

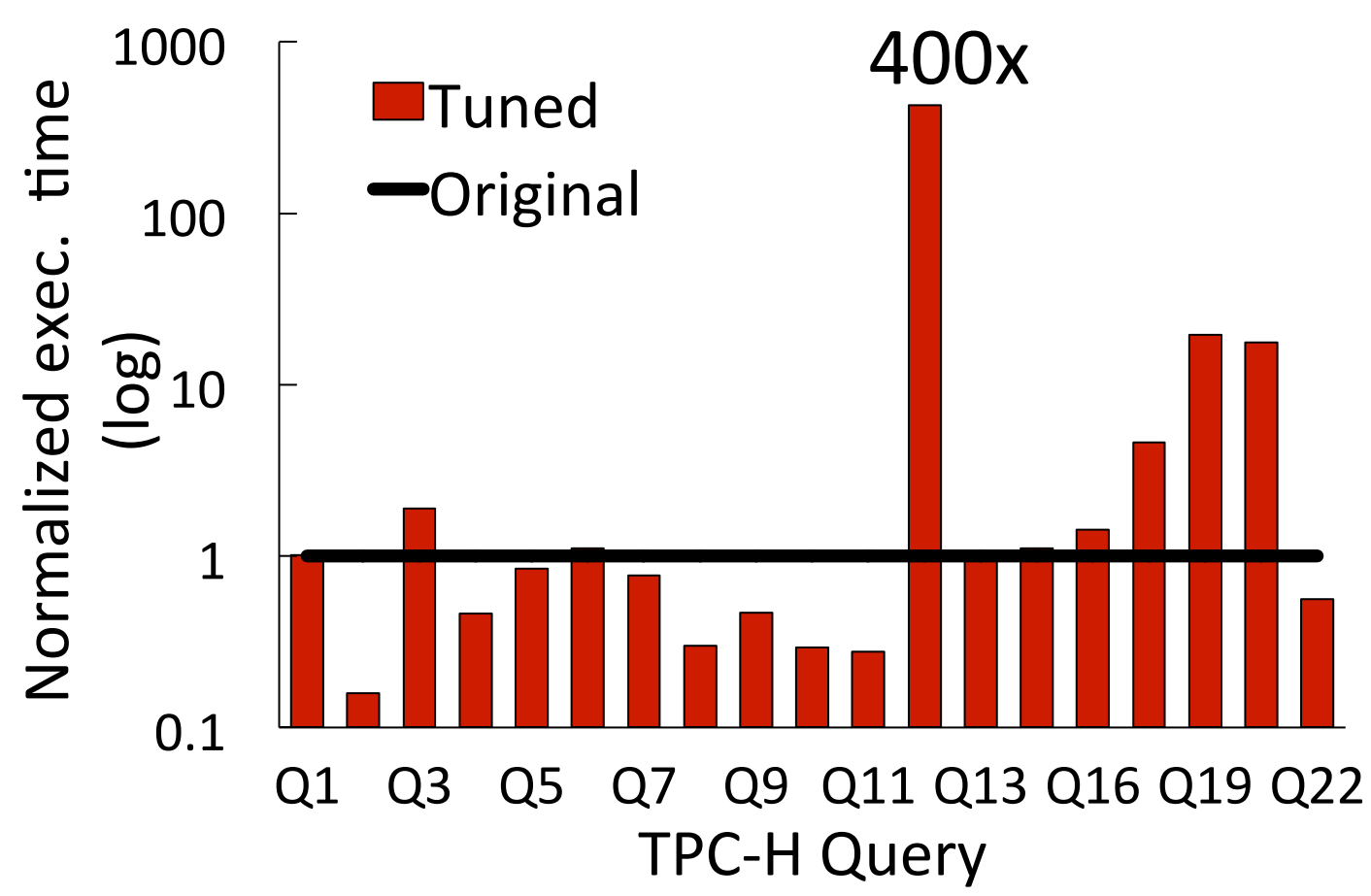
Smooth Scan: Statistics-oblivious Access Paths

Renata Borovica-Gajic, Stratos Idreos, Anastasia Ailamaki, Marcin Zukowski and Campbell Fraser

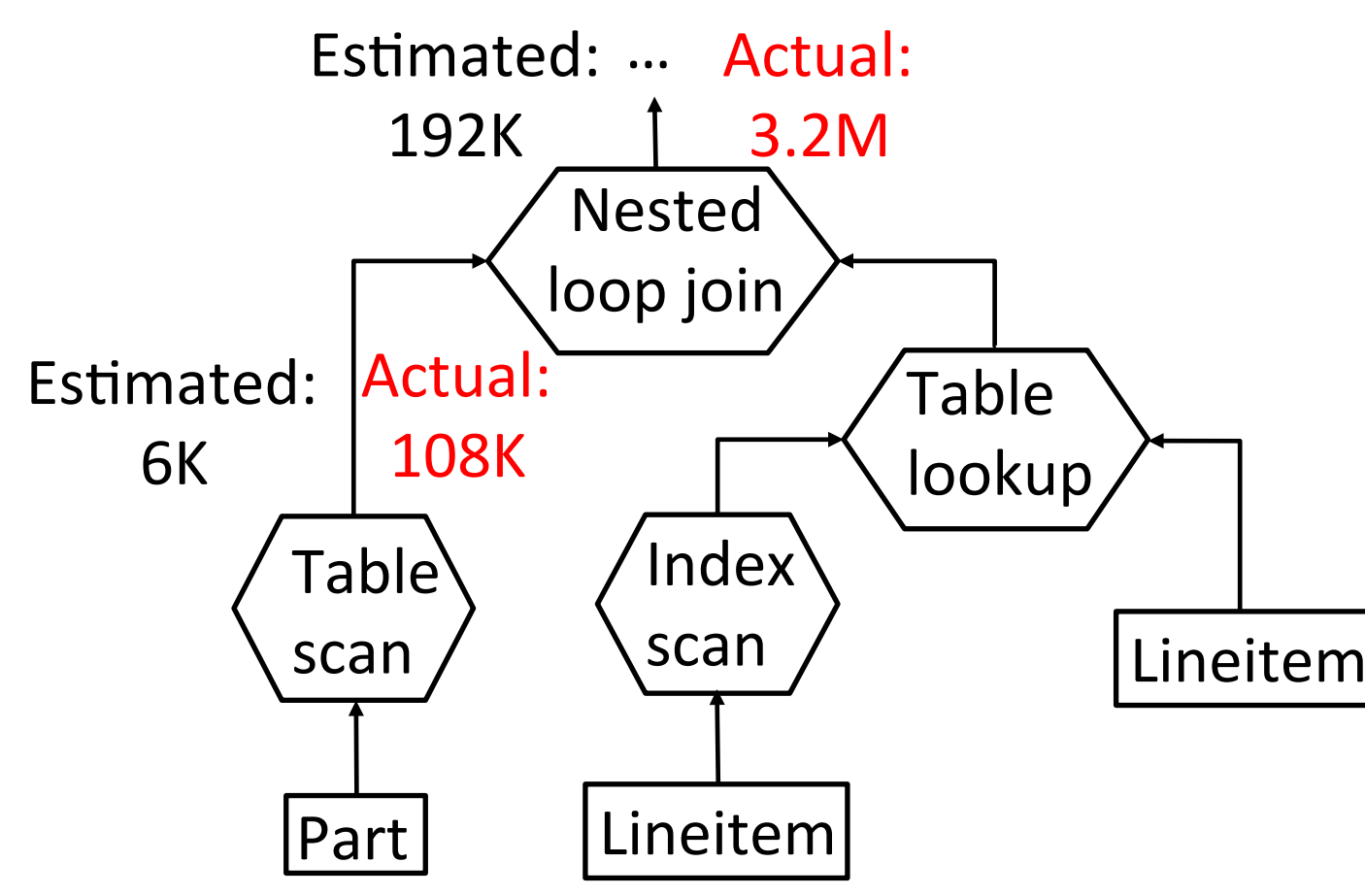
Need for Intra-Query Adaptivity

State of Affairs in Database Systems

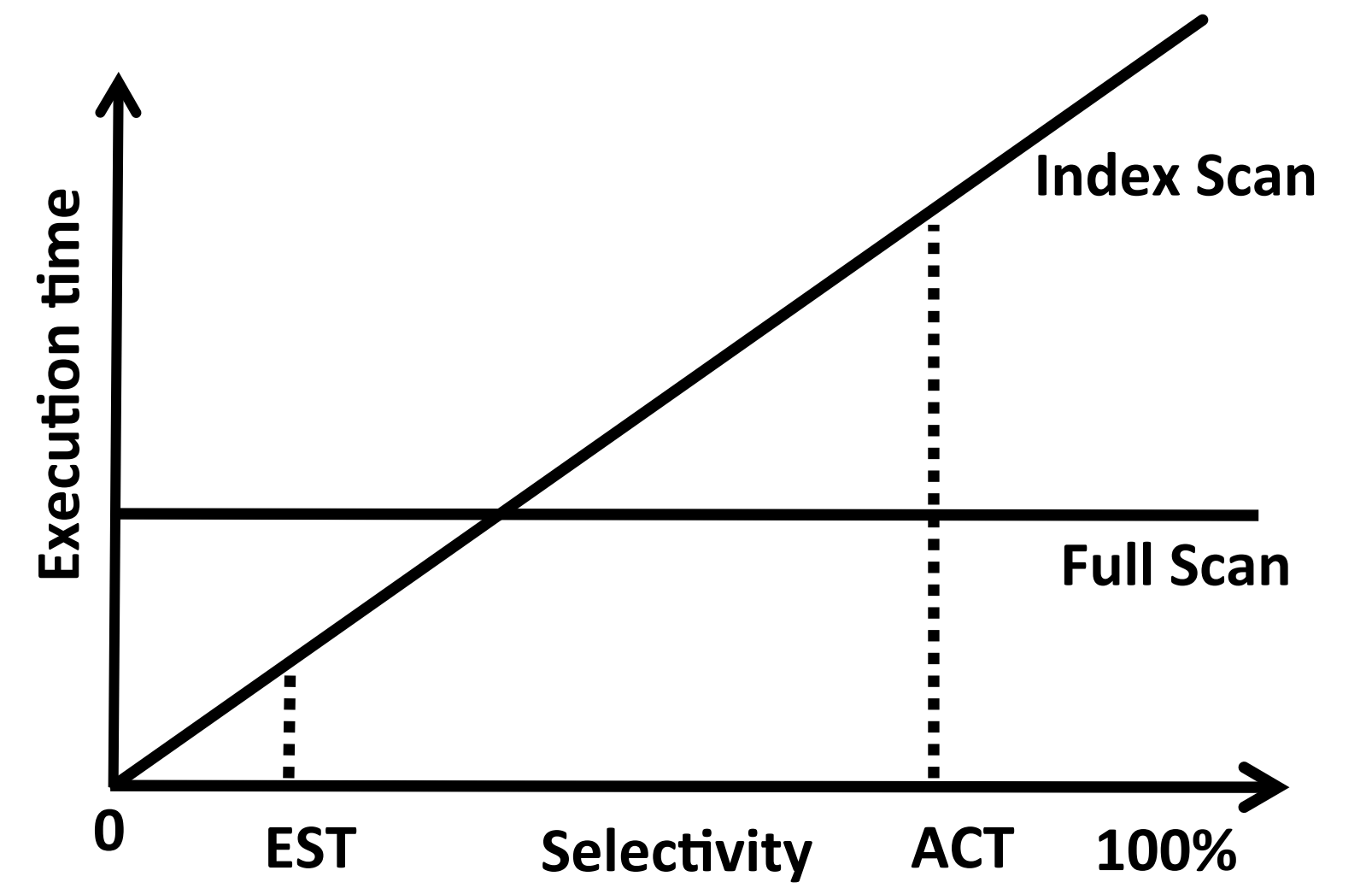
Setting: TPC-H, SF10, DBMS-X, Tuning tool 5GB space



Cause for sub-optimal plans



Access path selection problem



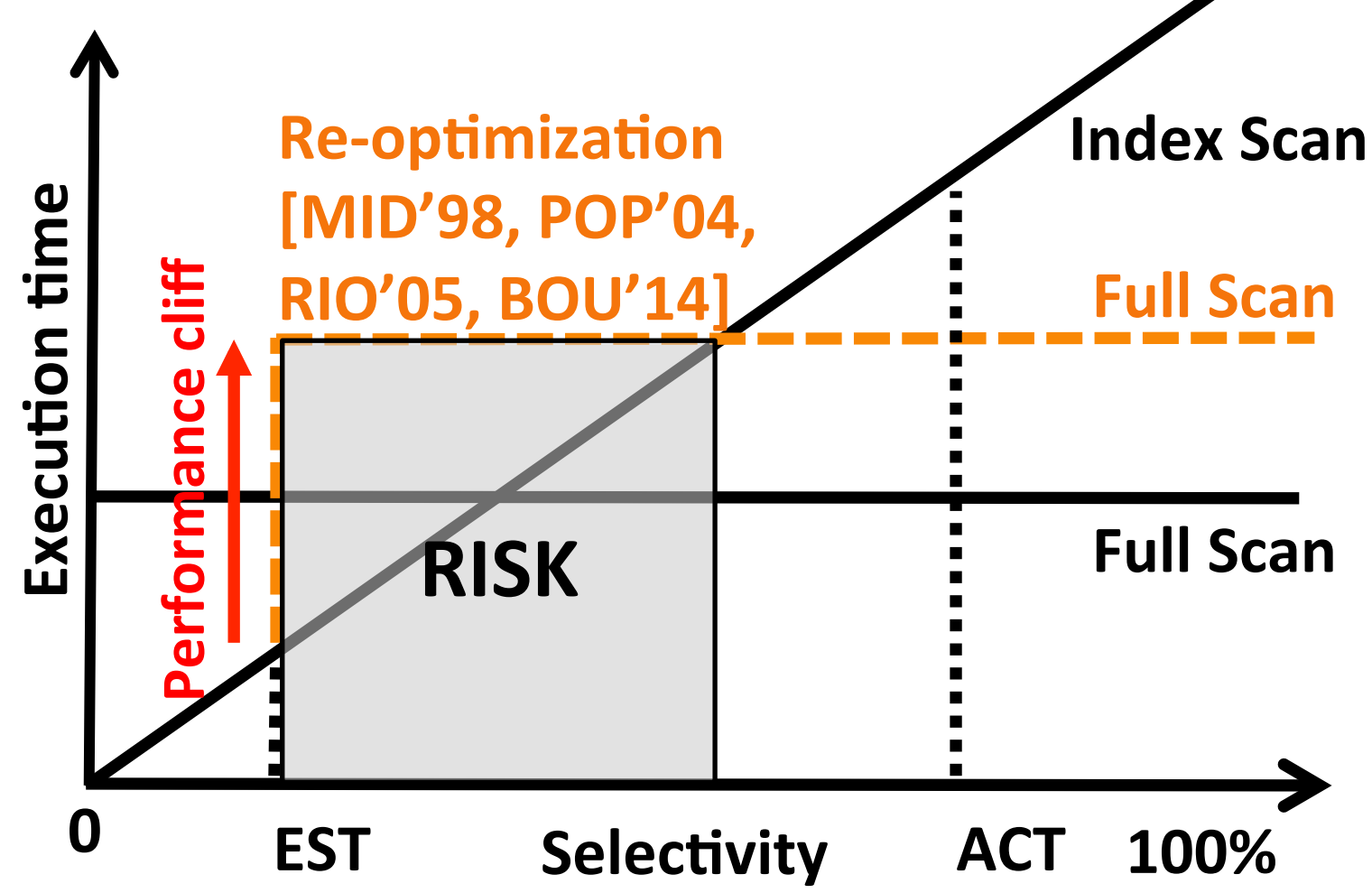
➤ Degradation due to sub-optimal access path choices

➤ Cardinality misestimates

➤ Statistics: unreliable advisor

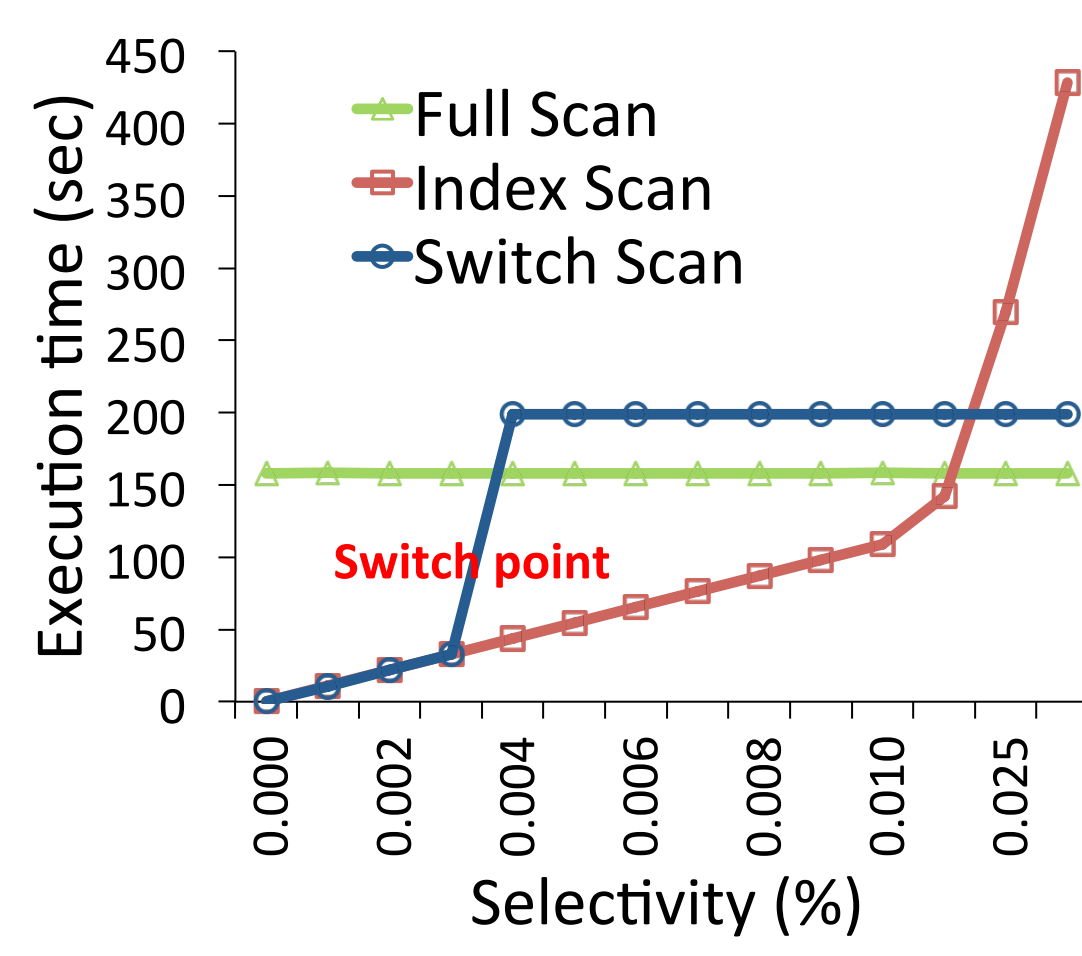
Adaptivity in Access Path Operators

Mid-query re-optimization



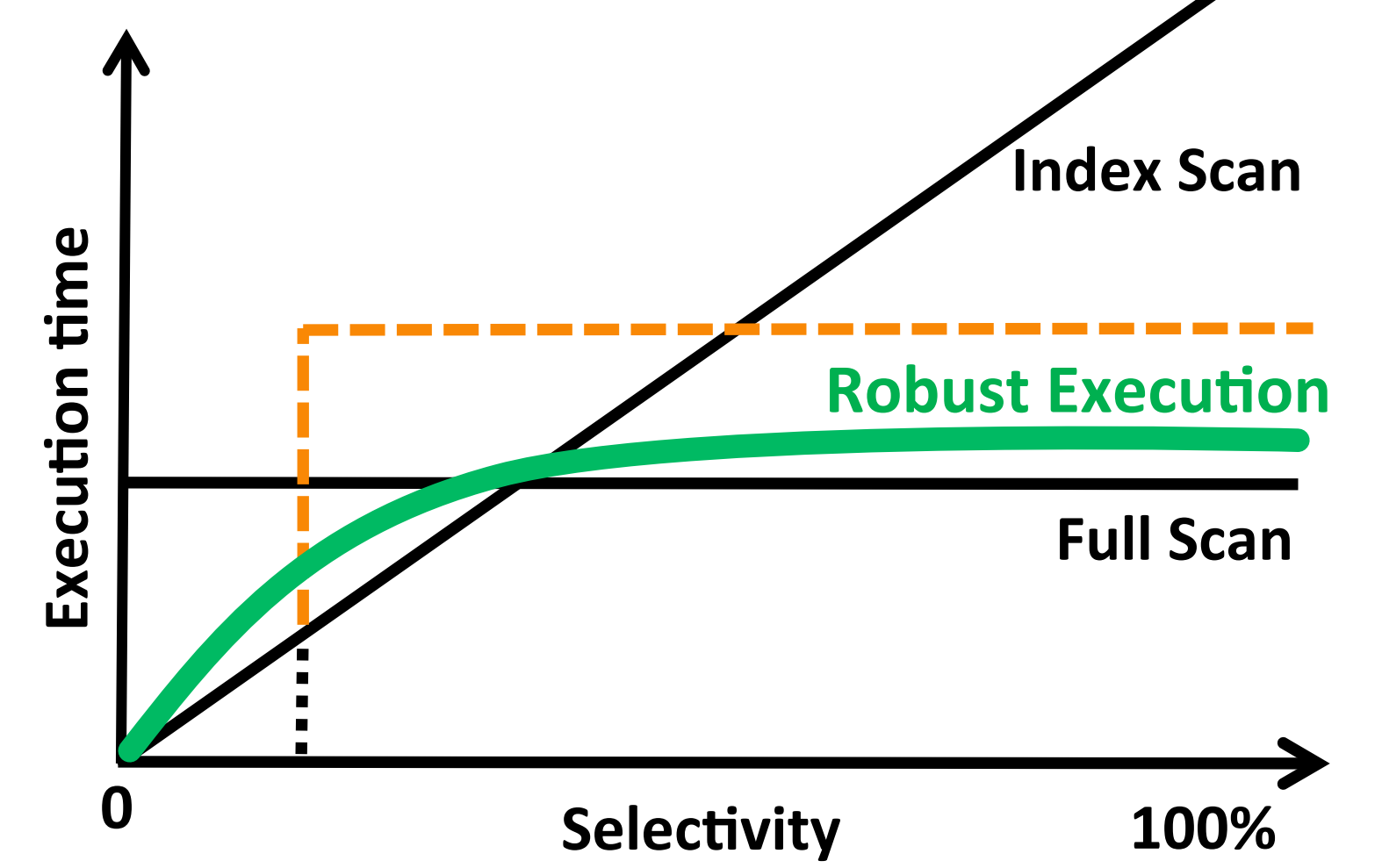
➤ Re-optimization risky

Re-optimization in action



➤ Violate user expectation

Quest for Robust Execution



➤ Near-optimal for all inputs

Smooth Scan in Nutshell

Statistics-oblivious access paths

SMOOTH SCAN

- Ignore optimizer's estimates
- Learn result distribution at run-time
- Adapt as you go

DESIGN GOALS

- Avoid performance cliffs & risk
- Continuous, gradual and smooth adaptation

➤ Adaptive, but smooth

Adaptivity with Smooth Scan

INSIGHT: Morph between Index and Sequential Scan

HOW?

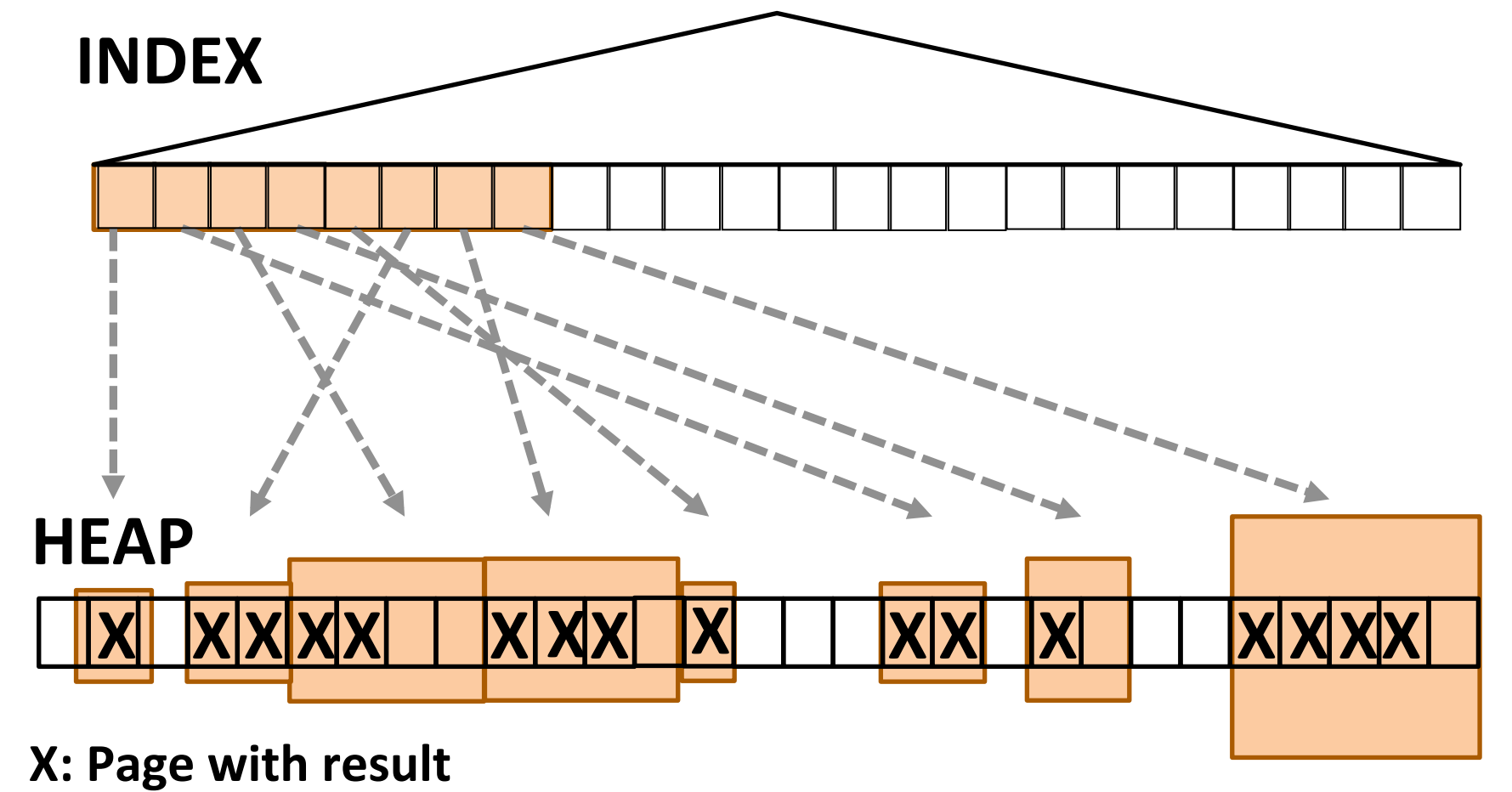
- Index Access
- Entire Page Probe
- Gradual Flattening Access

WHEN?

- Selectivity increase -> Mode Increase
- Selectivity decrease -> Mode Decrease

➤ Data driven adaptation

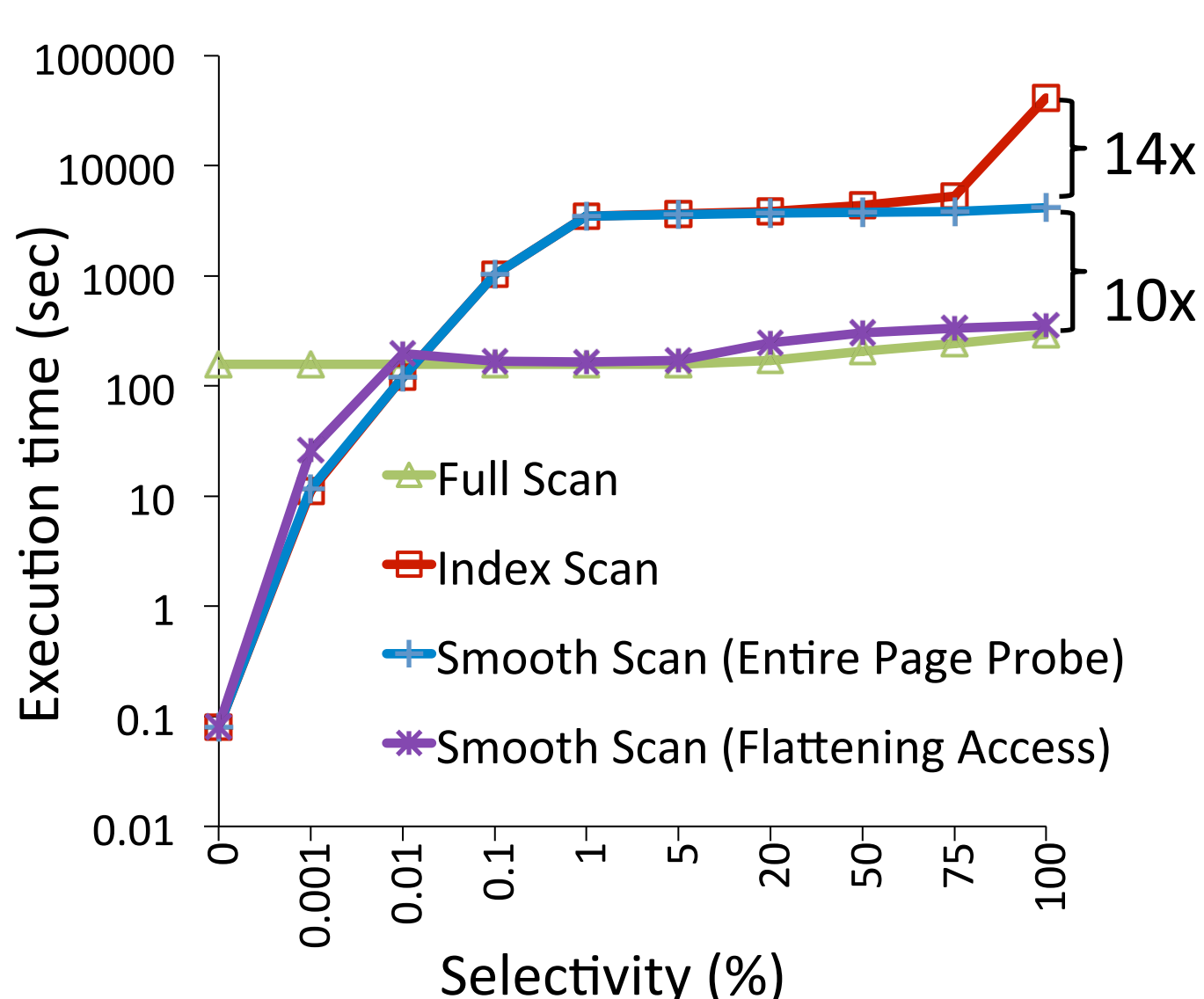
Region Snooping = Data driven adaptation



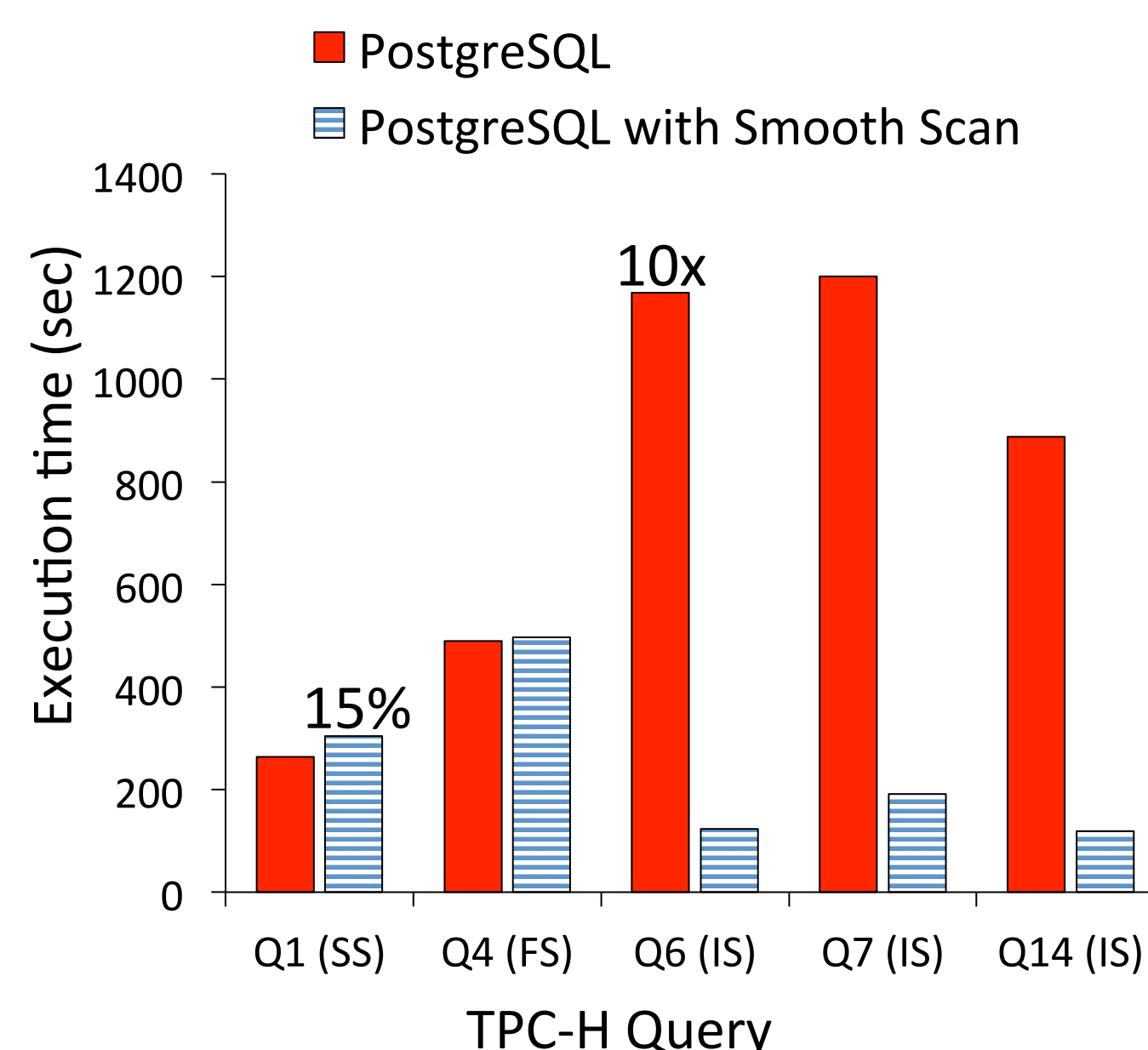
➤ Avoid repeated access ➤ Less random I/O

Smooth Scan in Action

Setting: 400M tuples, 25GB, Index(c2)
Query: select * from R where c2 < X%;



Setting: TPC-H, SF 10, PostgreSQL with Smooth Scan



Smooth Scan Summary

Operator morphing from one form to another

$$+ \text{Data driven adaptation} = \text{Robust query execution}$$