Automated Configuring of DBMS through Machine Learning

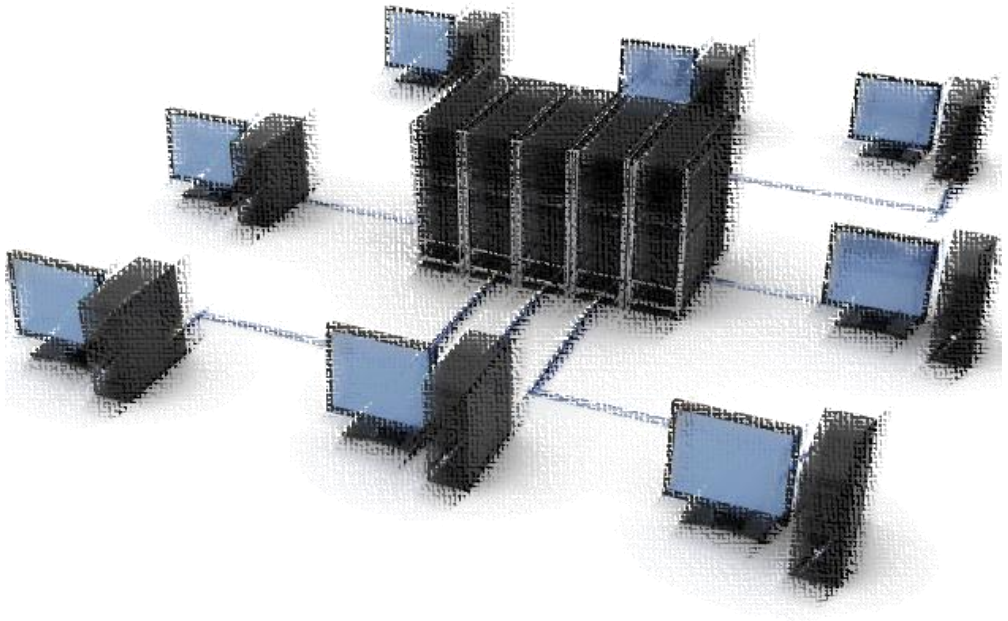
D.L.H.P. PRIYADARSHANA

SEMINAR PRESENTATION | PRINCIPLES OF DATA & KNOWLEDGE BASED SYSTEMS | WISE18

DBMS : WHY SO IMPORTANT?

Provide a way to **store and retrieve** database information '*conveniently*' and '*efficiently*'

File System	DBMS
Less security	More security
Best to store small amounts of data	Suitable to store large amounts of data
No restrictions on duplicate data	Avoid duplicate data
Cannot define relationships between records	Possible define relationships between data using *Primary Key* & *Foreign Key* constraints
Cannot perform criteria based data selection	Easy to search for data that follows a particular criteria



DBMS TUNING : WHY – WHAT – HOW ?

- Data contributes to an organization's competitive advantage:
  - Data : relevant, organized, accurate, easily accessible

    e.g., Access time is critical for a medical system application
- DBMS performance slows dramatically as both data and the business grow
  - Needs to be tuned!

WHY

- Configuration of memory and processing resources of the computer running the DBMS
  - Proper memory allocations for cache usage,
  - Setting the frequency for writing data to storage,
  - Assign network protocols to communicate, etc.

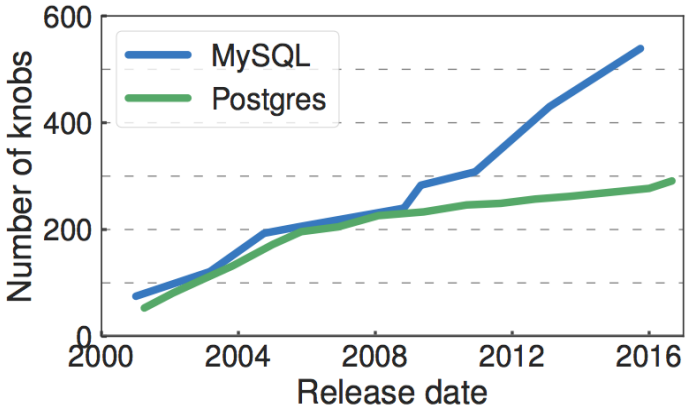
WHAT

- Involves tuning hundreds of ***configuration knobs*** controlling almost all aspects of DBMS — **HOW**

MANUAL VS AUTOMATED TUNING

Standard Approach ➡ Hire an Expert Database Administrator (DBA)

- manually adjust configuration knobs using experience + *'trial-and-error'* approach
- becoming increasingly difficult:
  - high number of configurable knobs
  - continuous knob settings
  - inter-dependent knobs
  - each deployment is configured from scratch!



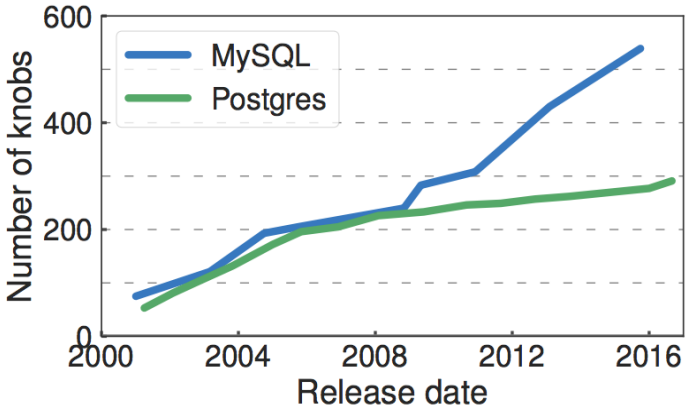
~250 new configuration options added to MySQL between 2012 and 2016 [1]

[1] Van Aken, Dana, et al. "Automatic Database Management System Tuning Through Large-scale Machine Learning." *International Conference on Management of Data*. ACM, 2017.

MANUAL VS AUTOMATED TUNING

Standard Approach ➡ Hire an Expert Database Administrator (DBA)

- manually adjust configuration knobs using experience + *'trial-and-error'* approach
- becoming increasingly difficult:
  - high number of configurable knobs
  - continuous knob settings
  - inter-dependent knobs
  - each deployment is configured from scratch!



~250 new configuration options added to MySQL between 2012 and 2016 [1]

➡ **Auto-tuners** – Tools which can automatically configure DBMS knobs

- Expected to automatically find knob configurations for best DBMS performance (high throughput, low latency, low cost, etc.)
- Weaknesses:
  - DBMS specific (PGTune for Postgres, Oracle SQL analyzer tool)
  - Require DBA's input (DB2 Performance Wizard tool by IBM)
  - DOES NOT REUSE PAST DATA!!!

[1] Van Aken, Dana, et al. "Automatic Database Management System Tuning Through Large-scale Machine Learning." *International Conference on Management of Data*. ACM, 2017.

MACHINE LEARNING FOR AUTO-TUNING

OtterTune ➡ An *'auto-tuner'* capable of reusing past tuning-data.

- Developed by a set of researchers in the 'Carnegie Mellon Database Group'

> *"Make it easy for anyone to deploy a DBMS,*
> *even with zero expertise in database administration!"*

What's so special 'bout it?

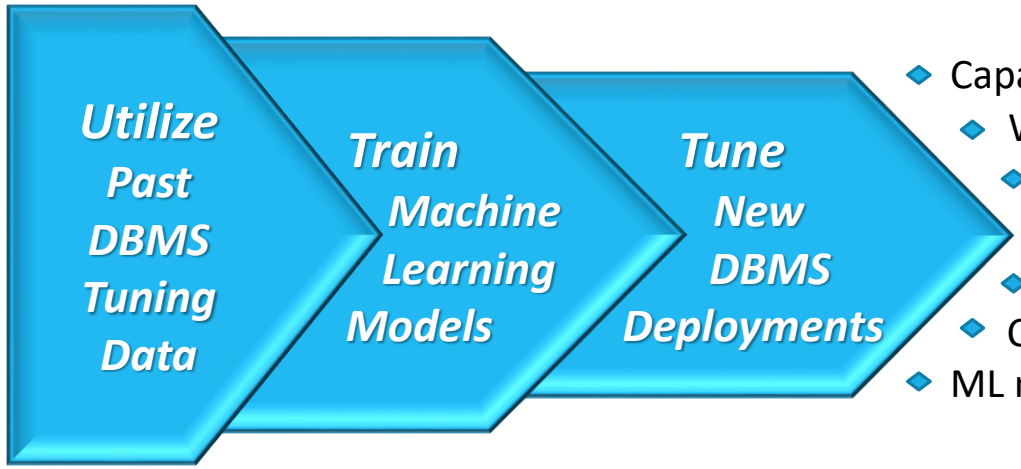
MACHINE LEARNING FOR AUTO-TUNING

OtterTune ➡ An *'auto-tuner'* capable of reusing past tuning-data.

- Developed by a set of researchers in the 'Carnegie Mellon Database Group'

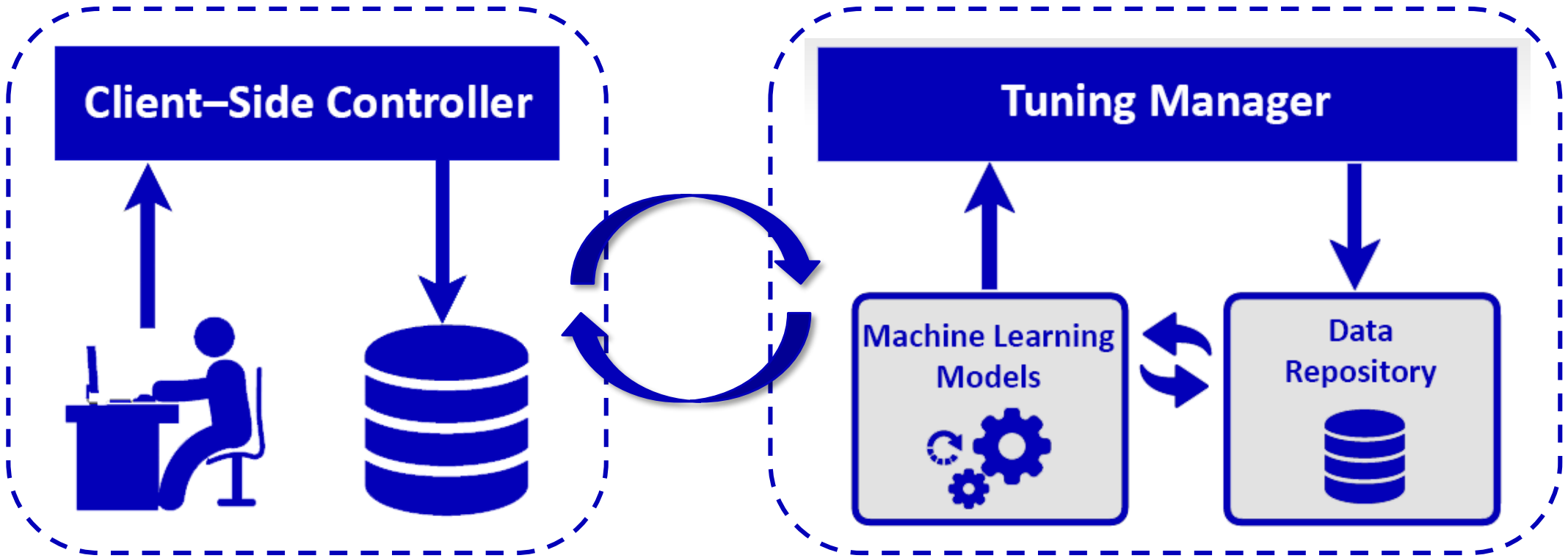
> *"Make it easy for anyone to deploy a DBMS, even with zero expertise in database administration!"*

What's so special 'bout it?

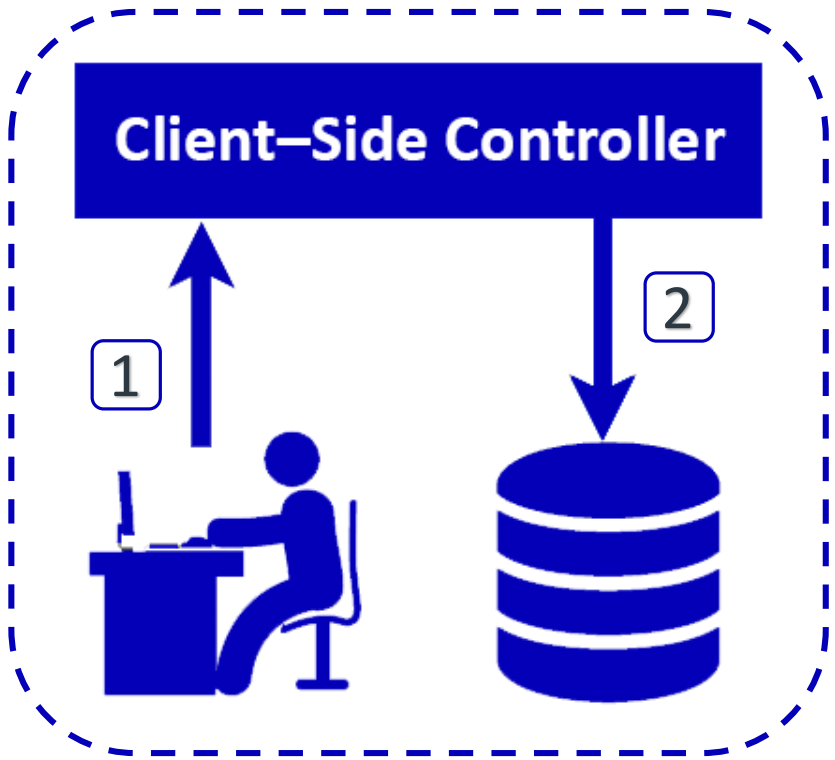


- Capable of reusing knowledge gained from previous tuning attempts
  - Works on any DBMS
    - Minimum input requiring from DBA
      - Use ML models at different stages of the process
    - Can produce best knob config. recommendations within a short time
  - Can incrementally predict better knob configurations using new tuning data
- ML models written in Python and it's open-source (https://github.com/cmu-db/ottertune)

ARCHITECHTURE



1ST COMPONENT : CLIENT–SIDE CONTROLLER



1 ➡ At the beginning of each tuning session,
DBA specify the '*Target Objective*' *(high throughput, low latency, etc.)*

2 ➡ Controller starts the *'observational period'*
on the target DBMS

- During *'Observational Period':*
  - Collect initial knob configurations & DBMS runtime statics
  - Execute workload traces or queries given by DBA
  - Maintain minimum interaction with DBMS
  - Collect same set of initially gathered internal metrics

Target Objective + Internal Metrics

2ND COMPONENT : TUNING MANAGER

Tuning Manager:

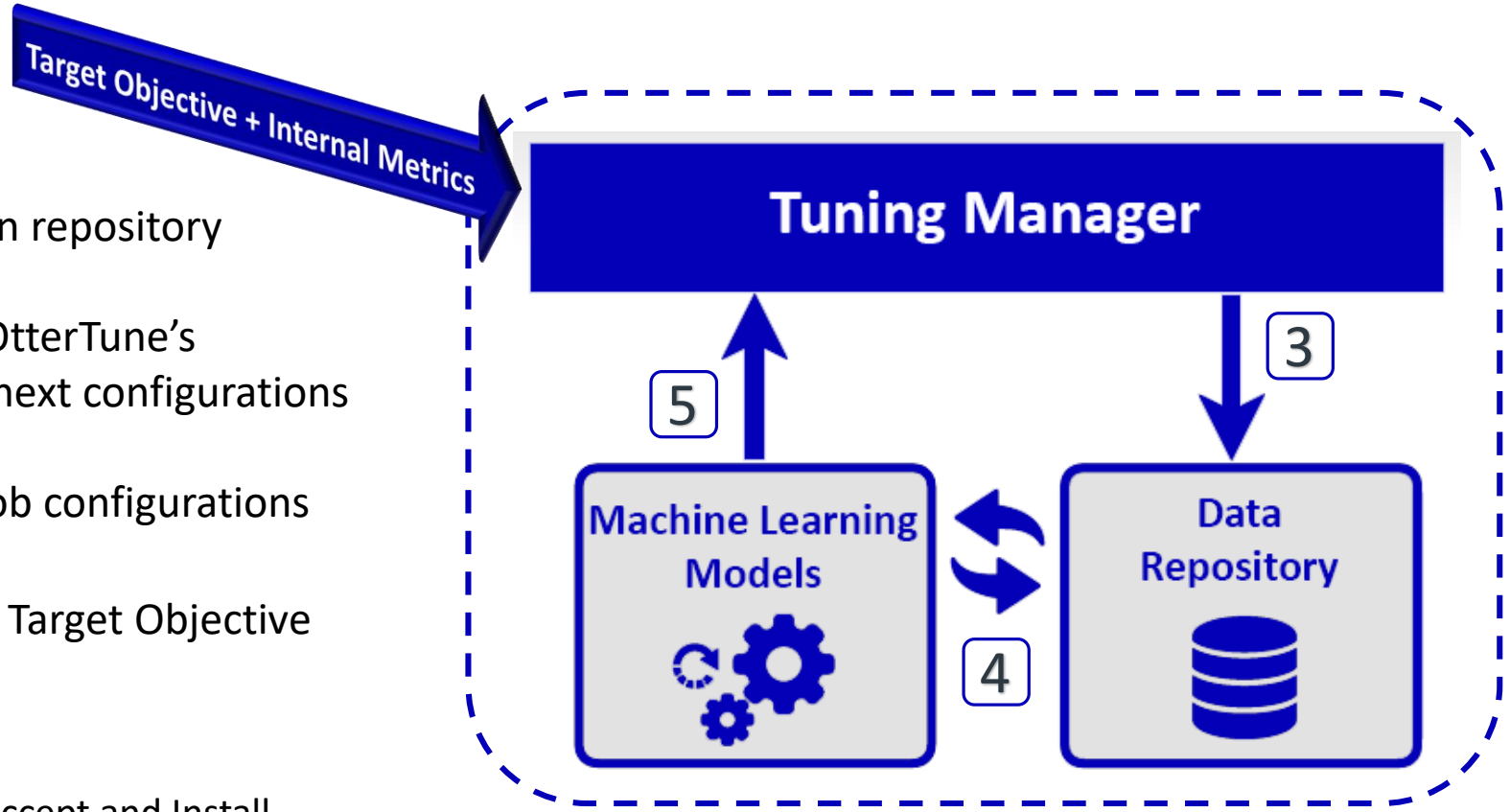
3 ➡ Store received information in repository

4 ➡ Use repository data within OtterTune's
'ML Pipeline' to compute next configurations

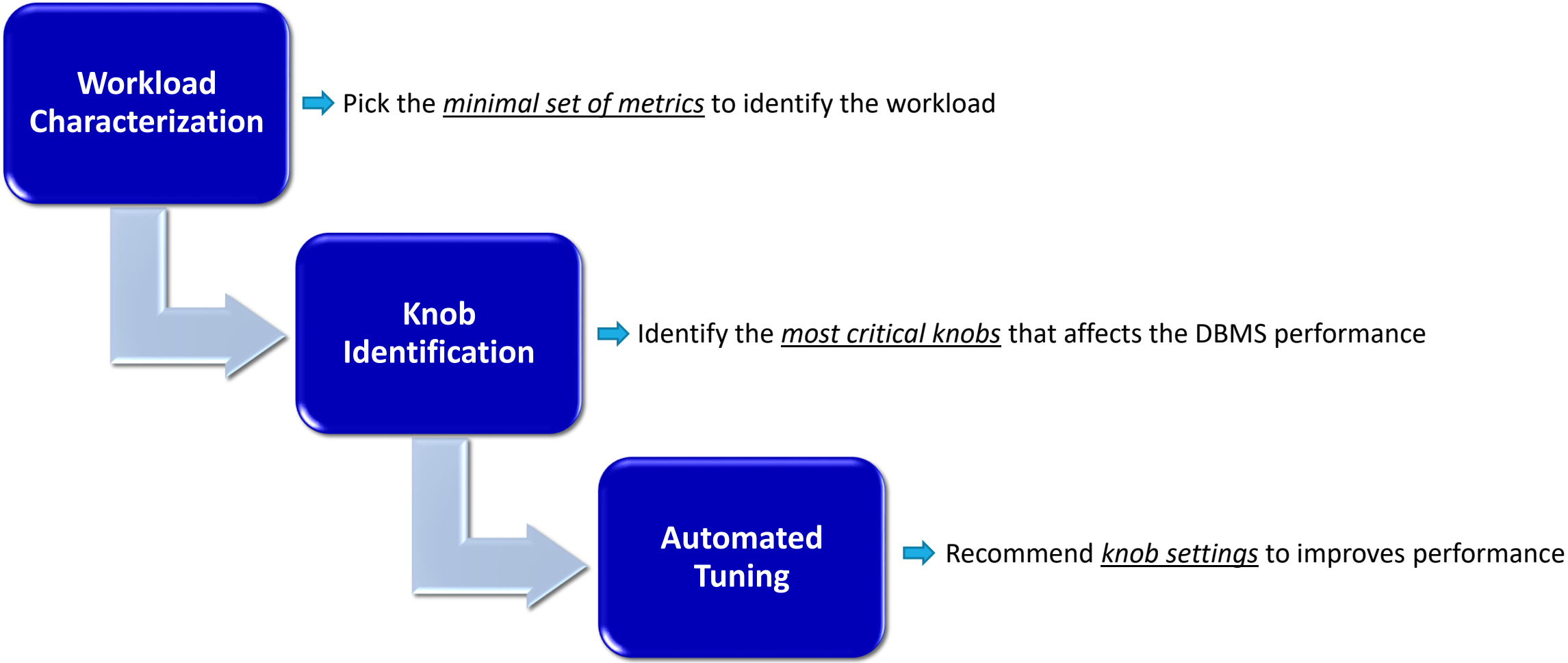
5 ➡ Return recommended knob configurations
+
Expected improvement in Target Objective

Controller

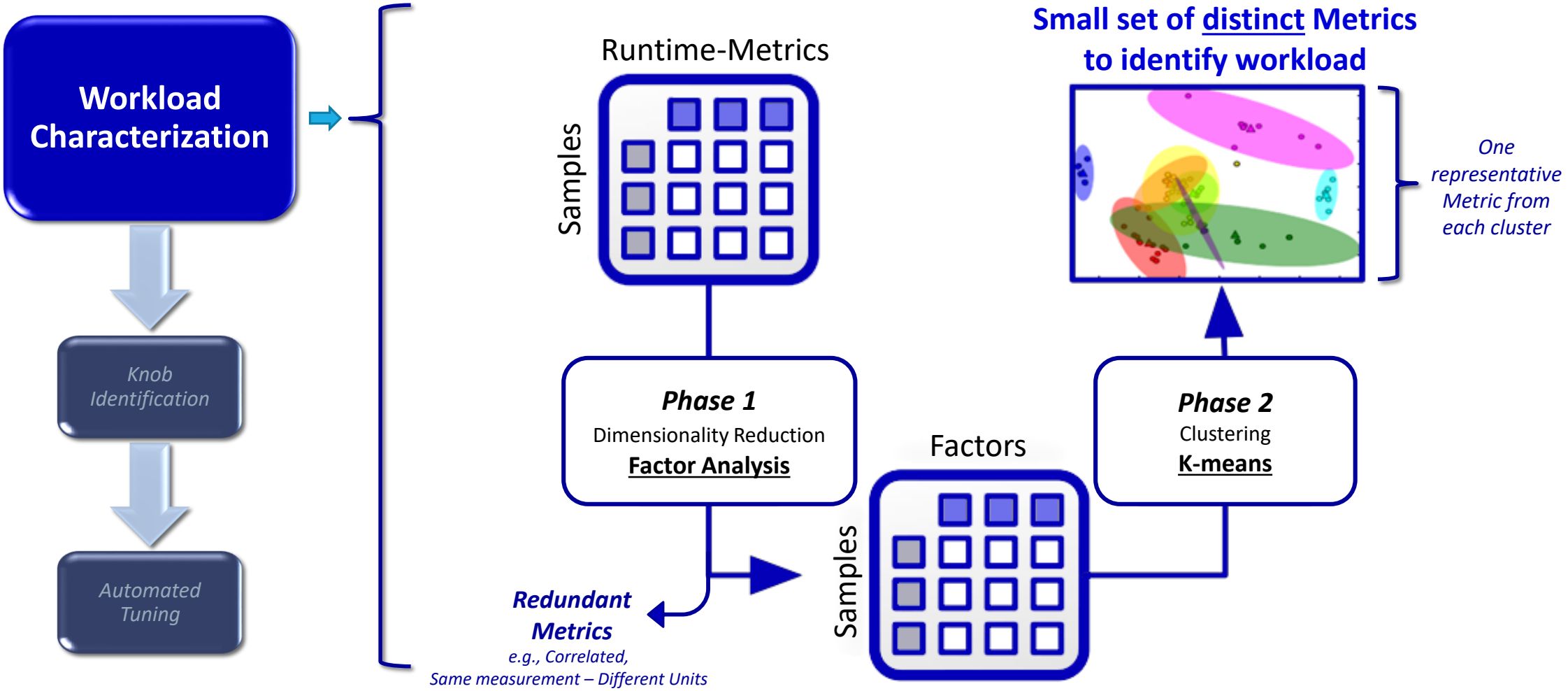
6 ➡ DBA review Recommendations
- Accept and Install
- Reiterate for better recommendations



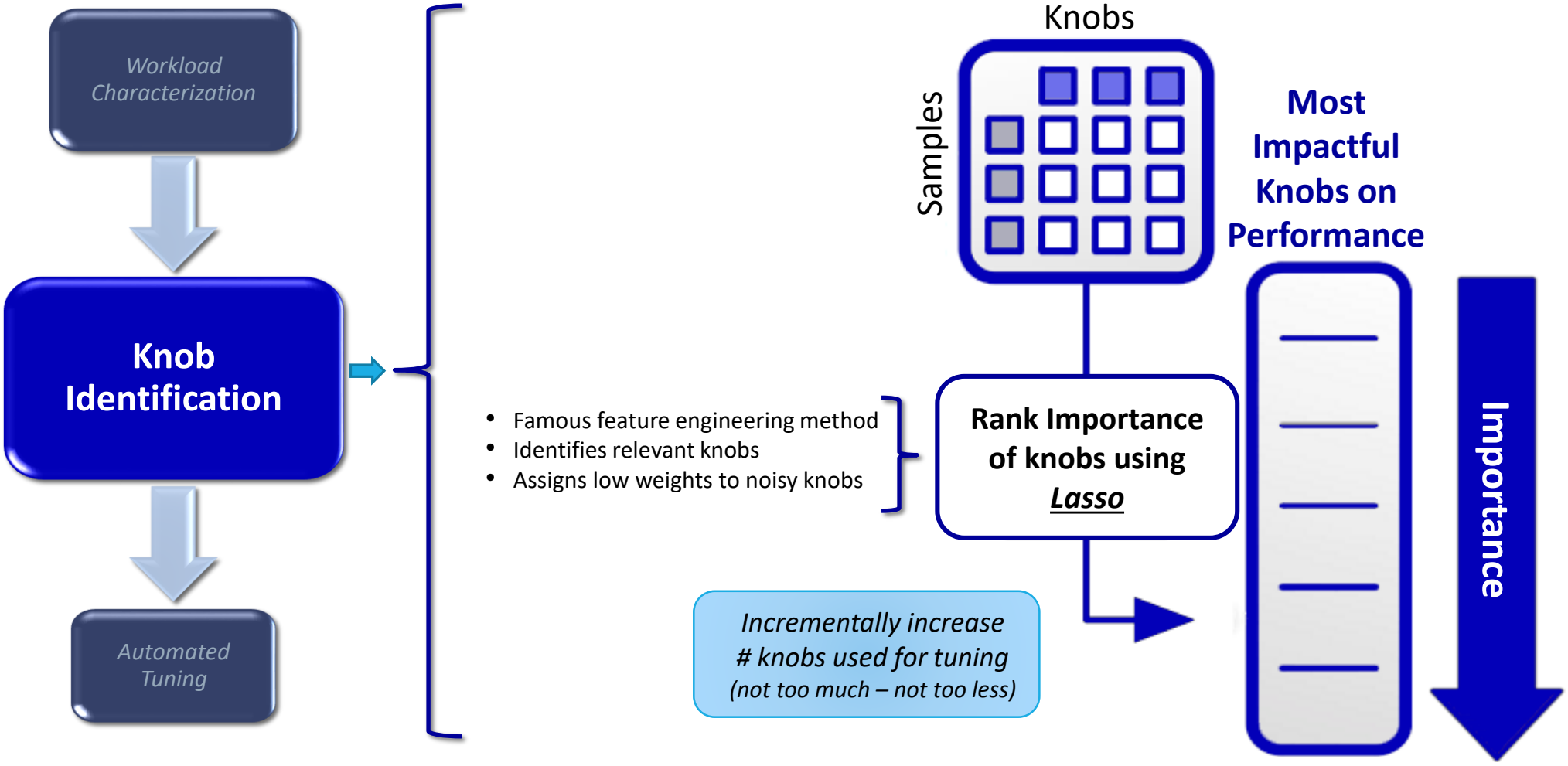
ML PIPELINE


Workload Characterization
Pick the *minimal set of metrics* to identify the workload
Knob Identification
Identify the *most critical knobs* that affects the DBMS performance
Automated Tuning
Recommend *knob settings* to improves performance

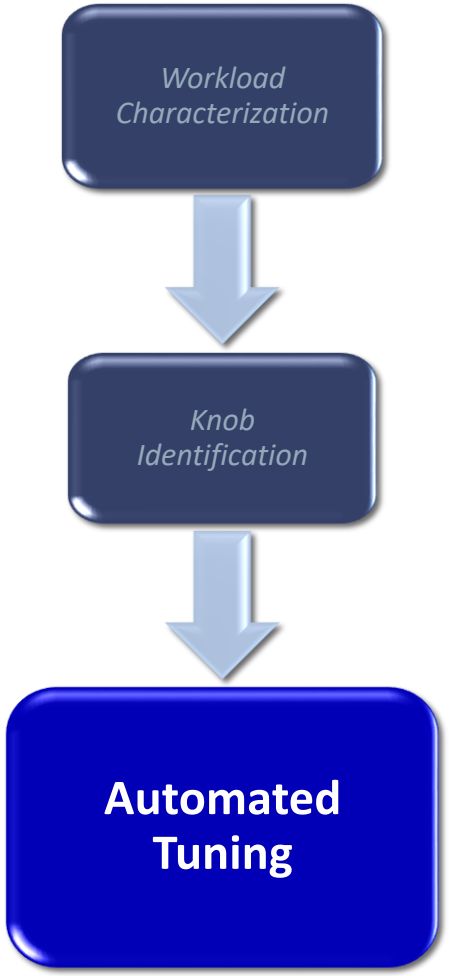
ML PIPELINE: WORKLOAD CHARACTERIZATION



ML PIPELINE: KNOB IDENTIFICATION



ML PIPELINE: AUTOMATED TUNING



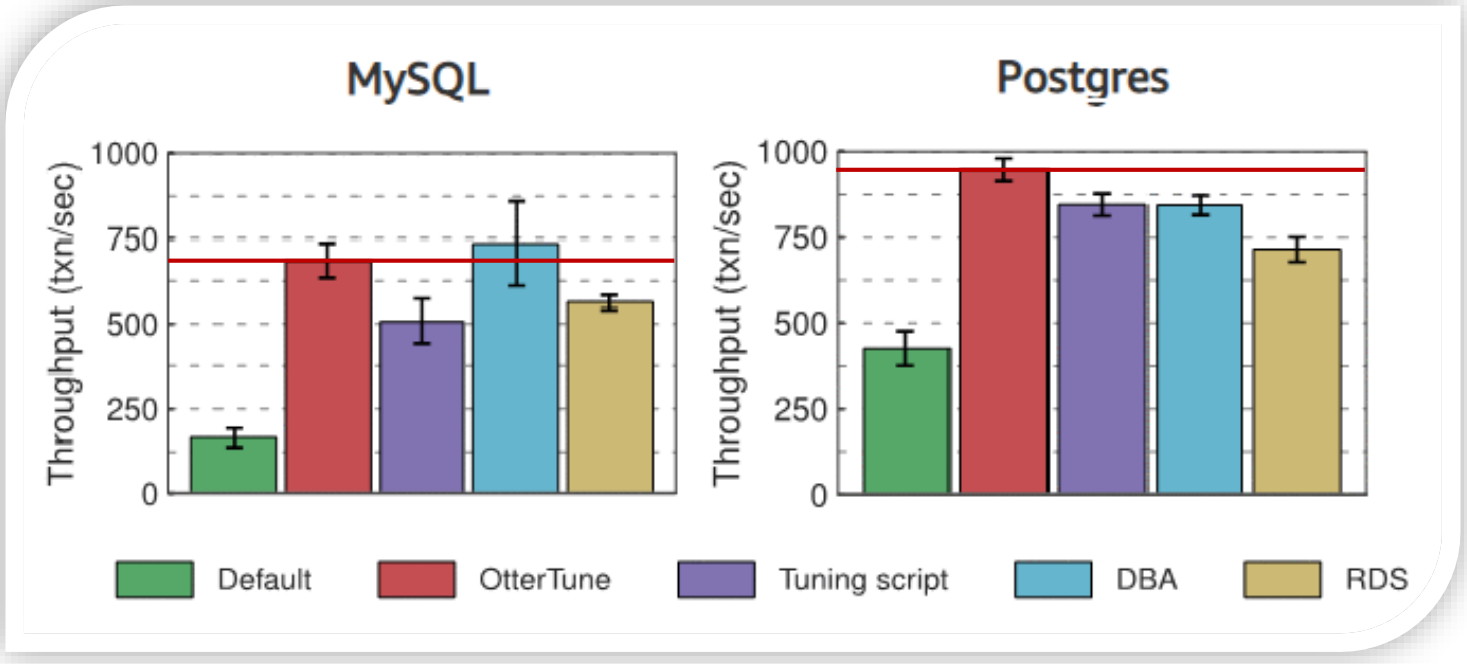
2 Step-Approach:

- **Phase 1 : Match target workload to similar workload in repository**
  - Measure similarity by computing '*average metric distance*'
- **Phase 2 : Generate configuration recommendations to optimize 'Target Objective'**
  - Similar workload data used to train 'Gaussian Process' model
    - – Predict 'Target Objective' values for sample set of possible configurations
    - – Optimize next knob configurations using *'Exploration vs Exploitation'* framework

      (*explore configurations near current best with the sole purpose of exploiting a better Target Objective)*

Expected Improvement + Knob Configuration Recommendations

OTHERS VS OTTERTUNE



Throughput of MySQL and Postgres for configurations generated by *different tuning agents*[1]

- Experiment conducted on Amazon EC2
  - Throughput equal or better than the rest!

OtterTune produce ***94%*** of DB experts generated configurations within ***60 mins!***[1]

[1] Van Aken, Dana, et al. "Automatic Database Management System Tuning Through Large-scale Machine Learning." *International Conference on Management of Data*. ACM, 2017.

THANK YOU!