Microsoft

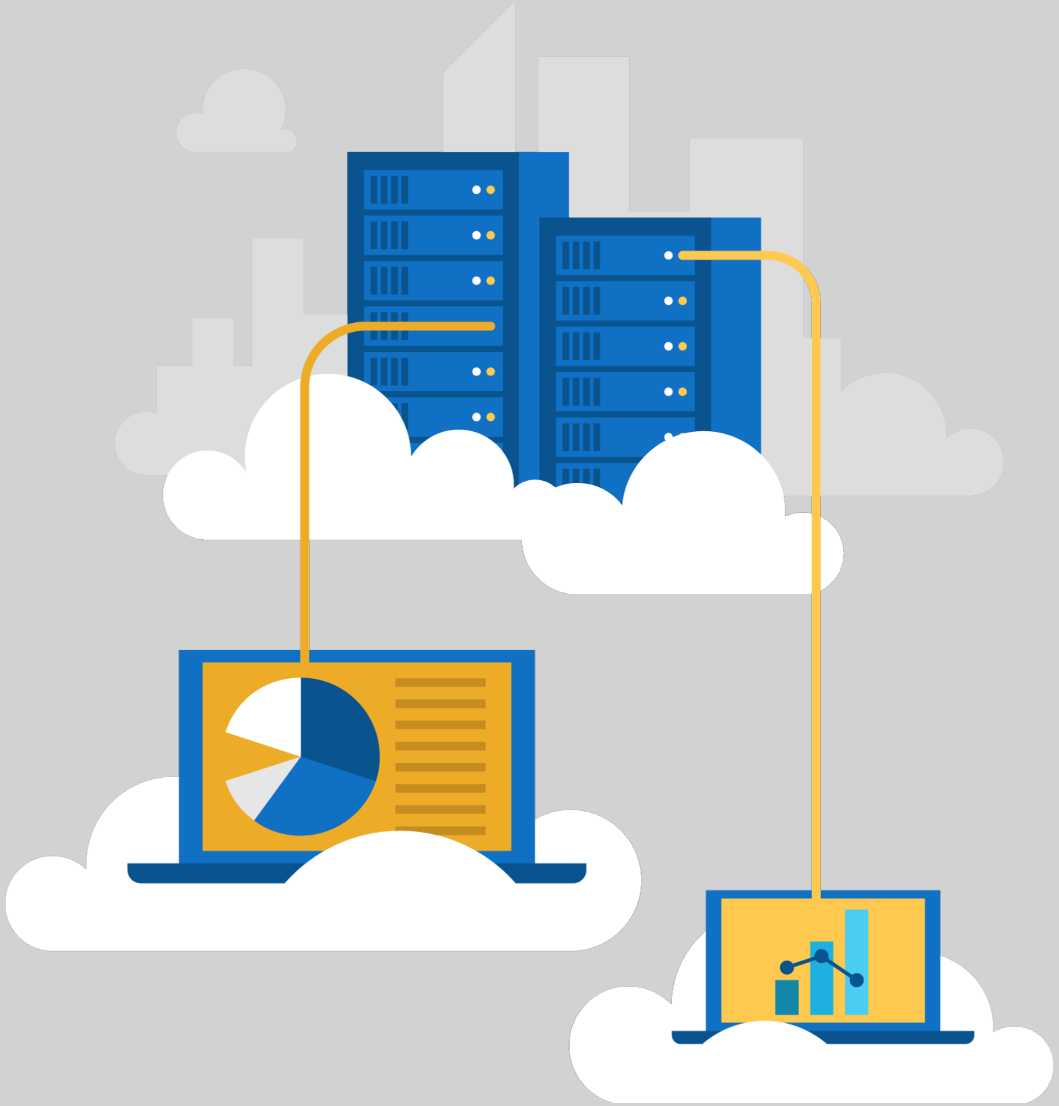
Microlearner:

A fine-grained Learning Optimizer for Big Data Workloads at Microsoft

Alekh Jindal, Shi Qiao, Rathijit Sen, Hiren Patel

Microsoft

April 2021



Cloud Trends

- Managed Query Engines
  - Easy to get started: no setup/installation
  - Serverless models: no resource provisioning
- Complex Workloads
  - Large and growing:
    Millions of queries, machines; Exabytes of data
  - Sophisticated:
    SQL + UDFs (Python, C#, Java) + ML …

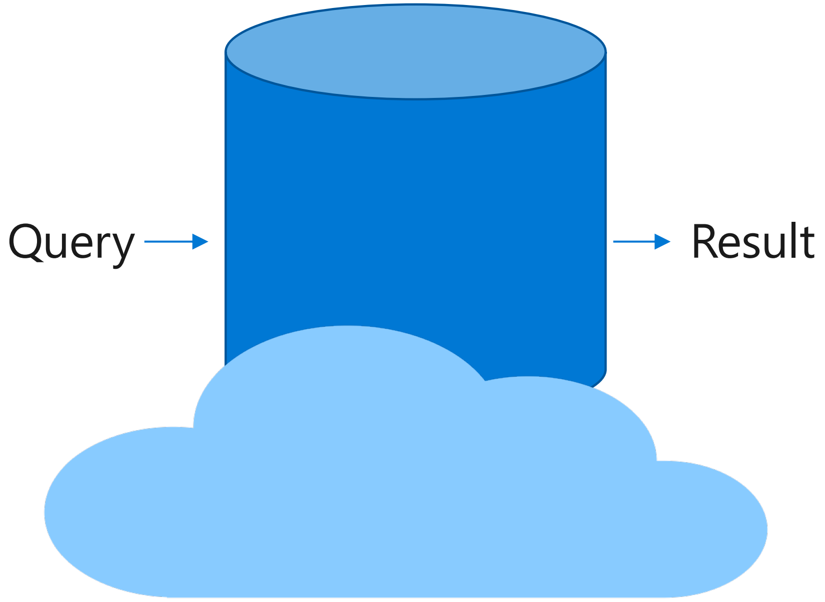
⇒Total cost of ownership (TCO) is important!

⇒Lots of moving parts!!

⇒Very hard to optimize!!!

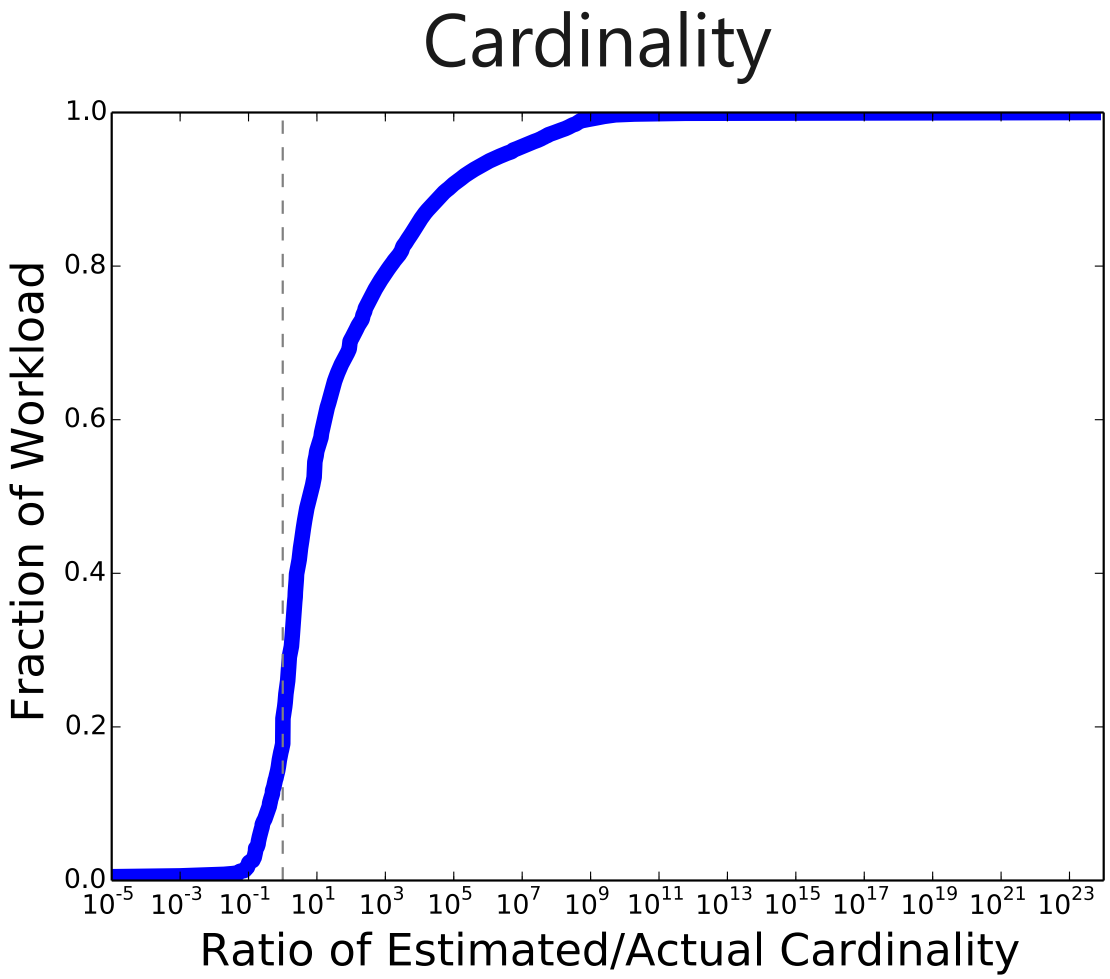
⇒Lack of expertise; DBAs!!!!

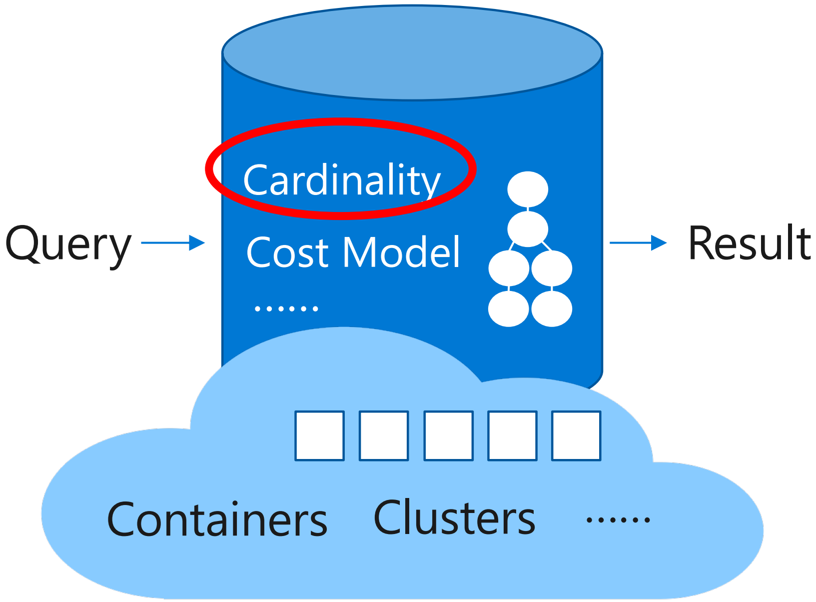
⇒Tough cloud developer life!!!!!



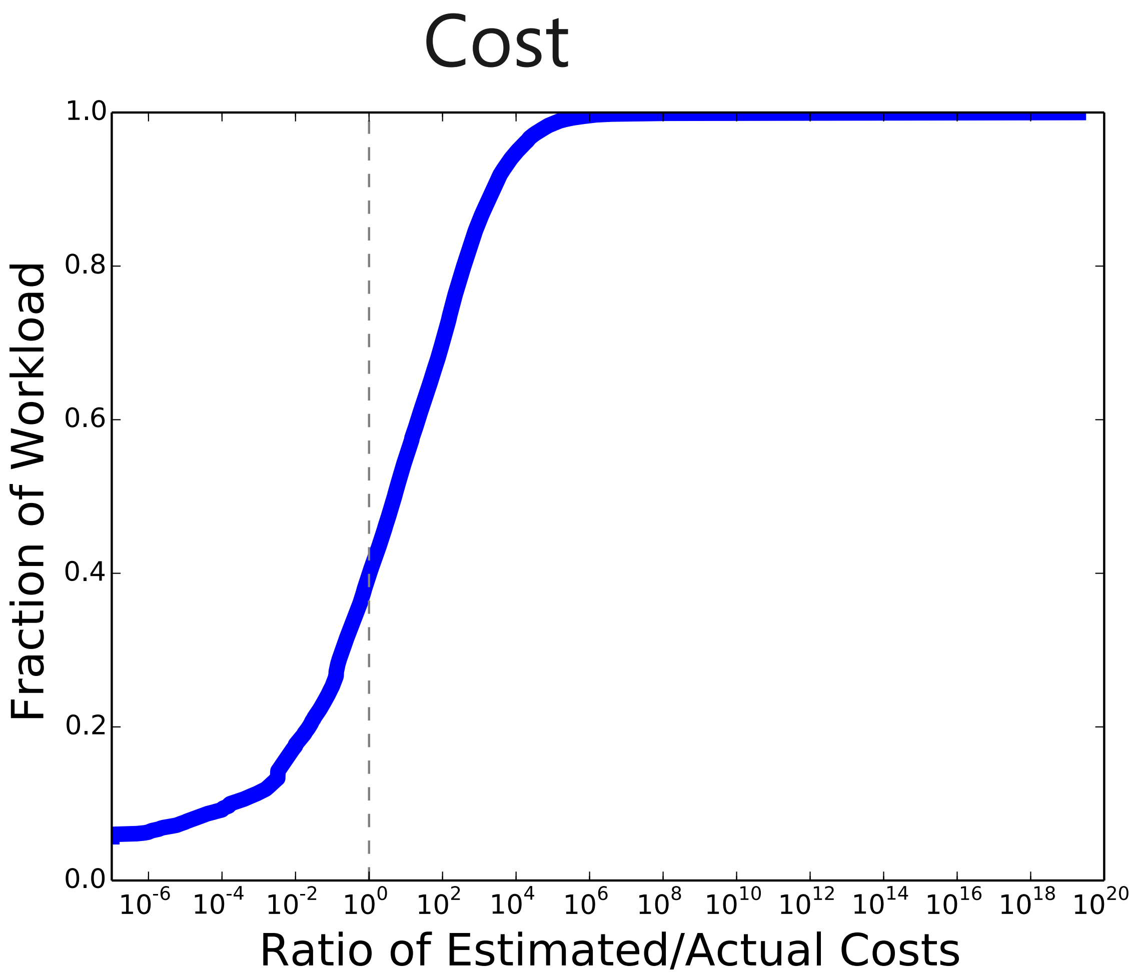
SCOPE

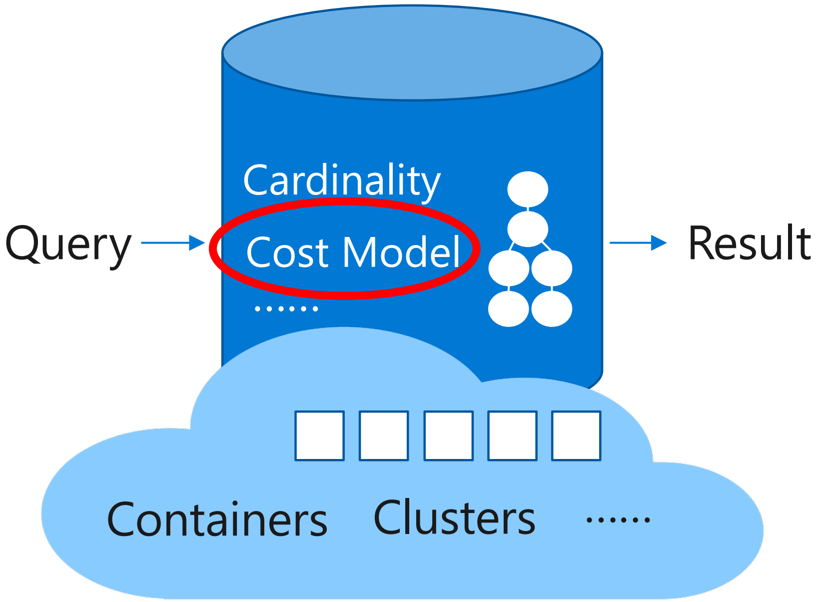
Current state of Key Decisions in SCOPE



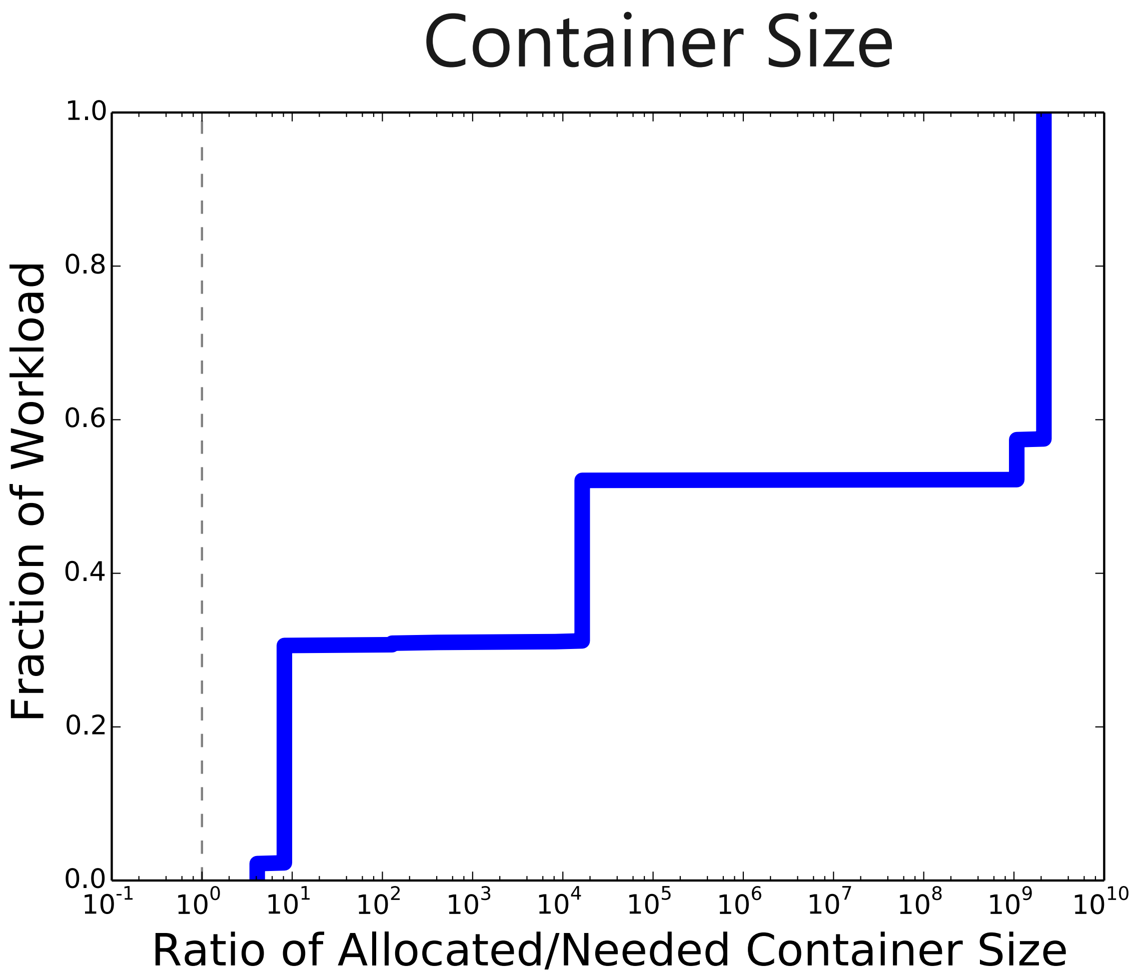


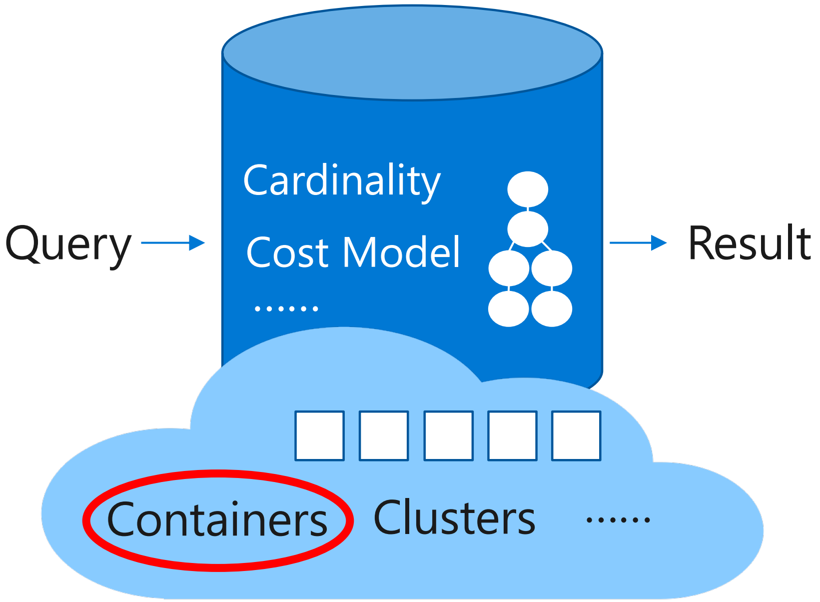
Current state of Key Decisions in SCOPE



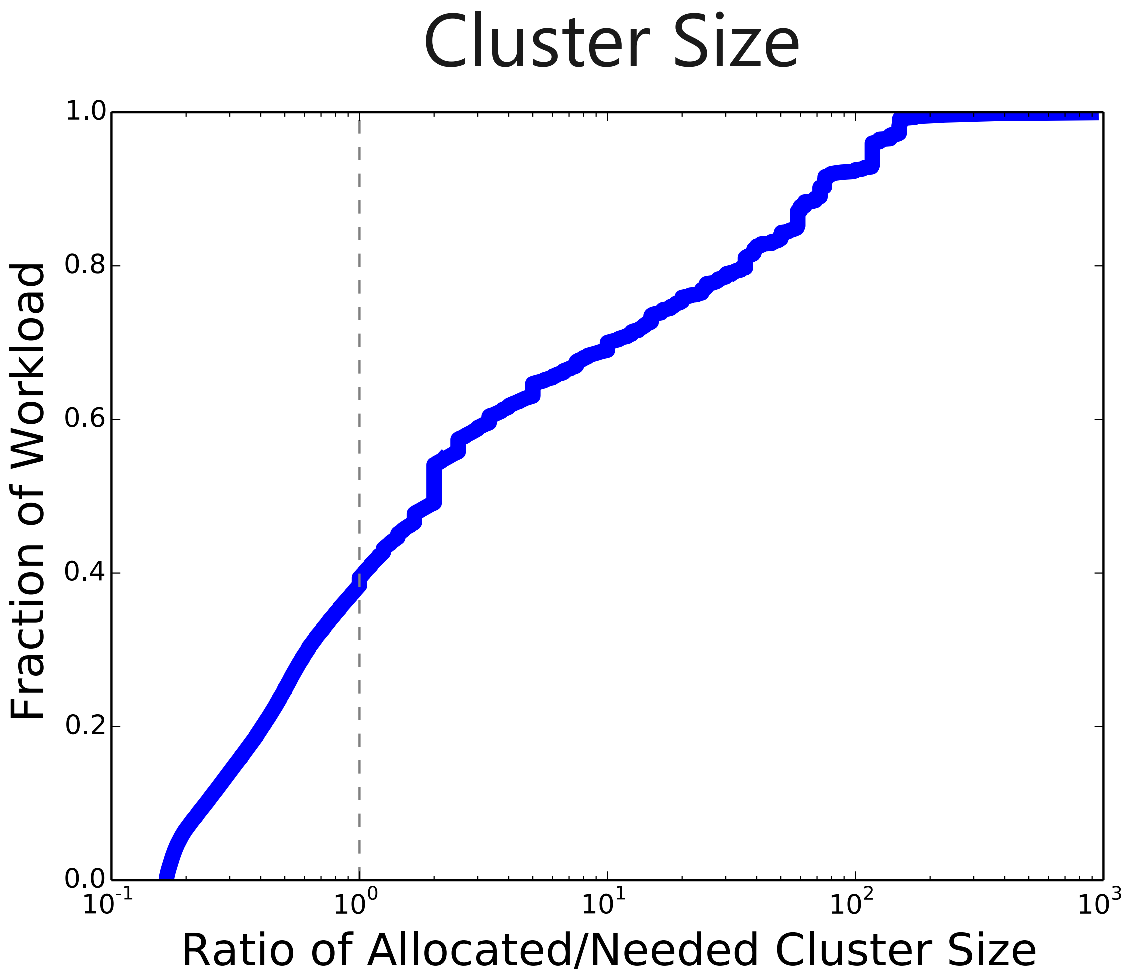


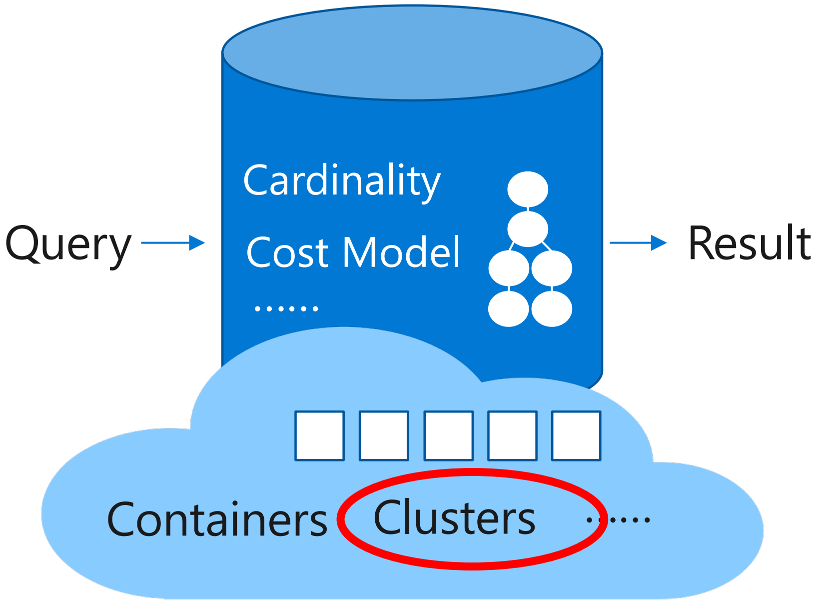
Current state of Key Decisions in SCOPE





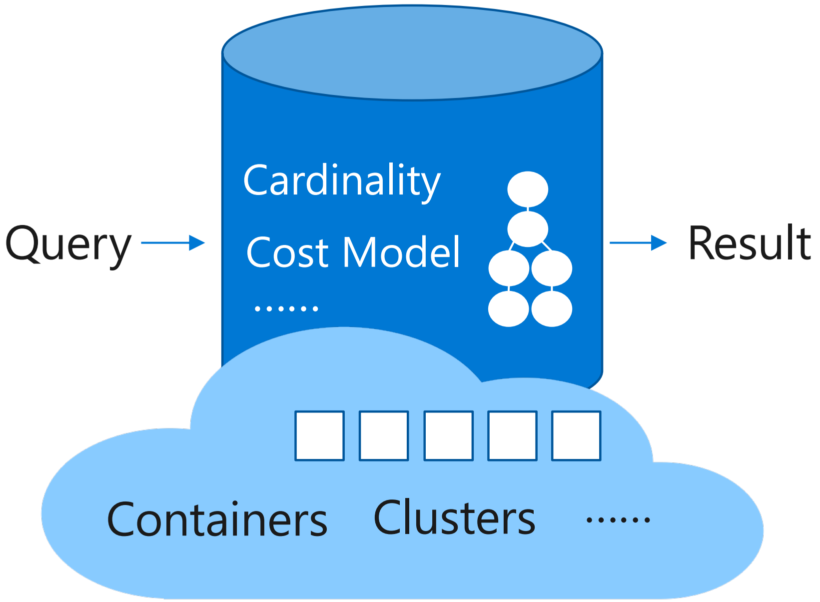
Current state of Key Decisions in SCOPE





Workload Challenges in SCOPE

- Large DAGs
  - 20% jobs have >50 operators; 3% have >500
- Large #tasks
  - 50% jobs have >100 tasks; 10% have >10K
- Structured + Unstructured data
  - 40% jobs have unstructured inputs
- Shuffle more important than join
  - 5x more shuffles than joins
- Optimizing data movement
  - 66% jobs have shuffle; 17% have >10 shuffle/job
- User defined operators
  - 40% jobs with table UDFs; 15% with scalar UDFs




BIG DATA
Workloads
Query Optimizer

Large workloads =>Learn It Up!

Workload ⇨

scikit learn

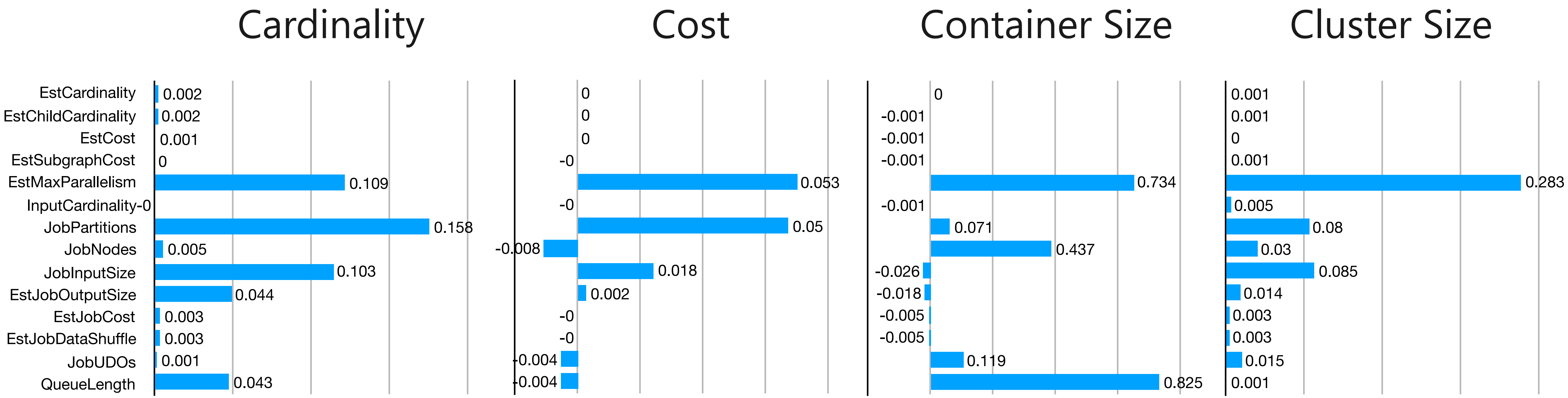
ML.NET

PyTorch

TensorFlow

⇨ Global Model

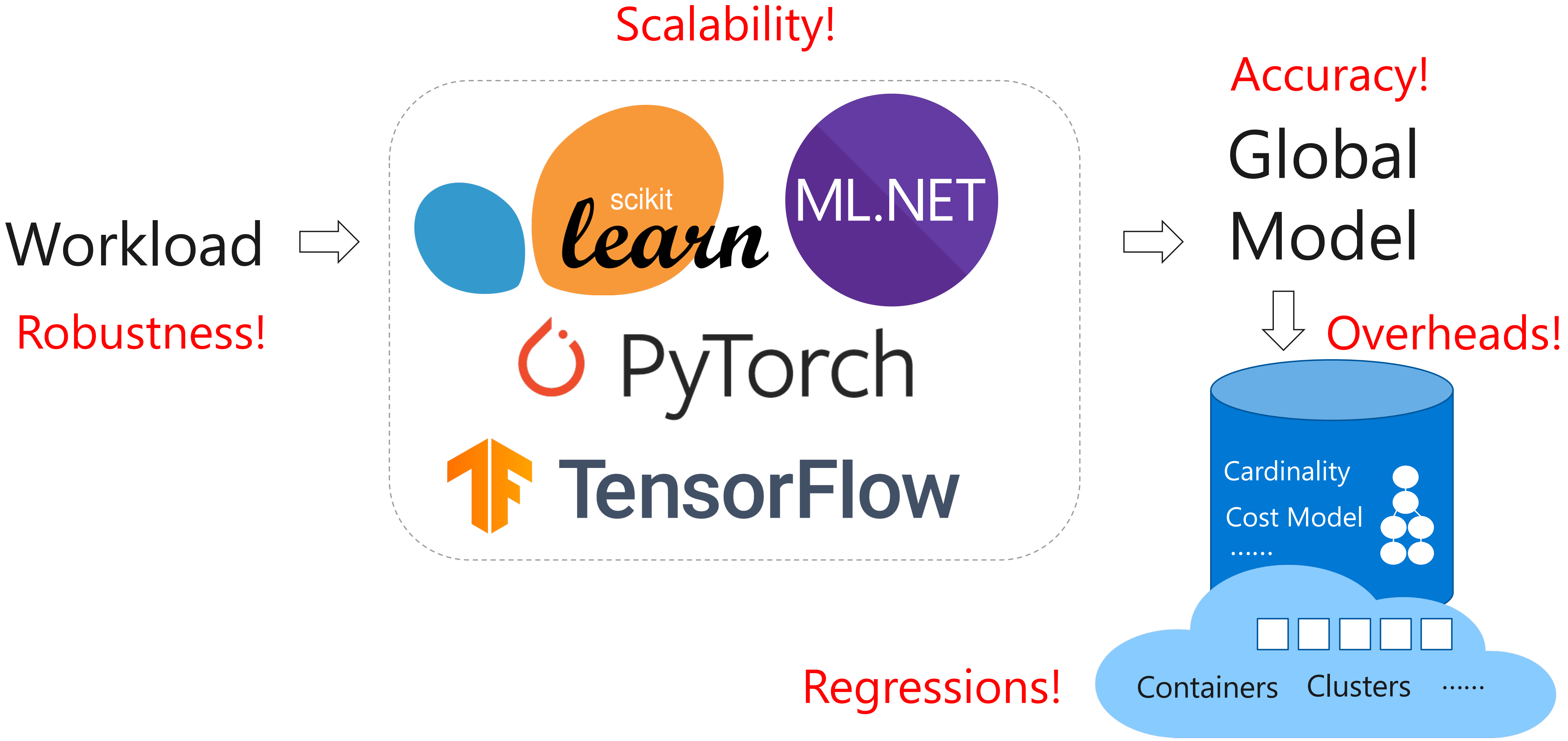
Global single-attribute correlations



- Workload diversity, complexity, evolution!

.... Oops!

Learning Challenges

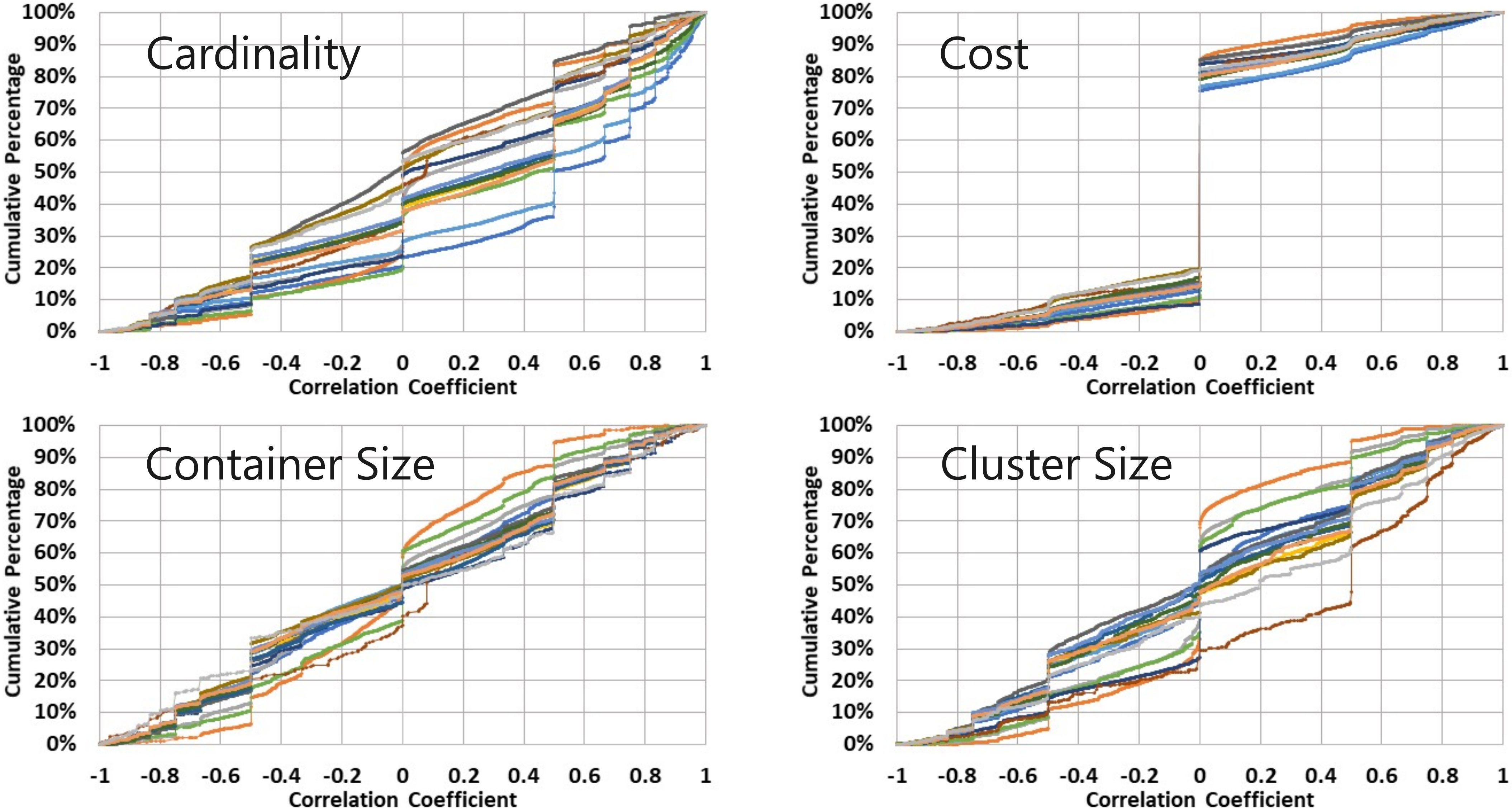



EVERGREEN
EVER GIVEN
PANAMA
MICROMODELS

A fine-grained learning approach

- Characterize workloads into smaller subsets
  - Identify and tag internal states as seen by the optimizer
  - User them to characterize later on
- Learn specialized *micromodels* for each subset
  - Targeted learning and feedback
  - Divide and conquer to manage the cloud complexity

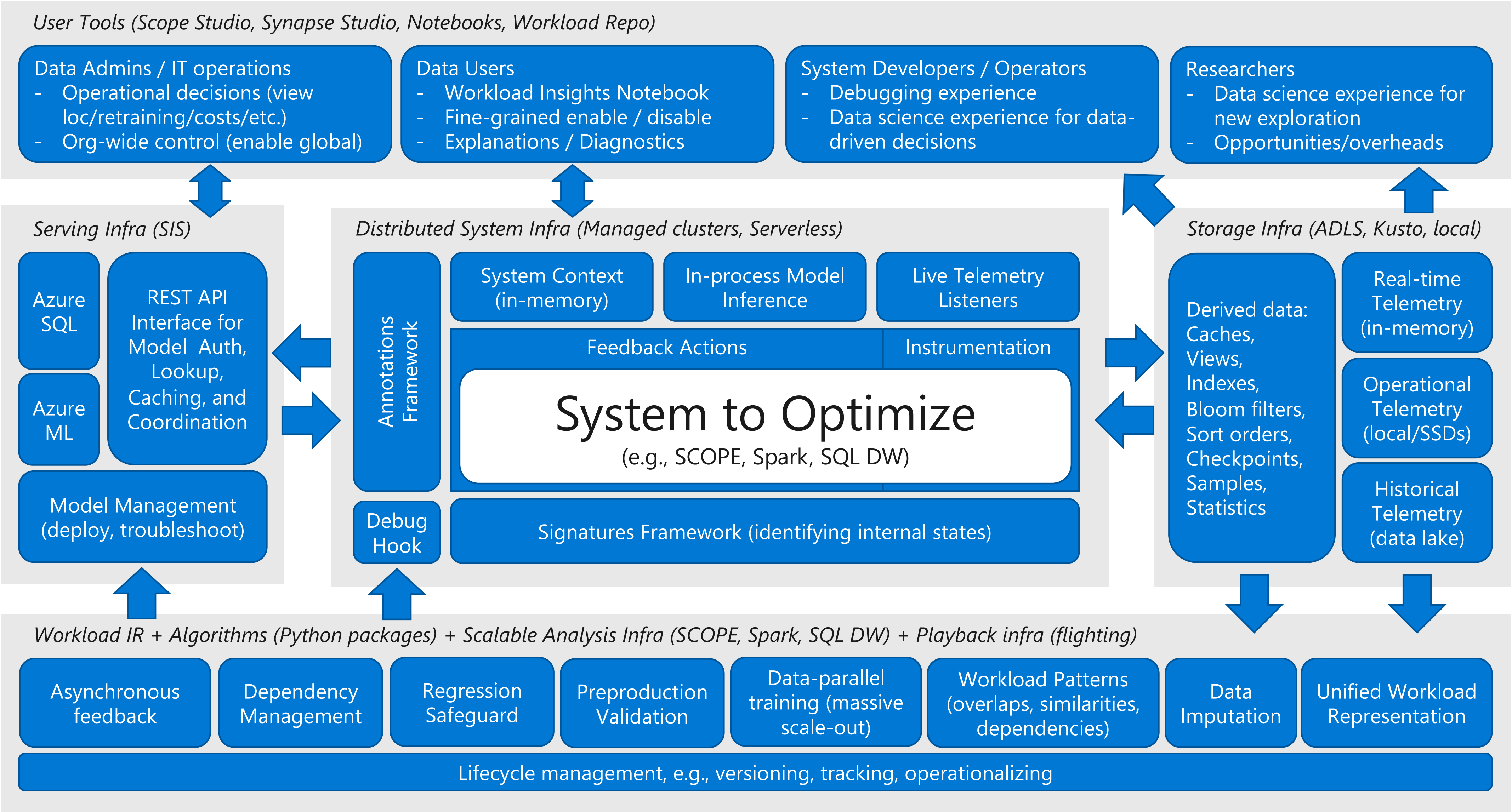
A fine-grained learning approach



Building Production Confidence

- Preproduction Validation
  - Identify the workload subset to experiment on
  - Run and compare performance before/after
- Avoiding regressions
  - Customer expectation: better or same performance
  - Latency, total processing time, resources, etc.
- Dependencies
  - What system version did the workload came from?
  - What other models does a learned model depend on?
- Tracking
  - When was it last trained?
  - What input was it trained on?
  - Can we purge the models trained on a given input?
- Retraining
  - What is the right retrain interval?

Peregrine: more detailed view



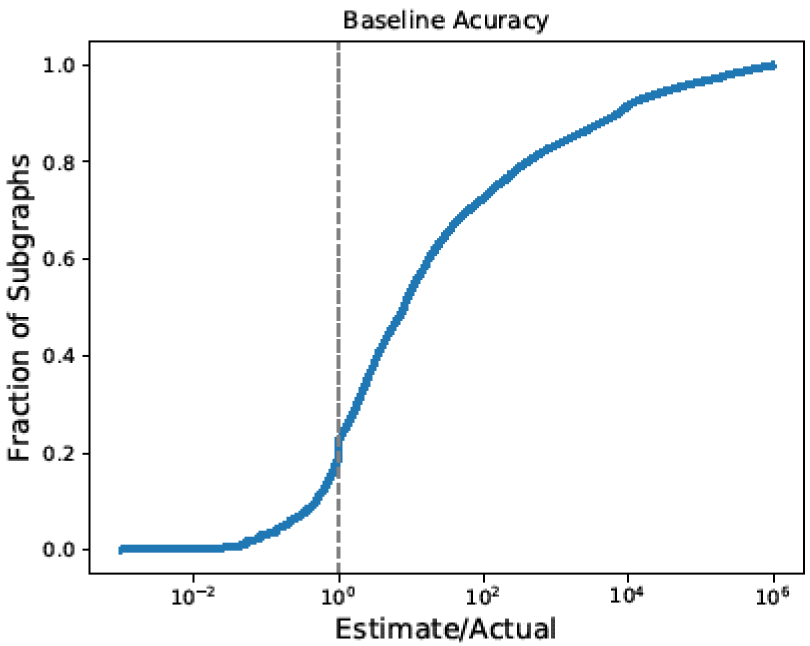
CardLearner on Production Workloads

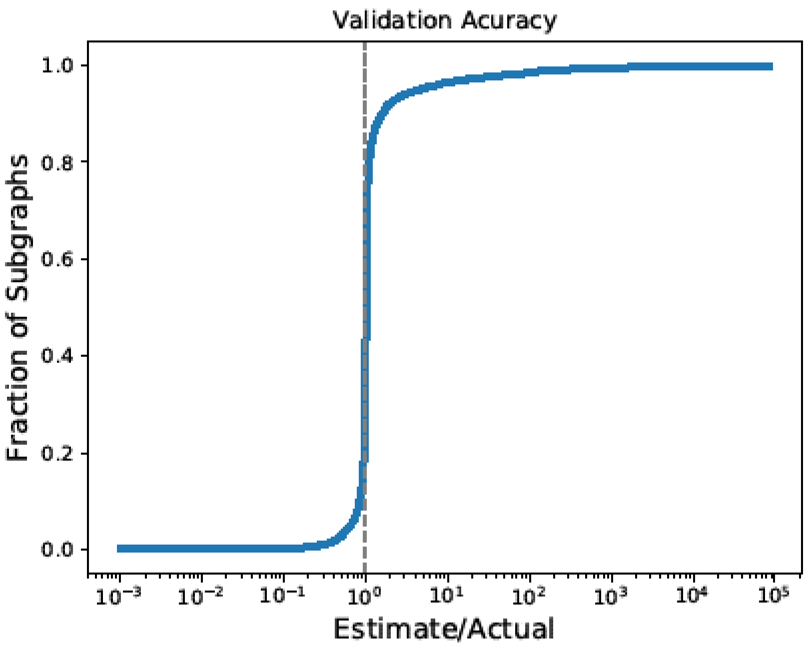
- Training
  - 6 day of workload (~564K jobs)
  - Recompile with the learned card models (~400K models)
  - Identify models causing plan changes (~52K models)
- Validation
  - 1 subsequent day of workload (~93K jobs)
- Filtering good models
  - Avg. Baseline/Actual difference >= 100%
  - Max. Validated/Actual difference <= 100%
  - Avg. Validated/Actual difference <= 10%
  - Max. Validated/Predicted difference <= 1%
  - ~10K models

Training: ~6 hours (200 containers)
Validation: ~1hour (200 containers)
Cumulative Model Size: 1.5MB

Towards a Learning Optimizer for Shared Clouds . Chenggang Wu, Alekh Jindal, Saeed Amizadeh, Hiren Patel, Wangchao Le, Shi Qiao, Sriram Rao. VLDB 2019.

Model Accuracy

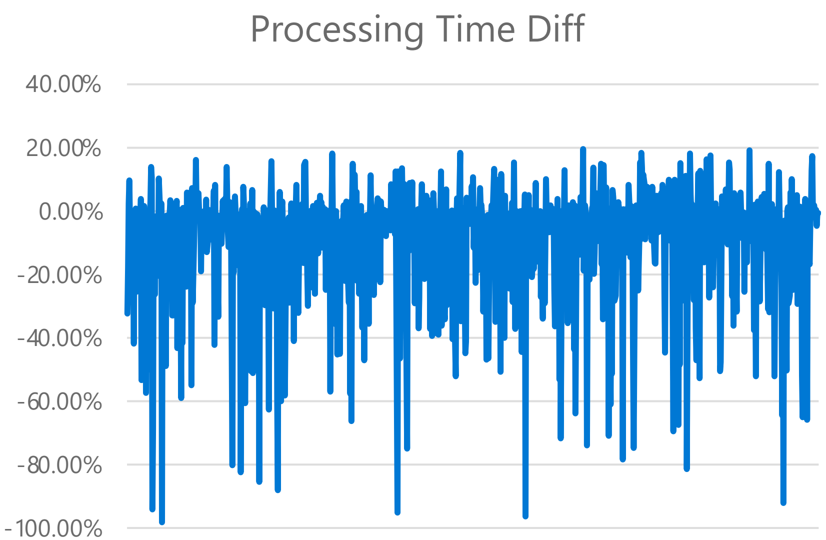


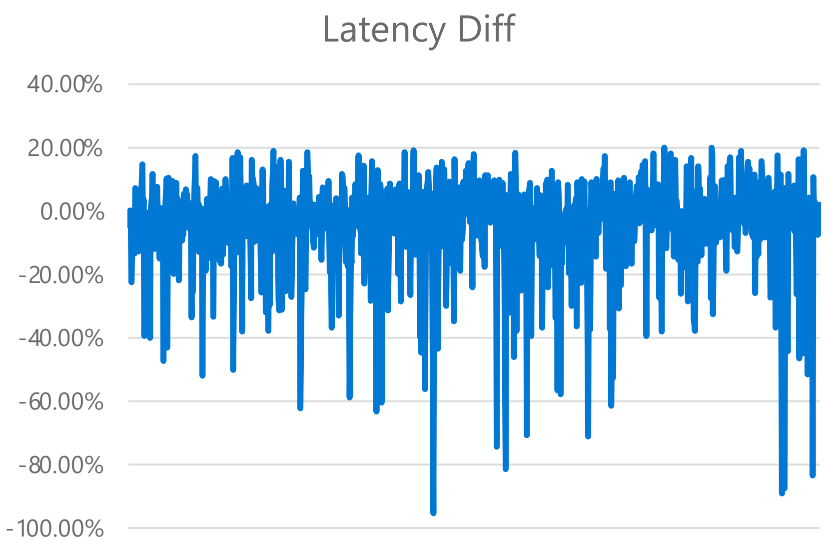


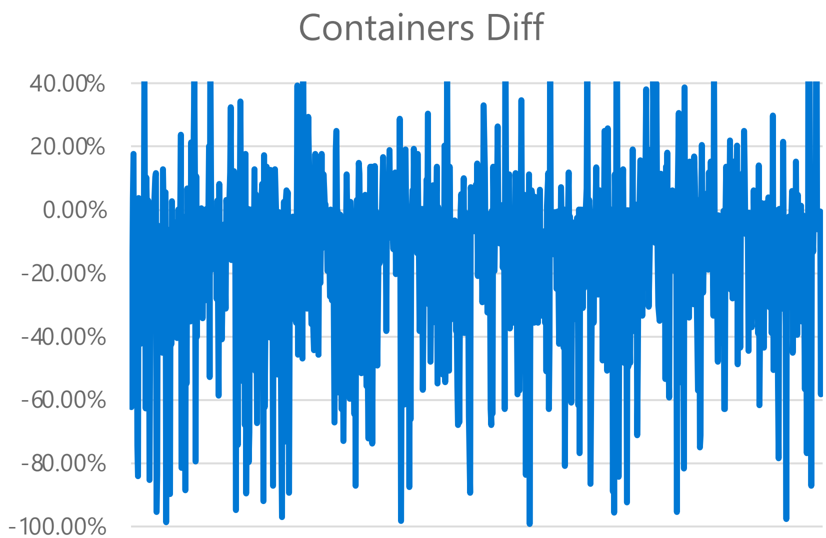
- 153M subexpressions
- 95th percentile Error
  Baseline: 465711%; CardLearner: 1%

Pre-production Experiments

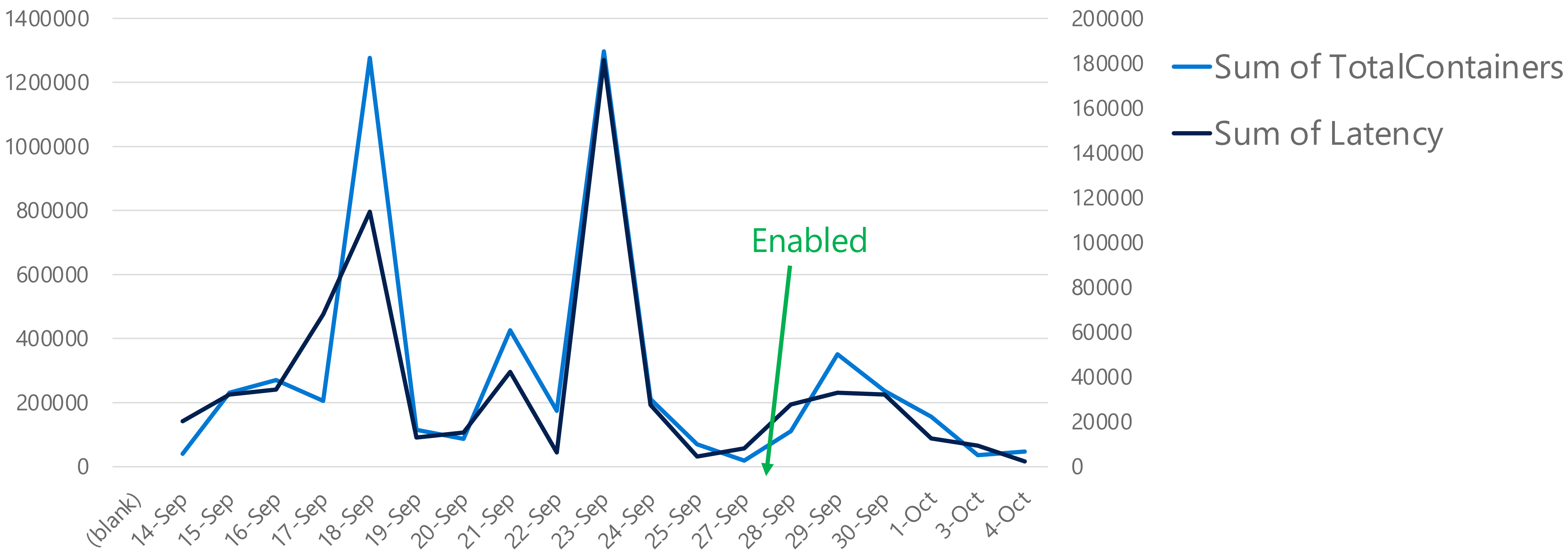
- 20K jobs could use the feedback
- 2518 pipelines flighted, 1282 with plan changes, 41 regressed (excluded)
- For 1241 pipelines, ~12K jobs/day
- Avg. improvement: 6.41% latency, 6.90% processing time, 8.29% containers







Production Deployment



Summary

- Cloud query engines have grown very sophisticated
  - => not easy to optimize
- We can leverage machine learning over massive cloud workloads
  - => not easy to build global models
- We present a fine-grained learning approach:
  - Characterize workloads into subsets
  - Learned micromodels over each subset
  - Easy to scale training to very large workloads
  - Smaller, cheaper models to score within the query engine
- We have built and deployed learned cardinalities in SCOPE
  - Large number of steps to avoid performance regressions
- Long journey: 2017 (intern)-> 2018 (integration)-> 2019 (perf)-> 2020 (deploy)

Microsoft