UNIVERSITY OF CAMBRIDGE

Accelerating the Configuration Tuning of Big Data Analytics with Similarity-aware Multitask Bayesian Optimization

Ayat Fekry, Lucian Carata,
Thomas Pasquier, Andrew Rice

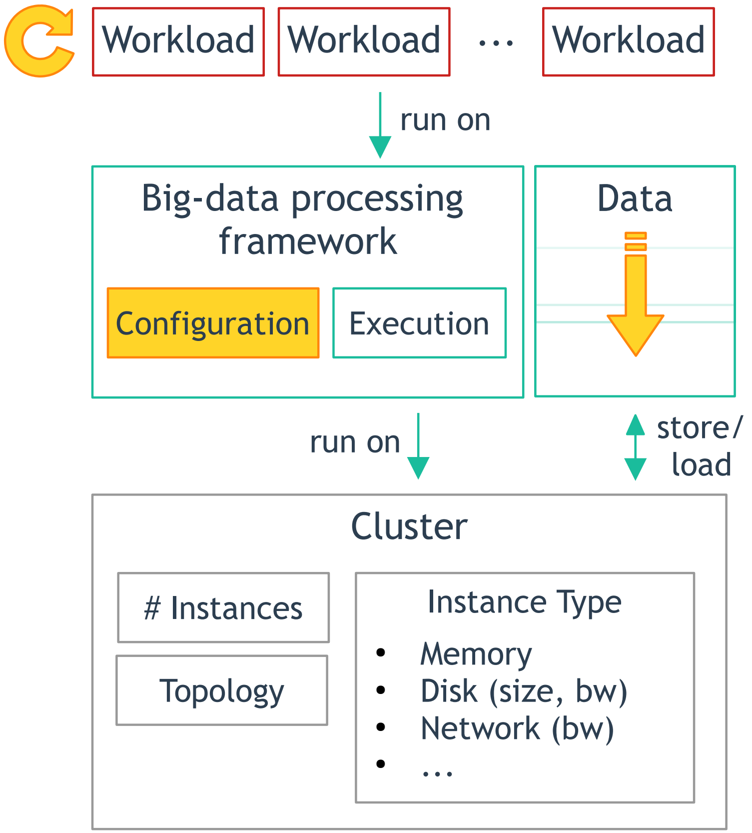
akmf3@cl.cam.ac.uk
lucian.carata@cl.cam.ac.uk

High-level problem overview

- **We want to:**
  - optimize configurations of data processing frameworks (Hadoop, Spark, Flink) in workload-specific ways.
  - allow amortization of tuning costs in realistic settings:
    - evolving input data (increase in size, change of characteristics)
    - an elastic cluster configuration

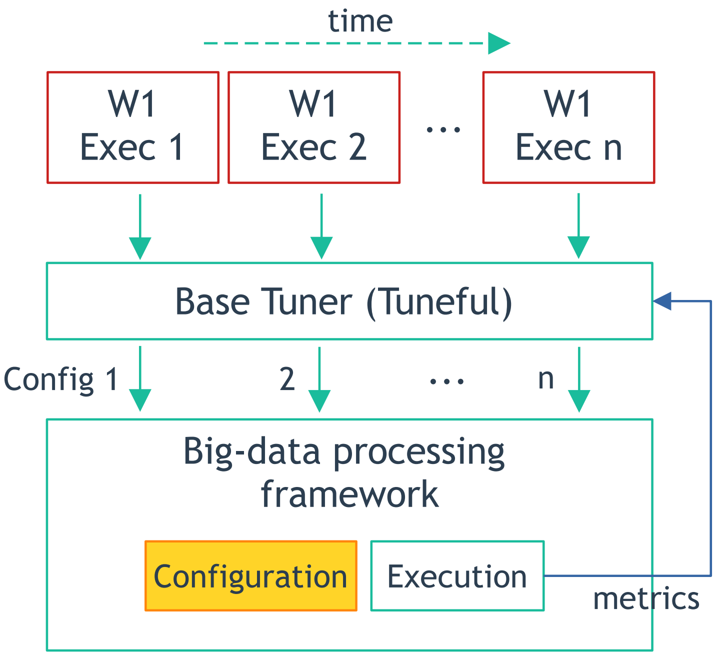
High-level problem overview

- **We want to:**
  - optimize execution of workloads in data processing frameworks (Hadoop, Spark, Flink)
  - allow amortization of tuning costs in realistic settings:
    - evolving input data (increase in size, change of characteristics)
    - an elastic cluster configuration
- **When assuming repeated workload execution**
  - daily/weekly/monthly reporting
  - incremental data analysis
  - frequent analytics queries/processing



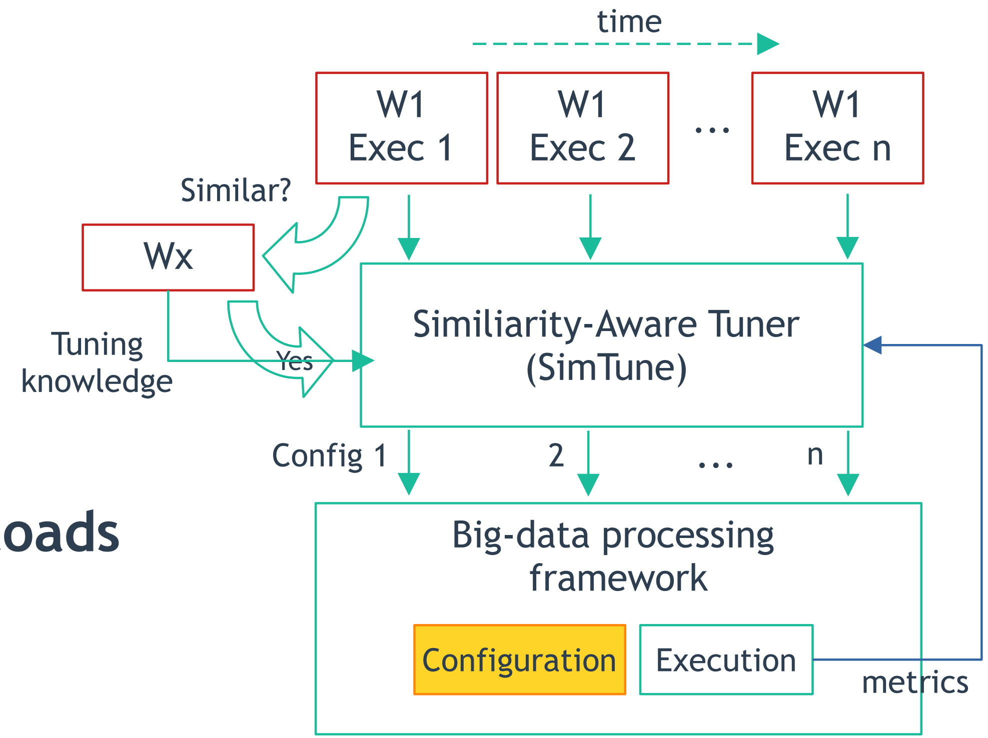
High-level solution overview

- How:
  - By incrementally tuning the configuration of the framework
    - per workload
    - determining and tuning only significant parameters
    - aim is to quickly converge to configurations close to optimum



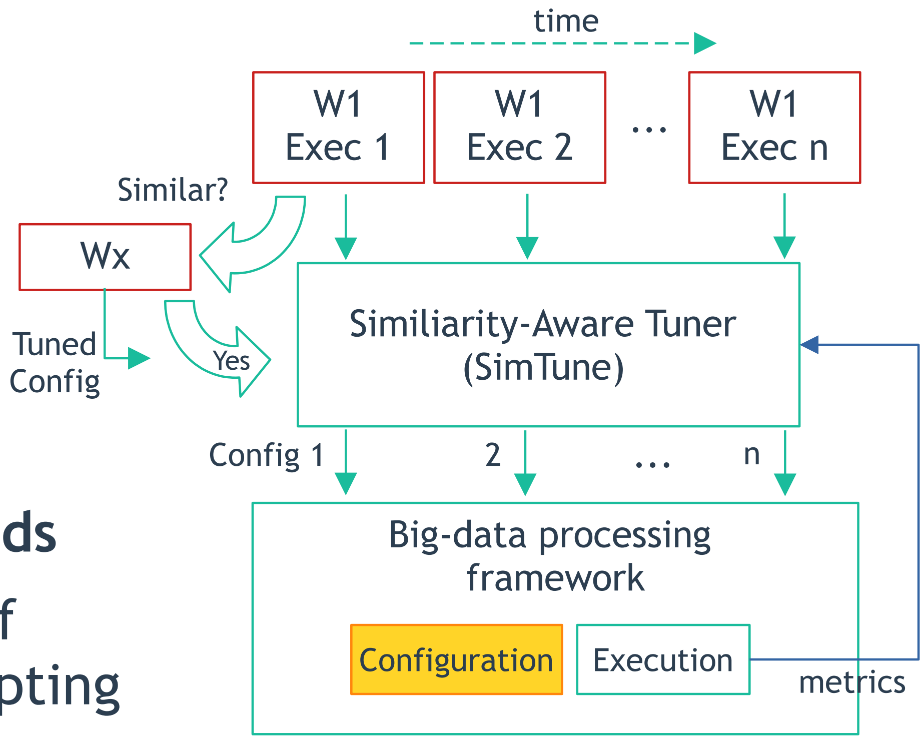
High-level solution overview

- **How:**
  - By incrementally tuning the configuration of the framework
    - per workload
    - determining and tuning only significant parameters
  - **By leveraging existing tuning knowledge across similar workloads**



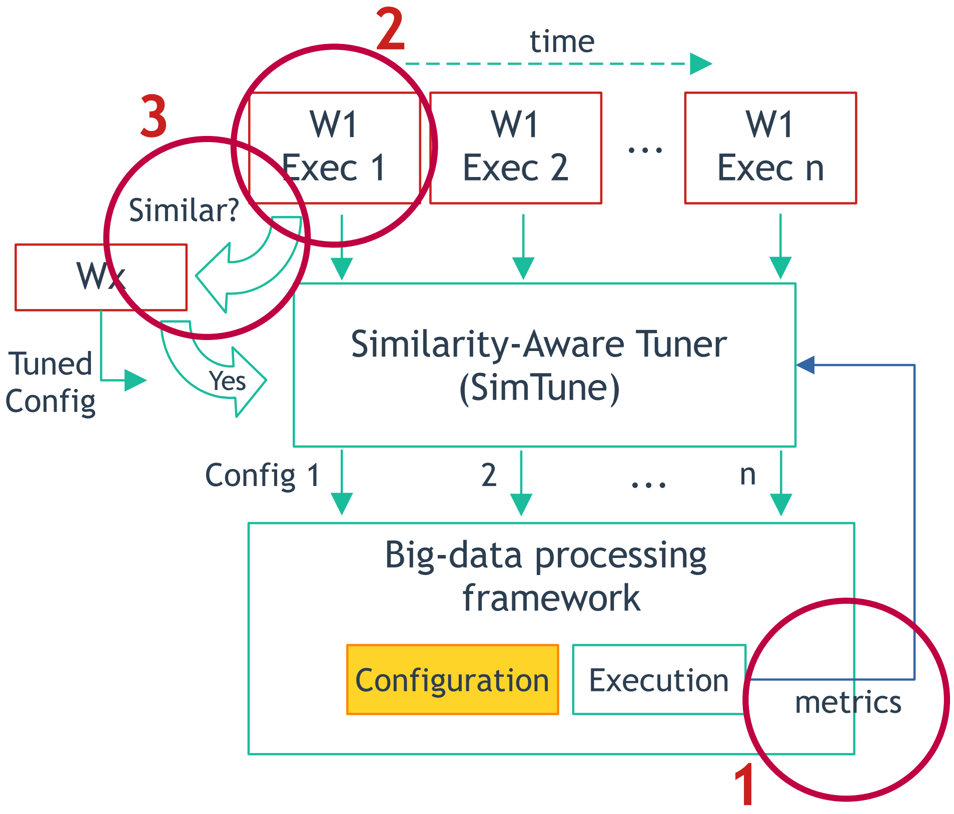
High-level solution overview

- **How:**
  - By incrementally tuning the configuration of the framework
    - per workload
    - determining and tuning only significant parameters
  - **By leveraging existing tuning knowledge across similar workloads**
  - By carefully combining a number of established ML techniques and adapting them to the problem domain



Required puzzle pieces

- **Workload characterization**
  1) Workload monitoring
  2) Workload representations
  3) Similarity analysis



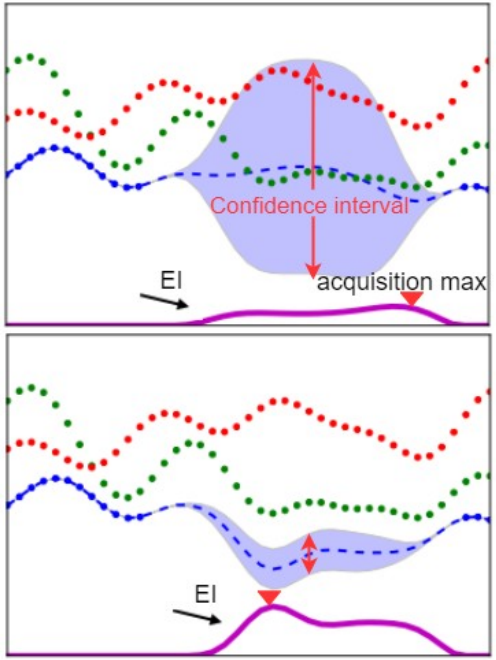
Required puzzle pieces

- **Workload characterization**
  1) Workload monitoring
  2) Workload representations
  3) Similarity analysis

- **Similarity-aware tuning**
  4) Multitask Bayesian Learning

Single task modeling for blue

Multitask modeling for blue
using knowledge about red and green



[1]

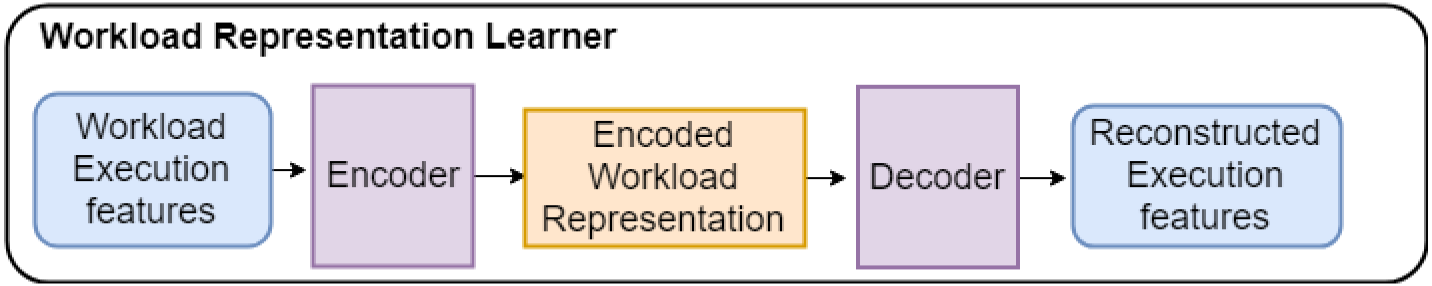
[1] K. Swersky et. all, *Multi-task bayesian optimization*

Workload characterization

- **Monitoring workload caracteristics & resource consumption**
  - Metric examples:
    - number of tasks per stage, input/output size, data spilled to disk, etc
    - CPU time, memory, GC time, serialization time, …
  - **Representing metrics in relative terms**
    - GC time as proportion of total CPU time
    - Amount of shuffled/disk spilled data as proportion of total input data

Workload characterization

- ## Workload representation
  - Would like a **low-dimensionality** representation because it’s difficult to come up with informative distance metrics in high-dimensional space
  - We propose an autoencoder based solution, where the low-dimensionality representation is **learned**
    - offline phase based on historic execution metrics
    - resulting encoding/decoding model can be reused



Workload characterization

- **Similarity analysis**
  - Given new workload, find a **source** (already tuned) workload
    - Closest in encoded representation space (using L_1 norm)
    - Distance computed on a fixed fingerprinting configuration for the new workload

Similarity-aware tuning

- Assume a source workload s was found for workload w
  1) Tune the same significant parameters as for s
  2) Retrieve Bayesian tuning model of s, T_s
  3) Add w as a new task to T_s
  4) Suggest the next (tuned) configuration sample, cs_w for w
  5) Update tuning model with metrics from executing w with configuration cs_w

Similarity-aware tuning

- Natural criteria for stopping the tuning
  - e.g: Acquisition function maximum (Expected Improvement) drops below 10%

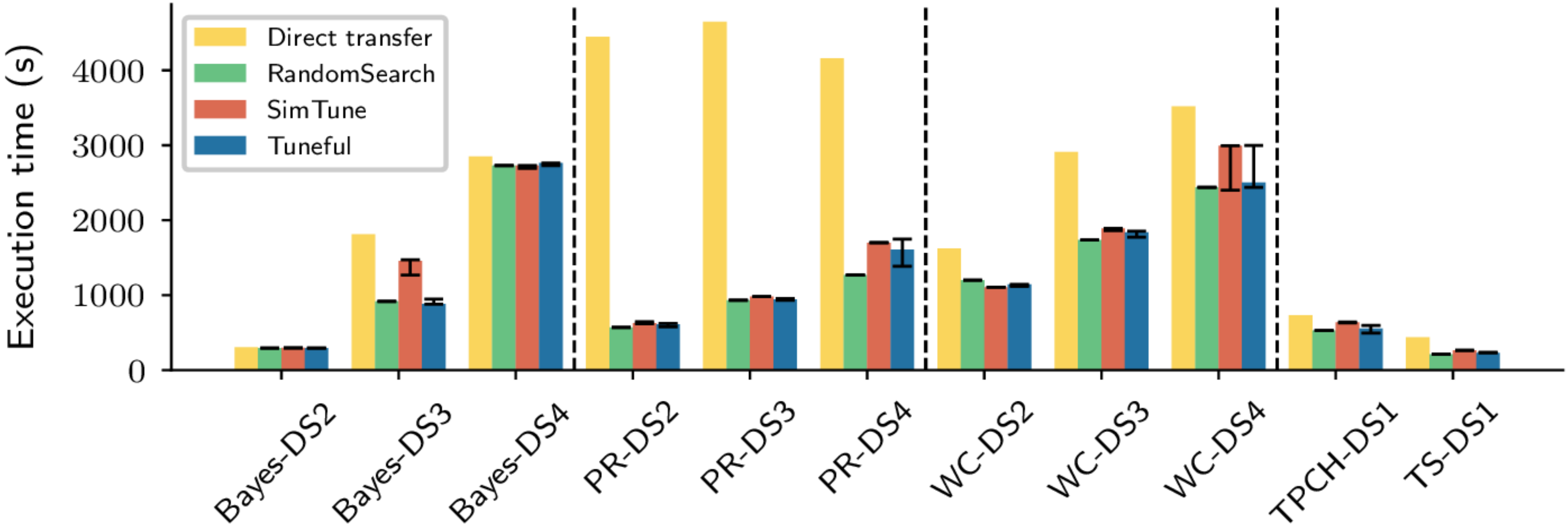
- Method able to detect inaccurate similar workload matching
  - Large difference between cost predicted by model and actual execution, across multiple executions

Experiments

pre-tuned (source) set

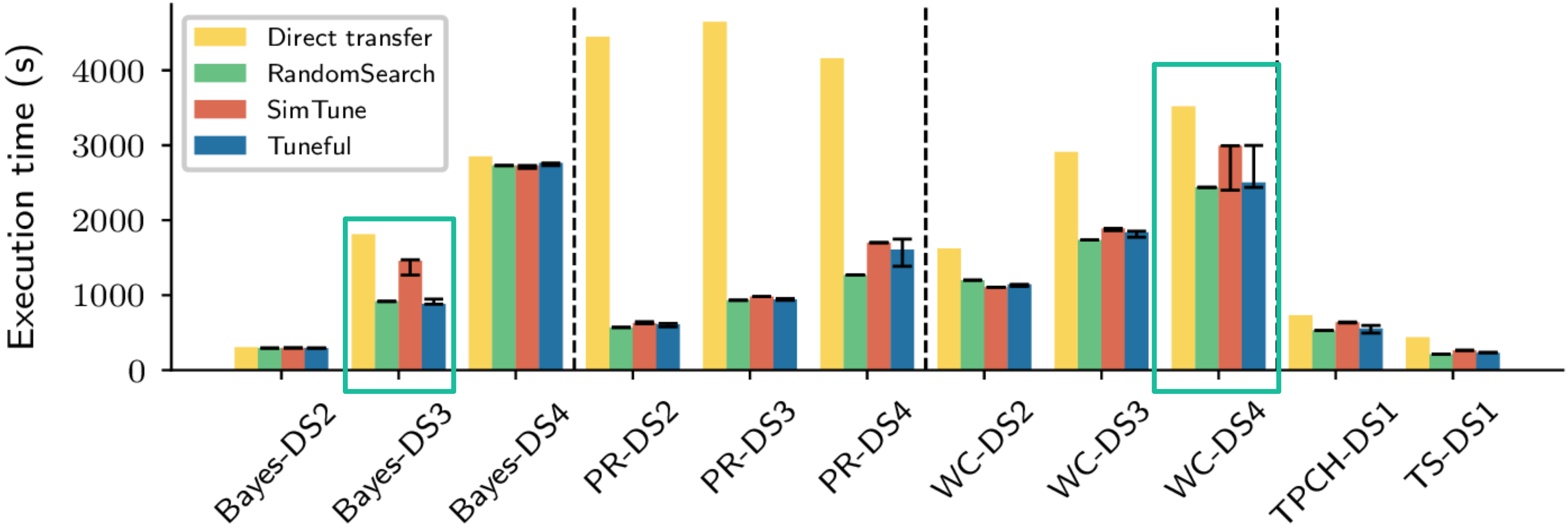
Workload (Abbrev)	Input data sizes (DS)					Units
	DS1	DS2	DS3	DS4	DS5	
PageRank (PR)	5	10	15	20	25	million pages
Bayes Classifier (Bayes)	5	10	30	40	50	million pages
Wordcount (WC)	32	50	80	100	160	GB
TPC-H Benchmark (TPCH)	20	40	60	80	100	GB (compressed)
Terasort (TS)	20	40	60	80	100	GB

Tuned execution times (at convergence)



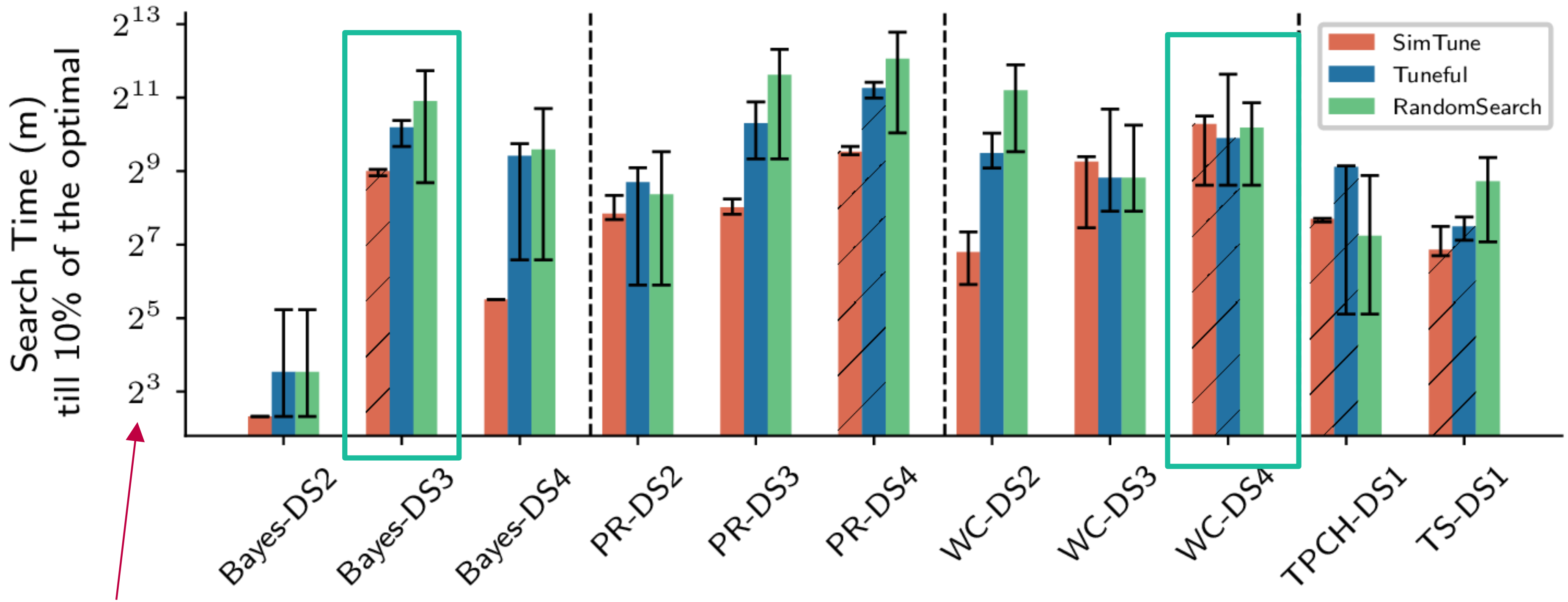
Source dataset: *-DS1

Tuned execution times (at convergence)



Source dataset: *-DS1

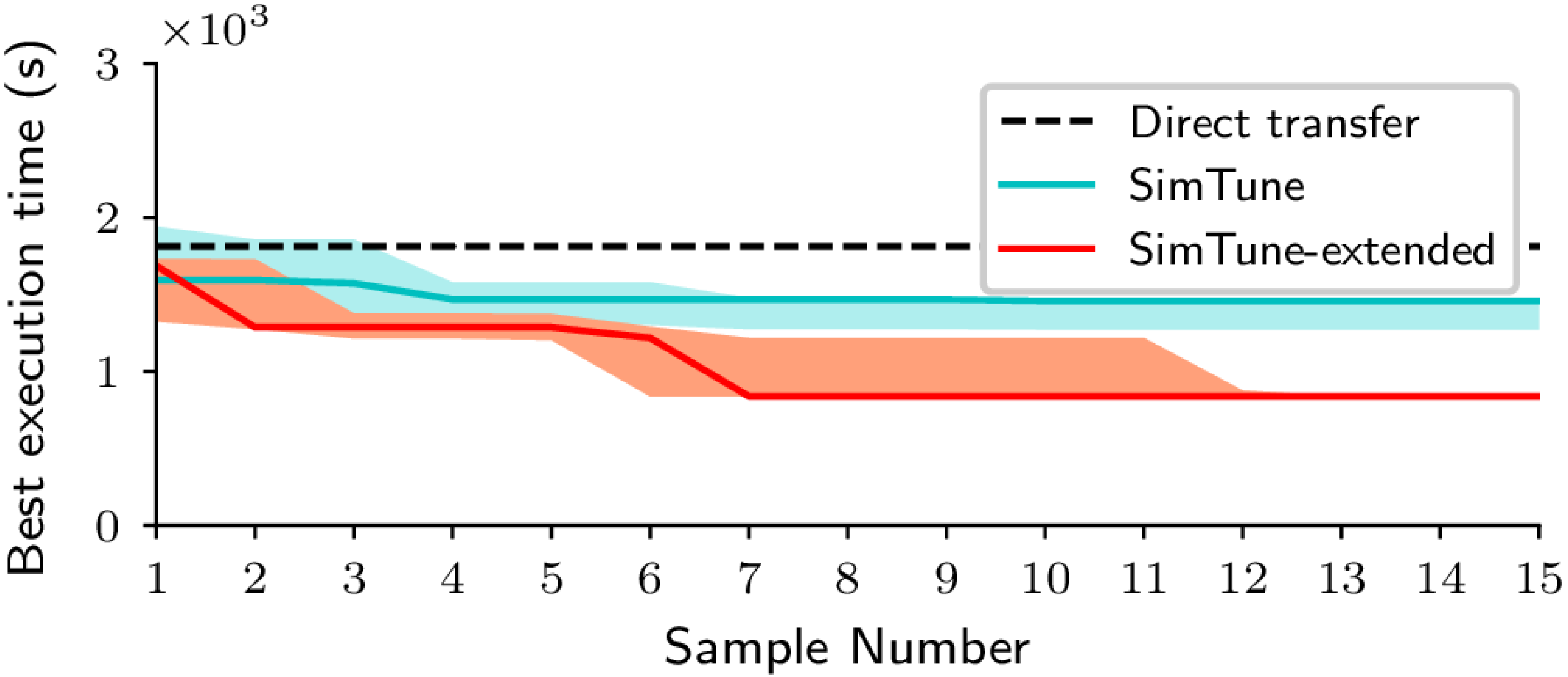
Time until finding best configuration



log axis

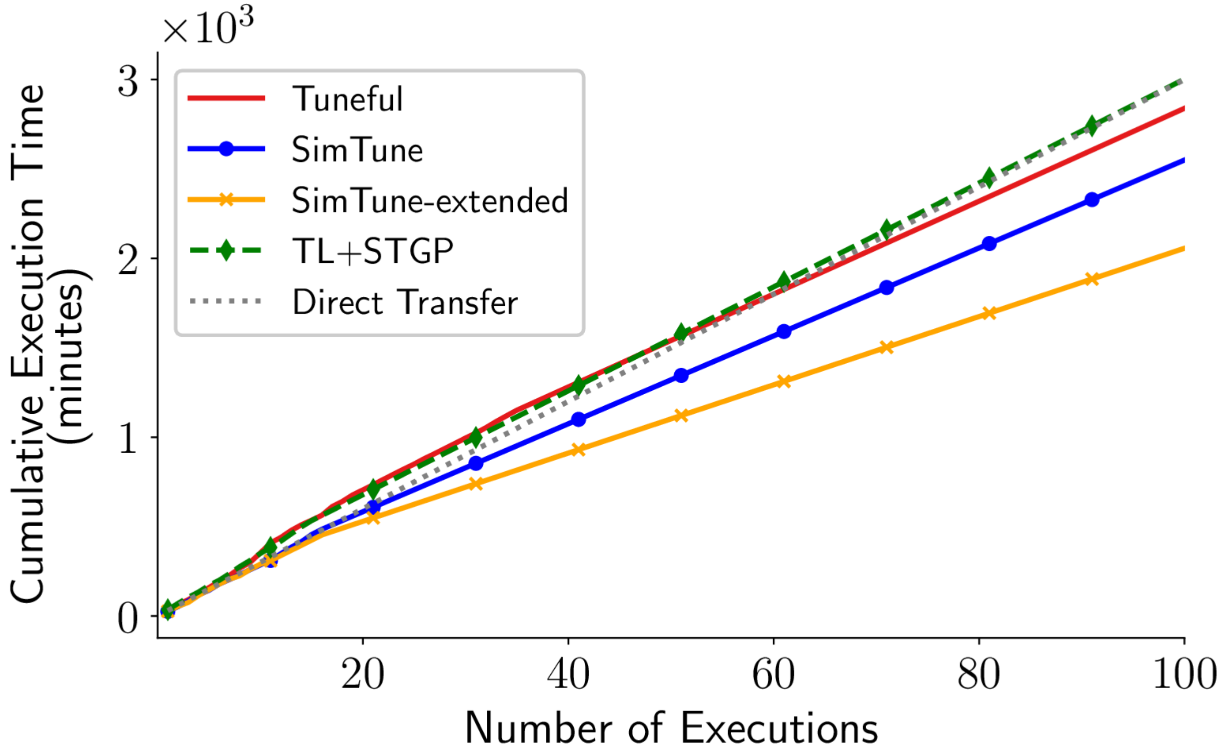
`Source dataset: *-DS1`

Extended tuned (source) dataset for Bayes-DS3



Source dataset: *-DS1 + Bayes DS2

Tuning cost amortization (Bayes-DS3)



SimTune source dataset: *-DS1
SimTune-extended source dataset: *-DS1 + Bayes-DS2

UNIVERSITY OF CAMBRIDGE

Thank you! Ready for questions!

https://github.com/ayat-khairy/simtune

Interested in discussing off-line or colaborating?

akmf3@cl.cam.ac.uk
lucian.carata@cl.cam.ac.uk