Make Wise Decisions for Your DBMSs: Workload Forecasting and Performance Prediction Before Execution

Zhengtong Yan[1], Jiaheng Lu[1](✉), Qingsong Guo[1], Gongsheng Yuan[1], Calvin Sun[2], and Steven Yang[2]

[1] Department of Computer Science, University of Helsinki, Helsinki, Finland
{zhengtong.yan, jiaheng.lu, qingsong.guo, gongsheng.yuan}@helsinki.fi
[2] Huawei Toronto, Toronto, Canada
{calvin.sun3, steven.yuan1}@huawei.com

1 Background and Motivation

The performance of a Database Management System (DBMS) is decided by the system configurations and the workloads it needs to process. To achieve *instance optimality* [13], database administrators and end-users need to choose the optimal configurations and allocate the most appropriate resources in accordance with the workloads for each database instance. However, the high complexity of time-varying workloads makes it extremely challenging to find the optimal configuration, especially for a cloud DBMS that may have millions of database instances with diverse workloads. There is no *one-size-fits-all* configuration that works for all workloads since each workload has varying patterns on configuration and resource requirements. If a configuration cannot adapt to the dynamic changes of workloads, there could be a significant degradation in the overall performance of a DBMS unless a sophisticated administrator is continuously re-configuring the DBMS.

An ideal solution to address the above challenges is the *autonomous* or *self-driving* DBMSs (e.g., `Oracle Autonomous Database` [12], `Peloton` [14], `NoisePage` [15], and `openGauss` [7]) which are expected to automatically and constantly configure, tune, and optimize themselves in accordance with the workload changes without any intervention from human experts. Since the optimal configuration setting is very dependent on the workload characteristics, thus the first and key step for an autonomous DBMS is to predict the future workload based on the historical data. Firstly, the DBMS should be able to forecast when the workload will significantly change (i.e., workload shift), how many workloads will arrive (i.e., arrival rate), and what is the next query that a user will execute (i.e., next query) in the future. The predicted workload information enables an autonomous DBMS to decide when and how to re-configure itself in a predictive manner before the workload changes occur. Secondly, an autonomous DBMS also needs to predict the query performance by estimating some essential run-time metrics before execution, such as how long a query will take to complete (i.e,

Table 1. A summary of the major topics in this tutorial.

Workload Characteristics		**Descriptions**	**Information**
Workload Forecasting	Workload Shift	*When* the workloads will change	Time
	Arrival Rate	*How many* workloads will arrive	Volume
	Next Query	*What* will be the next query or transaction	Change
Performance Prediction	Execution Time	*How long* the workloads will take to run	Duration
	Resource Usage	*How much resources* will be consumed	Resource

execution time) and how much resources will be consumed (i.e., resource utilization). Predicting the execution time and resource demand prior to execution is useful in many tasks, including admission control, query scheduling, progress monitoring, system sizing, and resource management [18].

In this tutorial, we will focus on 1) how to forecast the future workloads (e.g., workload shift detection, arrival rate prediction, and next query prediction), and 2) how to analyze the behaviors of the workloads (e.g., execution time prediction and resource usage estimation). We will provide a comprehensive overview and detailed introduction of the two topics, from state-of-the-art methods, real-world applications, to open problems and future directions. Specifically, we will not only discuss traditional methods, such as time-series analysis [3, 16], Markov modeling [4,5], analytical modeling [17,18], and experiment-driven methods [1], but also cover the state-of-the-art AI techniques, including machine learning [8], deep learning [10], reinforcement learning [11], and graph embedding [20]. Table 1 summarizes the major research topics that will be presented in this tutorial.

2 Brief Outline

We plan to deliver a 1.5 hour tutorial, which will be organized as follows:

Part I: Motivation and background (5 min)

Part II: Workload forecasting and performance prediction (60 min)

- Workload shift detection (10 min)
- Arrival rate forecasting (10 min)
- Next query forecasting (10 min)
- Execution time and resource usage prediction (30 min)

Part III: Case study of real-world applications (20 min)

Part IV: Open challenges and future directions (5 min)

3 Differences with Our Previous Tutorial

This tutorial was presented as a part in IEEE ICDE 2020 [19]. The previous tutorial had attracted significant interests from industry and academia. Compared with the previous one, this tutorial mainly focuses on workload forecasting and performance prediction with more valuable and deeper insights. In addition, we add some recent works about query encoders that are used to transform queries and query plans into feature vectors [6,9,13]. We also add two real-world applications about how to utilize the predicted query arrival rate for automatic index selection [8] and how to achieve high throughput of query scheduling based on query execution time prediction [2].

4 Target Audiences

This tutorial is intended for a broad scope of audience ranging from database systems researchers to industry practitioners, with a focus on workload forecasting and performance prediction. Those who are interested in self-driving and autonomous databases could also gain some useful knowledge from this tutorial. Basic knowledge in database workload and query optimization is sufficient to follow this tutorial. Some background in machine learning, reinforcement learning, and graph embedding techniques would be helpful.

5 Short Bibliographies

Zhengtong Yan is a doctoral student at the University of Helsinki. His research topics lie in autonomous multi-model databases with reinforcement learning.

Jiaheng Lu is a professor at the University of Helsinki. His main research interests lie in the Big Data management and database systems. He has published more than one hundred journal and conference papers.

Qingsong Guo is a postdoctoral researcher at the University of Helsinki. His research interests include multi-model databases and automatic management of big data with deep learning.

Gongsheng Yuan is a doctoral student at the University of Helsinki. His research topics lie in databases with quantum theory or reinforcement learning.

Calvin Sun is the Chief Database Architect at Huawei Cloud. He has 20+ years experience in developing several database systems, ranging from embedded database, large-scale distributed database, to cloud-native database.

Steven Yuan is the Director of Huawei Toronto Distributed Scheduling and Data Engine Lab. He leads an research team in big data and cloud domain, focusing on distributed scheduling and distributed database, from IaaS to PaaS.

6 Acknowledgment

The work is supported by Canada Huawei Research Funding and the Academy of Finland project (No. 310321).

References

1. Ahmad, M., Duan, S., Aboulnaga, A., Babu, S.: Predicting Completion Times of Batch Query Workloads Using Interaction-aware Models and Simulation. In: EDBT. pp. 449–460 (2011)

2. Amazon Redshift Workload Management (WLM): https://docs.aws.amazon.com/redshift/latest/dg/cm-c-implementing-workload-management.html
3. Higginson, A.S., Dediu, M., Arsene, O., Paton, N.W., Embury, S.M.: Database Workload Capacity Planning using Time Series Analysis and Machine Learning. In: SIGMOD. pp. 769–783 (2020)
4. Holze, M., Ritter, N.: Towards Workload Shift Detection and Prediction for Autonomic Databases. In: PIKM. pp. 109–116 (2007)
5. Holze, M., Ritter, N.: Autonomic Databases: Detection of Workload Shifts with n-Gram-Models. In: ADBIS. pp. 127–142. Springer (2008)
6. Jain, S., Howe, B., Yan, J., Cruanes, T.: Query2Vec: An Evaluation of NLP Techniques for Generalized Workload Analytics. arXiv:1801.05613 (2018)
7. Li, G., Zhou, X., Sun, J., Yu, X., Han, Y., Jin, L., Li, W., Wang, T., Li, S.: openGauss: An Autonomous Database System. Proceedings of the VLDB Endowment **14**(12), 3028–3042 (2021)
8. Ma, L., Aken, D.V., Hefny, A., Mezerhane, G., Pavlo, A., Gordon, G.J.: Query-based Workload Forecasting for Self-driving Database Management Systems. In: SIGMOD. pp. 631–645. ACM (2018)
9. Marcus, R., Papaemmanouil, O.: Flexible Operator Embeddings via Deep Learning. arXiv preprint arXiv:1901.09090 (2019)
10. Marcus, R., Papaemmanouil, O.: Plan-Structured Deep Neural Network Models for Query Performance Prediction. PVLDB **12**(11), 1733–1746 (2019)
11. Meduri, V.V., Chowdhury, K., Sarwat, M.: Evaluation of Machine Learning Algorithms in Predicting the Next SQL Query from the Future. ACM Transactions on Database Systems (TODS) **46**(1), 1–46 (2021)
12. Oracle Autonomous Database: https://www.oracle.com/autonomous-database/
13. Paul, D., Cao, J., Li, F., Srikumar, V.: Database Workload Characterization with Query Plan Encoders. arXiv preprint arXiv:2105.12287 (2021)
14. Pavlo, A., Angulo, G., Arulraj, J., Lin, H., Lin, J., Ma, L., Menon, P., Mowry, T., Perron, M., Quah, I., Santurkar, S., Tomasic, A., Toor, S., Aken, D.V., Wang, Z., Wu, Y., Xian, R., Zhang, T.: Self-Driving Database Management Systems. In: CIDR (2017)
15. Pavlo, A., Butrovich, M., Ma, L., Menon, P., Lim, W.S., Van Aken, D., Zhang, W.: Make Your Database System Dream of Electric Sheep: Towards Self-Driving Operation. PVLDB **14**(12), 3211–3221 (2021)
16. Taft, R., El-Sayed, N., Serafini, M., Lu, Y., Aboulnaga, A., Stonebraker, M., Mayerhofer, R., Andrade, F.: P-store: An Elastic Database System With Predictive Provisioning. In: SIGMOD. pp. 205–219. ACM (2018)
17. Wu, W., Chi, Y., Hacígümüş, H., Naughton, J.F.: Towards Predicting Query Execution Time for Concurrent and Dynamic Database Workloads. PVLDB **6**(10), 925–936 (2013)
18. Wu, W., Chi, Y., Zhu, S., Tatemura, J., Hacigümüs, H., Naughton, J.F.: Predicting Query Execution Time: Are Optimizer Cost Models Really Unusable? In: ICDE. pp. 1081–1092. IEEE (2013)
19. Yan, Z., Lu, J., Chainani, N., Lin, C.: Workload-Aware Performance Tuning for Autonomous DBMSs. In: ICDE. pp. 2365–2368. IEEE (2021)
20. Zhou, X., Sun, J., Li, G., Feng, J.: Query Performance Prediction for Concurrent Queries Using Graph Embedding. PVLDB **13**(9), 1416–1428 (2020)