M.E. from Asian Institute of Technology, Thailand. He obtained Ph.D. in Informatics from The Graduate University for Advanced Studies, Japan. He was a founder and leader of a national project Thailand's IPv6 testbed at National Electronics and Computer Technology Center. He is currently an assistant professor and an acting director of Ph.D.IT. program at Sripatum university. His main areas of research interests include cloud computing and distributed computing.



Masawee Masdisornchote received B.S. from Silpakorn university, Thailand and M.S. from King Mongkuts Institute of Technology Ladkrabang, Thailand. She is currently a full-time professor and a director of ICT department at Sripatum university. Her main research ares include opinion mining and cloud mining.



Putchong Uthayopas received his bachelor and master degree in Electrical Engineering from Chulalongkorn University in 1984 and 1988. He received master and Ph.D. in Computer Engineering from University of Louisiana in 1994 and 1996 accordingly. His research interest is in cluster computing, grid and cloud computing system and tools. He published more than 130 refereed publication in conferences and Journals. He is a co-founder of the Thai National Grid project. In 2012, he received a distinguish computer engineer award in system integration from the Engineering Institute of Thailand.

1 Sripatum university, 2410/2 Phaholyothin Road, Jatujak, Bangkok 10900, Thailand.

View full text

Copyright © 2015 Elsevier B.V. All rights reserved.

ELSEVIER

All content on this site: Copyright © 2026 Elsevier B.V., its licensors, and contributors. All rights are reserved, including those for text and data mining, AI training, and similar technologies. For all open access content, the relevant licensing terms apply.

RELX™