IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

A Desirable Strategy for Enhancing Cost Containment in Cloud Cache

Suganthi.J[1], Abinaya.B[2], R.G. Suresh Kumar[3], T. Nalini[4]

Master of Technology, Department of CSE, Rajiv Gandhi College of Engineering & Technology, Pondicherry, India[1,2]

Research Scholar, Vels University, Chennai, India[3]

Professor, Department of Computer Science and Engineering, Bharath University, Chennai, India[4]

Abstract: The cloud caching service can maximize its pricing scheme based on optimization .There are two major challenges while trying to define an optimal pricing scheme in caching service. The first is to define a simplified model of the price-demand dependency and the second is to establish the pricing scheme adaptable to modelling errors, time-dependent model changes and stochastic behaviour of the application. Also, the iterative optimization allows for re-definition of the parameters in the price-demand model, hence the pricing scheme should be adaptable to time changes. In this paper we propose an optimal pricing scheme and a method for the efficient computation of structure correlation by extending a cache-based query cost estimated subroutine and template-based workload compression technique is used. Pricing optimization proceeds in iterations on a sliding time-window that allows online corrections on the predicted demand, via re-injection of the real demand values at each sliding instant and hence the cost estimation and the consistency can be maintained.

Keywords*:* cloud cache, optimal pricing, template based workload compression, structure correlation.

I. INTRODUCTION

Cloud computing support computing resources (hardware and software) that are delivered as a service over a network and simply we pay for what we use(typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure in it.

Cloud computing confide remote services with specifying user's data, software and computation. Cloud computing purely relies on sharing of resources to achieve consistency and overall economies of scale similar to a utility (like the electricity grid) over a network**.** The foundation of cloud computing emerges the edge concept of converged infrastructure and services.

There are many services available in cloud computing such (IaaS),Platform as a service (PaaS),Software as a service (SaaS), are the main services used in cloud ,and the following are the additional services such as Storage as a service (STaaS), Security as a service (SECaaS), Data as a service (DaaS),Database as a service (DBaaS), virtual desktop infrastructure, API as a service (APIaaS), Backend as a service (BaaS).

A. Cloud Engineering

Cloud engineering is the application of engineering which focuses on cloud services and disciplines to cloud computing. It generates a systematic approach to the high-level concerns of commercialization, standardization, and governance in conceiving, developing, operating and maintaining cloud computing systems.

It is a method of encompassing diverse areas such as system engineering, software engineering performance.

II. RELATED WORK

Cloud computing, by providing the new trend for service infrastructures requires user multi-tenancy as well as minimal capital expenditure. In cloud services that provide large amounts of data are massively collected and queried, such as scientific data, where users typically pay for queried services. The cloud supports caching of data and high storage in order to provide quality query services. User payments cover query execution costs and maintenance of cloud infrastructure, and incur cloud profit [1]. An economic model for self-tuned cloud caching that aims the service of scientific data is reworked to policies and encourage high quality individual and refine the overall query services but also support the profit of the cloud. This economy achieves minimal capital expenditure and, meanwhile the services of multiple users are in an efficient and also resource-economic way [2].

Recently many techniques cache the optimizer's output and evaluate plans regarding the cached results, reducing the number of calls to the optimizer [3]. However, the cost of invoking the optimizer to fill the cache is nontrivial reduce scalability when running workloads with thousands of queries. The intermediate optimization results in a dynamic programming based optimizer to reduce the cache initialization overhead and demonstrate the accuracy and efficiency of their techniques by implementing them on the Postgre SQL open source query optimizer. For a star-schema workload, our techniques build the cost model 5 to 10 times faster than the conventional approach, while preserving accuracy. To demonstrate the effectiveness of the technique we build PINUM, a fast and low overhead proof of concept (POC) cost model for Postgre SQL DBMS. PINUM reduces the construction overhead of a query plan cache by a factor of at least 5, without compromising accuracy. The basic concepts of three branches of game theory, leader–follower, cooperative, and two-person nonzero sum games, are reviewed and applied to the study of the Internet pricing issue [4]. In particular, they have emphasized that the cooperative game (also called the bargaining problem) provides an overall picture for the issue. With a simple model for Internet quality of service (QoS), we demonstrate that the leader–follower game may lead to a solution that is not Pareto optimal and in some cases may be "unfair," and that the cooperative game may provide a better solution for both the Internet service provider (ISP) and the user. It will be overcome by cooperative game approach to Internet pricing. With a simple QoS model, we demonstrated that the leader–follower game may lead to a solution that is not Pareto optimal and in some sense may be unfair, but the cooperative approach can provide a better solution [4]. Grid economy provides a mechanism or inducement for resource owners to be part of the Grid, and encourages users to utilize resources optimally and effectively. Revenue Management (RM) is proposed for determine pricing of reservations in Grids in order to increase ports is used. The aim of revenue management is to periodically update the prices in response to market demands, by charging different fares to different customers for a same resource [5]. Common resources that can be reserved are compute nodes and network bandwidth. They presented a novel approach of using Revenue Management (RM) to determine pricing of reservations in a Grid system.

III. PROBLEM ANALYSIS

There are two major challenges when trying to define an optimal pricing scheme for the cloud caching service. The first is to define a simplified enough model of the price-demand dependency, to achieve a feasible pricing solution, but not an oversimplified model. For example, a static pricing scheme cannot be optimal if the demand for services has deterministic seasonal fluctuations. The second challenge is to define a pricing scheme that is adaptable to 1) modelling errors, 2) time-dependent model changes, and 3) stochastic behaviour of the application. The demand for services, for instance, may depend in a non-predictable way on factors that are external to the cloud application, such as socioeconomic situations.

A. static pricing

A static pricing scheme cannot be optimal due to fluctuations in the services and very with the demand for services .Static pricing results in an unpredictable and, therefore, uncontrollable behaviour of profit. This economy is limited to offering budget options to the users, and does not propose any pricing scheme. Other solutions for similar frameworks focus on job scheduling and bid negotiation**,** issues orthogonal to optimal pricing.

IV. PROPOSED SCHEME

The cloud caching service can maximize its profit using an optimal pricing scheme. Optimal pricing necessitates an appropriately simplified price-demand model that incorporates the correlations of structures in the cache services. The pricing scheme should be adaptable to time changes.

A. Price adaptively to time changes

Profit maximization is pursued in a finite long-term horizon. The horizon includes sequential non-overlapping intervals that allow for scheduling structure availability. At the beginning of each interval, the cloud redefines availability by taking offline some of the currently available structures and taking online some of the unavailable ones. Pricing optimization proceeds in iterations on a sliding time-window that allows online corrections on the predicted demand, via re-injection of the real demand values at each sliding instant. Also, the iterative optimization allows for re-definition of the parameters in the price-demand model, if the demand deviates substantially from the predicted.

B. Modelling structure correlation

Our approach models the correlation of cache structures as a dependency of the demand for each structure on the price of every available one. Pairs of structures are characterized as competitive, if they tend to exclude each other, or collaborating, if they coexist in query plans. Competitive pairs induce negative, whereas collaborating pairs induce positive correlation. Otherwise correlation is set to zero. The index-index, index column, and column-column correlations are estimated based on proposed measures that can estimate all three types of correlation. We propose a method for the efficient computation of structure correlation by extending a cache-based query cost estimation module and a template-based workload compression technique.

V. SYSTEM DESIGN

A. Input design

The input design act as a mediator between the information system and the user. It embraces the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form. Processing can be done by examining the computer to read data from a written or printed document. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple.

B. Output design

A quality output is one, which meets the requirements of the end user and presents the information clearly. The results of processing the system are communicated to the users and to other system through outputs. Thus the output determines show the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized manner; the right output must be developed

IJARCCE

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

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

while ensuring that each output element is designed in a way such that it is easy for the people to understand.

2. When analyzing the design of the computer output, they should identify the specific output that is needed to meet the requirements.

3. Select the most appropriate methods for presenting information.

4. Create document, report, or other formats that contain information produced by the system. The output form should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the future.
- Identify the important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

VI. IMPLEMENTATION

Implementation is the process of the theoretical design is turned out into a working system. Thus it is considered to be the most important stage in emphasizing a successful new system in giving the user, the confidence that the new system will work and be effective. This stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing the methods to achieve changeover.

A. System architecture

In the system architecture the coordinator or computer consist of the shared file system and four CPU which is given as cpu1, cpu2, cpu3, cpu4. These four CPU are connected to the shared file system within the coordinator. Here we are using the database as the backend. The user will interact with the coordinator through internet and the coordinator will use the back end.

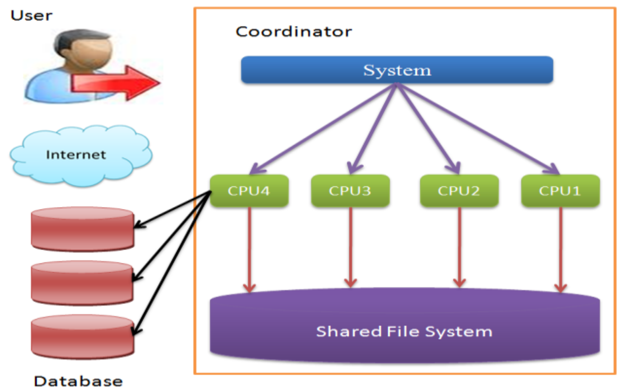

Fig. 1 System Architecture

B. Algorithm

Global: cache structures S, prices P, availability Δ
Query Execution ( )
ifqcan be satisfied in the cache then
(result, cost)←runQueryInCache(q)
else
(result, cost)←runQueryInBackend (q)
end if
S←addNewStructures()
returnresult, cost
optimalPricing(horizon T, intervals t[i], S)
(Δ,P)←determineAvailability&Prices (T, t,S)
returnΔ,P
main()
execute in parallel tasks T1 and T2:
T1:
forevery new i do
slide the optimization window
OptimalPricin(T, t[i], S)
end for
T2:
While new query q do
(Result, cost)←query Execution (q)
end while
ifqexecuted in cache then
Charge cost to user
else
Calculate total price and charge price to user
end if

C. Query execution

The cloud cache is a full-fledged DBMS along with a cache of data that reside permanently in back-end database. The goal of the cloud cache is to tender cheap efficient multi-user querying on the back-end data, while maintaining the cloud provider profitable. Service of queries are performed by executing either in the cloud cache or in the back-end database. The faster the execution, the more data structures it employs, and therefore, the more expensive the service. Thus the cloud infrastructure provides sufficient amount of storage space for a large number of cache structures and maintaining the cost effectively.

D. Optimal pricing

We assume that each structure is built from groove in the cloud cache, as the cloud may not have administration rights on existing back-end structures. Nevertheless, cheap computing and parallelism on cloud infrastructure may raise the performance of structure creation. For a column, the building cost is the cost of transferring it from the back end and combining it with the currently cached columns. This cost may contain the cost of net grating the column in the existing cache table. Since sorting is the most important step in building an index, the cost of building an index is approximated to the cost of sorting the indexed columns. In case of multiple cloud databases, the cost of data movement is incorporated in the building cost. Hence, building a column or an index in the cache has a one-time static cost, whereas their maintenance yields a storage cost that is linear with time.

VII. CONCLUSION

This work proposes a new original pricing scheme designed for cloud cache that offers querying services and scope at the maximization of the cloud profit. We define a suitable Price-demand model and we express the optimal pricing problem.

The proposed solution allows- on one hand long-period profit maximization, and, on the other, dynamic calibration to the undeniable behaviour of the cloud application, while the optimization process is in progress. We discuss qualitative aspects of the solution and a variation of the problem that allows the consideration of user satisfaction together with profit maximization. The success of the pricing solution is ensured with the proposal of a method that estimates the interrelationship of the cache services that are used in a time-efficient manner.

REFERENCES

[1] Linhai He and Jean Walrand, "Pricing Differentiated Internet Services" in IEEE Transactions on Cloud computing, March 2011.

[2] Debabrata Dash, VerenaKantere and Anastasia Ailamaki, "An economic model for self-tuned cloud caching", in IEEE Transactions on Cloud Computing, July 2011.

[3] Debabrata Dash, Ioannis Alagiannis, Cristina Maier, and Anastasia Ailamaki, "Caching All Plans with Just One Optimizer Call", in IEEE Transactions on Cloud Computing, January 2012.

[4] Xi- Ren Cao, Hong-Xia Shen, Rodolfo Milito, and PatricaWirth.n, "Internet Pricing With a Game Theoretical Approach- Concepts and Examples", in IEEE Transactions on Cloud Computing, April 2011.

[5] Anthony Sulistio, KyongHoon Kim and Rajkumar Buyya, "Using Revenue Management to Determine Pricing of Reservations", in IEEE Conference, Jan 2012.

[6] Mark D. Ryan, "Cloud Computing Privacy Concerns on our Doorstep", in IEEE Transactions on Cloud Computing, September 2012.

[7] Xiong.N, Jiangxi, Han.W, "Green cloud computing schemes based on networks: a survey", in IEEE Transactions on Cloud Computing, December 2012.