Benchmarking Query Execution Robustness

Janet Wiener, Harumi Kuno, Goetz Graefe
HP Labs, Palo Alto

LABS hp

Outline

- Problem
- What is robustness?
- Metrics to quantify robustness
- A few measurements

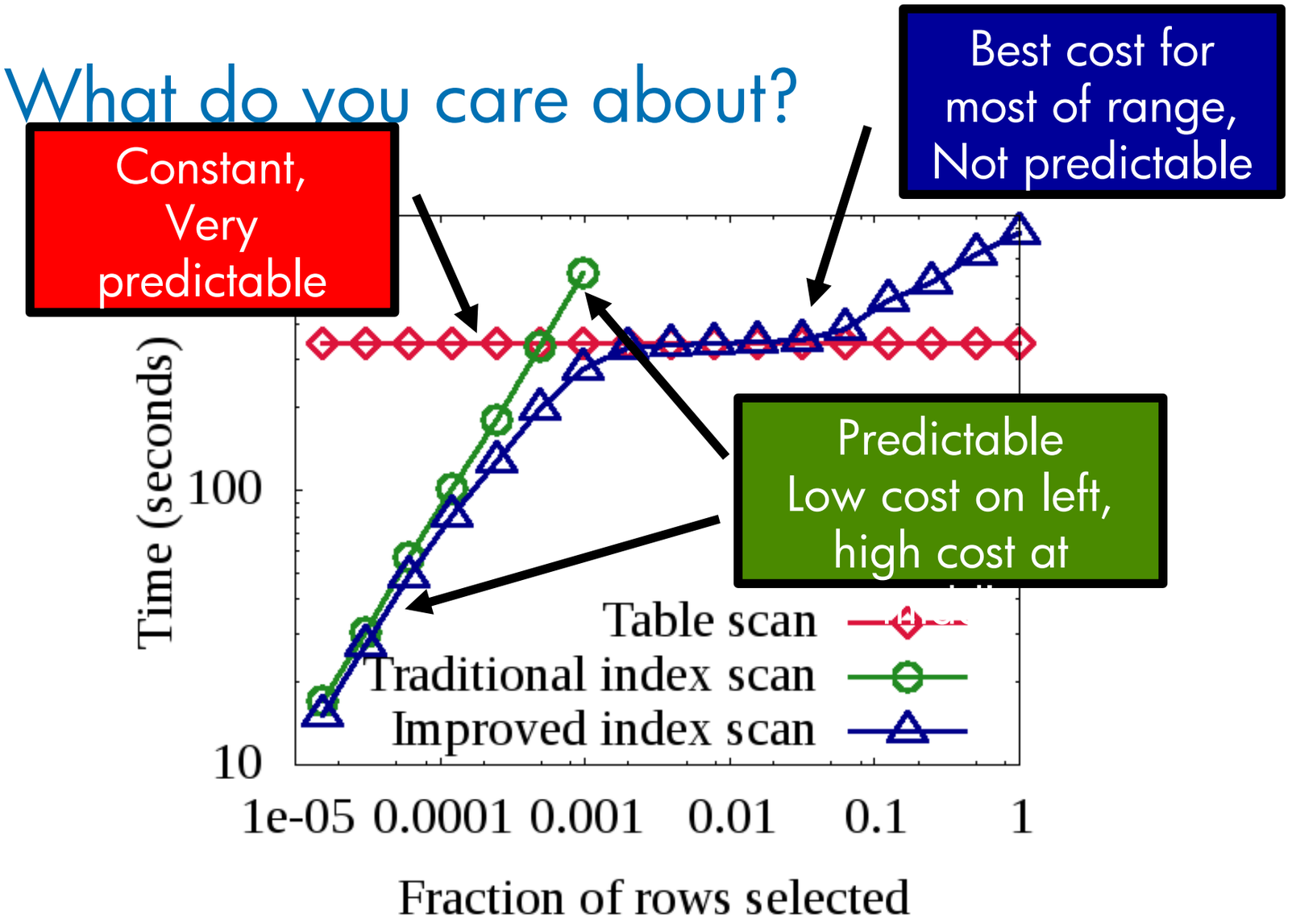
Problem

- Hard to tune database for all possible runtime conditions
- Unexpected conditions becoming more common
- …and lead to unexpected behavior
  - "This query takes 2 hours – except once in a while when it takes 20 hours. Why?"
- Goal: measure potential for surprises
- Ultimate goal: eliminate surprises

Robustness perspectives

- Query optimizer
  - Choose good plan for expected conditions
- Query executor
  - Process given plan efficiently under different runtime conditions
- Physical database design
  - Choose design that leads to robust performance
- Workload management
  - Characterize how query performance surprises affect overall database performance

What do you care about?



LABS hp

Who can use robustness results?

- Customers
  - Gauge risk of unexpected conditions on performance
  - Choose database with predictable performance
- Software developers
  - Make algorithms more robust
- Hardware vendors
  - Gauge performance with specific hardware
  - Improve sizing, provisioning tools
- Tool vendors
  - Assist in physical database design

Robustness complements current benchmarks

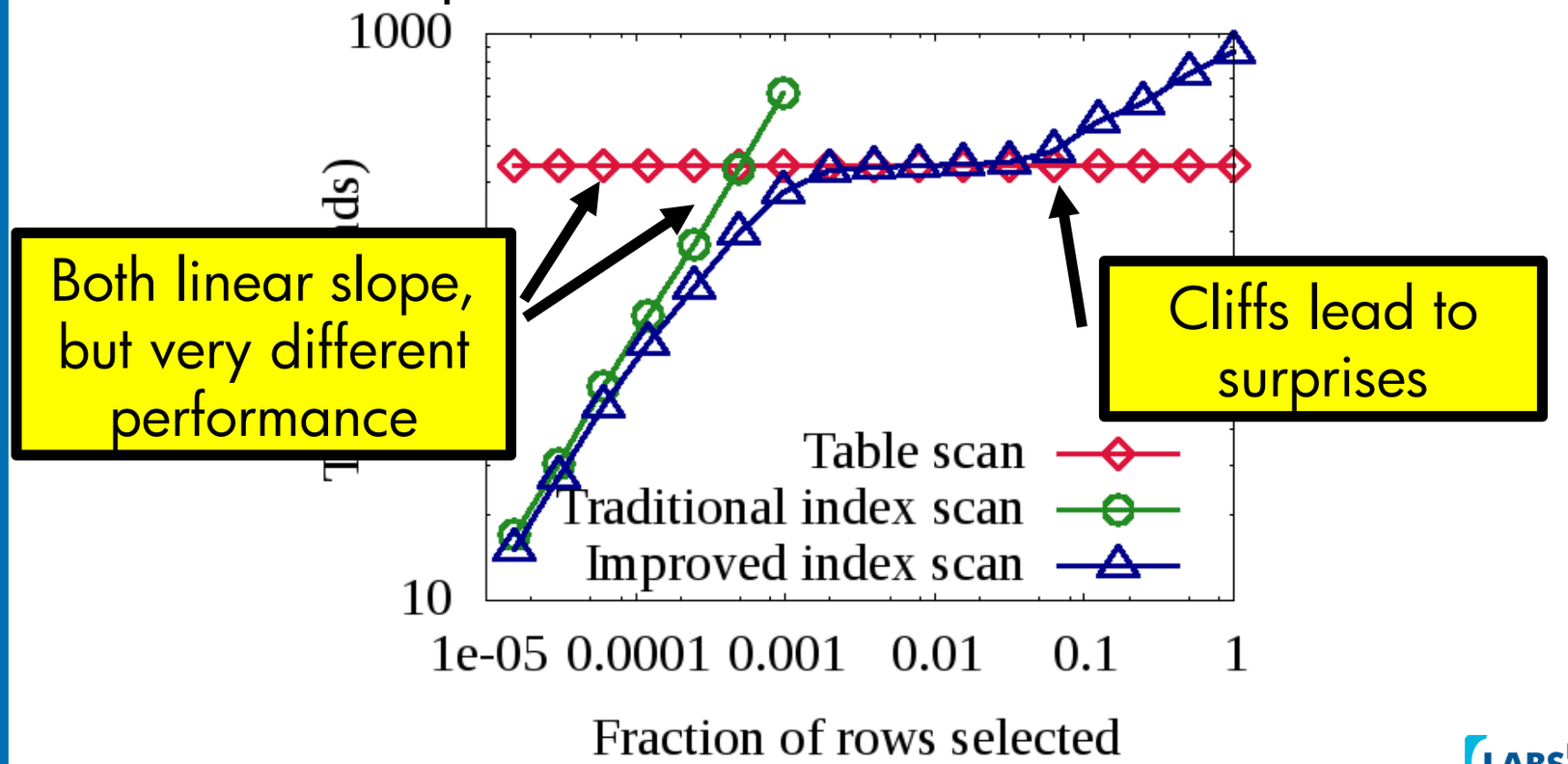
- Current benchmarks measure speed of queries under fixed conditions
  - And report performance or price-performance
- Query execution robustness measures speed of query plans under variety of conditions
  - Force a query plan
  - Isolate compilation vs running time
  - Look at shape of performance curve

Variable runtime conditions

- Data sizes
- Resource availability
  - Memory
  - Buffer pool
  - I/O bandwidth
- Concurrency conflicts
  - Locks

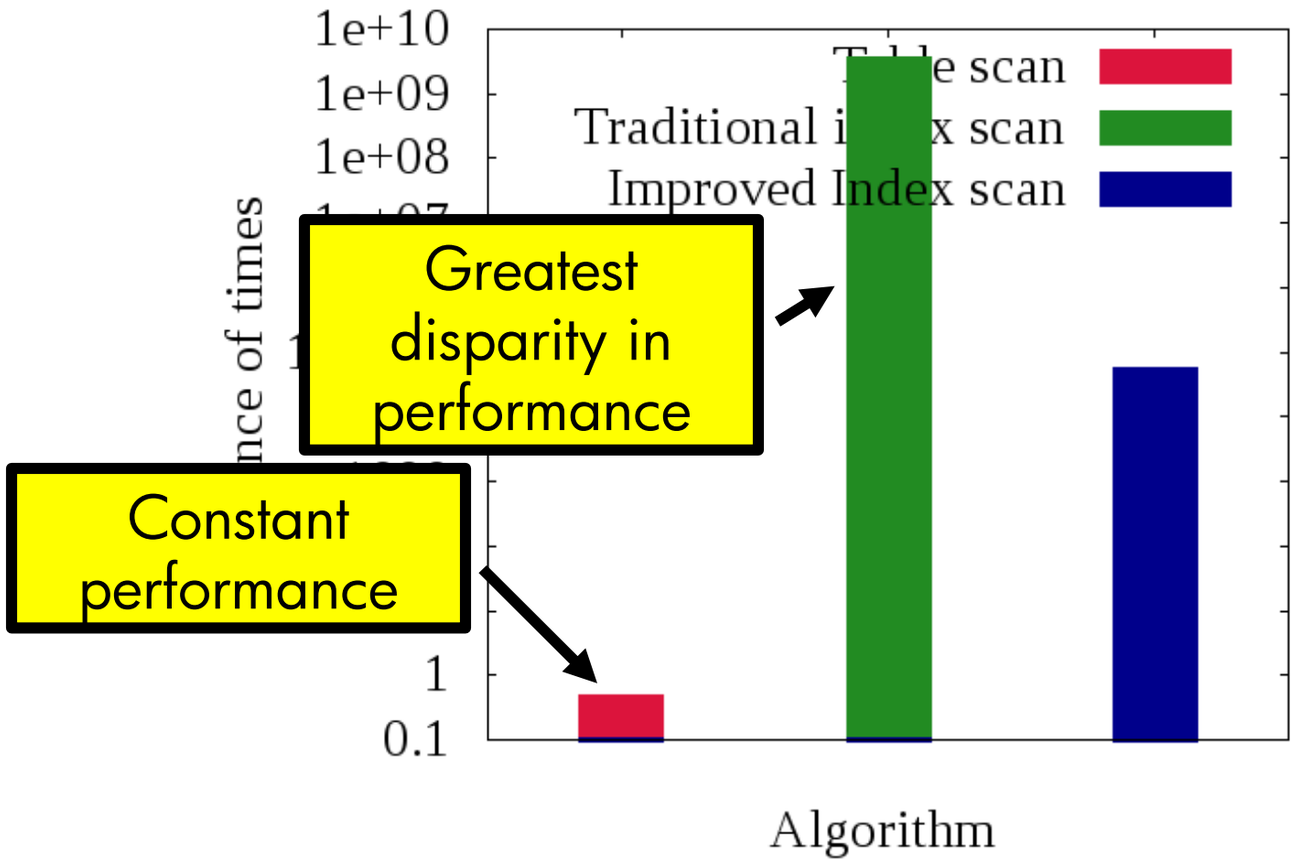
Robustness metrics: Graceful degradation

- How do conditions impact performance?
- Measure slope of curves, cliffs in curves



Robustness metrics: Consistency

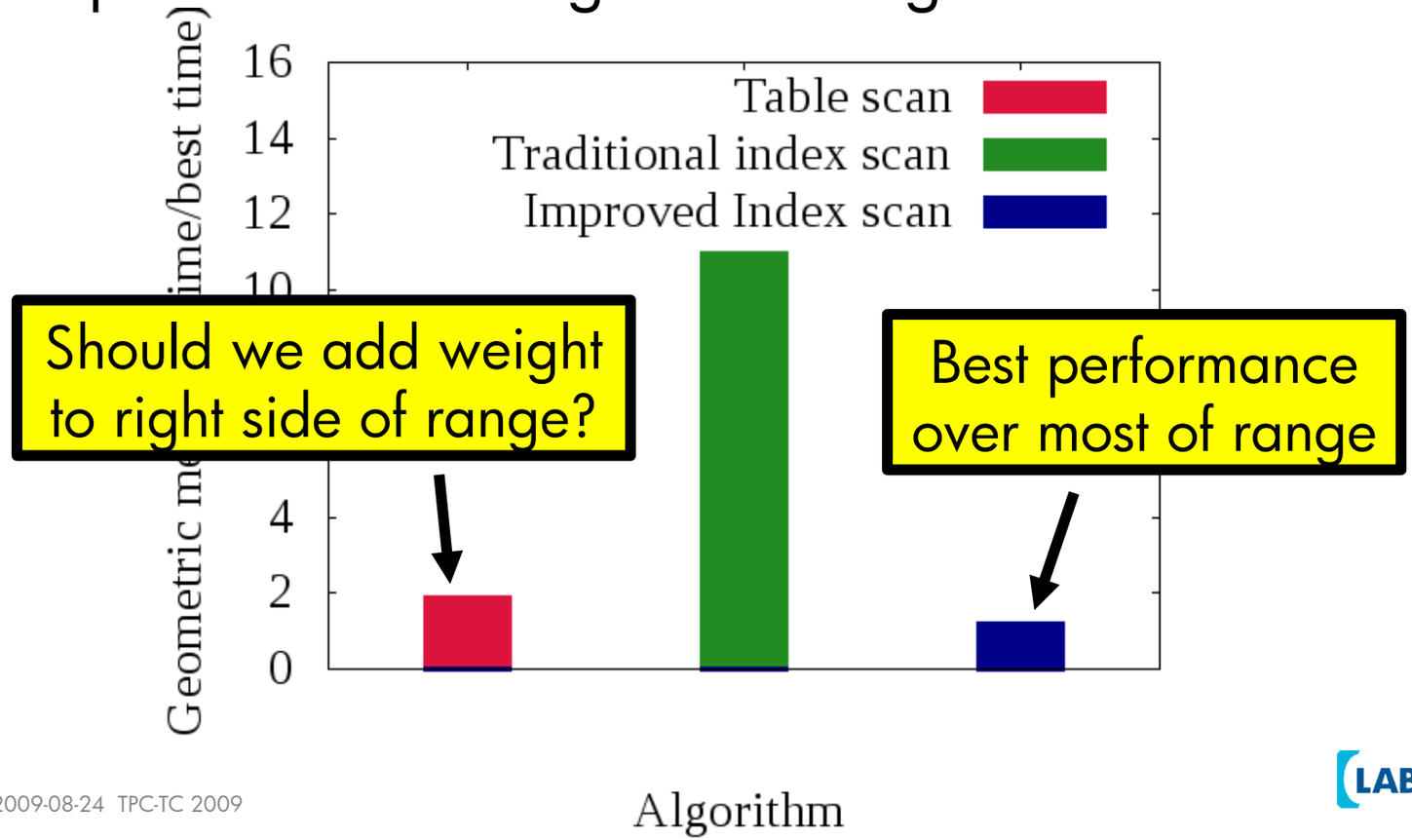
- How does performance vary across conditions



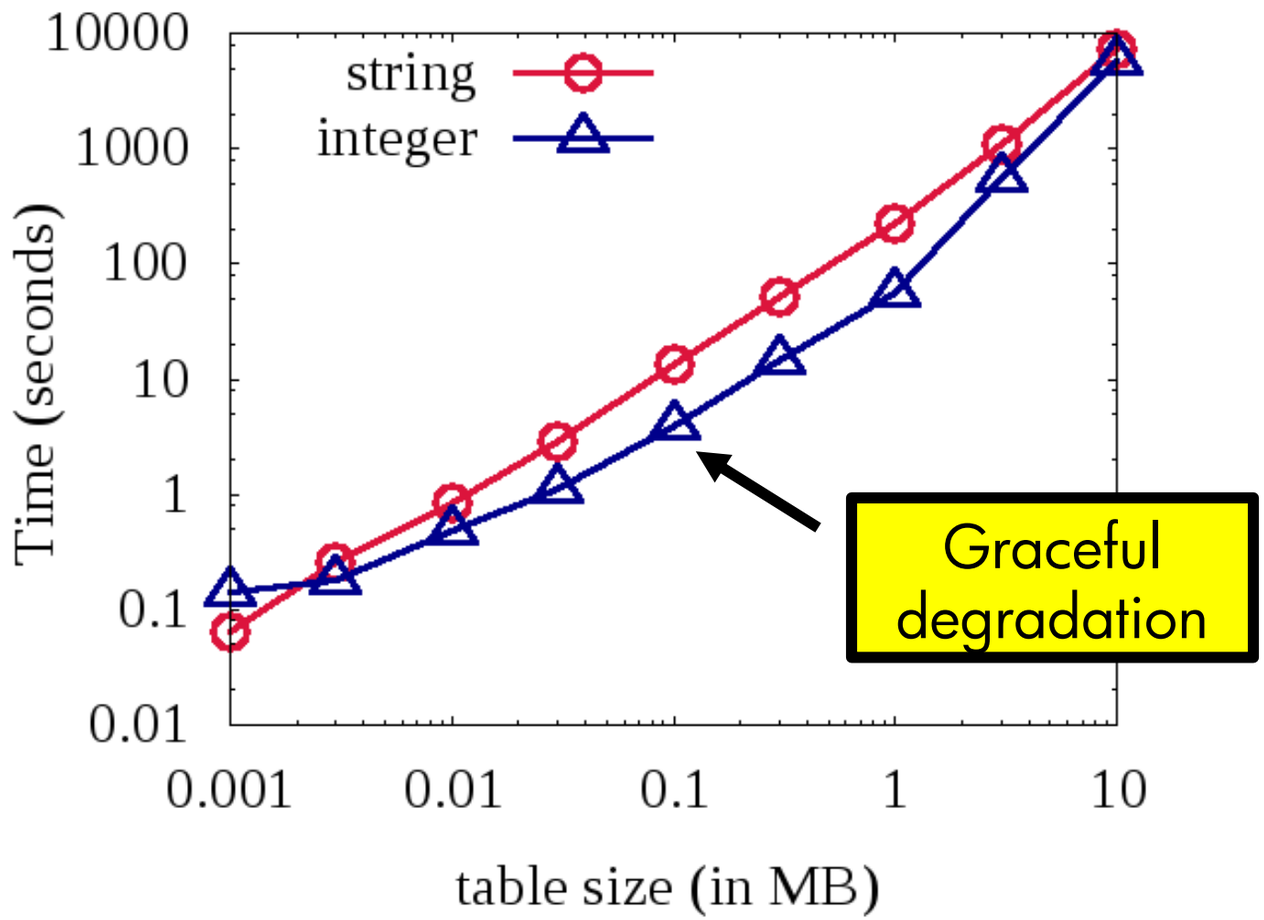
LABS hp

Robustness metrics: Optimality

- How does performance vary between algorithms?
- Compare to "best" algorithm for given conditions

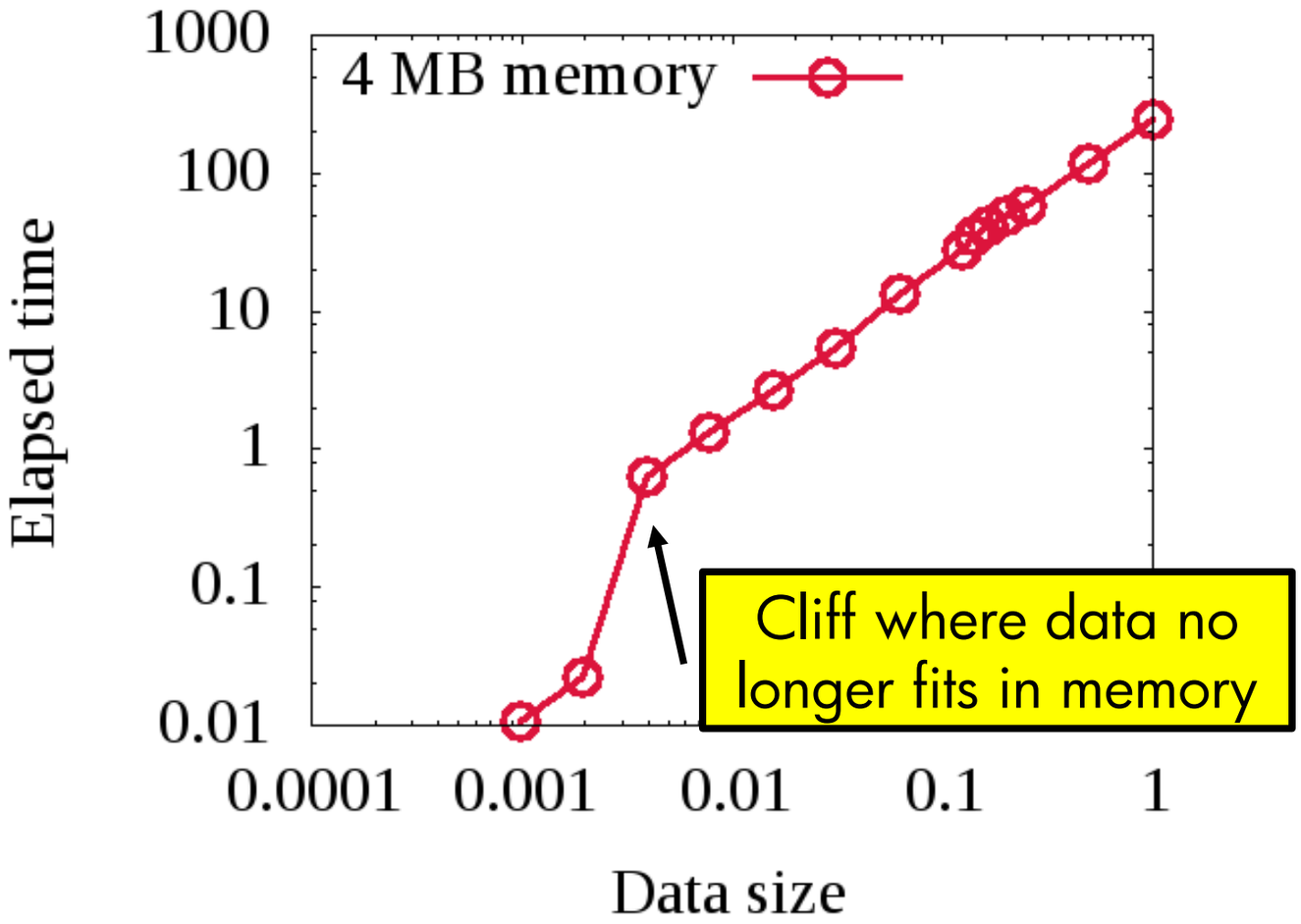


Sort results: Database X



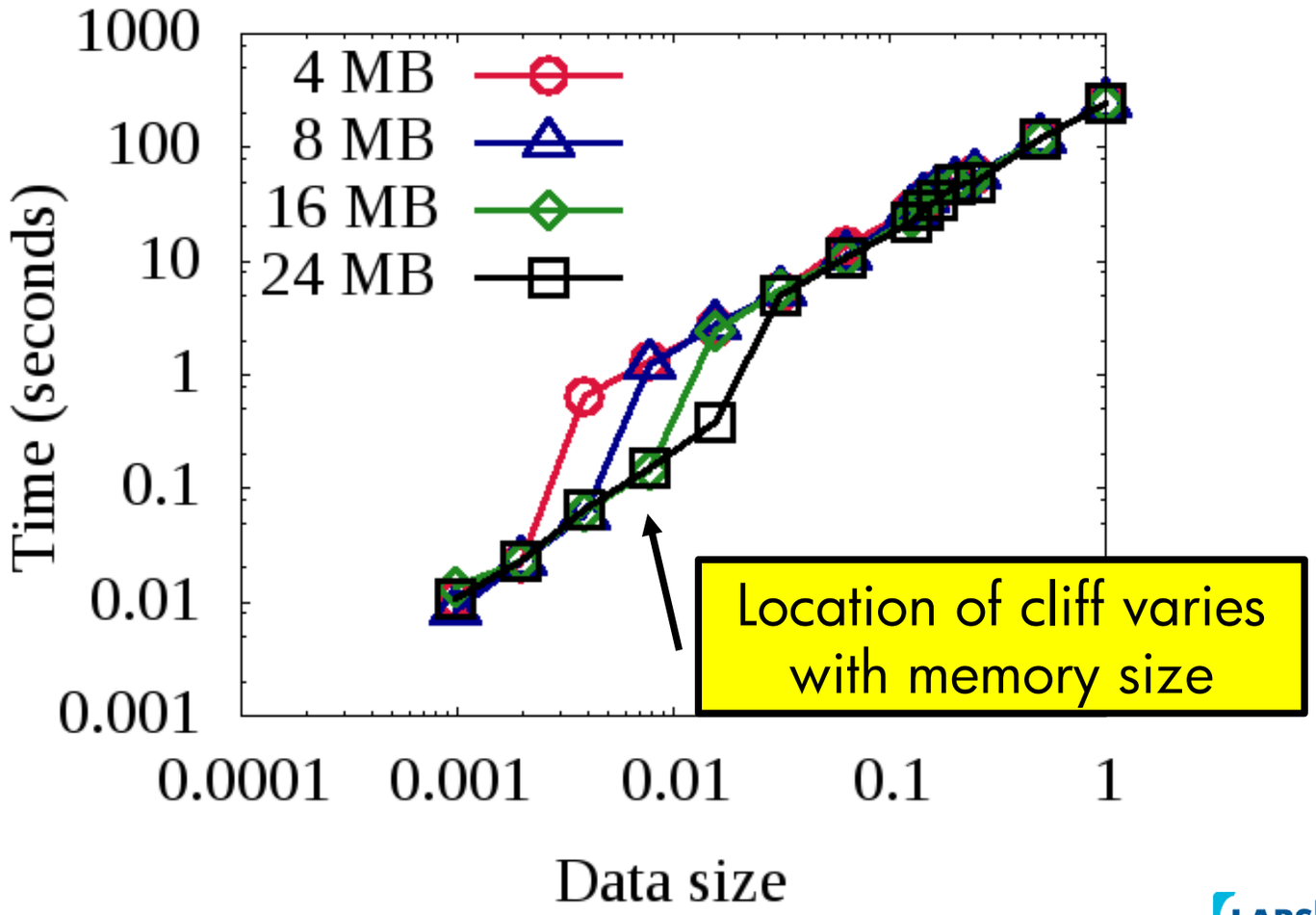
LABS hp

Sort results: Database Y



LABS hp

Sort results: Database Y



Our research at HP Labs

- Query execution robustness
  - Complements query optimization, physical design
- Measure multiple algorithms
  - Scan, sort done; joins next
- On multiple databases
  - Different algorithms, implementations
- Results so far include
  - Algorithms that lack robustness
  - Techniques to improve robustness

Conclusions

- Surprises happen for many reasons!
  - Not always easy to understand
- Robustness metrics provide new information
  - To customers, software developers, hardware vendors
  - Change priorities and decisions
- Robustness must measure "real-life" performance
- Include robustness in future benchmarks
  - So we can motivate and protect improvements