Dynamic Management of Resources and Workloads for RDBMS in Cloud: a Control-theoretic Approach

Pengcheng Xiong
Georgia Institute of Technology
266 Ferst Drive
Atlanta, GA 30332
xiong@gatech.edu
Expected graduation date: Fall 2012
Advisor: Professor Calton Pu

ABSTRACT

As cloud computing environments become explosively popular, dealing with unpredictable changes, uncertainties, and disturbances in both systems and environments turns out to be one of the major challenges facing the concurrent computing industry. My research goal is to dynamically manage resources and workloads for RDBMS in cloud computing environments in order to achieve "better performance but lower cost", i.e., better service level compliance but lower consumption of virtualized computing resource(s).

Nowadays, although control theory offers a principled way to deal with the challenge based on feedback mechanisms, a controller is typically designed based on the system designer's domain knowledge and intuition instead of the behavior of the system being controlled. My research approach is based on the essence of control theory but transcends state-of-the-art control-theoretic approaches by leveraging interdisciplinary areas, especially from machine learning. While machine learning is often viewed merely as a toolbox that can be deployed for many data-centric problems, my research makes efforts to incorporate machine learning as a full-fledged engineering discipline into control-theoretic approaches for realizing my research goal.

My PhD thesis work implements two solid systems by leveraging machine learning techniques, namely, ActiveSLA and SmartSLA. ActiveSLA is an automatic controller featuring risk assessment admission control to obtain the most profitable service-level compliance. SmartSLA is an automatic controller featuring cost-sensitive adaptation to achieve the lowest total cost. The experimental results show that both of the two systems outperform the state-of-the-art methods.

Categories and Subject Descriptors

D.4.8 [**OPERATING SYSTEMS**]: Performance—*Measurements,Modeling and prediction*; K.6.0 [**MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS**]: General—*Economics*

General Terms

Algorithms, Experimentation, Performance

Keywords

Cloud computing, Database-as-a-service, Machine learning

SIGMOD/PODS'12 PhD Symposium, May 20, 2012, Scottsdale, AZ, USA.

1. INTRODUCTION

Cloud computing is the delivery of computing as a service whereby shared resources, software and information are provided as a utility (like the electricity grid) over a network (typically the Internet) [1]. As an emerging computing paradigm, Cloud computing brings both opportunities and challenges for data management services [4, 5]. On one hand, data management service providers enjoy the opportunity of cost reduction through tenant consolidation. They always pursue high resource utilization, because the higher the utilization by consolidating multiple clients in shared infrastructures, the lower the hardware cost, operational cost and maintenance cost. On the other hand, data management service providers suffer from time-varying workload because resource utilization should not be pushed too high or the service-level compliance under time-varying workload could be jeopardized. The goal of my thesis dissertation is to understand and analyze this resource sharing dilemma due to the tension between high resource utilization and service-level compliance under workload variation for database service providers.

Following the guidance from my advisor Professor Calton Pu, I study this dilemma as an optimization problem, i.e., how to maximize SLA-based profit [15, 16]. We prefer SLA-based profit over other metrics such as average query execution time because it considers not only the revenue but also the cost, which determine a database service provider's final profit. The revenue is generated by the service level compliance between a data management service provider and a customer. It may also include SLA penalty cost when the service level agreement is violated, e.g., too high query execution latency. The cost includes infrastructure cost (virtualized resource cost) and adaptation cost.

In multitenant databases, there are several different levels of sharing [3], including private virtual machine (VM), private database, private table, and shared table. I study two cases so far for my PhD thesis, i.e., shared table and private virtual machine. I explore an adaptive control-theoretic approach, i.e., admission control and resource allocation to maximize SLA-based profit for the two cases. My contributions can be summarized in the following thesis statement:

Thesis statement: *State-of-the-art control theoretic approaches to automated resources and workloads management of RDBMS can be enhanced by leveraging machine learning techniques. Automatic control featuring risk assessment admission control and cost-sensitive adapatation obtains the most profitable service-level compliance, achieves the lowest total cost, outperforming the state-of-the-art methods.*

The thesis statement will be demonstrated by the work in this paper, i.e., ActiveSLA [16] and SmartSLA [15].

ActiveSLA demonstrates that a state-of-the-art control theoretic approach enhanced by risk assessment admission control obtains the most profitable service-level compliance. Different from the traditional admission control, ActiveSLA estimates the probability risk for each query to meet/miss the service-level agreement after it is admitted. Based on the risk assessment and the expected profit/penalty, ActiveSLA determines whether or not to admit the query to obtain the most profitable service-level compliance. Due to the risk assessment admission control enhancement, experimental results show that ActiveSLA is able to make admission control decisions that can obtain at least 20% more profit than several state-of-the-art methods.

SmartSLA demonstrates that a state-of-the-art control theoretic approach enhanced by cost-sensitive adaptation achieves the lowest total cost. Different from the traditional resource allocation, SmartSLA not only takes into consideration of the cost due to SLA violation and the cost of virtualized resources but also the adaptation cost for resource allocation. Based on the cost models, SmartSLA determines the resource allocation and takes actions in a cost-sensitive way to minimize the total cost. Due to the cost-sensitive adaptation enhancement, experimental results show that SmartSLA is able to make resource allocation decisions that can save at least 20% cost than several state-of-the-art methods.

2. ACTIVESLA: ADAPTIVE ADMISSION CONTROL FOR CLOUD DATABASES

ActiveSLA is an automatic control system featuring risk assessment admission control to help Cloud database service providers make admission control decisions to obtain the most profitable service-level compliance.

2.1 Problem definition

Although Cloud computing enables sharing of resources and costs across a large pool of customers, the bursty workloads from multiple customers can potentially overload the system and as a result cause significant performance degradation among all customers. The performance degradation will break service-level agreement and make the Cloud service provider lose profit. As a candidate solution for system overload, admission control can stall the new requests (e.g., [7]) or reject (e.g., [13]) the new requests when the system is near an overload condition until the system condition improves. However, the state-of-the-art admission control techniques do not work directly towards the main goal of database service providers—namely to maximize their profit by satisfying different SLAs for their clients. There are two major challenges, i.e., (1) Merely estimating the query execution time is not enough to make profit-oriented decisions. For two queries that have the same estimated query execution time, the probabilities of them meeting and missing their deadline may be totally different, which result in the diverse admission decisions. (2) Because of diverse SLAs, we may have to make different admission control decisions even when the queries have the same deadline and the same probability of meeting the deadline. For example, we are more likely to admit a query with higher profit than a query with lower profit under the same deadline and the same probability of meeting this deadline.

2.2 Our contribution

Our ActiveSLA successfully overcomes the above challenges by leveraging decision theory which is concerned with identifying the profit and penalty values, uncertainties and other issues relevant to an admission decision. The main contributions of ActiveSLA are twofold:

(1) ActiveSLA shows, by using both theoretical reasoning and empirical evaluation, how appropriate machine learning techniques can be successfully used to answer a key question of admission control for database service providers: "What is the probability for a query to meet or miss its deadline?" ActiveSLA uses machine learning techniques to (a) take many query related features as well as database system related features into consideration, (b) recognize complex patterns from the data in an automatic way, and (c) provide detailed probabilities for different outcomes.

More specifically, ActiveSLA has three unique characteristics: (a) it uses a non-linear learning method, (b) it is based on a classification model, and (c) it includes more comprehensive features, i.e., query type and mix (the number of currently running queries), query features (the number of sequential I/O and the number of non-sequential I/O), database and system conditions (buffer cache, system cache, transaction isolation level, and CPU, memory, and disk status).

(2) ActiveSLA makes decisions in a holistic fashion by considering (a) the probability for a new query to meet its deadline under the current system condition, (b) the profit consequences of alternative actions and outcomes, and (c) the potential impact of admitting a just arrived query on the currently running queries as well as on the future queries.

More specifically, ActiveSLA makes single-query as well as multiple-query profit-oriented decisions. For a single query, ActiveSLA makes profit-oriented admission control decisions by using the standard decision theory under a general SLA. Then we show that under a commonly used SLA form, namely step-function SLA, the decisions can be made in a more efficient way. For multiple queries, ActiveSLA takes into account the interference among clients (queries), who are competing with each other for the shared system resources.

2.3 Implementation and evaluation

Our ActiveSLA is implemented as two modules as shown in Figure 1. First, a prediction module is built to estimate the probability for a new query to finish the execution before its deadline. Second, based on the predicted probability, a decision module is built to determine whether or not to admit the given query into the database system. The decision is made with the profit optimization objective, where the

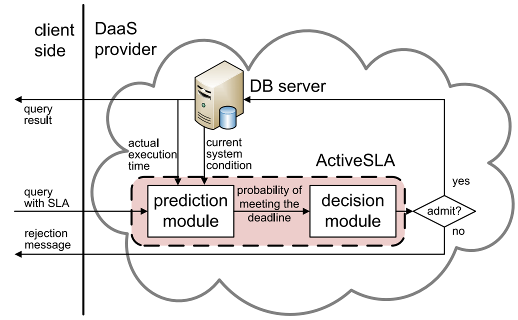

Figure 1: System architecture of ActiveSLA.

expected profit is derived from the service-level agreements between a service provider and its clients.

Our ActiveSLA is evaluated by extensive real system experiments with standard database benchmark TPC-W, under different traffic patterns such as static and dynamic traffic patterns, different RDBMS settings such as read commit and serialization, and different SLAs such as gold and silver SLAs. The evaluation results demonstrate that ActiveSLA can make a more precise prediction of whether the query will meet or miss the deadline and make more profit by obtaining better service-level compliance. For example, the prediction error for one of the state-of-the-art methods (Q-Cop) [13] is around 25% when we use a step-wise SLA of 30s. However, the prediction error for our ActiveSLA is around 13%, which cuts the prediction error almost by half. The more precise prediction also makes our ActiveSLA generate 20% more profit than other state-of-the-art methods.

3. SMARTSLA: VIRTUALIZED RESOURCE MANAGEMENT FOR CLOUD DATABASES

SmartSLA is an automatic control system featuring cost-sensitive adaptation to help Cloud database service providers make resource allocation decisions to achieve the lowest total cost. Compared with the previous section where we consider the case of shared table for multitenant databases, we consider the case of private virtual machine in this section, where each tenant database runs in its own virtual machine. This level of sharing allows us to explicitly control the system resources allocated for each VM, or the corresponding tenant. The current virtualization technologies allow packing of a large number of VMs into physical machines thereby increasing the cost efficiency of infrastructure resources [12]. While it seems quite attractive to consolidate multiple tenants into a physical machine, it requires careful planning and management in order to satisfy tenants' SLAs.

3.1 Problem definition

Let us consider an illustrative example shown in Fig. 2. In this example, we assume that there are two kinds of clients, e.g., a gold one and a silver one for the cloud database service provider. As their workload demand changes, they add or remove database slaves. The clients share the hardware resources where master and slaves are contained in a separate VM which is common in many web applications hosted on large clusters [17]. The service provider charges an agreed-upon fee if it delivers the service by meeting the SLAs and pays a penalty if it fails to meet the SLAs. Consequently, a failure to deliver on SLAs results in higher penalty for the gold client. In reality, of course, there may be more than two kinds of clients.

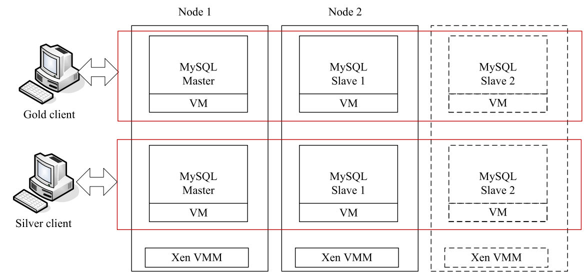

Figure 2: An illustration example where two clients are hosted in private virtual machines.

Thus, in order to obtain the most profit, it is important for the service provider to achieve the lowest total cost, which includes SLA penalty cost, infrastructure cost (virtualized resource cost) and adaptation cost. On one hand, the service provider should intelligently distribute limited resources, such as CPU and memory, among competing clients. On the other hand, some other resources, although not strictly limited, have an associated cost. Database replication is such an example. Adding additional database replicas not only involves direct cost (e.g., adding more nodes), but also has initiation cost (e.g., data migration) and maintenance cost (e.g., synchronization). The key issues to the successful management of resources are as follows:

Local Analysis : The first issue is to identify the right configuration of system resources (e.g., CPU, memory etc.) for a client to meet the SLAs while optimizing the revenue. Answers to such a question are not straightforward as they depend on many factors such as the current workload from the client, the client-specific SLAs, and the type of resources.

Global Analysis : The second issue that a service provider has to address is the decision on how to allocate resources among clients based on the current system status. For example, how much CPU share or memory should be given to the gold clients versus the silver clients, when a new database replica should be started, etc. Answers to such decisions obviously rely on the result of the above *Local Analysis* decisions.

3.2 Our contribution

Our SmartSLA successfully addresses the above issues by adaptive resource management. The main contributions of SmartSLA are twofold:

(1) SmartSLA shows, by using both statistical analysis and modeling, how appropriate machine learning techniques can be successfully used to answer two key questions of resource management: "How is the system performance in terms of SLA penalty cost correlated with the system configuration?" and "How can we accurately predict the system performance in terms of the SLA penalty cost?" SmartSLA uses statistical methods to discover the marginal distribution of SLA penalty cost on each of the parameters of CPU share, memory size, client workload, and replica number. It then uses machine learning techniques to model the system performance in terms of the SLA penalty cost with the combination of these parameters.

More specifically, SmartSLA has two observations. (1) From the statistical analysis, we can see that, some parameters such as CPU share, client workload, and replica number impact the SLA penalty cost in a near-linear fashion; some other parameters such as the memory size, impact the SLA penalty cost in a strongly nonlinear way. (2) A series of mature machine learning techniques, such as linear regression, regression tree and additive regression are investigated for predicting the system performance. The nonlinear machine learning technique additive regression has better performance than linear regression mainly due to the nonlinearity that we have observed from statistical analysis.

(2) SmartSLA makes decisions in a two-level fashion. On the first level, we allocate CPU and memory shares while we assume that the number of replicas is fixed and we only consider the SLA penalty cost. On the second level, we tune the number of database replicas to reduce the total cost, where the total cost includes not only the SLA penalty cost but also the infrastructure and the adaptation costs.

More specifically, SmartSLA combines an SLA penalty cost model, an infrastructure cost model and an adaptation cost model to minimize the total of cost which is composed of SLA penalty cost, infrastructure cost and the adaptation cost.

3.3 Implementation and evaluation

Our SmartSLA is implemented as two modules as shown in Figure 3. The first one is a system modeling module, which learns a model for the relationship between the resource allocation and expected cost for a single client. The second one is a resource allocation module, which dynamically makes decisions on the changes of the resource allocation among clients.

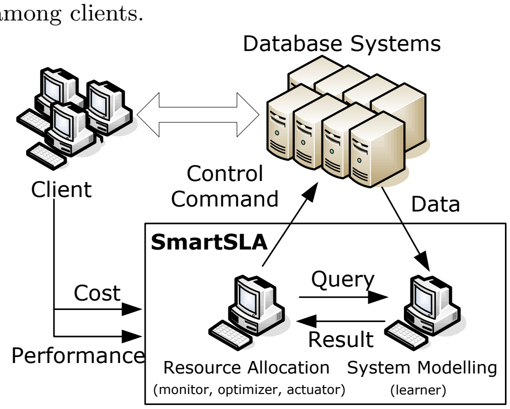

Figure 3: The architecture of our test bed.

Our SmartSLA is evaluated by extensive real system experiments with standard database benchmark TPC-W, under different traffic patterns such as static and dynamic traffic patterns and different SLAs such as gold and silver SLAs.

The evaluation results demonstrate that SmartSLA can make a more precise prediction of system performance in terms of the SLA penalty cost when nonlinear machine learning techniques are used. For example, the relative absolute error for linear regression is 44.0% while it is 28.6% with the additive regression. SmartSLA can reduce the total cost when multi-level control decisions are made. For example, under our experimental setting, the baseline cost is 2902. This value is reduced to 2464 when the first level controller is used. The value is further reduced to 2284 when the second level controller is also activated.

4. RELATED WORK

Dynamic management of resources and workloads for RDBMS is always one of the hottest topics in the Cloud computing. In this section, I summarize and compare the work related to my PhD research according to two aspects, i.e., admission control and resource allocation.

4.1 Admission control

Admission control can be used as an overload management technique to achieve the most profitable service-level compliance. Most of the state-of-the-art techniques are based on rejecting incoming work to a service by refusing to accept new requests. For example, Schroeder et al. [11] dynamically adjust the lowest MPL that corresponds to the best application performance. Popovici and Wilkes [10] use simulation to develop scheduling policies to make profit in the uncertain resource environment. The admission control mechanisms in the above work are general admission control mechanisms which can be not only used in general applications or systems, but also used in database management systems. Contrast to these admission control mechanisms which are oblivious to query types and query mixes, Q-Cop [13], QShuffler [2] and Gatekeeper [7] take into consideration the different requirement for different type of queries when admission control decisions are made in database management systems. For example, Q-Cop [13] is a prototype system for improving admission control decisions that considers a combination of the load on the database management system, the number of concurrent queries being executed, the actual mix of queries being executed, and the expected time a user may wait for a reply before they or their browser give up (i.e., time out).

ActiveSLA has the advantage of [11] where the decision module dynamically tunes the best MPL as there are different optimal MPLs for different workloads. ActiveSLA also has the advantage of [13, 2] where the query type and query mix are taken into consideration. However, ActiveSLA distinguishes itself from the above work in two major aspects. (1) It estimates the probability for a new query to meet/miss the service-level agreement after it is admitted. ActiveSLA builds a non-linear classification model to predict this probability rather than a linear regression model that is used in Q-Cop [13]. Moreover, besides query type and query mix that are used in existing work, ActiveSLA also takes into consideration query features as well as the database-specific and system-level metrics, which further help to improve the prediction accuracy. (2) The admission control decisions made by ActiveSLA are steered by service-level-agreements and expected profit. Therefore, differentiated services, which are very important in the Cloud databases, are provided.

4.2 Resource allocation

The virtual resource management in cloud environments has been studied with goals such as QoS awareness, performance isolation and differentiation as well as higher resource utilization. There are a plethora of work towards optimal CPU and memory partitioning with respect to the performance guarantees. For example, Pradeep et al. [9] develop an adaptive resource control system that dynamically adjusts the resource shares to applications in order to meet application-level QoS goals while achieving high resource utilization in the data center. Urgaonkar et al. [14] present techniques for provisioning CPU and network resources in shared hosting platforms. Most of the previous work uses linear model to design and implement the controller. However, there will be oscillation and the system will be unstable once the operation point moves out of the linear area. For example, as we show in this paper, there is a significant non-linear relationship between the performance and some of the system metrics. Compared with the previous work, we use machine learning technique to overcome the non-linear obstacle.

Besides the metrics that can be tuned for general systems, there are also lots of special database system intrinsic metrics that can be tuned to improve the database performance. For example, Duan et al. [6] tune the parameters of a database in order to get a better database performance. The most important problem is that the search space is huge and the optimal configuration is hard to find. Ganapathi et al. [8] use a machine learning technique called KCCA to predict metrics such as elapsed time, records used, disk I/Os, etc. Compared with KCCA, we focused on popular and easy-to-use techniques such as linear regression and boosting. Moreover, KCCA is sensitive to some modeling parameters such as the definition of the Euclidean distance and the scale factor. Although well-tuned parameters can give good prediction, bad parameter settings may cause significant degradation in the model's predictive power.

The work presented in Soror et al. [12] is most closely related to ours. There are two significant differences. (1) They model the problem as a service differentiation problem under the resource constraints. However, we model the problem as a two level optimization/control problem. Compared with theirs, we consider the cloud environments where database service provider can enjoy more flexibility to extend their resources. (2) They model the relationship between the performance and the system metrics like CPU and memory individually. However, in our work, we combine the system metrics, the number of replicas, and the arrival rate as the multiple input. Consequently our model is comprehensive to capture various relationships among system metrics and performance. Moreover, we also consider the adaptation cost related to database systems and provide a model for replica tuning.

5. CONCLUSIONS

In this paper, I summarize my up-to-date PhD thesis work. Firstly, I specify the problem of dynamic management of resources and workloads for RDBMS in Cloud that I am focusing on. Secondly, I describe the control-theoretic methodology that I plan to follow. Thirdly, I present in details of how to use admission control to manage workloads and how to use two-level resource allocation to manage virtualized resources. Finally, I summarize the contributions that I have made so far and compared my work with the state-of-the-art works.

The remaining work includes: (1) Besides RDBMS, I plan to extend control-theoretic approach to deal with different types of database systems to manage data and serve queries, e.g., NoSQL databases. (2) I plan to combine the management of resources and workloads together in a holistic way and explore the possibility of composition of dynamic admission control and resource allocation. (3) I also plan to investigate dynamic SLA suggestion and negotiation (e.g., when a user issues a query with a high SLA, let the system come back with a lower SLA at a lower cost, and see if the user is willing to take).

Acknowledgment

This research has been partially funded by National Science Foundation by IUCRC/FRP (1127904) , CISE/CNS (1138666), RAPID (1138666), CISE/CRI (0855180), NetSE (0905493) programs, and gifts, grants, or contracts from DARPA/I2O, Singapore Government, Fujitsu Labs, Wipro Applied Research, and Georgia Tech Foundation through the John P. Imlay, Jr. Chair endowment. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or other funding agencies and companies mentioned above.

6. REFERENCES

[1] http://en.wikipedia.org/wiki/Cloud_computing.

[2] M. Ahmad, A. Aboulanaga, and S. Babu. Modeling and exploiting query interactions in database systems. In *Proc. of CIKM*, 2010.

[3] S. Aulbach, T. Grust, D. Jacobs, A. Kemper, and J. Rittinger. Multi-tenant databases for software as a service: schema-mapping techniques. In *Proc. of SIGMOD*, 2008.

[4] M. Balazinska, B. Howe, and D. Suciu. Data markets in the cloud: An opportunity for the database community. *PVLDB*, 4(12):1482–1485, 2011.

[5] G. Chen, H. T. Vo, S. Wu, B. C. Ooi, and M. T. Özsu. A framework for supporting dbms-like indexes in the cloud. *PVLDB*, 4(11):702–713, 2011.

[6] S. Duan, V. Thummala, and S. Babu. Tuning database configuration parameters with iTuned. In *Proc. of the VLDB*, 2009.

[7] S. Elnikety, E. Nahum, J. Tracey, and W. Zwaenepoel. A method for transparent admission control and request scheduling in e-commerce web sites. In *Proc. of WWW*, 2004.

[8] A. Ganapathi, H. Kuno, U. Dayal, and J. L. Wiener. Predicting multiple metrics for queries: Better decisions enabled by machine learning. In *Proc. of ICDE*, 2009.

[9] P. Padala, K. Hou, X. Zhu, M. Uysal, Z. Wang, S. Singhal, A. Merchant, and K. Shin. Automated control of multiple virtualized resources. In *Proc. of EuroSys*, 2009.

[10] F. I. Popovici and J. Wilkes. Profitable services in an uncertain world. In *Proc. of SC*, 2005.

[11] B. Schroeder, M. Harchol-Balter, A. Iyengar, E. M. Nahum, and A. Wierman. How to determine a good multi-programming level for external scheduling. In *Proc. of ICDE*, 2006.

[12] A. Soror, K. S. U. Minhas, A. Aboulnaga, P. Kokosielis, and S. Kamath. Automatic virtual machine configuration for database workloads. In *Proc. of SIGMOD*, 2008.

[13] S. Tozer, T. Brecht, and A. Aboulnaga. Q-cop: Avoiding bad query mixes to minimize client timeouts under heavy loads. In *Proc. of ICDE*, 2010.

[14] B. Urgaonkar, P. Shenoy, and T. Roscoe. Resource overbooking and application profiling in shared hosting platforms. In *Proc. of OSDI*, 2002.

[15] P. Xiong, Y. Chi, S. Zhu, H. J. Moon, C. Pu, and H. Hacigumus. Intelligent management of virtualized resources for database systems in cloud environment. In *Proc. of ICDE*, 2011.

[16] P. Xiong, Y. Chi, S. Zhu, J. Tatemura, C. Pu, and H. Hacigumus. Activesla: A profit-oriented admission control framework for database-as-a-service providers. In *Proc. of SOCC*, 2011.

[17] F. Yang, J. Shanmugasundaram, and R. Yerneni. A scalable data platform for a large number of small applications. In *Proc. of CIDR*, 2009.