A Learning-based Service for Cost and Performance Management of Cloud Databases

Ryan Marcus
Brandeis University
ryan@cs.brandeis.edu

Sofiya Semenova
Brandeis University
sofiya@brandeis.edu

Olga Papaemmanouil
Brandeis University
olga@cs.brandeis.edu

***Abstract*—Data management applications deployed on IaaS cloud environments must simultaneously strive to minimize cost and provide good performance. Balancing these two goals requires complex decision-making across a number of axes: resource provisioning, query placement, and query scheduling. While previous works have addressed each axis in isolation for specific types of performance goals, this demonstration showcases WiSeDB, a cloud workload management advisor service that uses machine learning techniques to address all dimensions of the problem for *customizable* performance goals. In our demonstration, attendees will see WiSeDB in action for a variety of workloads and performance goals.**

Video: http://youtu.be/YAKRxSoUs18

I. Introduction

Despite the fast growth of cloud computing, taking full advantage of IaaS (Infrastructure-as-a-Service) clouds remains a complex task for data management applications. Today, application developers are expected to make a myriad of decisions related to virtual machine (VM) provisioning and workload scheduling in order to meet their performance goals while minimizing resource usage fees paid to cloud providers. Existing research has tackled this challenge by dividing the problem into smaller pieces, such as *resource provisioning* [9], [11], *query placement* [5], and *query scheduling* [3]. Further, while a broad range of performance criteria are covered by these systems (e.g., response time [2], average workload latency [5], percentile [4]), each offers a solution tuned for *specific performance metrics*. While studying each dimension of the problem independently for a limited set of performance metrics reduces the complexity of the problem, it fails to deliver effective deployments with respect to cost *and* performance, as these sub-problems are strongly interconnected. Unifying these solutions into a working framework is also not straightforward, as each technique makes different assumptions about workload characteristics and performance goals.

This demonstration will showcase WiSeDB [6], [7], a cost and performance management service for IaaS-deployed data management applications that overcomes the above challenges. WiSeDB leverages machine learning techniques to acquire insight into the complex interplay of resource provisioning and workload scheduling decisions and their impact on the application's monetary cost and performance. Using this insight, it delivers customized models (*strategies*) for tackling these tasks. WiSeDB's learning framework delivers a number of unique capabilities. First, it is *metric-independent*, i.e., it allows applications to define custom application-level performance goals. Second, it provides *end-to-end strategies* for processing incoming workloads, i.e., it indicates: (a) the cloud resources to be provisioned (e.g., number/type of VMs), (b) the distribution of resources among the queries (e.g., which VM will execute a given query) and (c) the execution order of these queries within each VM. Finally, WiSeDB is *cost-aware*, i.e., its models are trained to minimize the monetary cost of deploying data management applications on IaaS clouds.

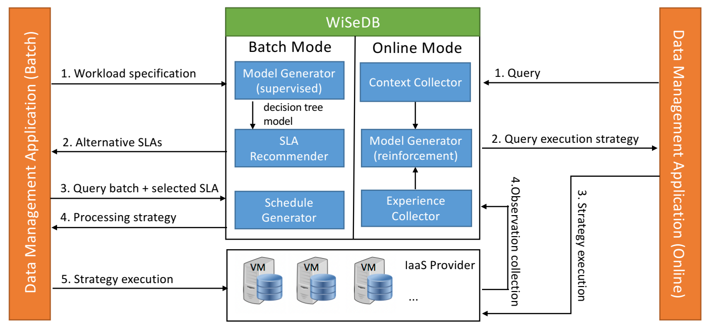


Fig. 1: Architecture

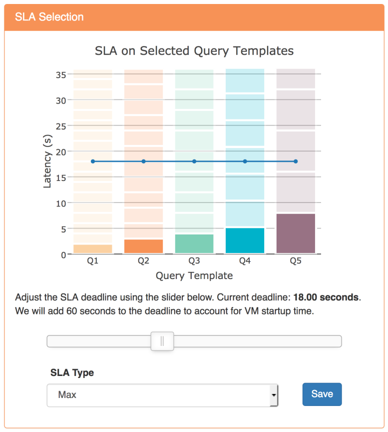
II. System Overview

WiSeDB employs a learning approach to tackle the workload scheduling and resource provisioning challenges of IaaS-deployed data-centric applications[1]. Its architecture is shown in Figure 1. Applications interact with WiSeDB by customizing a Service Level Agreement (SLA) to express their desired performance goal (e.g., percentile, total execution time, deadline per query, etc). WiSeDB recommends custom SLA-aware strategies for executing incoming queries. These strategies strive to minimize the monetary cost of using the IaaS resources as well as any penalty fees caused by SLA violations.

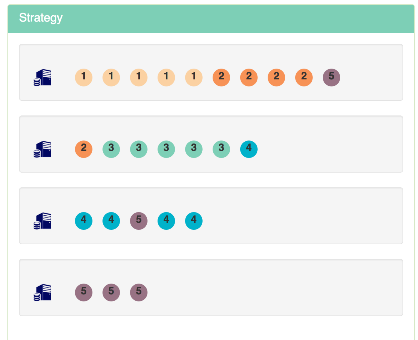
WiSeDB can generate low-cost strategies for executing both batch query workloads (*batch mode*) and queries arriving one at a time (*online mode*). In batch mode, we assume known query types (and query execution times), while in online mode we relax these assumptions and we can handle previously-unseen queries. Hence, the online mode is designed for dynamic environments.

Batch mode relies on *supervised learning* (decision tree classifiers) to generate application-specific strategies. Given a strategy customized to a given SLA and workload specifications (query types and execution times), WiSeDB also leverages the model's training data to generate with very low overhead numerous alternative SLAs, allowing users to

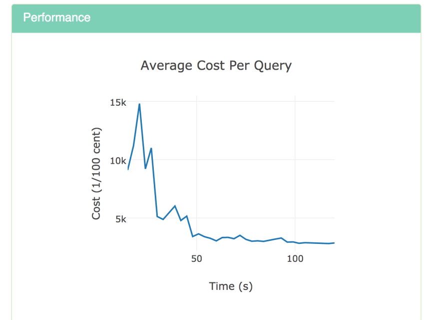
[1] We assume OLAP (read-only) workloads executed on fully replicated DBs.



(a) Selecting an SLA



(b) Example execution strategy



(c) Cost over time of online mode

Fig. 2: WiSeDB demo screenshots

explore cost vs. performance trade-offs common in cloud environments [8]. For each incoming query batch, the user can pick any of the recommended SLAs and WiSeDB provides a custom recommendation including (a) number and type of VMs to reserve, (b) dispatching and execution order of queries to these VMs. More details can be found in [7].

Online mode relies on *reinforcement learning* (contextual mult-armed bandits) to identify low-cost actions for each incoming query. Given the query's context and SLA, it selects an action according to the probability of that action being the "best" (minimizing the overall monetary cost). Example actions include executing a query on an already-provisioned VM or reserving a new VM. The system continuously learns from its past actions by incorporating new *observations* into its model, thus improving its recommendations over time. Observations include information on the cost of the action as well as the action's context (e.g., resource availability features on VMs, query features). More details can be found in [6].

III. Demonstration

Our demonstration will allow users to interact with WiSeDB's two operating modes. In batch mode, users will be able to select their (assumed known) query templates (TPC-H templates) and observe the expected latency per query. This will help them set their performance goals in the next step (i.e., picking a performance metric from the drop-down list and adjusting its threshold through the slider shown in Figure 2a). WiSeDB recommends a set of alternative SLAs for the user to pick. The user can view the strategy for its selected SLA (Figure 2b) and execute it on the cloud for randomly generated query batches. The interface will compare the monetary cost of the strategy learned by WiSeDB and compare it to common heuristics. Additionally, users can inspect and interact with the decision tree model generated.

In online mode, the user selects the desired performance metric for the SLA (similarly to the batch mode). Our demo will then generate random query sequences to execute on the cloud. Users may choose TPC-H templates to include in the sequence and adjust their arrival rates. As queries arrive, users will be able to see live the scheduling decisions of WiSeDB as well as a plot of WiSeDB's average cost per query (Figure 2c). The interface allows users to study the context collected from each decision as well as the model on which each decision is based. As WiSeDB incorporates more observations from past decisions, users will observe the average cost converging.

In both modes, users are able to dynamically change the queries composing each workload, select and modify SLAs on-the-fly, and observe the actual monetary impact of various scheduling decisions in real-time. Further, users will be able to inspect the models generated by WiSeDB to understand the strategies discovered by the system and effectiveness of our learning-based approach.

Our demonstration will be done using two laptops running the front-end interface as well as the WiSeDB service. Our data management application will be a TPC-H [10] 10GB database and our query workloads will consist of TPC-H templates. All cloud deployments will be realized on the Linode cloud [1].

IV. Conclusion

The WiSeDB demonstration will allow conference attendees to interact with our learning-based service. Users will be able to see the monetary and performance impact of provisioning and query scheduling decisions in real-time. Further, they will be able to inspect the machine learning models generated by WiSeDB and understand the resource and workload management strategies discovered by the system.

References

[1] Linode, https://linode.com.

[2] Y. Azar et al. Cloud scheduling with setup cost. In *SPAA '13*.

[3] Y. Chi et al. iCBS: Incremental cost-based scheduling under piecewise linear SLAs. *PVLDB '11*.

[4] W. Lang et al. Towards multi-tenant performance SLOs. In *ICDE '14*.

[5] Z. Liu et al. PMAX: Tenant placement in multitenant databases for profit maximization. In *EDBT '13*.

[6] R. Marcus et al. Releasing cloud databases from the chains of performance prediction models. In *CIDR '17*.

[7] R. Marcus et al. WiSeDB: A learning-based workload management advisor for cloud databases. *PVLDB '16*.

[8] J. Ortiz et al. Changing the face of database cloud services with personalized service level agreements. In *CIDR '15*.

[9] J. Ortiz et al. PerfEnforce demonstration: Data analytics with performance guarantees. In *SIGMOD '16*.

[10] M. Poess et al. New TPC benchmarks for decision support and web commerce. *SIGMOD '00*.

[11] J. Rogers et al. A generic auto-provisioning framework for cloud databases. In *ICDEW '10*.