Towards a Benchmark for Learned Systems

Laurent Bindschaedler
Andreas Kipf
Tim Kraska

Ryan Marcus
Umar Farooq Minhas

Video of Presenter

MIT CSAIL

DSAIL
Data Systems and AI Lab

Microsoft

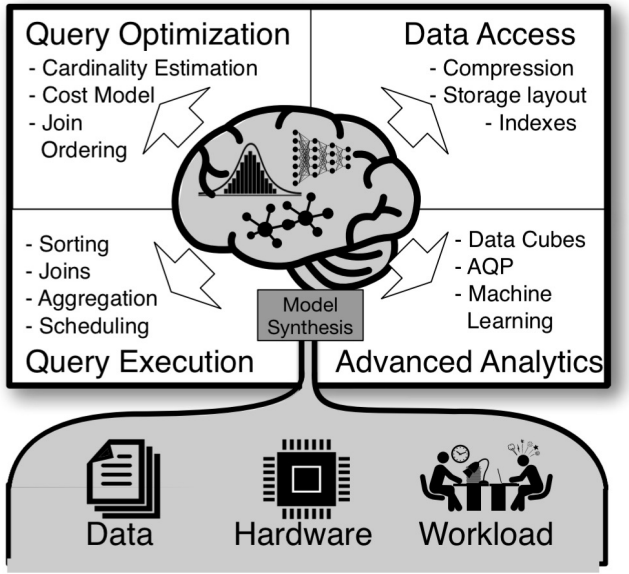
Massachusetts Institute of Technology

Learned Data Management Systems

Leverage machine learning techniques to
- Synthesize optimized data structures or components
- Help existing components perform better
- Tune configuration knobs

⇒ Adaptability to changing conditions

⇒ Instance-optimization or specialization



Learned systems show great promise in the lab
- Orders-of-magnitude performance improvements
- Unmatched adaptability and capacity to specialize

Still unclear how well they perform in production…

Video of Presenter

How to Evaluate Learned Systems Benefits?

Evaluating learned systems introduces many new challenges

- Fixed synthetic data/workload are too easy to “learn”
- Average performance insufficient to understand adaptability
- Overfitting to the benchmark
- Need to incorporate training cost
- Compare widely different designs

Addressing these challenges requires

- Realistic (and hard to predict) datasets and workloads
- New metrics to capture adaptability, specialization, …
- Comparing system cost with human costs (DBA, …)

Video of Presenter

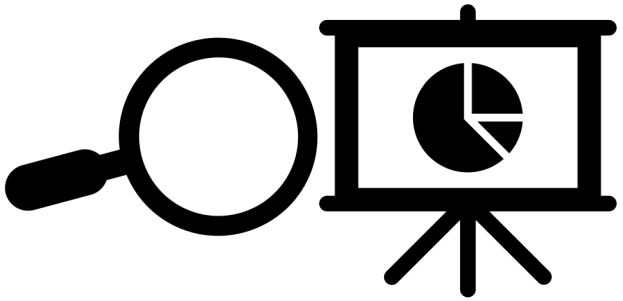
We need a Benchmark for Learned Systems!

Better benchmarking is the best way to

- Understand and compare learned systems characteristics
- Provide empirical evidence about the benefits of learned systems
- Address concerns about worst-case performance

Our goal is not replace existing benchmarks…

… but rather to complement them



Video of Presenter

Outline

1. Traditional Benchmark Challenges

2. Towards a New Benchmark
   a) Configuration and Execution
   b) Workload and Data
   c) New Metrics

3. Conclusions

Outline

1. Traditional Benchmark Challenges

2. Towards a New Benchmark
   a) Configuration and Execution
   b) Workload and Data
   c) New Metrics

3. Conclusions

Issues with Traditional Benchmarks

Traditional benchmarks measure average performance

- Synthetic workload representative of real-world application
- Measure end-to-end performance
- (Often) forbid leveraging workload knowledge

TPC®

Y! CSB

⇒ Insufficient/incompatible with learned systems

/!\ Traditional benchmark results remain relevant

Video of Presenter

1. Fixed Workload/Database are Easy to Learn

Since learned systems are adaptable, it is very easy to

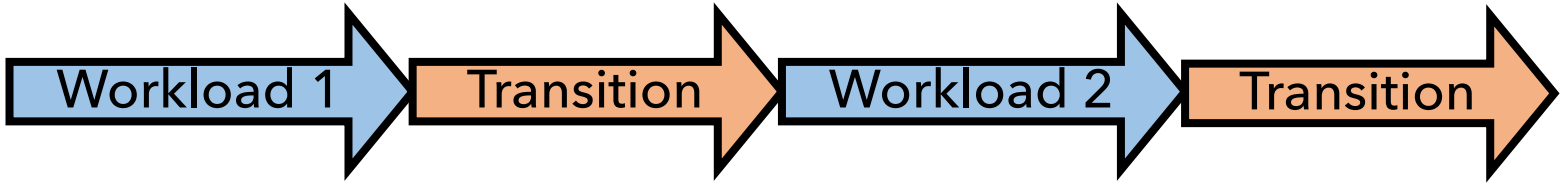
- Learn the characteristics of stable or predictable workloads
- Overfit to a fixed data distribution

⇒ Unfair advantage when comparing with traditional systems

⇒ Uninteresting results

Ideally: benchmark exhibits different behaviors

- Single execution run with transitions
- Analyze system behavior during transitions



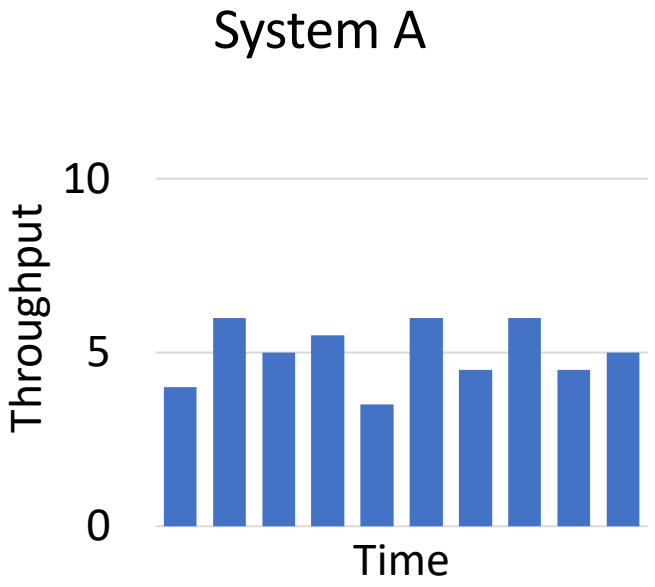
Video of Presenter

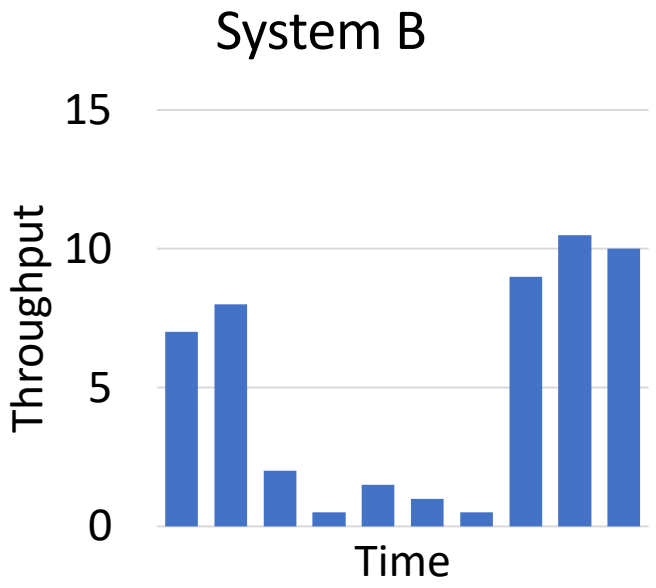
2. Average Metrics Fail to Capture Adaptability

Traditional benchmarks focus on average throughput

- Acceptable when data/workload fixed…
- … but average performance hides too much information

We cannot compare systems based on average performance





Video of Presenter

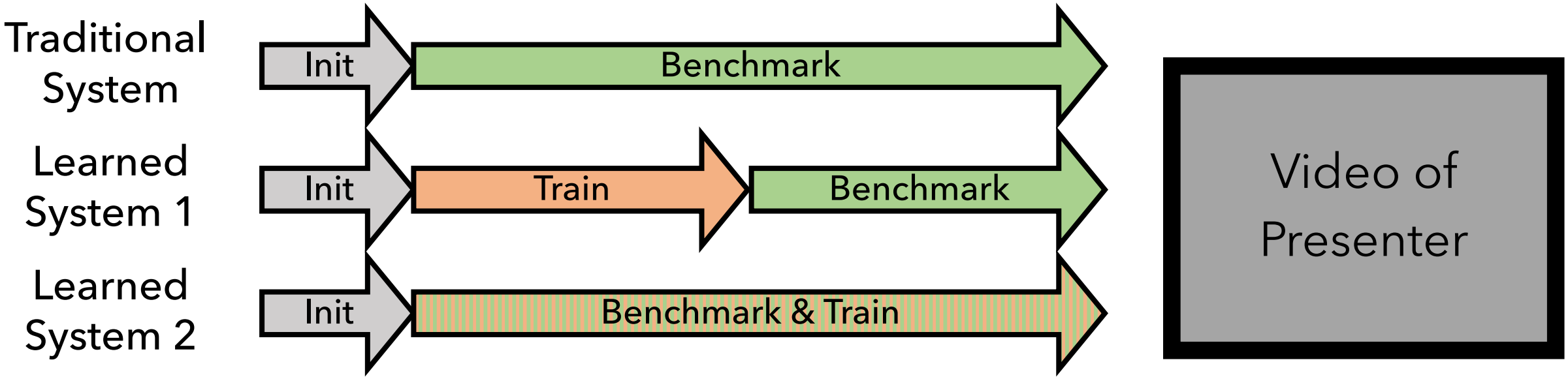
3. Shoud Not Ignore Model Training

Traditional benchmarks do not account for model training

- "Load & run for X minutes" execution model
- System performance reported for entire execution

Learned systems require training (offline or online)

- More training improves performance at the cost of extra overhead

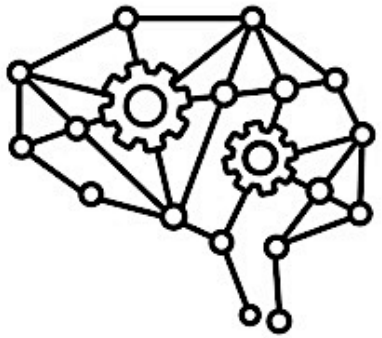


4. Cannot Ignore the Human Cost Anymore

Total Cost of Ownership (TCO) is largely ignored currently

However, the main goal of learned systems is to reduce the TCO!
⇒ TCO must be taken into account
⇒ Need to compare learned systems savings with “human” costs

How to estimate the TCO of a learned system?



vs.

Video of Presenter

Outline

1. Traditional Benchmark Challenges

2. Towards a New Benchmark
   a) Configuration and Execution
   b) Workload and Data
   c) New Metrics

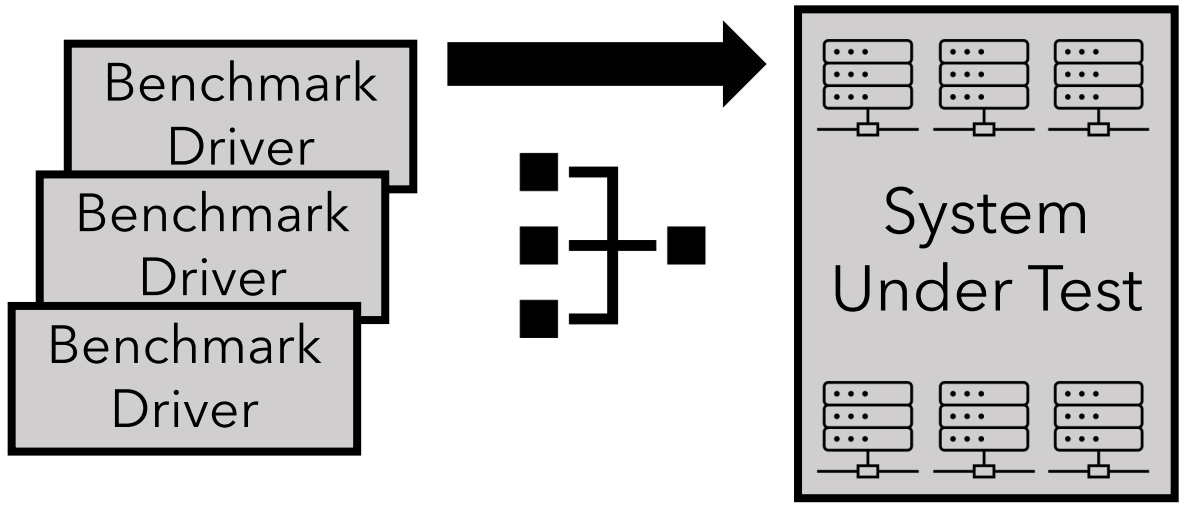
3. Conclusions

Configuration and Execution

Typical configuration (e.g., scale factor) + learning-specific config:

- Configuring mix of data/workloads
- Configuring workload transitions (transition duration or retraining)
- Training time or online training overhead

Execution should incorporate (and measure!) training



Video of Presenter

Realistic Workload and Database

Clear need for datasets and workloads representative of real world

Approach #1: dataset/workload "grading" tool

- Evaluates the relevance of a given dataset
- Favor datasets exhibiting skew or varying query load

Approach #2: synthetic data/workload generation

- Privacy-preserving techniques
- Machine learning techniques

Video of Presenter

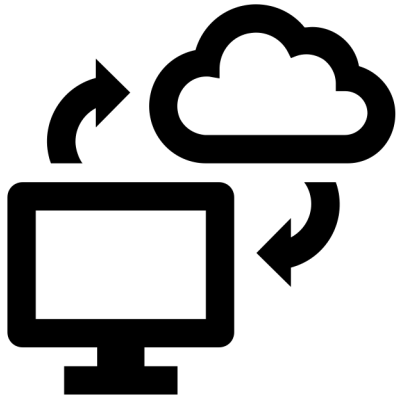
Benchmark-as-a-Service

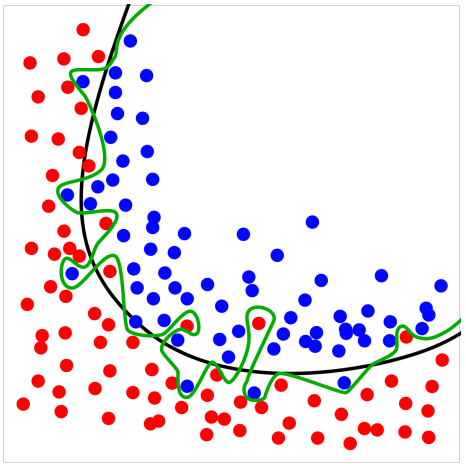
Deploy the benchmark as a cloud service

- Easy to evaluate systems in a unified environment
- Execute workloads and datasets that may not be public
- Execute “hold-out” workload to measure out-of-sample performance

BaaS could complement the benchmark

e.g., BaaS run required for inclusion in official benchmark results

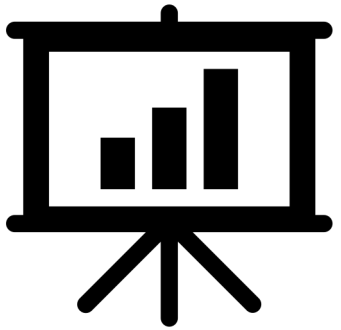




Video of Presenter

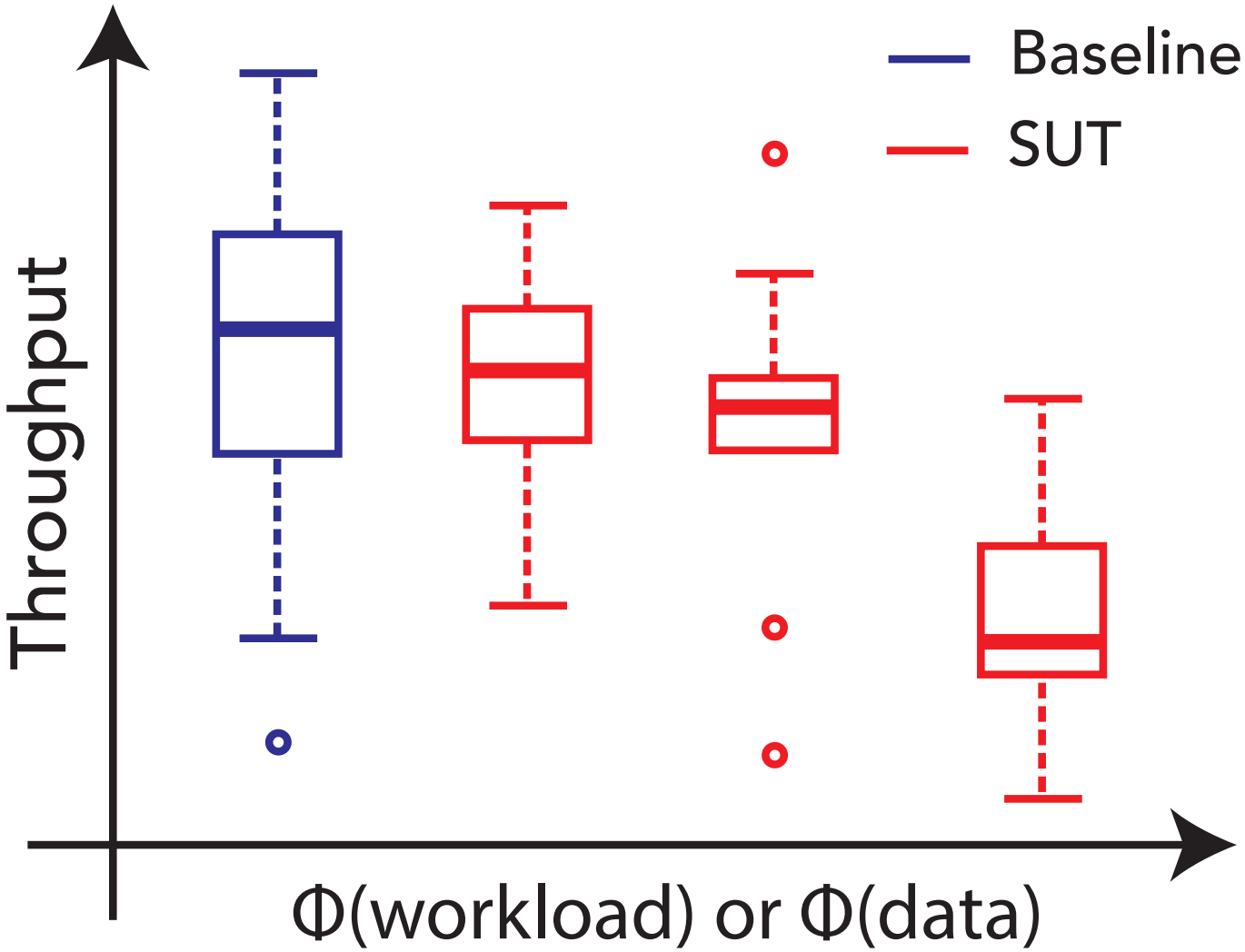
New Metrics

- Throughput per workload or data distribution
- Cumulative queries over time
- Service-Level Agreement violations
- Throughput per cost



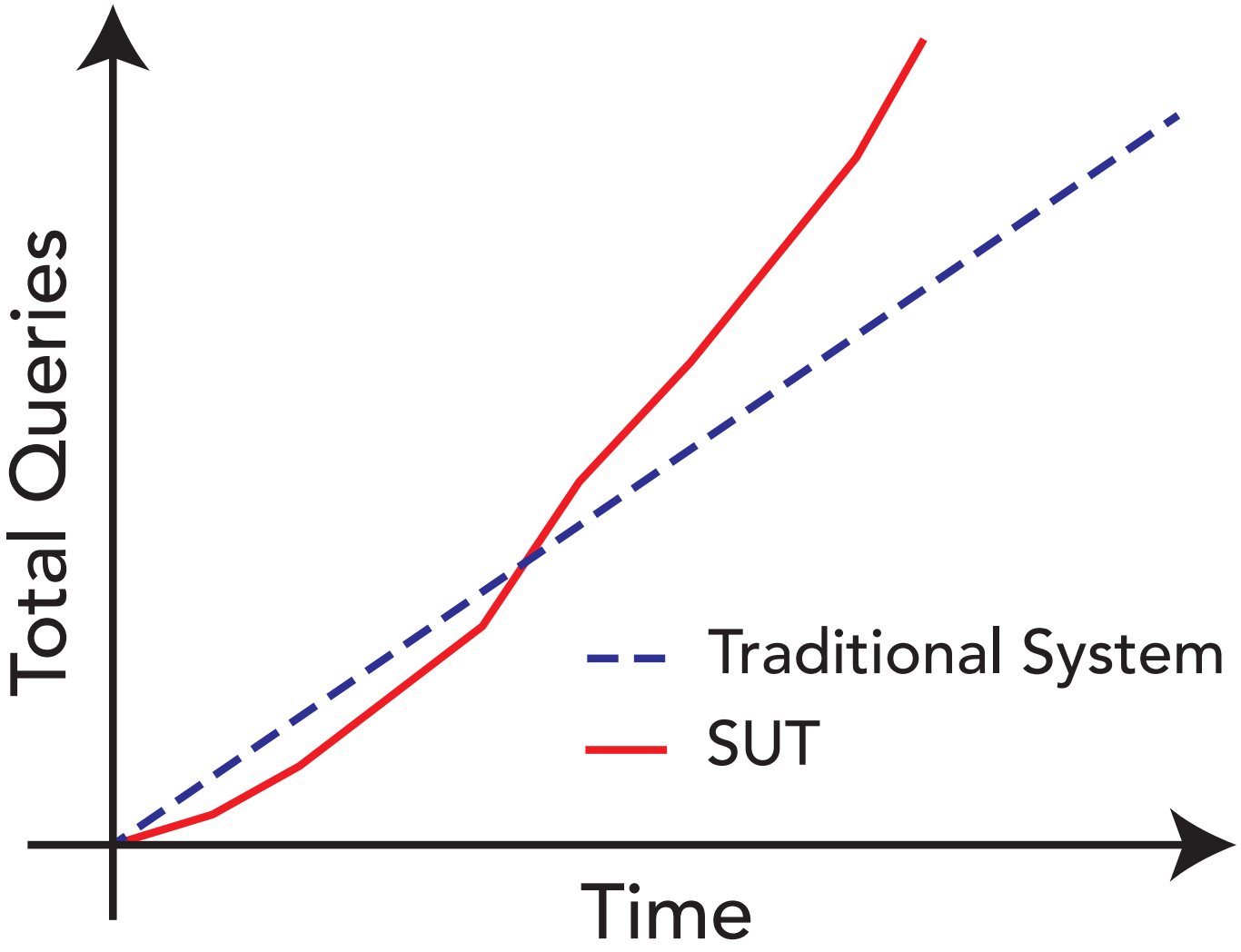
Video of Presenter

Throughput per Workload or Data



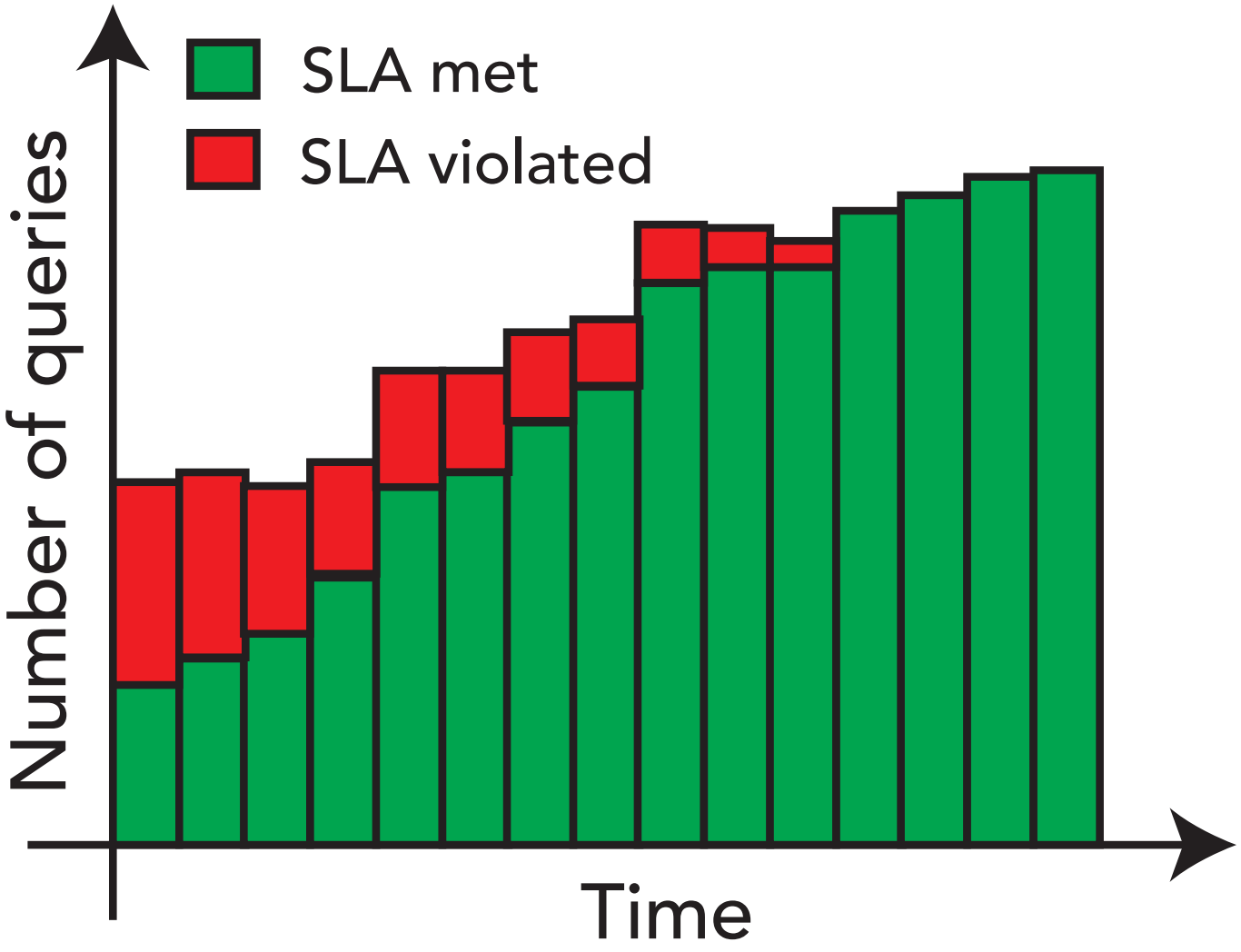
Video of Presenter

Cumulative Queries over Time



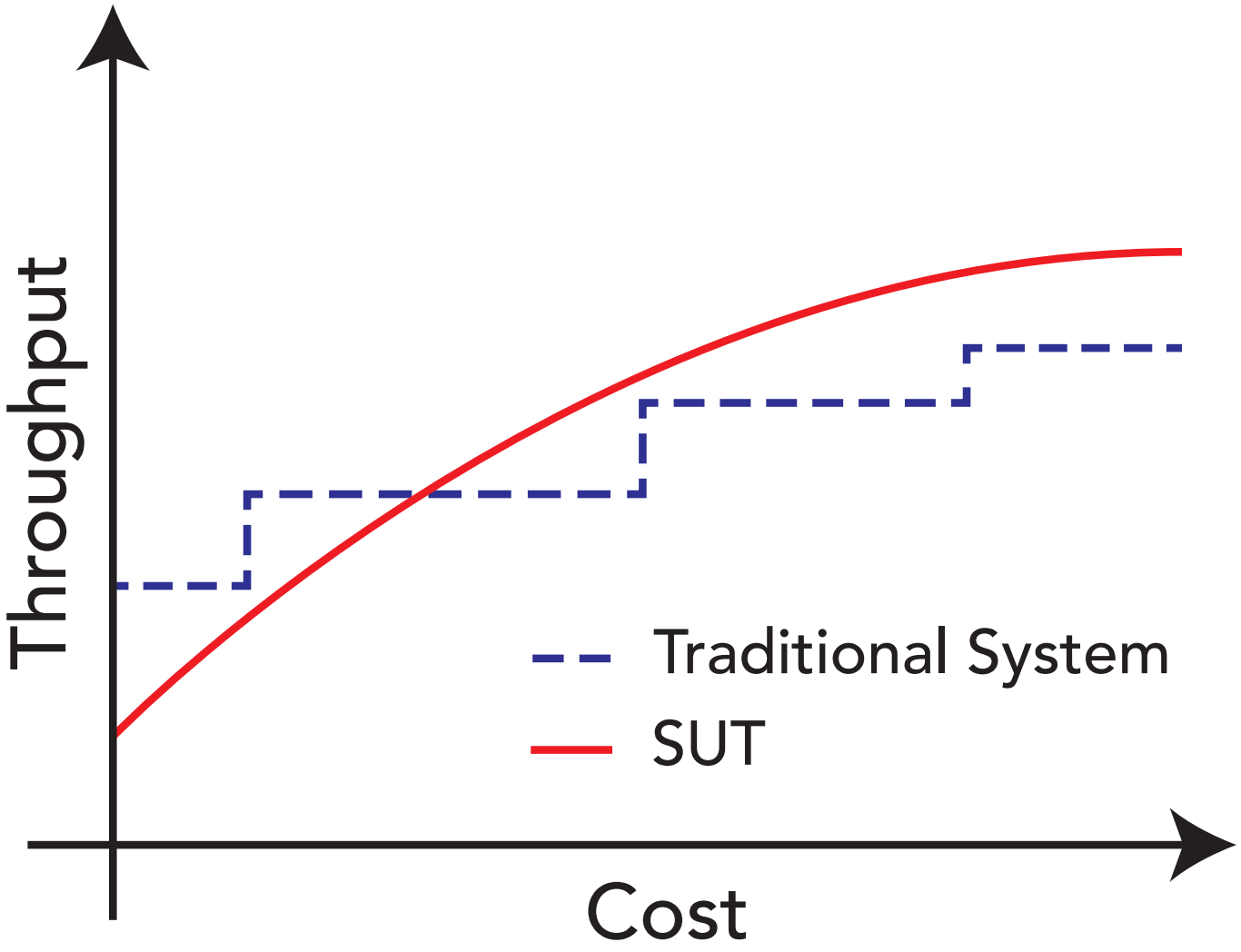
Video of Presenter

Service-Level Agreement Violations



Video of Presenter

Throughput per Cost



Video of Presenter

Outline

1. Traditional Benchmark Challenges

2. Towards a New Benchmark
   a) Configuration and Execution
   b) Workload and Data
   c) New Metrics

3. Conclusions

Conclusions

Adoption of learned systems will require convincing practitioners

- Better benchmarking is the best way to address their concerns

This paper = preliminary ideas and challenges

Next step: build a first version of the benchmark

binds.ch/tabfls

bindscha@mit.edu

Video of Presenter

MIT CSAIL

DSAIL Data Systems and AI Lab

FNSNF

Microsoft

Google

intel