VLDB 2019 Tutorial

Speedup your Analytics: Automatic Parameter Tuning for Databases and Big Data Systems

Jiaheng Lu, University of Helsinki
Yuxing Chen, University of Helsinki
Herodotos Herodotou, Cyprus University of Technology
Shivnath Babu, Duke University / Unravel Data Systems

Outline

Motivation and Background

History and Classification

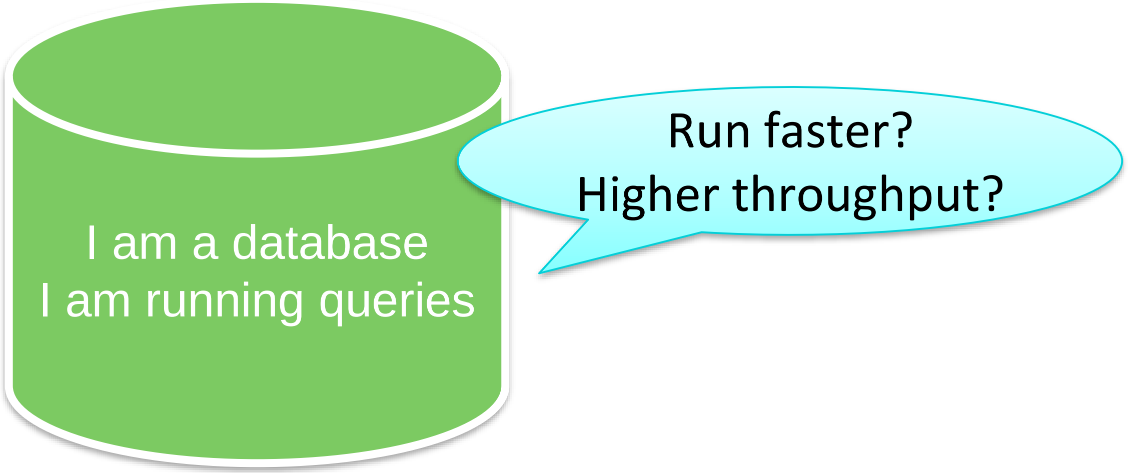
Parameter Tuning on Databases

Parameter Tuning on Big Data Systems

Applications of Automatic Parameter Tuning

Open Challenges and Discussion

What and How to Tune?



- What to configure?
  - Which parameters (knobs)?
  - Which are most important?

- How to tune (to best throughput)?
  - Increase buffer size?
  - More parallelism on writing?

Figure. Tuning guitar knobs to right notes (frequencies)

What to Tune – Some Important Knobs for throughput

Threads

Timeout Settings

Memory Cache

Parameter Name	**Brief Description and Use**	**Deafult**
bgwriter_delay	Background writer's delay between activity rounds	200ms
bgwriter_lru_maxpages	Max number of buffers written by the background writer	100
checkpoint_segments	Max number of log file segments between WAL checkpoints	3
checkpoint_timeout	Max time between automatic WAL checkpoints	5min
deadlock_timeout	Waiting time on locks for checking for deadlocks	1s
default_statistics_target	Default statistics target for table columns	100
effective_cache_size	Effective size of the disk cache accessible to one query	4GB
shared_buffers	Memory size for shared memory buffers	128MB

What are the Important Parameters and How to Choose

- Affect the performance most (manually)
  - Based on expert experiences
  - Default documentation

Parameters have strong correlation to performance are important!

Performance-sensitive parameters are important!

If you want higher throughput, better tuning memory-related parameters

What are the Important Parameters and How to Choose

- Affect the performance most
- **Strongest correlation** between parameters and objective function (model)
  - Linear regression model for independent parameters:
    - Regularized version of least squares – Lasso (*OtterTune 2017)*
      - Interpretable, stable, and computationally efficient with higher dimensions
  - Deep learning model (*CBDTune 2019)*
    - The important input parameters will gain higher **weights** in training

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_N X_N$$

Weights Knobs

How to Tune – Key Tuning Goals

- **Avoidance:** to identify and avoid error-prone configuration settings
- **Ranking:** to rank parameters according to the performance impact
- **Profiling:** to classify and store useful log information from previous runs
- **Prediction:** to predict the database or workload performance under hypothetical resource or parameter changes
- **Tuning:** to recommend parameter values to achieve objective goals

How to Tune – Tuning Methods

Methods	**Approach**	**Methodology**	**Target Level**
Rule-based	SPEX (2013)	Constraint inference	Avoidance
	Xu (2015)	Configuration navigation	Ranking
Cost-model	STMM (2006)	Cost model	Tuning
Simulation-based	Dushyanth (2005)	Trace-based simulation	Prediction
	ADDM (2005)	DAG model & simulation	Profiling, tuning
Experiment driven	SARD (2008)	P&B statistical design	Ranking
	iTuned (2009)	LHS & Guassian Process	Profiling, tuning
Machine Learning	Rodd (2016)	Neural Networks	Tuning
	OtterTune (2017)	Guassian Process	Ranking, tuning
	CDBTune (2019)	Deep RL	Tuning
Adaptive	COLT (2006)	Cost Vs. Gain analysis	Profiling, tuning

Relational Database Tuning Methods

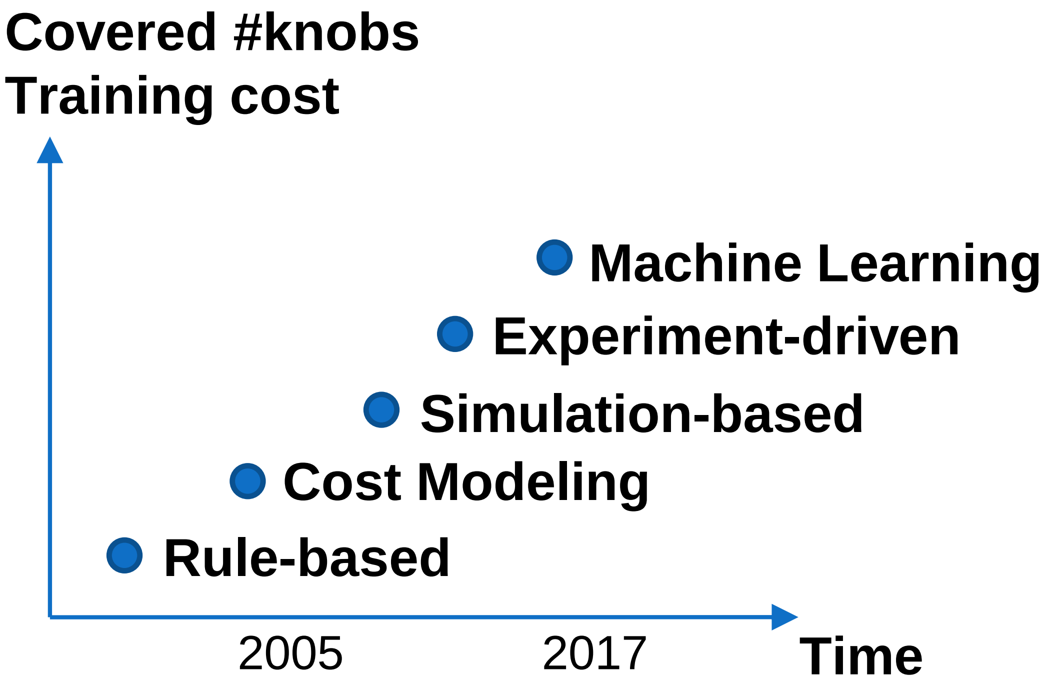


Figure. Developing trend: putting more training cost to uncover more knobs

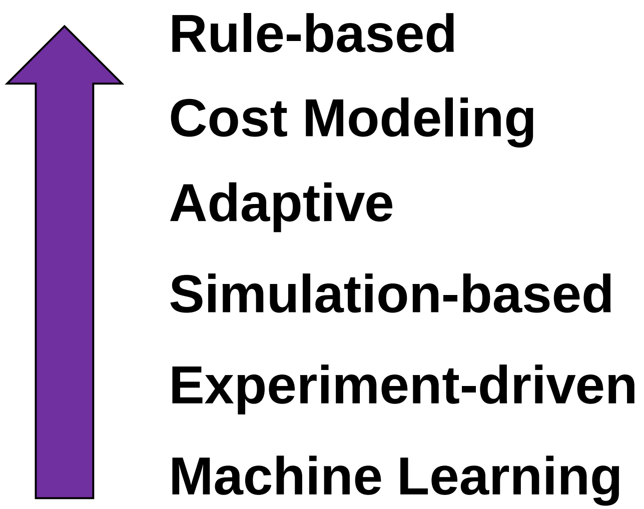


Figure. Required expert knowledge on system

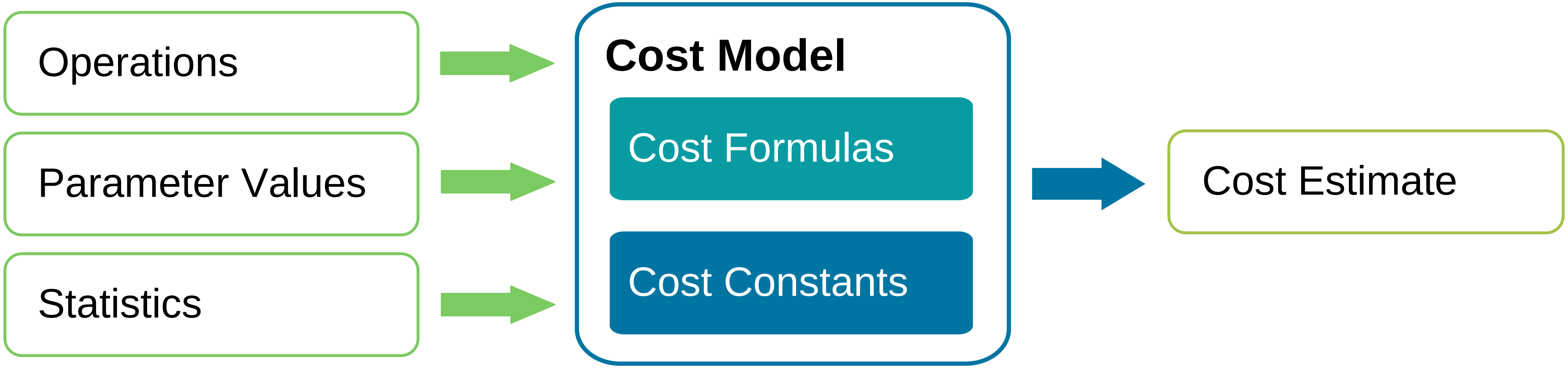
Tuning Method: Rule-based

- Tuning based on rules derived from DBAs' expertise, experience, and knowledge, or Rule of Thumb default recommendation

Rules	Guarantee cache memory to accelerate queries …	Better not change the deadlock timeout if …	Default settings work most of the cases …
Trusted Parties	Expert	Rule of Thumb	Documents

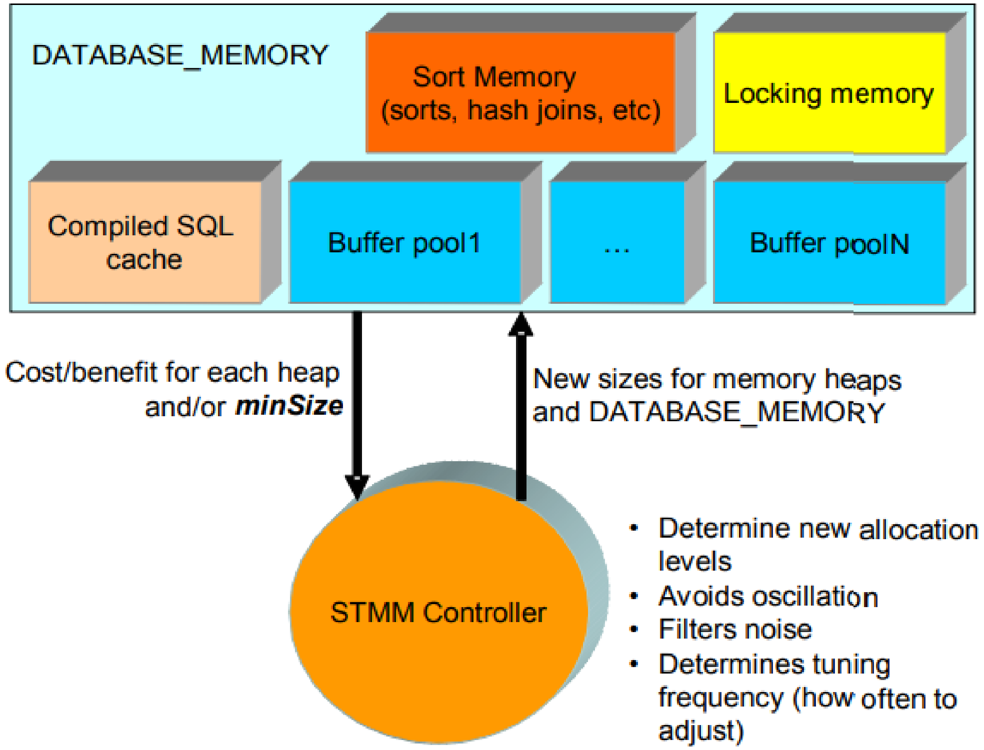
Tuning Method: Cost Modeling

➢ A cost model establishes a performance model by cost functions based on the deep understanding of system components



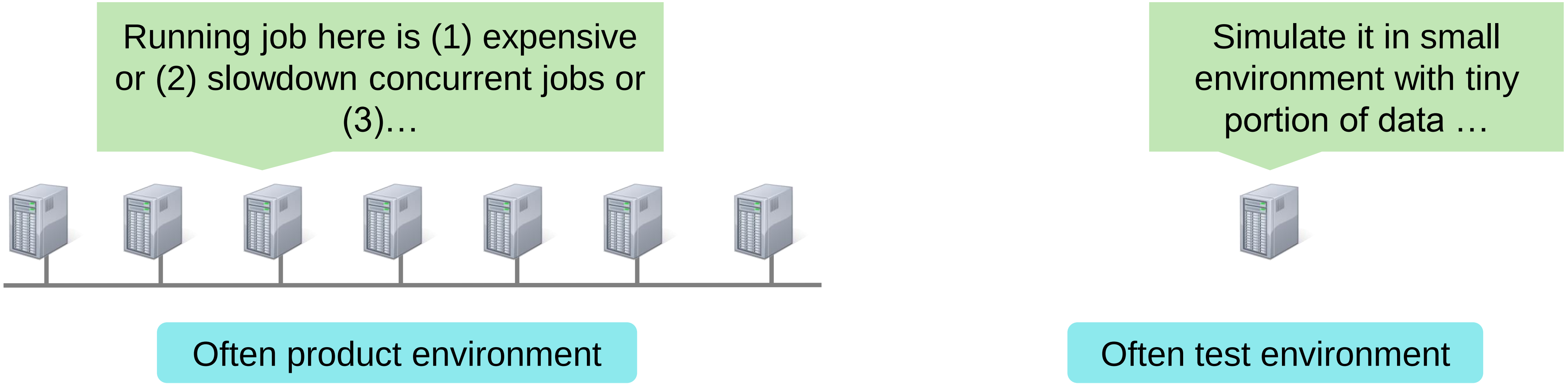
Tuning Method: Cost Modeling (STMM)

- STMM: Adaptive Self-Tuning Memory in DB2 (2006)
  - Reallocates memory for several critical components(e.g., compiled statement cache, sort, and buffer pools)



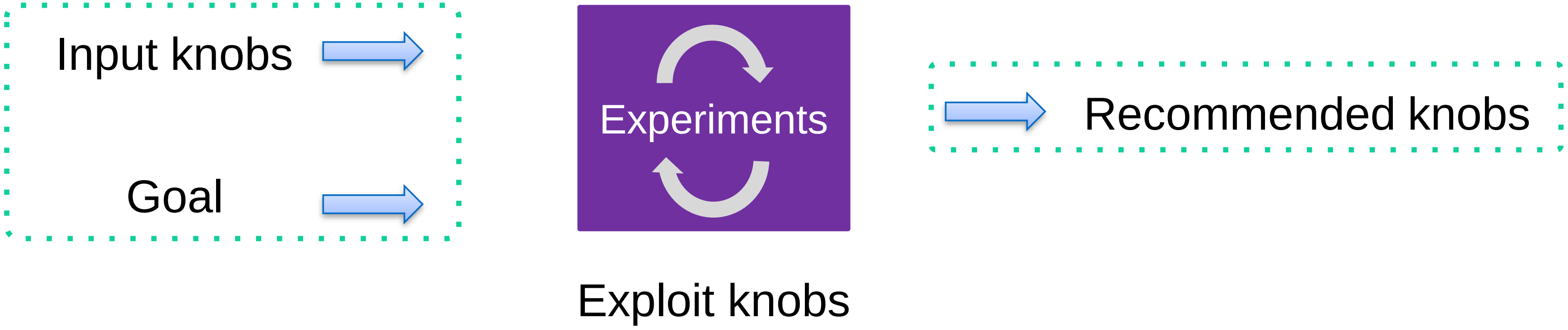
Tuning Method: Simulation-based

- A simulation-based approach simulates workloads in one environment and learns experience or builds models to predict the performance in another.



Tuning Method: Experiment-driven

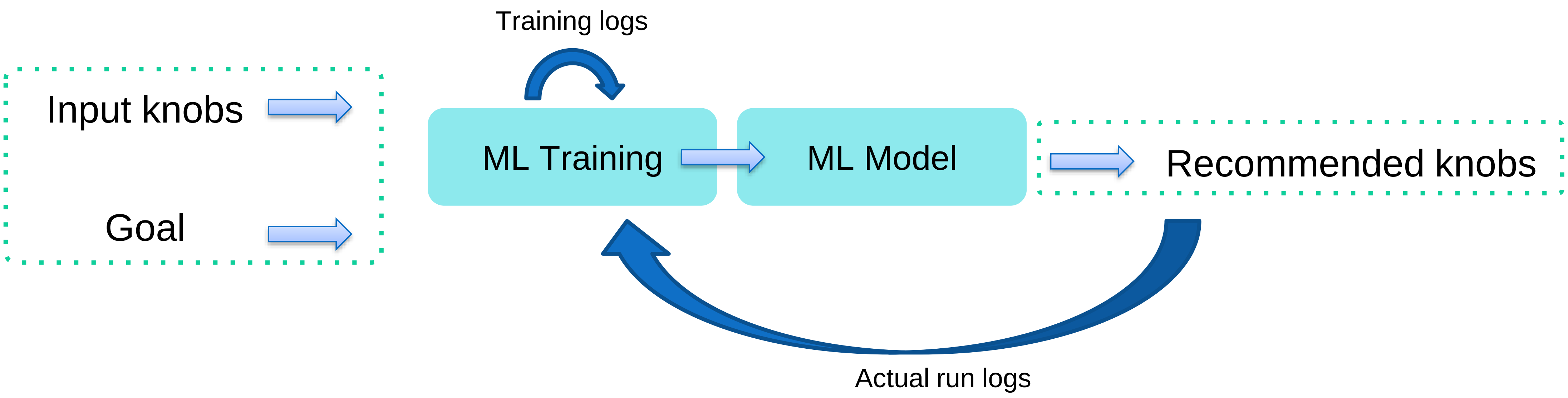
- An experiment-driven approach relies on repeated executions of the same workload under different configuration settings towards tuning parameter values



Classic paper: Tuning Database Configuration Parameters with iTuned. 2009

Tuning Method: Machine Learning

- Machine Learning (ML) approaches aim to tune parameters automatically by taking advantages of ML methods.



Tuning Method: Machine Learning (OtterTune 2017)

- **Factor Analysis:** transform high dimension parameters to few factors
- **Kmeans:** Cluster distinct metrics
- **Lasso:** Rank parameters
- **Gaussian Process:** Predict and tune performance

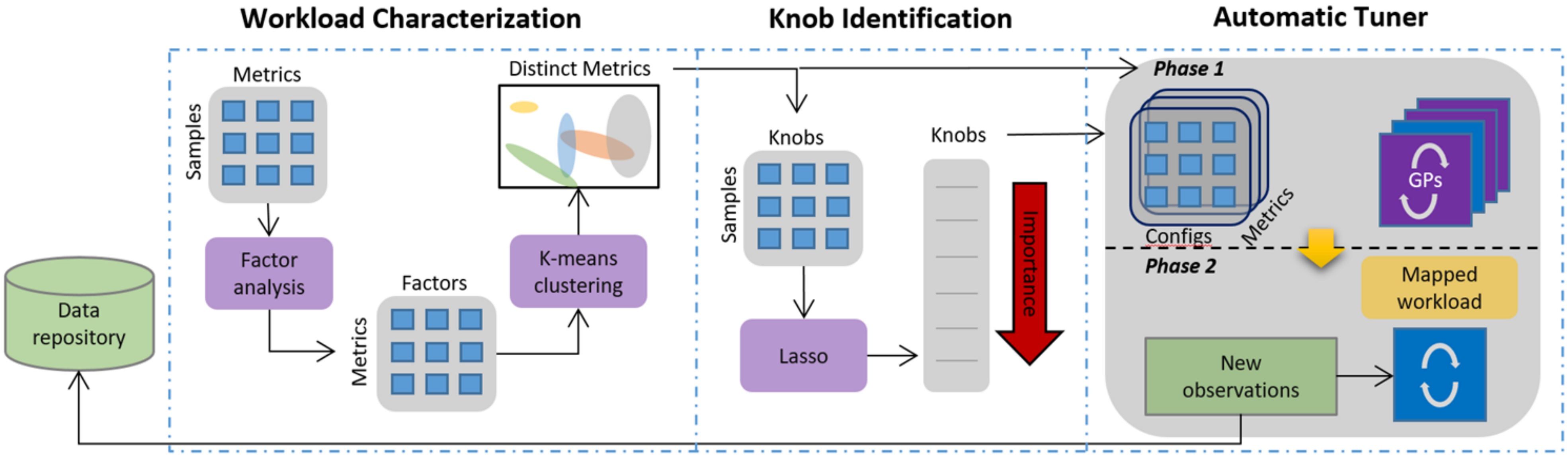


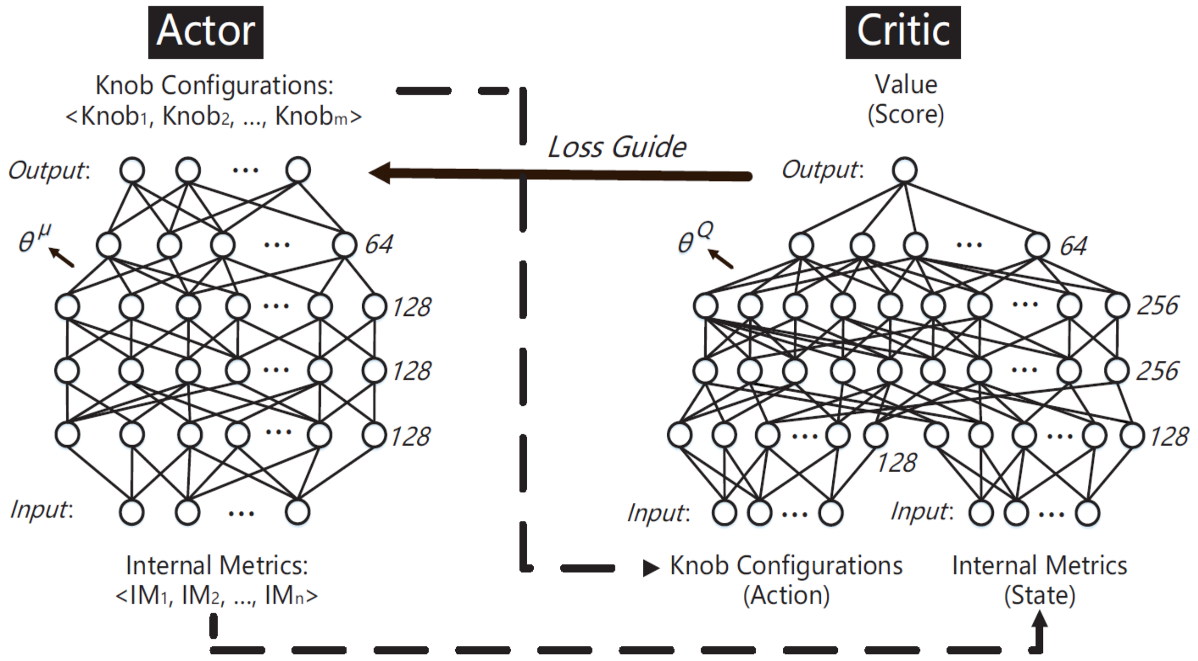
Figure. OtterTune system architecture

Tuning Method: Machine Learning (CDBTune 2019)

- **Reinforcement learning**
  - **State:** knobs and metrics
  - **Reward:** performance change
  - **Action:** recommended knobs
  - **Policy**: Deep Neural network
- **Key idea**
  - Feedback: try-and-error method
    - Recommend -> good/bad
  - Deep deterministic policy gradient
    - Actor critic algorithm

Reward: Throughput and latency performance change Δ from time t − 1 and the initial time to time t

Figure. CDBTune Deep deterministic policy gradient



Tuning Method: Adaptive

- An adaptive approach changes parameter configurations online as the environment or query workload changes

Figure. CLOT (2006) strategy

The Differences of Tuning Database & Big Data Systems in research papers

	Relational Database	Big Data System
Parameters	More parameters on memory	More parameters on vcores
Resource	Often fixed resources	Now more varying resources
Scalability	Often single machine	Often many machines in a distributed environment
Metrics	Throughput, latency	Time, resource cost

References (1/1)

- Narayanan, Dushyanth, Eno Thereska, and Anastassia Ailamaki. "Continuous resource monitoring for self-predicting DBMS." 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. IEEE, 2005.
- Dias, K., Ramacher, M., Shaft, U., Venkataramani, V., & Wood, G.. Automatic Performance Diagnosis and Tuning in Oracle. In CIDR (pp. 84-94), 2005
- Schnaitter, K., Abiteboul, S., Milo, T., & Polyzotis, N. Colt: continuous on-line tuning. In Proceedings of the 2006 ACM SIGMOD international conference on Management of data (pp. 793-795). ACM, 2006. (**CLOT**)
- Storm, Adam J., et al. "Adaptive self-tuning memory in DB2." Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, 2006. (**STMM**)
- Debnath, Biplob K., David J. Lilja, and Mohamed F. Mokbel. "SARD: A statistical approach for ranking database tuning parameters." 2008 IEEE 24th International Conference on Data Engineering Workshop. IEEE, 2008
- Babu, S., Borisov, N., Duan, S., Herodotou, H., & Thummala, V. Automated Experiment-Driven Management of (Database) Systems. In HotOS, 2009.
- Duan, Songyun, Vamsidhar Thummala, and Shivnath Babu. "Tuning database configuration parameters with iTuned." Proceedings of the VLDB Endowment 2.1: 1246-1257, 2009. (**iTuned**)
- Rodd, Sunil F., and Umakanth P. Kulkarni. "Adaptive tuning algorithm for performance tuning of database management system." arXiv preprint arXiv:1005.0972, 2010.
- Van Aken, D., Pavlo, A., Gordon, G. J., & Zhang, B.. Automatic database management system tuning through large-scale machine learning. In Proceedings of the 2017 ACM International Conference on Management of Data(pp. 1009-1024). ACM, 2017. (**OtterTune**)
- Zhang, J., Liu, Y., Zhou, K., Li, G., Xiao, Z., Cheng, B., & Ran, M. An end-to-end automatic cloud database tuning system using deep reinforcement learning. In Proceedings of the 2019 International Conference on Management of Data (pp. 415-432). ACM, 2019. (**CBDTune**)