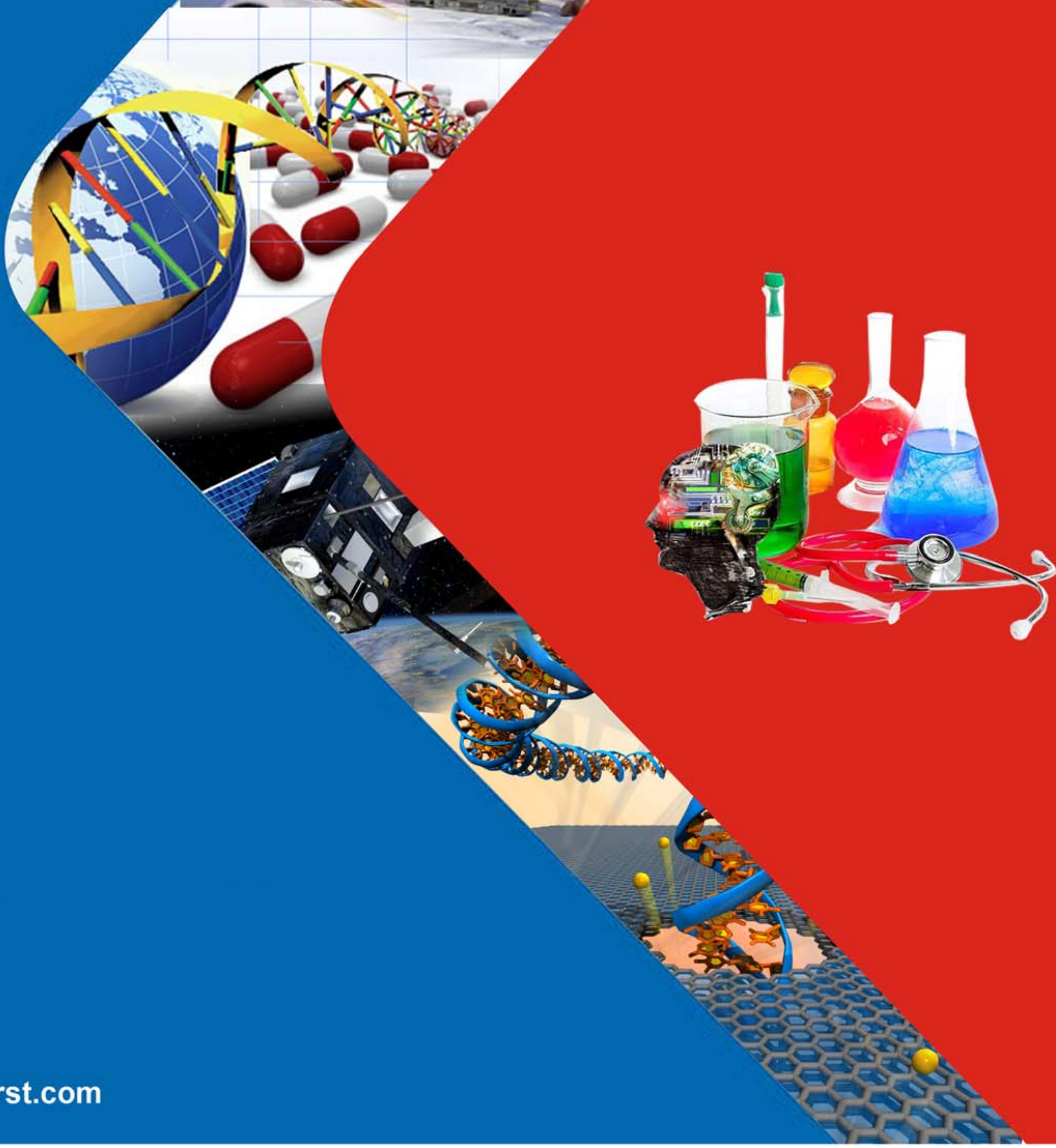
International Journal of Engineering Research and Science & Technology

IJERST

ISSN : 2319-5991
Vol. 3, No. 1
February 2014



www.ijerst.com

Email: editorijerst@gmail.com or editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 3, No. 1, February 2014

Research Paper

DEMAND PRICING & RESOURCE ALLOCATION IN MARKET-BASED COMPUTE GRIDS

Malini K[1]* and V Udhaya Kumar[1]

**Corresponding Author: Malini K* ✉ kmalini80@gmail.com*

Cloud applications that offer data management services are emerging. Such clouds support caching of data in order to provide quality query services. The users can query the cloud data, paying the price for the infrastructure they use. Cloud management necessitates an economy that manages the service of multiple users in an efficient, but also, resource economic way that allows for cloud profit. Naturally, the maximization of cloud profit given some guarantees for user satisfaction presumes an appropriate price-demand model that enables optimal pricing of query services. The model should be plausible in that it reflects the correlation of cache structures involved in the queries. Optimal pricing is achieved based on a dynamic pricing scheme that adapts to time changes. This paper proposes a novel price-demand model designed for a cloud cache and a dynamic pricing scheme for queries executed in the cloud cache. The pricing solution employs a novel method that estimates the correlations of the cache services in an time-efficient manner.

Keywords: Cloud data management, Data services, Cloud service pricing

INTRODUCTION

The leading trend for service infrastructures in the IT domain is called cloud computing, a style of computing that allows users to access information services. Cloud providers trade their services on cloud resources for money. The quality of services that the users receive depends on the utilization of the resources. The operation cost of used resources is amortized through user payments. Cloud resources can be anything, from infrastructure (CPU, memory, bandwidth, network), to platforms and applications deployed on the infrastructure. Cloud management necessitates an economy, and, therefore, incorporation of economic concepts in the provision of cloud services. The goal of cloud economy is to optimize: (i) user satisfaction and (ii) cloud profit. While the success of the cloud service depends on the optimization of both objectives, businesses typically prioritize profit. To maximize cloud profit we need a pricing scheme that guarantees user satisfaction while

[1] Department of Computer Science and Engineering, PRIST University Pondicherry, India.

adapting to demand changes .Recently, cloud computing has found its way into the provision of web services]. Information, as well as software is permanently stored in Internet servers and probably cached temporarily on the user side. Current businesses on cloud computing such as Amazon Web Services and Microsoft Azure have begun to offer data management services: the cloud enables the users to manage the data of back-end databases in a transparent manner.

Applications that collect and query massive data, like those supported by CERN, need a caching service, which can be provided by the cloud. The goal of such a cloud is to provide efficient querying on the back-end data at a low cost, while being economically viable, and furthermore, profitable. Figure 1 depicts the architecture of a cloud cache. Users pose queries to the cloud through a coordinator module, and are charged on-the-go in order to be served. The cloud caches data and builds data structures in order to accelerate query execution. Service of queries is performed by executing them either in the cloud cache (if necessary data are already cached) or in a back-end database. Each cache structure (data or data structures) has an operating (i.e. a building and a maintenance) cost. A price over the operating cost for each structure can ensure profit for the cloud. In this work we propose a novel scheme that achieves optimal pricing for the services of a cloud cache.

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy n company strength. Once these things r satisfied, ten next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into account for developing the proposed system. We have to analysis the Cloud Computing Outline Survey

PROPOSED SYSTEM

This paper proposes a novel price-demand model designed for a cloud cache and a dynamic pricing scheme for queries executed in the cloud cache. The pricing solution employs a novel method that estimates the correlations of the cache services in a time-efficient manner. The experimental study shows the efficiency of the solution. The operation cost of used resources is amortized through user payments. Cloud resources can be anything, from infrastructure (CPU, memory, bandwidth, network), to platforms and applications deployed on the infrastructure. Cloud management necessitates an economy, and, therefore, incorporation of economic concepts in the provision of cloud services. The goal of cloud economy is to optimize: (i) user satisfaction and (ii) cloud profit. While the success of the cloud service depends on the optimization of both objectives, businesses typically prioritize profit. To maximize cloud profit we need a pricing scheme that guarantees user satisfaction while adapting to demand changes. Service of queries is performed by executing them either in the cloud cache (if necessary data are already cached) or in a back-end database used in cloud cache.

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. Modeling Structure Correlations

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.

Advantage

- A novel demand-pricing model designed for cloud caching services and the problem formulation for the dynamic pricing scheme that maximizes profit and incorporates the objective for user satisfaction.
- An efficient solution to the pricing problem, based on non-linear programming, adaptable to time changes.
- A correlation measure for cache structures that is suitable for the cloud cache pricing scheme and a method for its efficient computation.

C. Algorithm

Global: cache structures *S*, prices *P*, availability Δ

Query Execution ( )

if *q* can be satisfied in the cache **then**

(result, cost)•!runQueryInCache (q)

else

(result, cost)←runQueryInBackend (q)

end if

S←addNewStructures ()

return result, cost

optimalPricing (horizon *T*, intervals *t*[*i*], *S*)

(Δ,*P*)←determineAvailability&Prices (T, t,S)

return Δ,*P*

main ()

execute in parallel tasks T1 and T2:

T1:

for every 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

if *q* executed in cache **then**

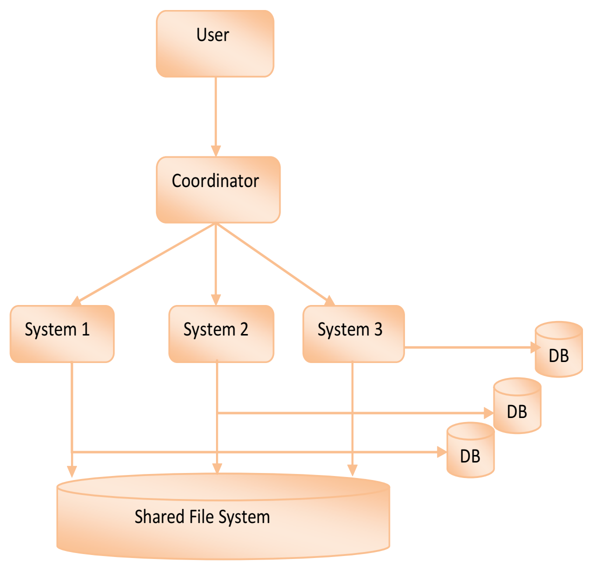
Charge *cost* to user

else

Calculate total price and charge price to user

end if

Figure 1: Architecture



IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

MODULES

A. Query Execution

The cloud cache is a full-fledged DBMS along with a cache of data that reside permanently in back-end databases. The goal of the cloud cache is to offer cheap efficient multi-user querying on the back-end data, while keeping the cloud provider profitable. Service of queries is performed by executing them either in the cloud cache or in the back-end database. Query performance is measured in terms of execution time. The faster the execution, the more data structures it employs, and therefore, the more expensive the service. We assume that the cloud infrastructure provides sufficient amount of storage space for a large number of cache structures. Each cache structure has a building and a maintenance cost.

B. Optimal Pricing

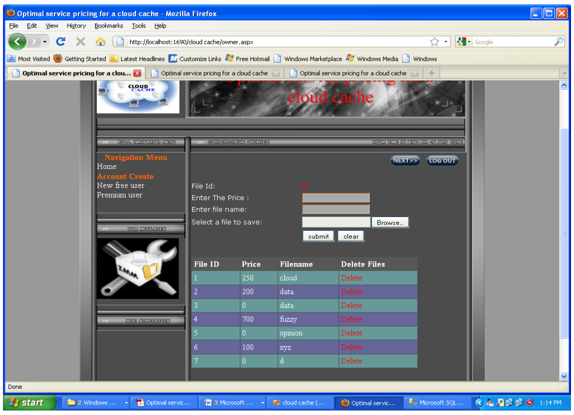
We assume that each structure is built from scratch in the cloud cache, as the cloud may not have administration rights on existing back-end structures. Nevertheless, cheap computing and parallelism on cloud infrastructure may benefit the performance of structure creation. For a column, the building cost is the cost of transferring it from the backend and combining it with the currently cached columns. This cost may contain the cost of nte grating the column in the existing cache table. For indexes, the building cost involves fetching the data across the Internet and then building the index in the cache.

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. The maintenance cost of a column or an index is just the cost of using disk space in the cloud. 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.

SCREEN SHOTS

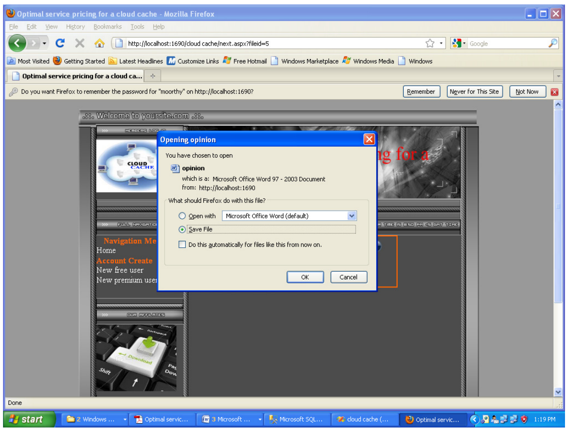
Price adaptively to time changes:

Figure 2: Screen Shot-Price Adaptively to Time Changes



MODELING STRUCTURE CORRELATIONS

Figure 3: Screen Shot-Modeling Structure Correlations



FUTURE ENHANCEMENT

The prediction of future behavior of a system is subject to unpredictable perturbations. Hence, the longer the horizon is, the more error-prone the optimization procedure is, as the prediction accuracy of the behavior of demand, tends to decrease with time.

CONCLUSION

This work proposes a novel pricing scheme designed for a cloud cache that offers querying services and aims at the maximization of the cloud profit. We define an appropriate Price-demand model and we formulate the optimal pricing problem. The proposed solution allows: on one hand, long-term profit maximization, and, on the other, dynamic calibration to the actual behavior 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 viability of the pricing solution is ensured with the proposal of a method that estimates the correlations of the cache services in an time-efficient manner.

REFERENCES

1. Applied Microsoft®.NET Framework Programming (Pro-Developer) by Jeffrey Richter.

2. Bruno N and Chaudhuri S (2007), An Online Approach to Physical Design Tuning, ICDE'07.

3. Ch. Chen, Muthucumaru Maheswaran and Michel Toulouse (2002), "Supporting co-allocation in an auctioning-based resource allocator for grid systems", in IPDPS.

4. Choenni S, Blanken H M and Chang T (1993), "On the selection of secondary indices in relational databases", *DKE*, Vol. 11, No. 3.

5. Behrouz A Forouzan (2006), *Data Communications and Networking*, McGraw-Hill.

6. Gabriel R Bitran and Rene Caldentey (2002), "An overview of pricing models for

revenue management", in *Manufacturing & Service Operations Management*, Vol. 5, pp. 203–209

7. Varena Kantere, Debabrata Dash, Gregory Francois, Sofia Kyriakopolou and Anastasia Ailmaki (2011), "Optimal Service Pricing for a Cloud Cache", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No.9.

8. Xi-Ren Cao, Hong-Xia Shen, Milito R and Wirth P (2002), "Internet pricing with a game theoretical approach: concepts and examples", *Networking, IEEE/ACM Transactions on*, Vol. 10, No. 2, pp. 208-216.

Sites Referred

1. http://aws.amazon.com/.
2. http://code.google.com/appengine/.
3. http://tomopt.com/tomlab/.
4. http:/www.cern.ch/.
5. http://www.gogrid.com/.
6. http://www.microsoft.com/azure/.

International Journal of Engineering Research and Science & Technology
Hyderabad, INDIA. Ph: +91-09441351700, 09059645577
E-mail: editorijlerst@gmail.com or editor@ijerst.com
Website: www.ijerst.com

ISSN 2319-5991