To tune or not to tune

Thomas Pasquier
tfjmp@cs.ubc.ca
https://tfjmp.org

The team

- **Ayat Fekry**, PhD student
- **Lucian Carata**, Senior Research Associate
- **Andrew Rice**, Professor
- **Andy Hopper**, Professor

About me

- Assistant Professor at the University of Bristol
- Moving to UBC in Summer 2021
- Area of research
  - Provenance-based Security/Auditing/IDS (SoCC, CCS, NDSS, USENIX Sec)
  - Self-tuning data processing framework (KDD, ICDCS)
  - Microsoft Cloud Computing Research Centre (http://www.mccrc.org/)
  - Reproducibility of Scientific Results
- Observing and understanding what computer systems do

About me

- Assistant Professor at the University of Bristol
- Moving to UBC in Summer 2021
- Area of research
  - Provenance-based Security/Auditing/IDS (SoCC, CCS, NDSS, USENIX Sec)
  - Self-tuning data processing framework (KDD, ICDCS)
  - Microsoft Cloud Computing Research Centre (http://www.mccrc.org/)
  - Reproducibility of Scientific Results
- Observing and understanding what computer systems do
- **Systems background**

Let’s talk about Tuneful

Talk based on the following publications

- Ferky et al. “**Towards Seamless Configuration Tuning of Big Data Analytics**”, ICDCS 2019
- Fekry et al. “**Tuneful: An Online Significance-Aware Configuration Tuner for Big Data Analytics**”, arxiv 2020
- Fekry et al. “**To Tune or Not to Tune? In Search of Optimal Configurations for Data Analytics**”, KDD 2020
- Fekry et al. “**Accelerating the Configuration Tuning of Big Data Analytics with Similarity-aware Multitask Bayesian Optimization**”, BigData 2020

Backed by experiments

- 7429h of Spark execution (see KDD)
- Over Amazon Web Service and Google Cloud Platform
  - No Microsoft yet ;)

https://github.com/ayat-khairy/tuneful-data

Motivation

- Discussing with scientist and colleagues
- Using data analytics platform is easy
- … using them efficiently is hard
  - **How do I configure this thing?**
- Wasted budget
  - **How do I save money?**
- 40% of jobs are recurrent

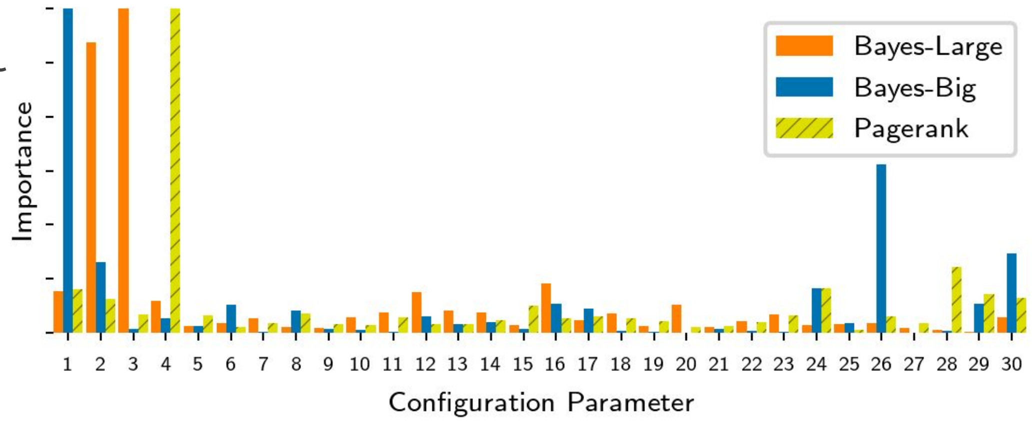
How can we help?

Challenges

Challenges: configuration parameters

One model does not fit all

Amazon/Google provide Configuration for Spark Cluster (from experiment **25% to 63% slower** than optimal)



Significant parameters analysis on HiBench Workloads

Challenges: finding the right configuration

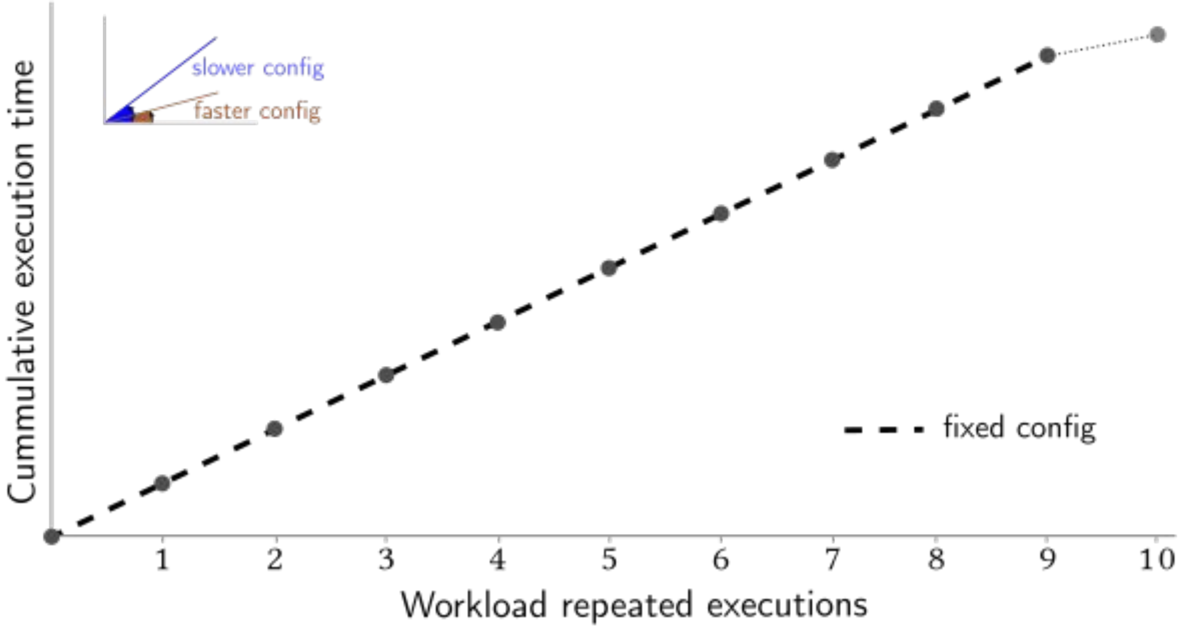
- Using a good enough configuration?
- Building a general model?
  - Needs hours of data, only feasible by cloud providers (maybe)
- Tuning for my specific workload?
  - Is it worth the cost?

Our idea

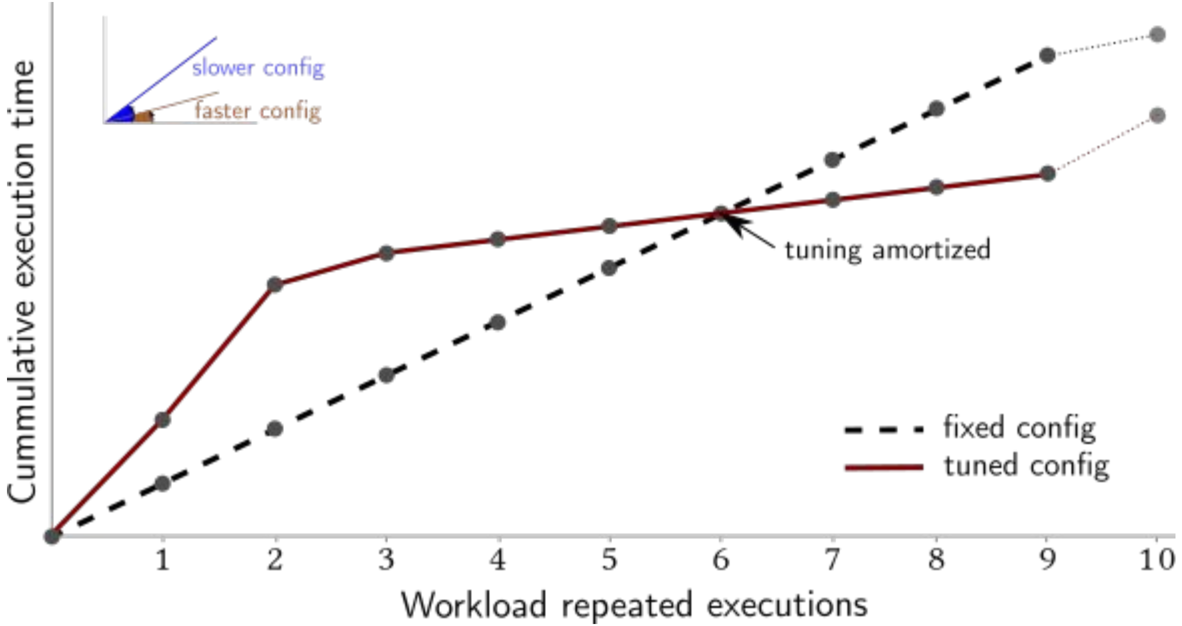
- Given a user and a cluster
- Assumption that most tasks occur more than once

Can we identify a better configuration while doing useful work?

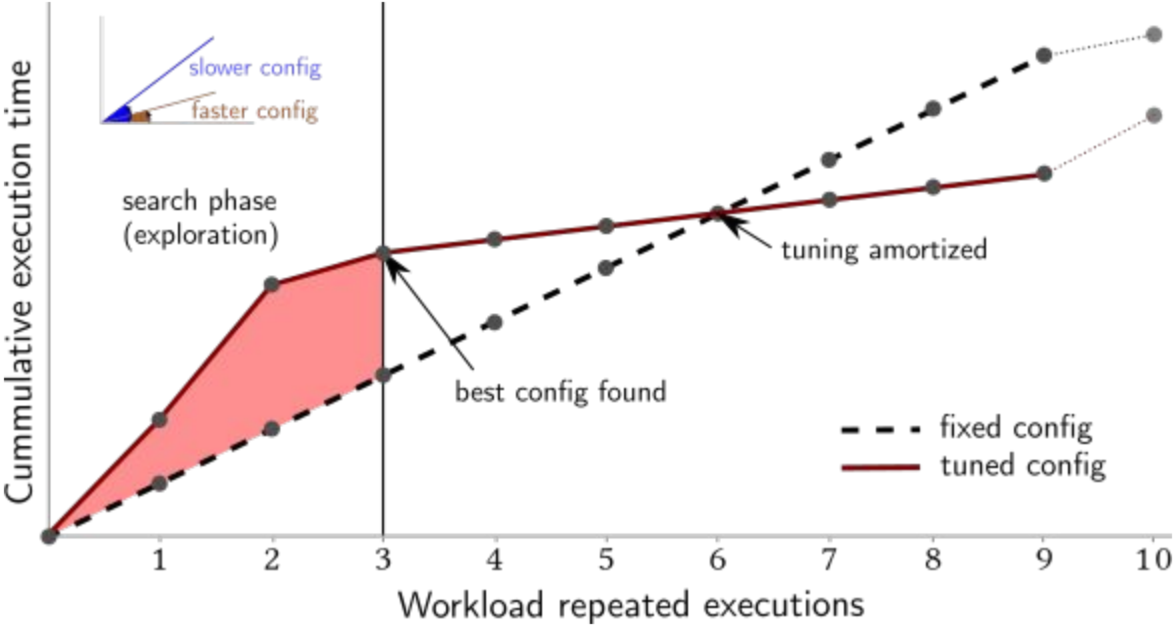
Cost amortization model



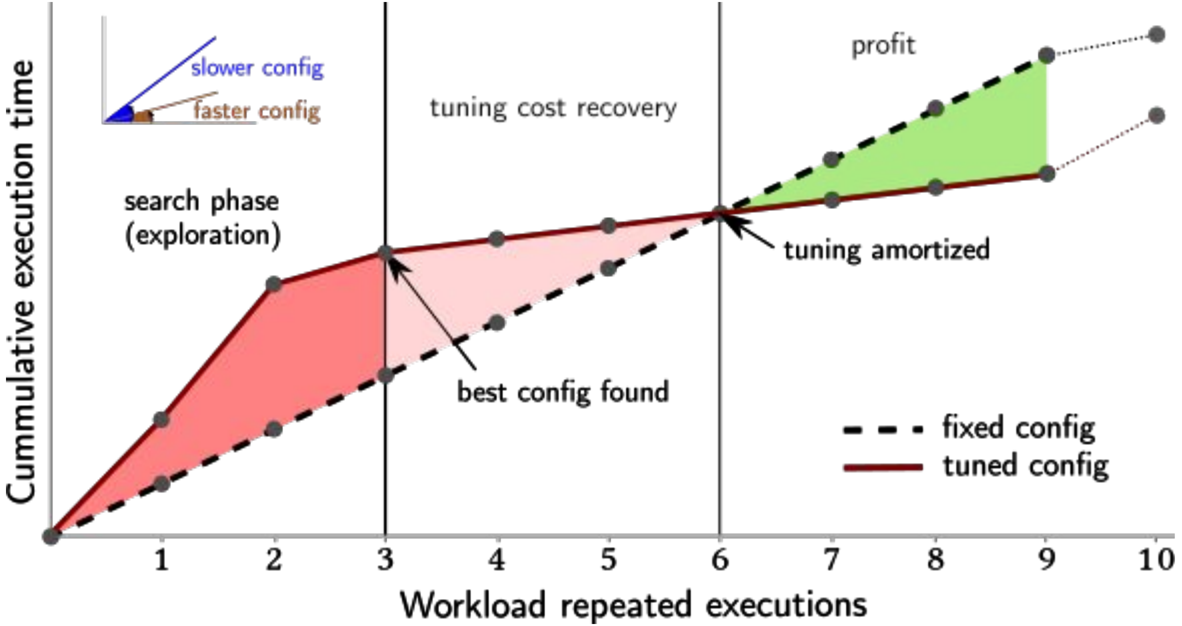
Cost amortization model



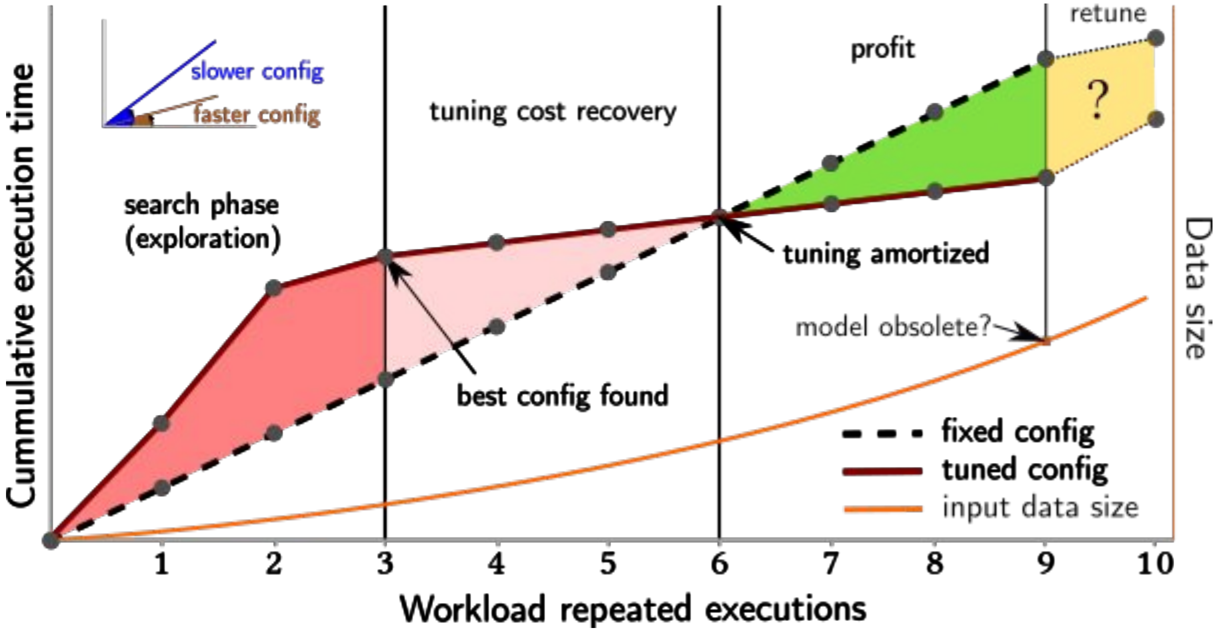
Cost amortization model



Cost amortization model



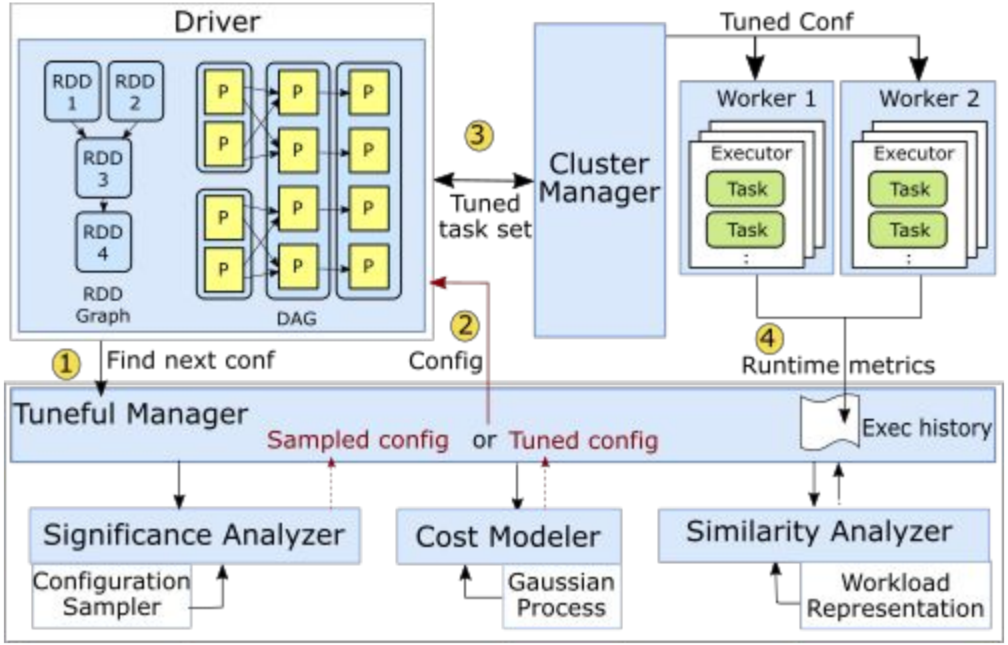
Cost amortization model



Solving the challenges

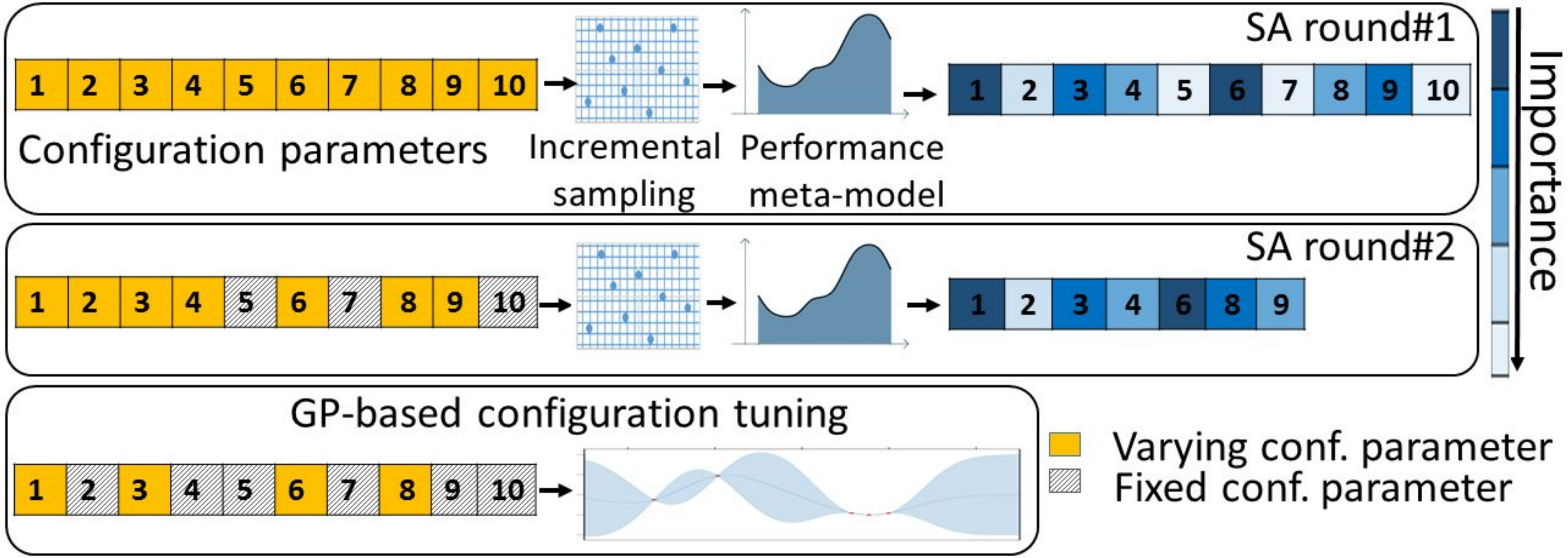
Overall architecture

- Spark extension
- Zero-knowledge tuning
- Significance-aware
- Similarity-aware
- Low exploration time
- … faster cost amortization



https://github.com/ayat-khairy/tuneful-code

Overview



SA round#1
1 2 3 4 5 6 7 8 9 10
Configuration parameters
Incremental sampling
Performance meta-model
1 2 3 4 5 6 7 8 9 10
SA round#2
1 2 3 4 5 6 7 8 9 10
1 2 3 4 6 8 9
Importance
GP-based configuration tuning
1 2 3 4 5 6 7 8 9 10
Varying conf. parameter
Fixed conf. parameter

Multi-round Sensitivity Analysis

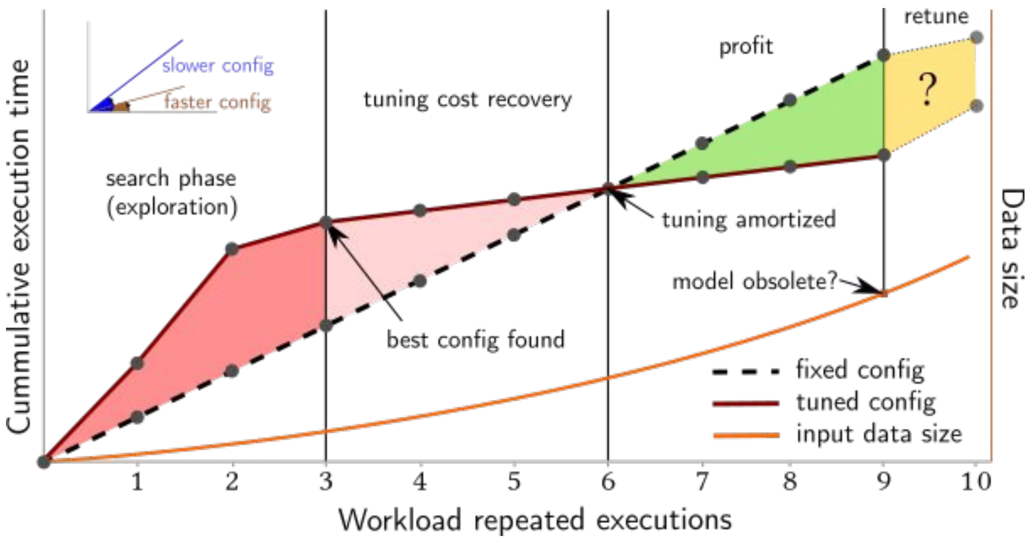
- Naive approach run an extensive benchmark
- Instead we sample a few configuration point
- Build model to predict execution time
  - Random Forest
- Empirically, we know few parameters are influential
- … **model does not need to be very accurate**
- Gini importance to find influential parameters
  - Features contributions based on how many times it is used in a tree split
- Each round we eliminate X% unimportant parameters (i.e. “fix” them)
- Run again for another round

Gaussian Process

- This time we need accuracy
- Use the significant parameters
- Predict execution time at n+1
- Rapidly converge towards optimal configuration

- When prediction consistently differ from observation
  - Tuning needs to be redone
  - Can be caused by change in dataset, cluster hardware etc.

Gaussian Process



- When prediction consistently differ from observation
  - Tuning needs to be redone
  - Can be caused by change in dataset, cluster hardware etc.

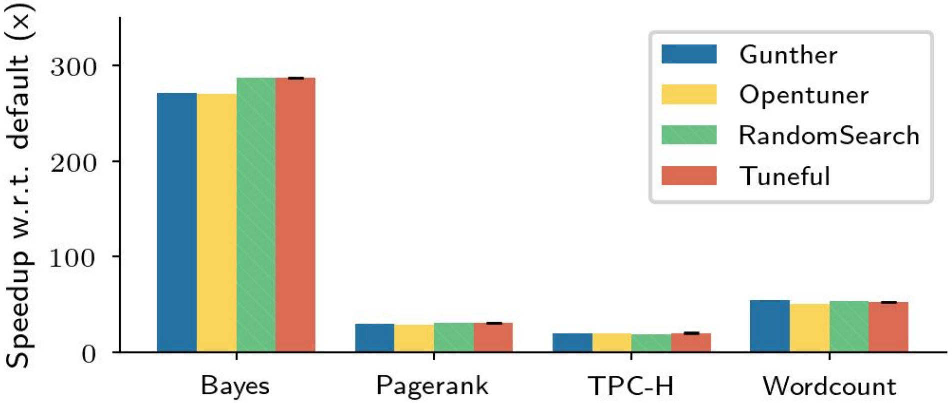
Budget - based on empirical study

- Significant parameters exploration
  - 20 samples (2 rounds at 10)
  - Empirically correct results when compared to expensive Recursive Feature Elimination* as ground truce
- Configuration Tuning
  - 15 Samples
  - Empirically good configurations

* Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. Machine learning. 2002.

Finding good configuration

- Tuneful 35 executions budget
- All other 100 executions
- Gunther*
  - Genetic algorithm
- Opentuner+
  - Ensemble of search techniques
  - Hill climbing, differential evolution and pattern search

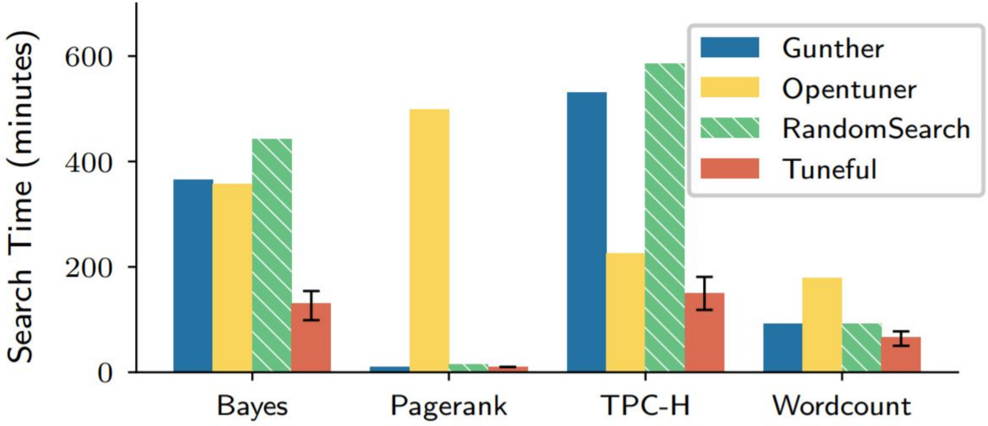


*Guangdeng et al. Gunther: Search-based auto-tuning of MapReduce.

+Jason et al. Opentuner: An extensible framework for program autotuning.

Reaching 10% of optimum

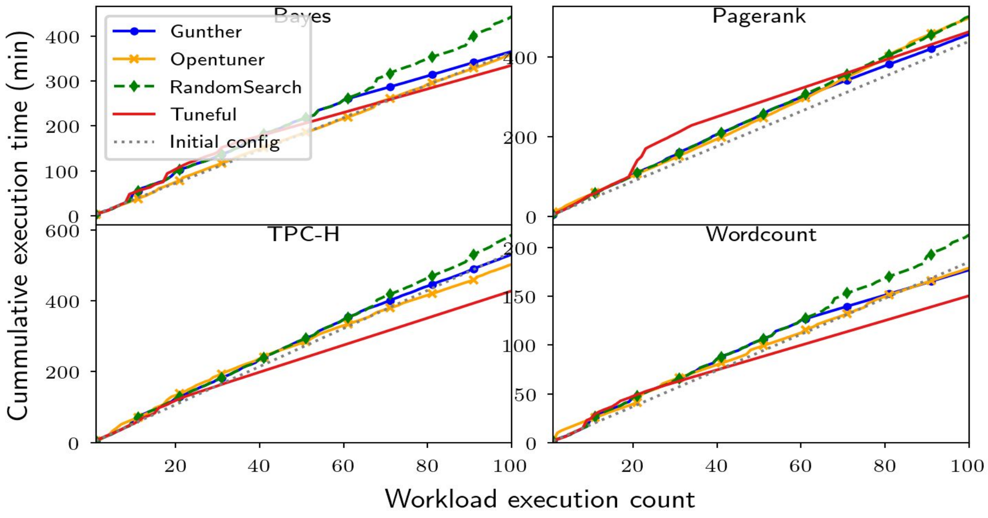
- Same budget
- Time to get to 10% of optimum
- What matters is not only the number of samples but how fast they execute



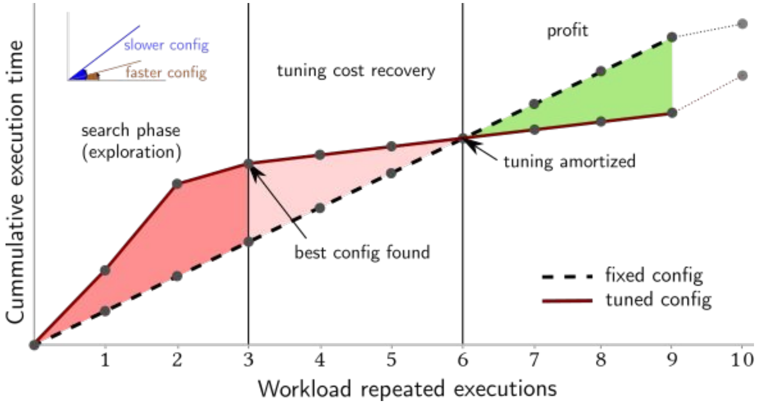
GP Converge towards the optimum and therefore reduce cost

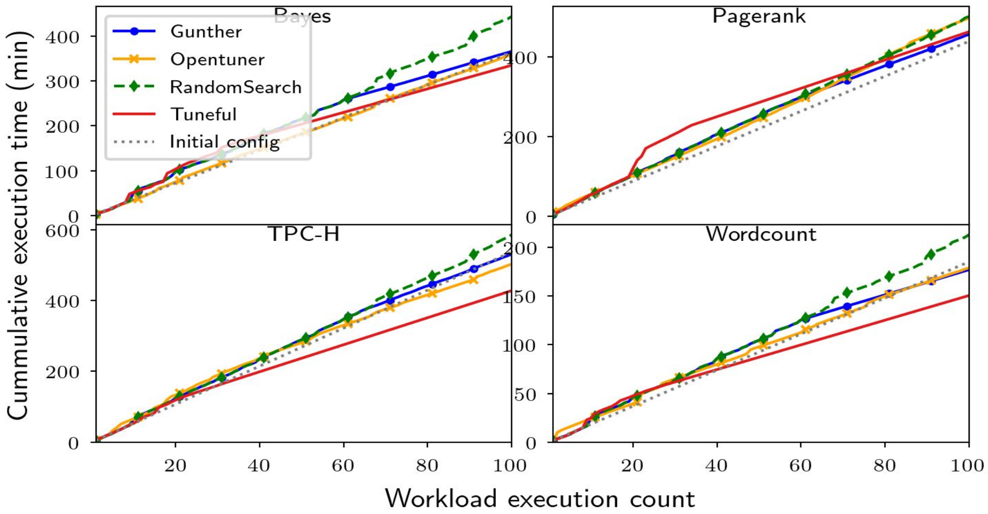
Cost Amortisation

- Let the algorithms run and see if we save Money
- Plot cumulative cost
- Spoiler: random search won’t ;)
- Gunther and Opentuner converge to some local minima eventually
- Tuneful has a spike in cost at the start of the GP, then stabilise to close to optimal



Cost Amortisation





- Tuneful has a spike in cost at the start of the GP, then stabilise to close to optimal

Optimization

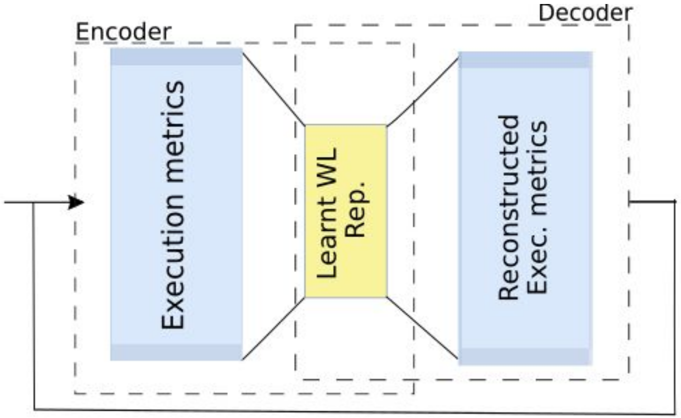
What could we improve?

- We configure each workload independently
- We do not learn from other workloads running on our cluster

Maybe we should?

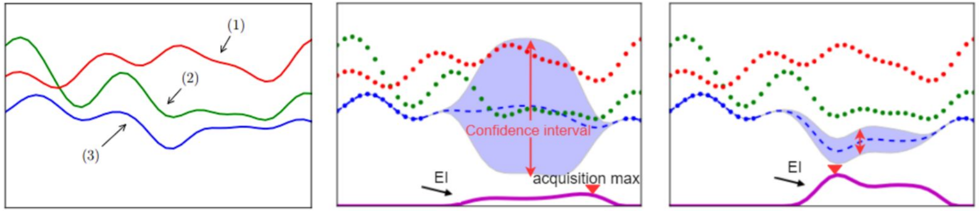
Tuneful evaluation: limited-knowledge tuning

- Same setting as before
- Cluster ran workloads for a while
- We captured execution metrics
- Similarity between workload via lower dimension projection
- Assume similar workload have similar execution parameters
- Use Multi Task Gaussian Process to optimize config.



Multi Task Gaussian Process

- We identified similar workload
  - same significant parameters
- We use Multi Task Gaussian Process (MTGP)
- Each workload is a task in MTGP
- Allow to find a good configuration much Faster
  - No SA
  - 10 round for GP as before



(a) Multitask GP sample functions (b) Independent GP Modelling (c) Multitask GP Modelling

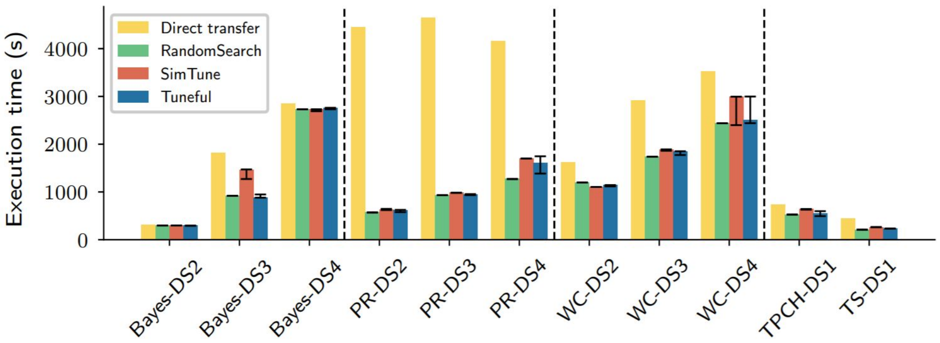
Fig. 1: (a)The actual functions of a Multitask GP with three tasks. Task 2 and 3 are strongly correlated, 1 and 3 are anti-correlated, and 1 and 2 are not correlated. (b) Independent single tasked GP modelling for Task 3. (c) Multitask GP modelling for Task 3, utilizing the other tasks (figure source is [36] with minor edits applied for more clarity).

Finding good configuration

- Tuneful (zero-knowledge)
- Direct transfer
- Random Search
- Simful (limited-knowledge tuneful) a.k.a. Transfer Learning + MTGP

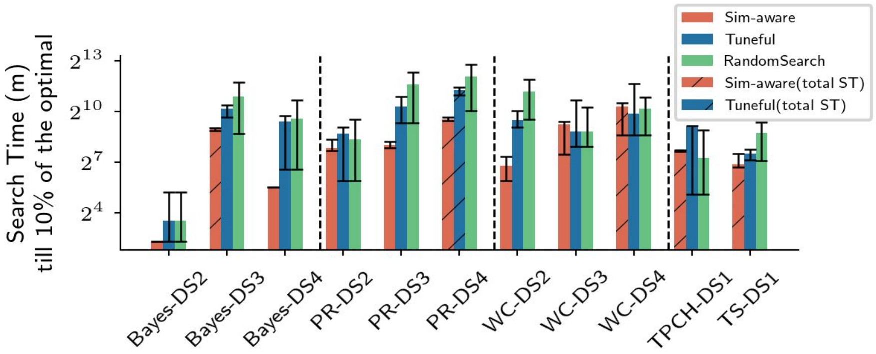
Budget:

- random search 100
- Tuneful 25
- Simtune 10



Tuneful evaluation: limited-knowledge tuning

- Measure how many minutes We need to find configuration at 10% of the optimum.

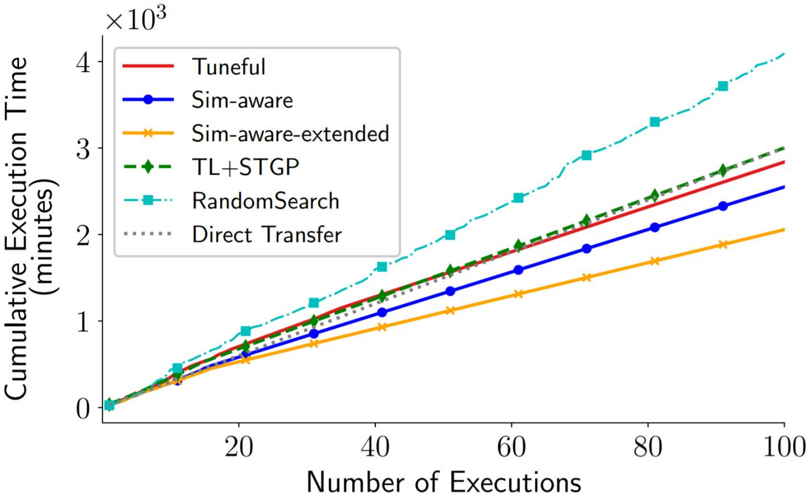


Shorter sample execution time

Simtune does generally much better!

More workloads (tasks in MTGP), better?

- Random Search
- Tuneful
- Direct Transfer
- TL + STGP
  - only significant parameters
- SimTune (5 tasks)
- SimTune-extended (8 tasks)



Simtune performs better

Able to leverage information from more workloads

Future work

- Modifying significant parameters analysis
  - Li et al. **“Statically Inferring Performance Properties of Software Configurations”** EuroSys 2020
  - May remove the need for costly sensitivity analysis
- Further engineering and deployment
  - Does it work in real life?
- Can we learn across clusters?
- Application beyond Spark? (probably yes)

… hiring students for fall 2021 at UBC

looking for collaboration!

Thank you!

tfjmp@cs.ubc.ca
https://tfjmp.org