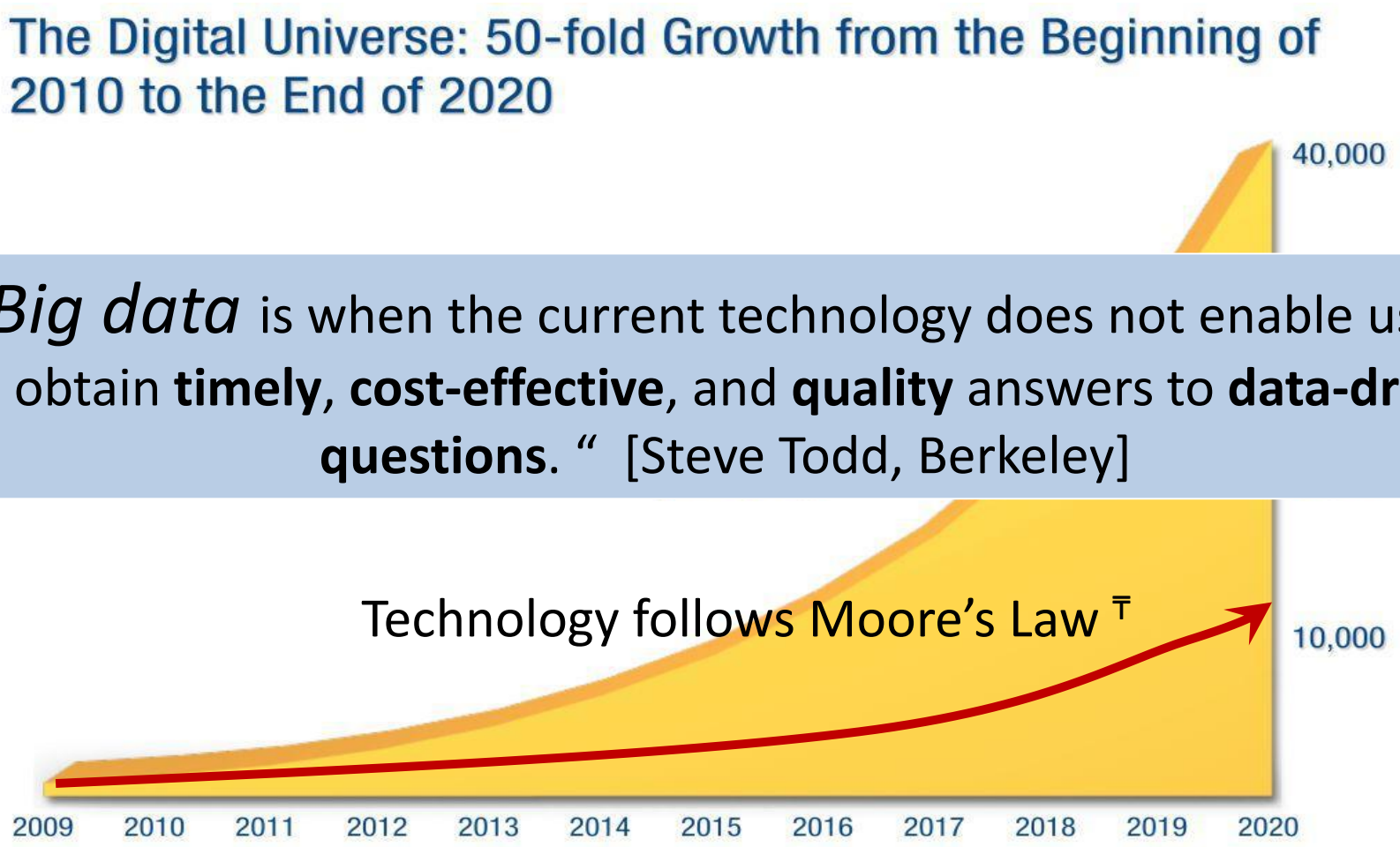
Towards timely, predictable and cost-effective data analytics

Renata Borovica-Gajić

Melbourne School of Engineering
Department of Computing & Information Systems

Big data proliferation



"*Big data* is when the current technology does not enable users to obtain **timely**, **cost-effective**, and **quality** answers to **data-driven questions**. " [Steve Todd, Berkeley]

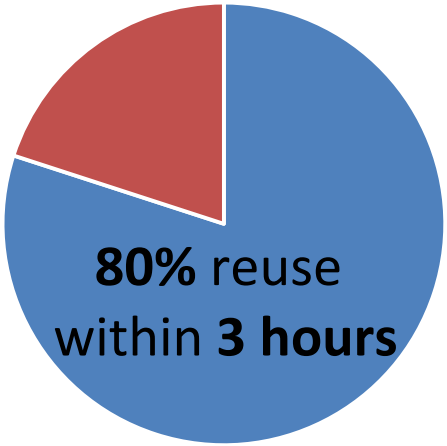
Technology follows Moore's Law ∓

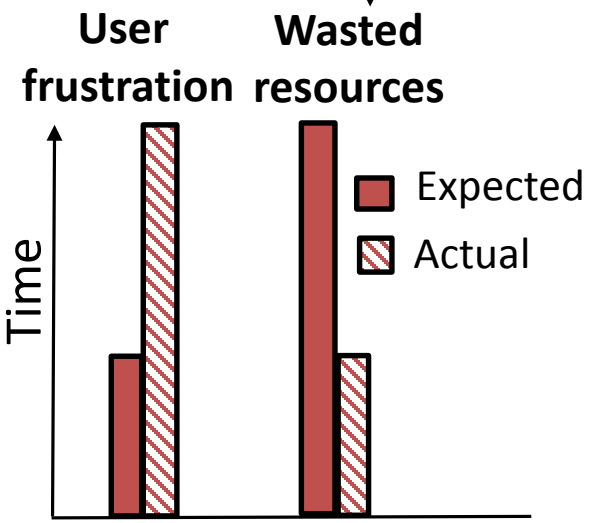
* "The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East", 2012, IDC

∓ "Trends in big data analytics", 2014, Kambatla et al

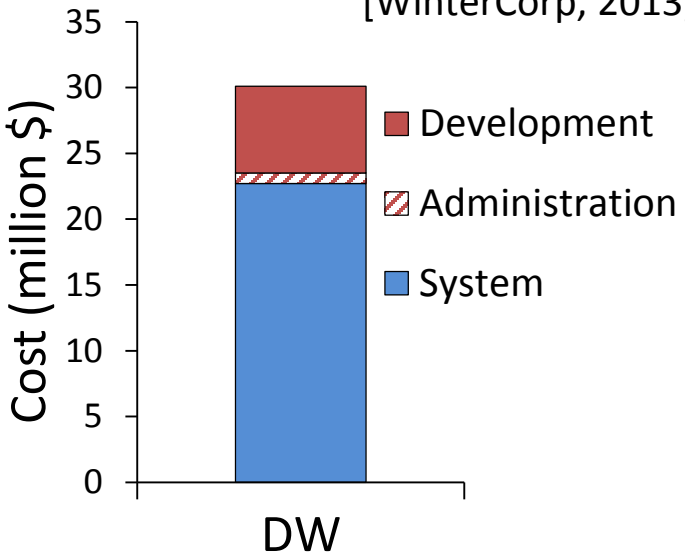
What business analysts want

Timely, predictable, cost-effective queries





[WinterCorp, 2013]



Minimal data-to-insight time

Predictable response time

Low infrastructure cost

Research challenge

As traditional DBMS rely on **predefined assumptions** about workload, data and storage, changes cause **loss of performance** and **unpredictability**.

Insight

Query execution must **adapt** at three levels

(to **workload**, **data** and **hardware**) to stabilise and **optimise performance** and **cost**.

Outline

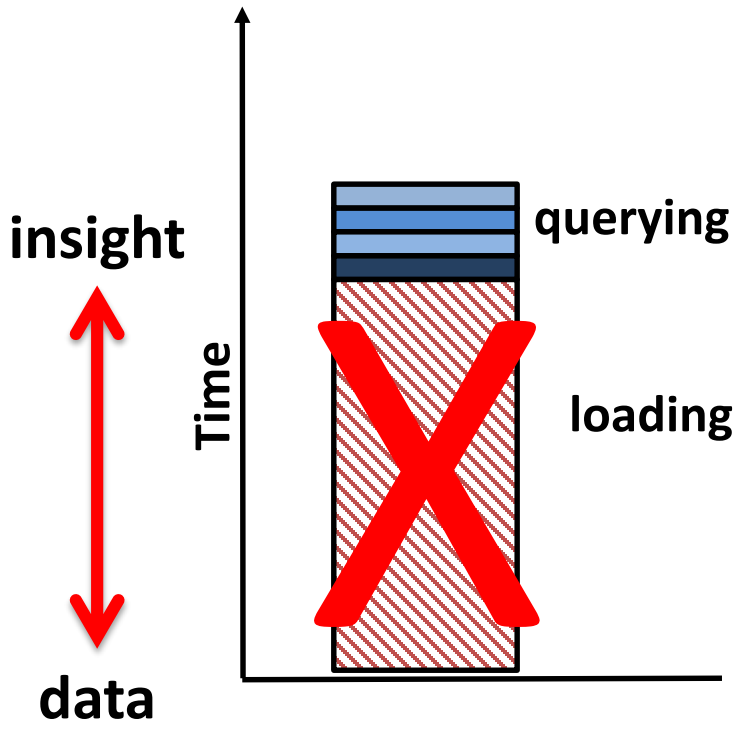
- **Minimise data-to-insight time**
  - Workload-driven adaptation [CACM'15, SIGMOD'12, VLDB'12]
- **Improve predictability of response time**
  - Data-driven adaptation [ICDE'15, DBTest'12]
- **Reduce analytics cost**
  - Cold storage & hardware-driven adaptation [VLDB'16]

Outline

- **Minimise data-to-insight time**
  - Workload-driven adaptation [CACM'15, SIGMOD'12, VLDB'12]
- **Improve predictability of response time**
  - Data-driven adaptation [ICDE'15, DBTest'12]
- **Reduce analytics cost**
  - Cold storage & hardware-driven adaptation [VLDB'16]

Data-to-insight time

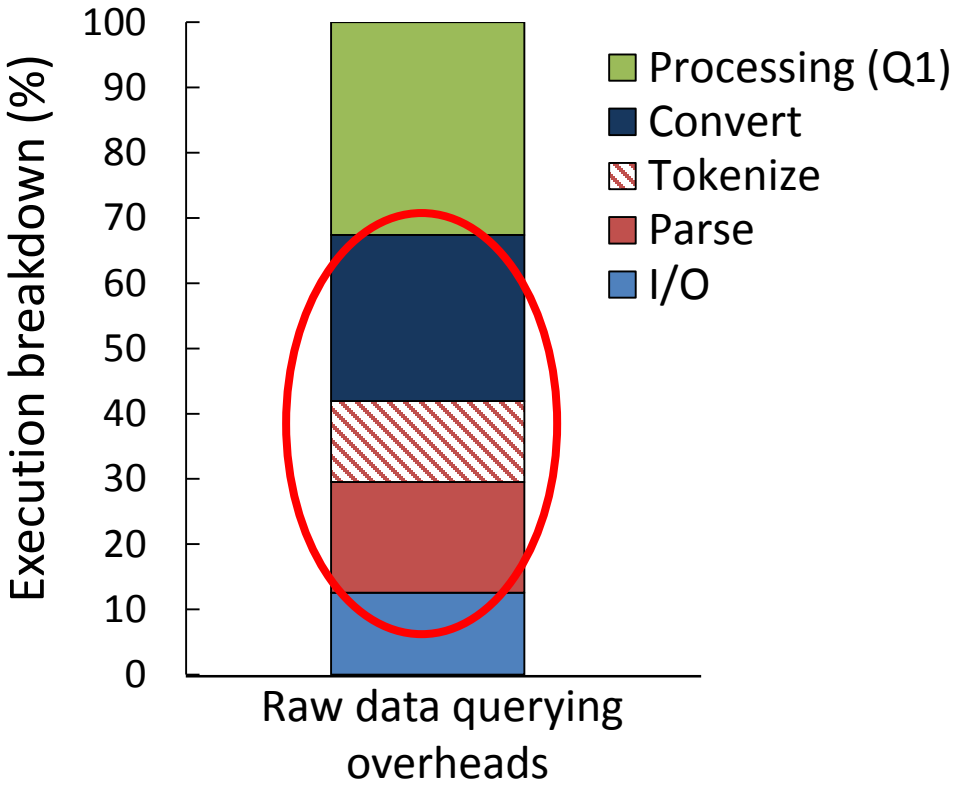
Traditional query stack



Time to first insight too long

Does not scale with data growth

Raw data querying stack

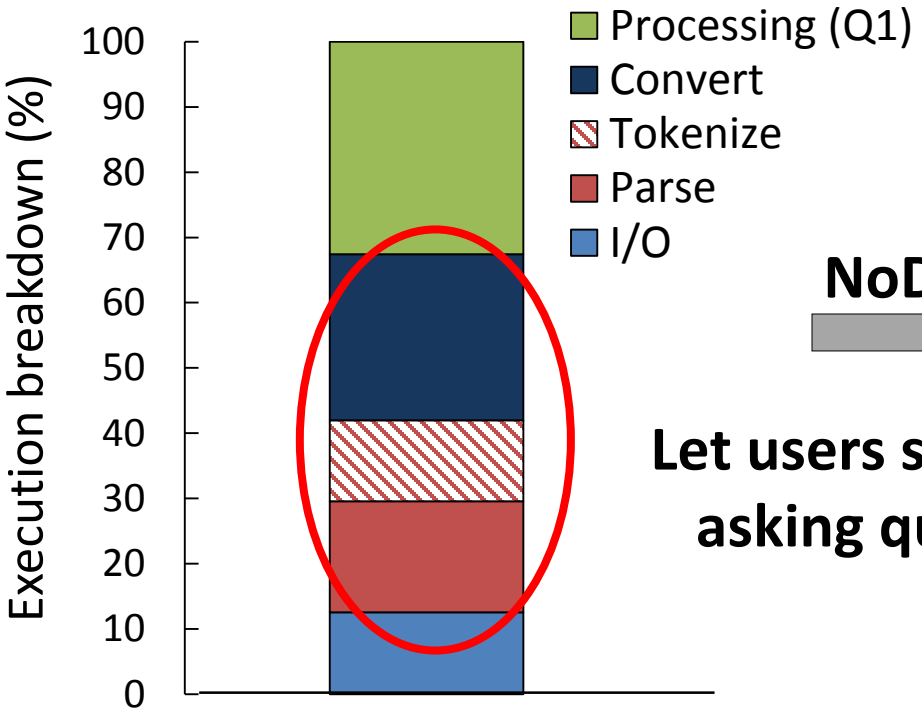


Overheads too high

Current technology ≠ efficient exploration

Optimise raw data querying stack

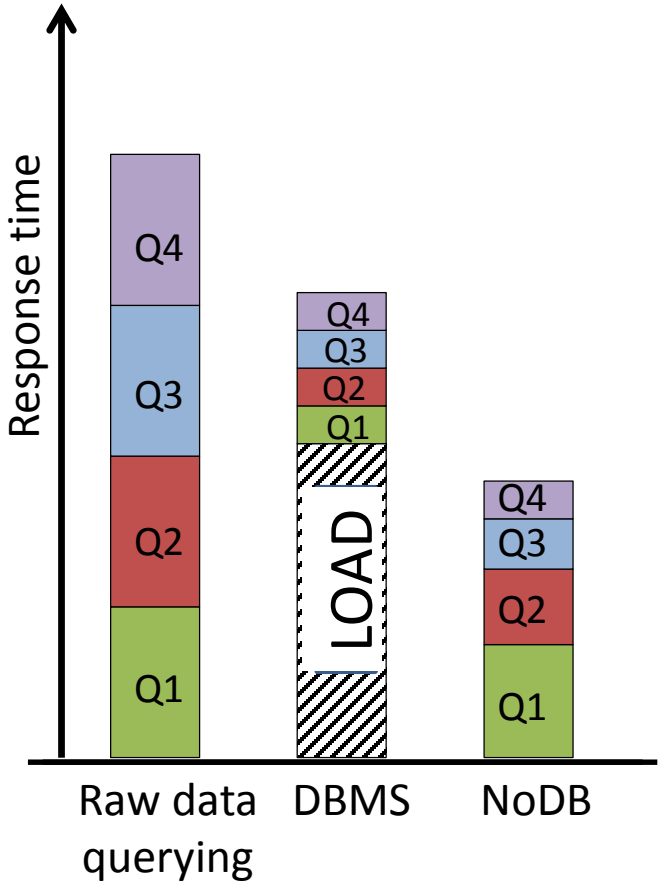
Raw data querying stack



Not everything needed for Q1

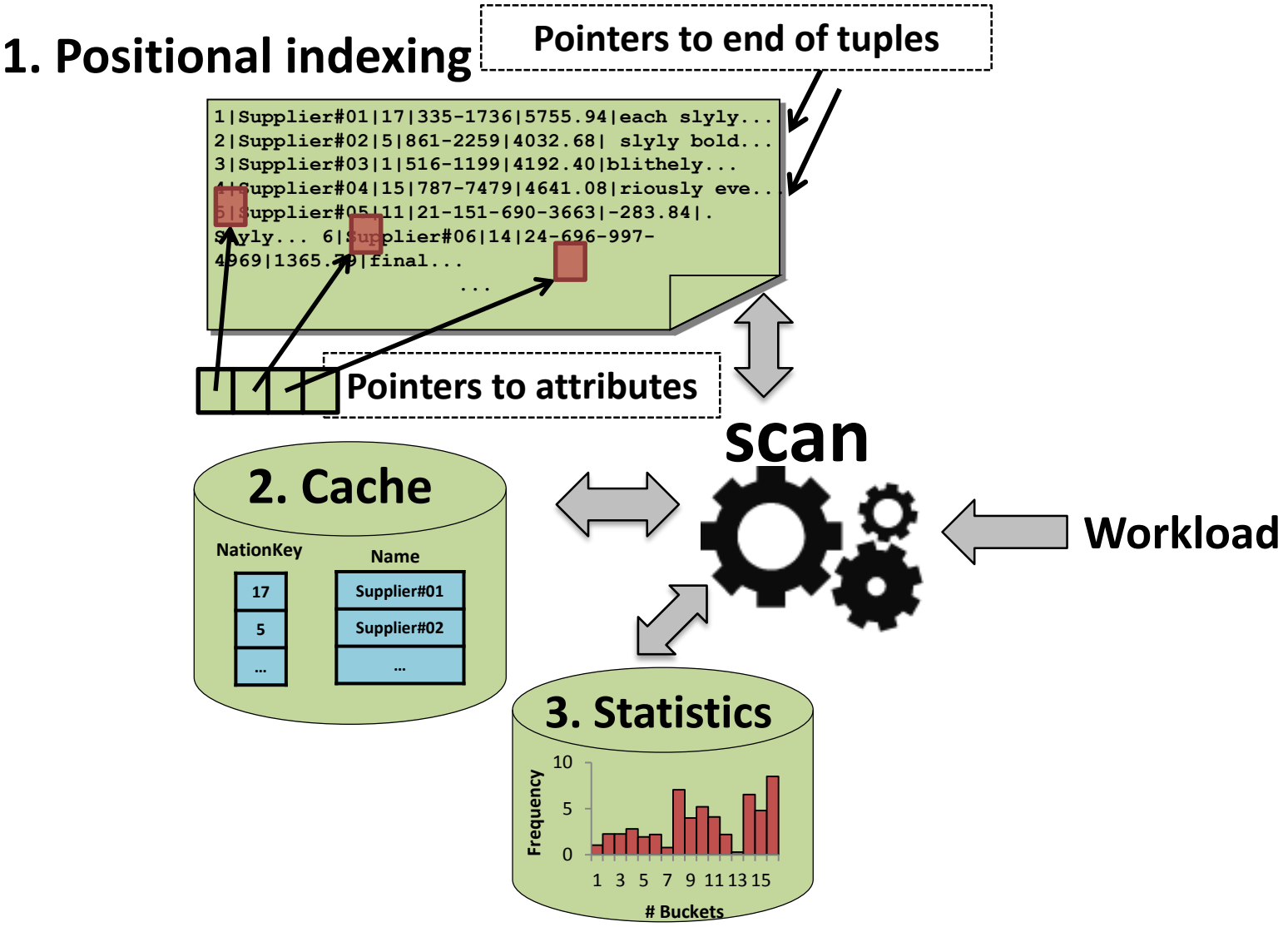
NoDB

Let users show by asking queries



NoDB: Workload-driven data loading & tuning

PostgresRaw: NoDB from idea to practice

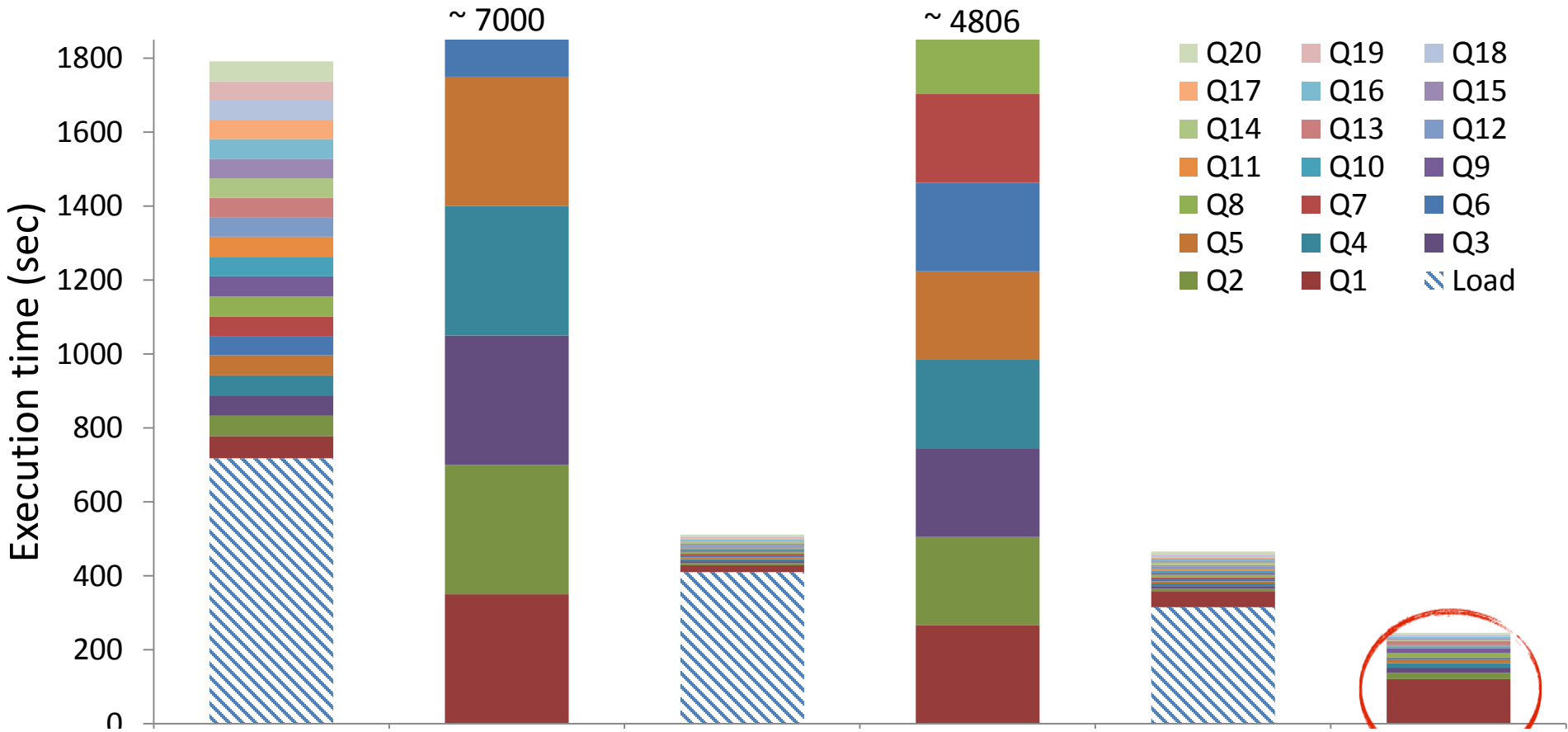


Adjust to queries = progressively cheaper access

PostgresRaw in action

Setting: 7.5M tuples, 150 attributes, 11GB file

Queries: 10 arbitrary attributes per query, vary selectivity



Data-to-insight time halved with PostgresRaw

Per query performance comparable to traditional DBMS

Summary of PostgresRaw

- **Query processing** engine over **raw data files**
- Uses user **queries** for **partial** data **loading** and **tuning**
- **Comparable performance** to traditional DBMS

IMPACT

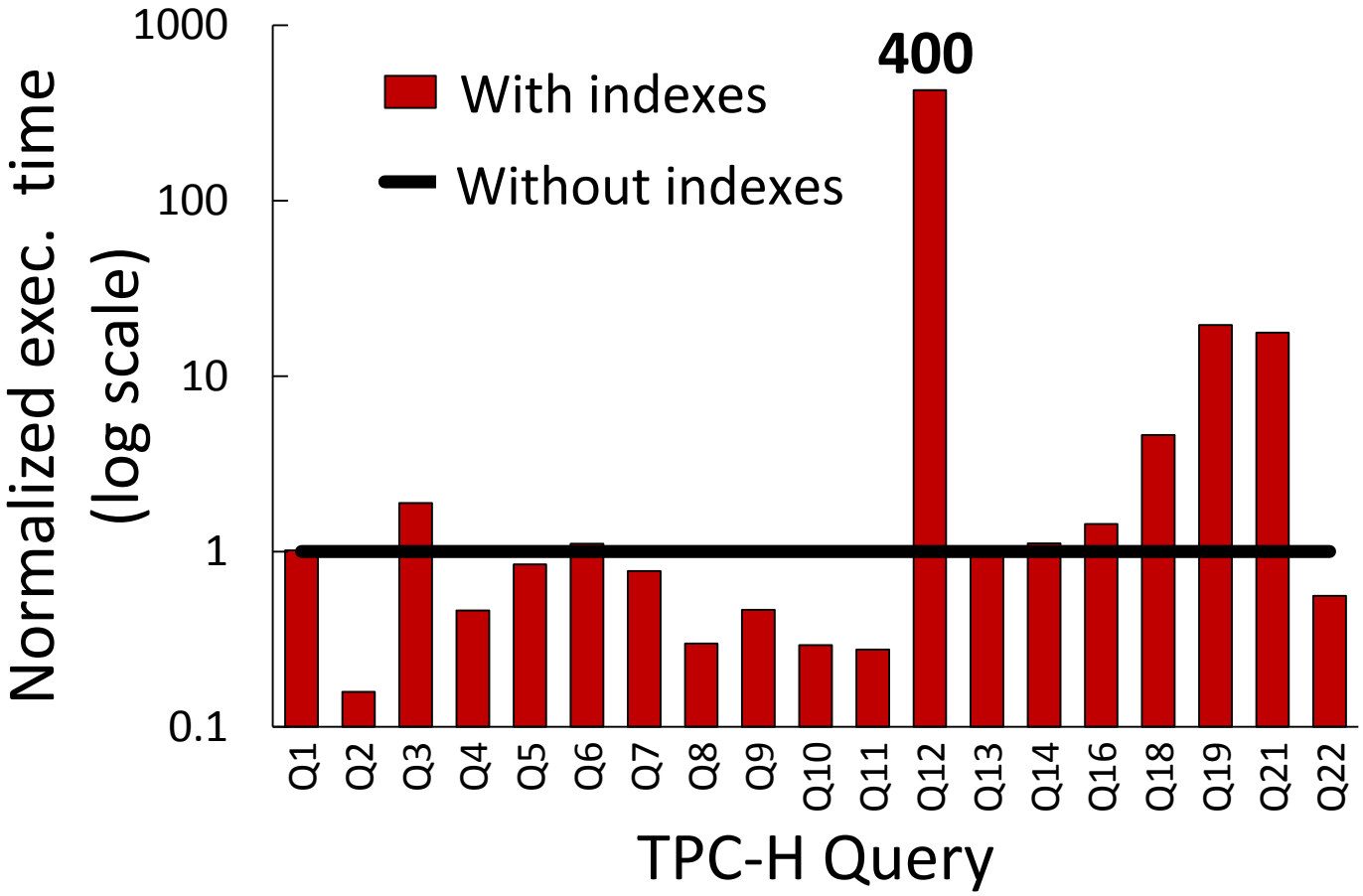
- Enables **timely data exploration** with **0 initialisation**
- **Decouples** user **interest** from **data growth**

Outline

- **Minimise data-to-insight time**
  - Workload-driven adaptation [CACM'15, SIGMOD'12, VLDB'12]
- **Improve predictability of response time**
  - Data-driven adaptation [ICDE'15, DBTest'12]
- **Reduce analytics cost**
  - Cold storage & hardware-driven adaptation [VLDB'16]

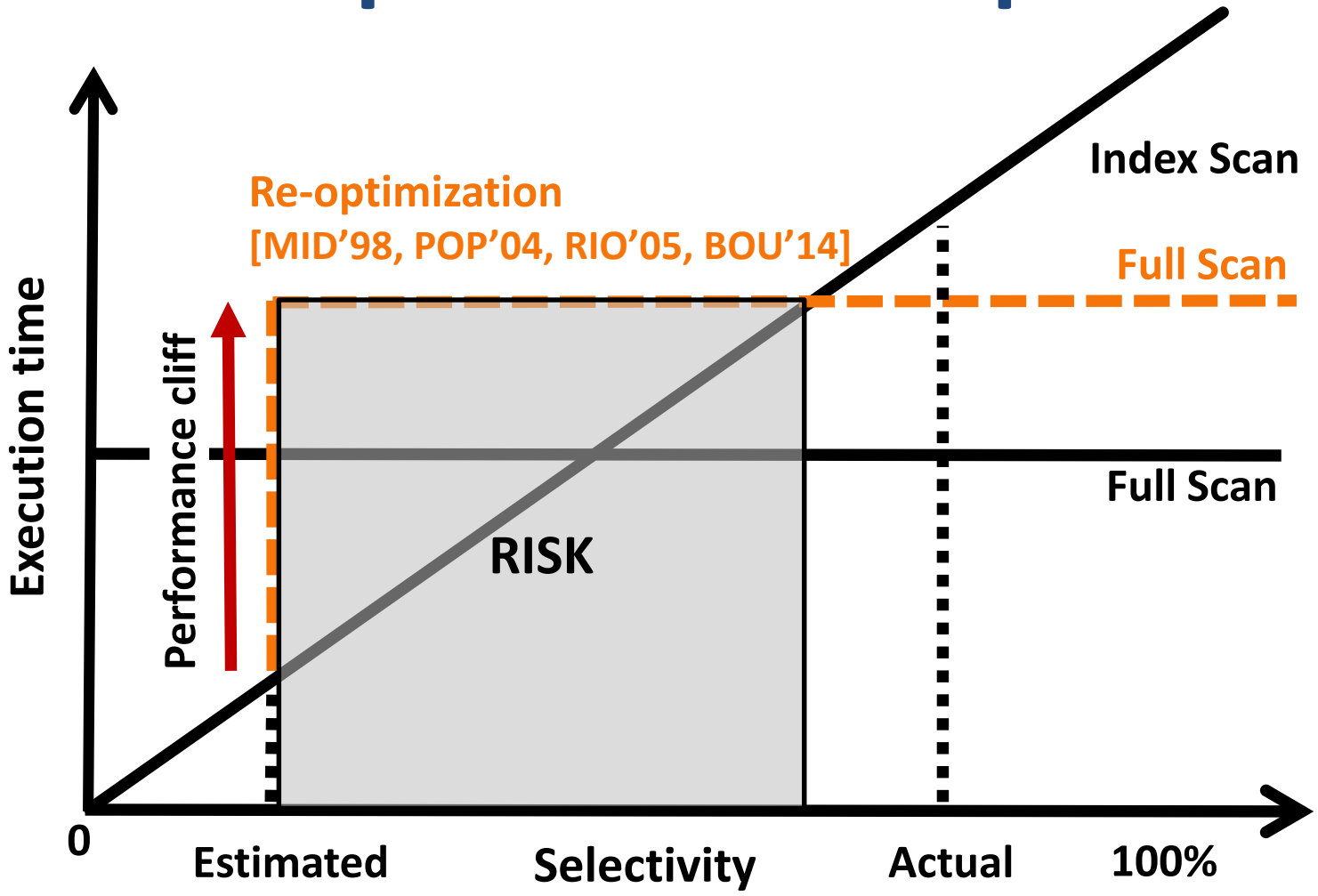
Index: with or without?

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



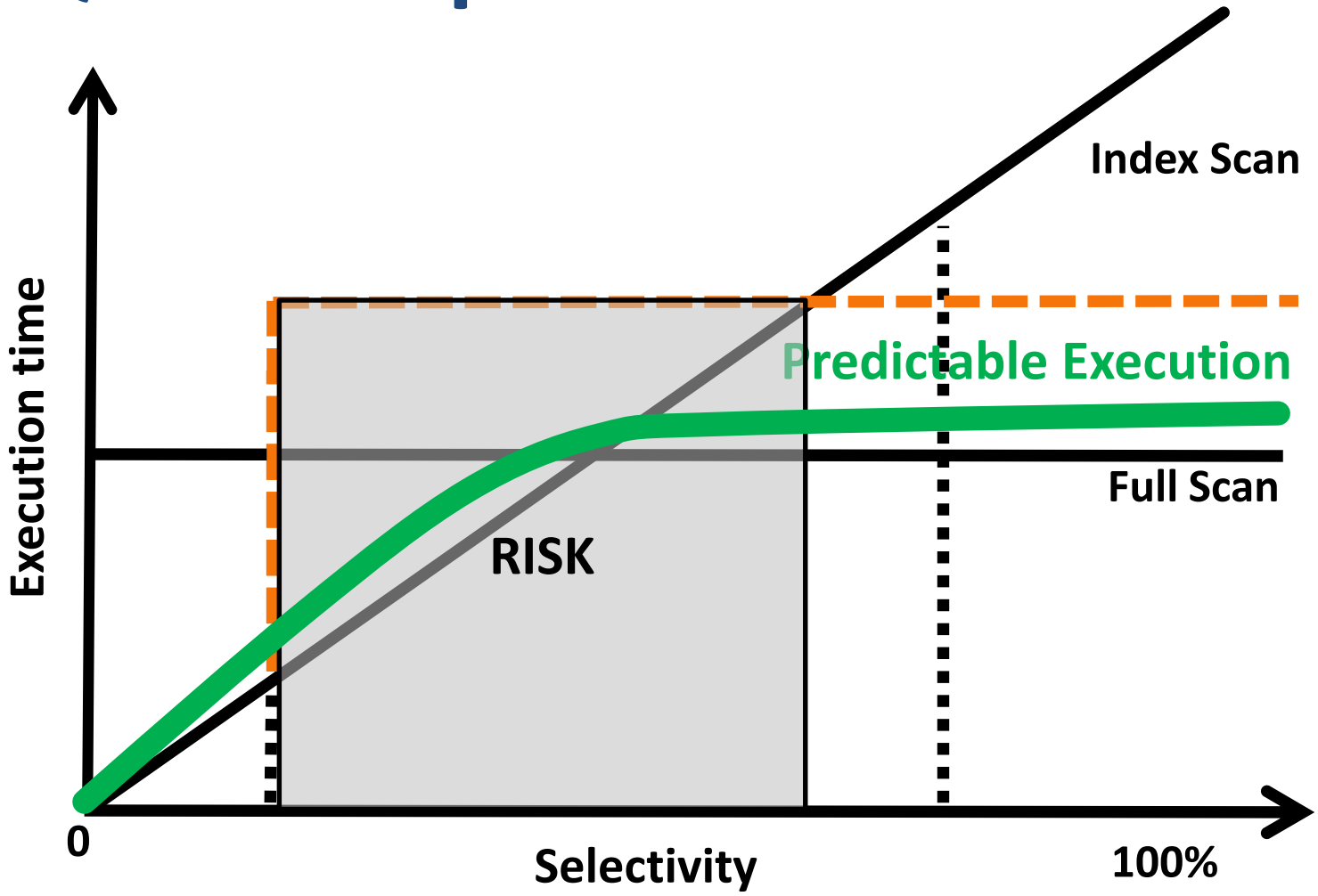
Performance hurt after tuning

Access path selection problem



Statistics: unreliable advisor
Re-optimization: risky

Quest for predictable execution



Removing variability due to (sub-optimal) choices

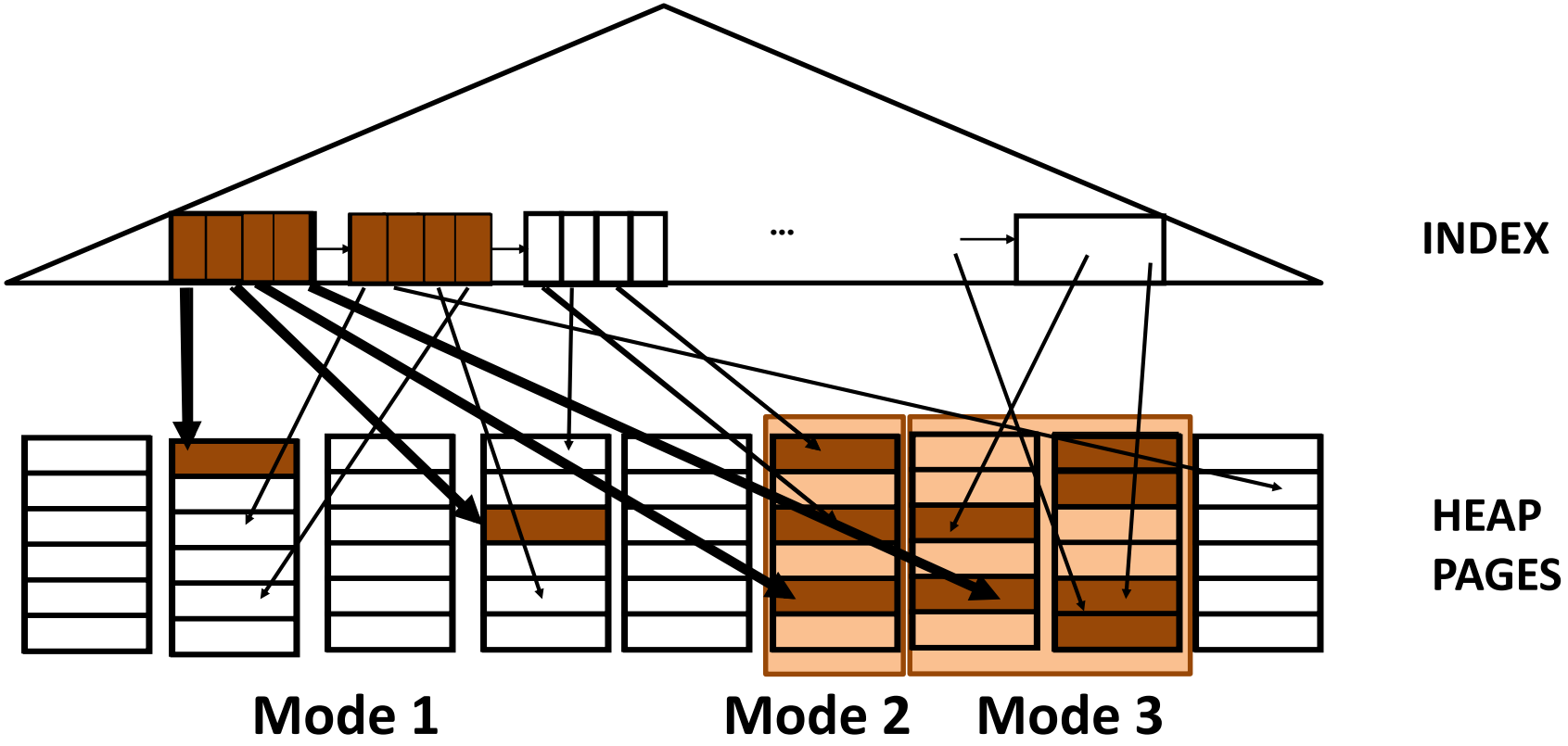
Smooth Scan

Morph between Index and Sequential Scan based on **observed result** distribution

Morphing mechanism

Modes:

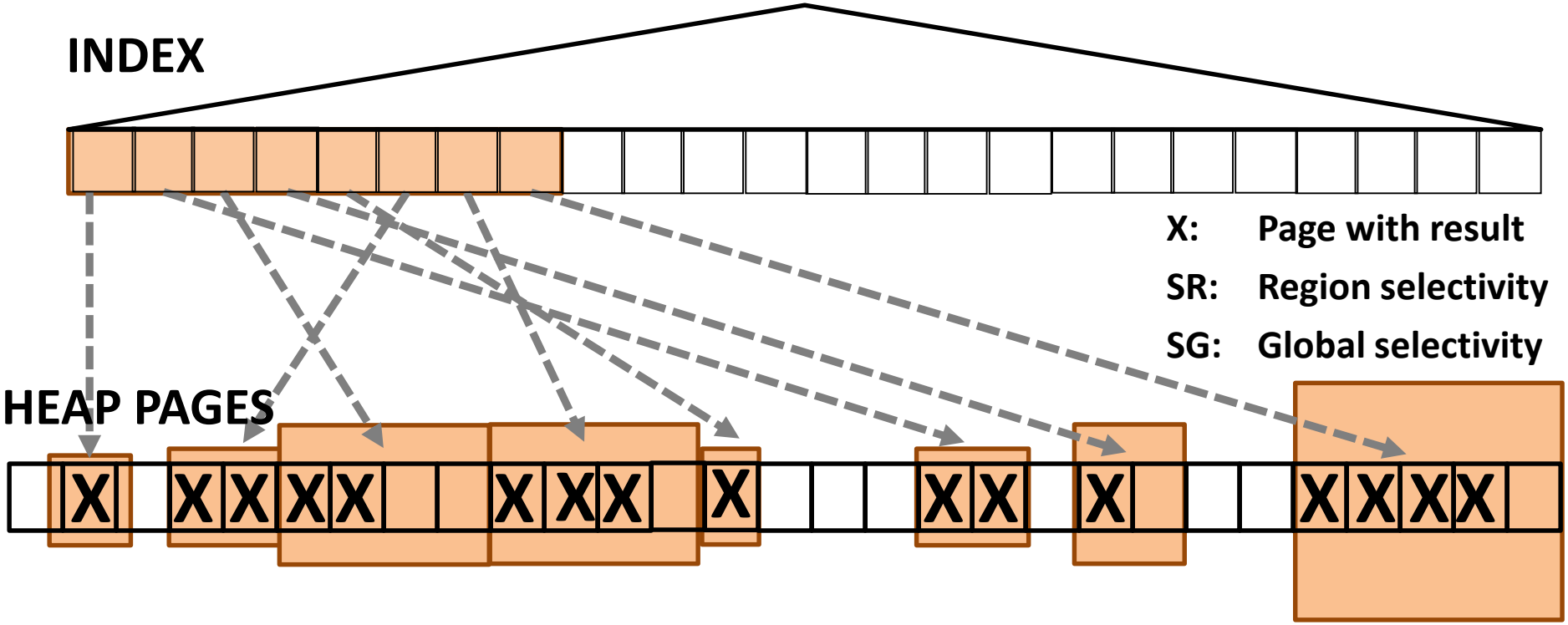
1. **Index Access:** Traditional index access
2. **Entire Page Probe:** Index access probes entire page
3. **Gradual Flattening Access:** Probe adjacent region(s)



Morphing policy

- Selectivity Increase -> Mode Increase
- Selectivity Decrease -> Mode Decrease

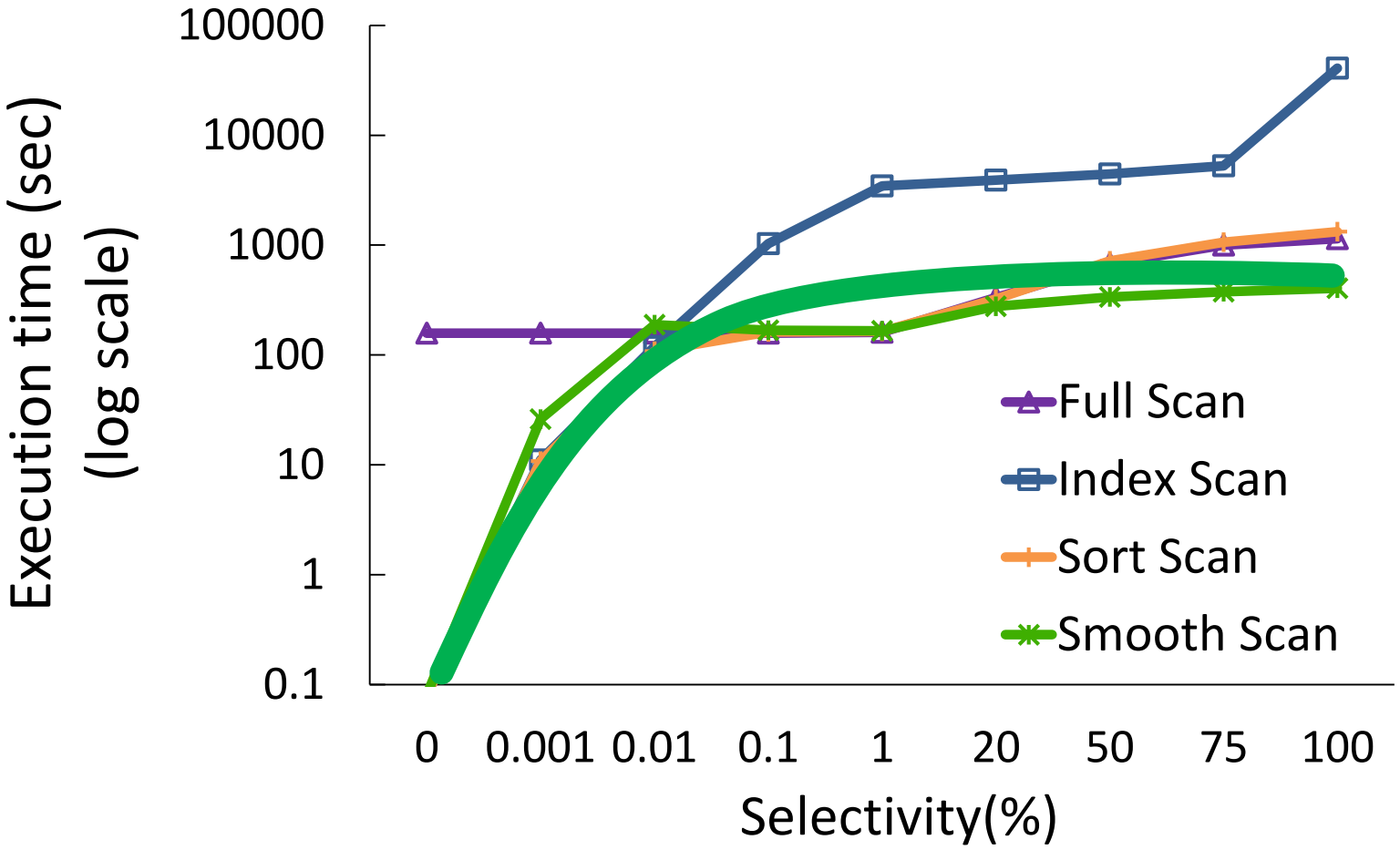
SEL_region >= SEL_global
SEL_region < SEL_global



Region snooping = Data-driven adaptation

Smooth Scan in action

Setting: Micro-benchmark, 25GB table, Order by, Selectivity 0-100%



Near-optimal over entire selectivity range

Summary of Smooth Scan

- **Statistics-oblivious** access path
- Uses region snooping to **morph** between alternatives
- **Near-optimal performance** for all selectivities

IMPACT

- **Removes** access path selection **decision**
- Improves **predictability** by **reducing variability** in query execution

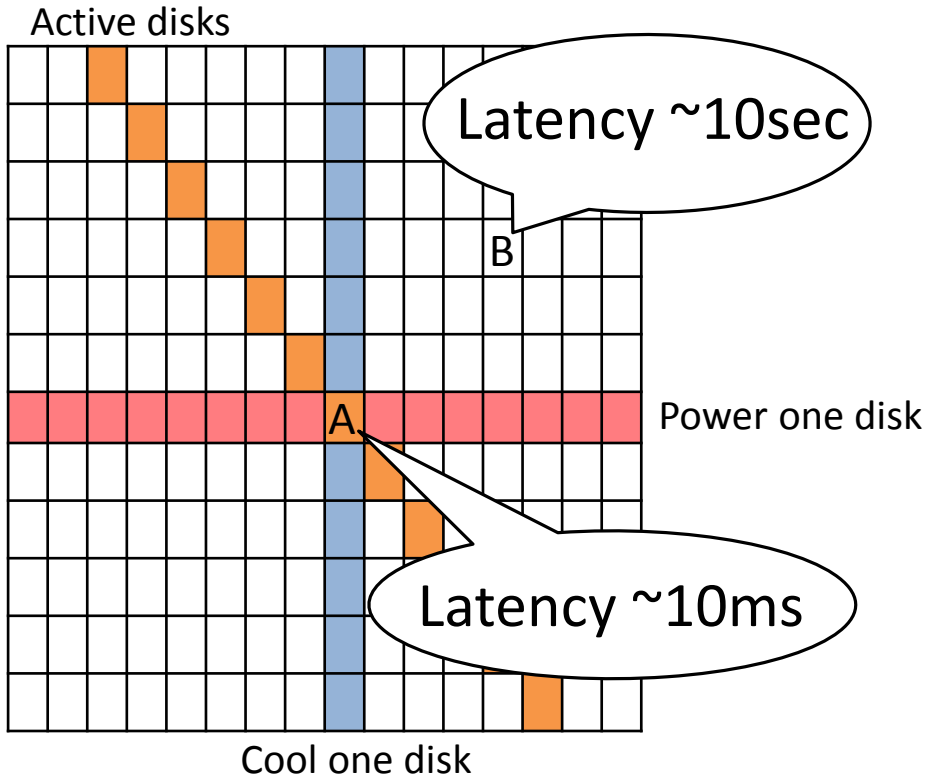
Outline

- **Minimise data-to-insight time**
  - Workload-driven adaptation [CACM'15, SIGMOD'12, VLDB'12]
- **Improve predictability of response time**
  - Data-driven adaptation [ICDE'15, DBTest'12]
- **Reduce analytics cost**
  - Cold storage & hardware-driven adaptation [VLDB'16]

Proliferation of cold data

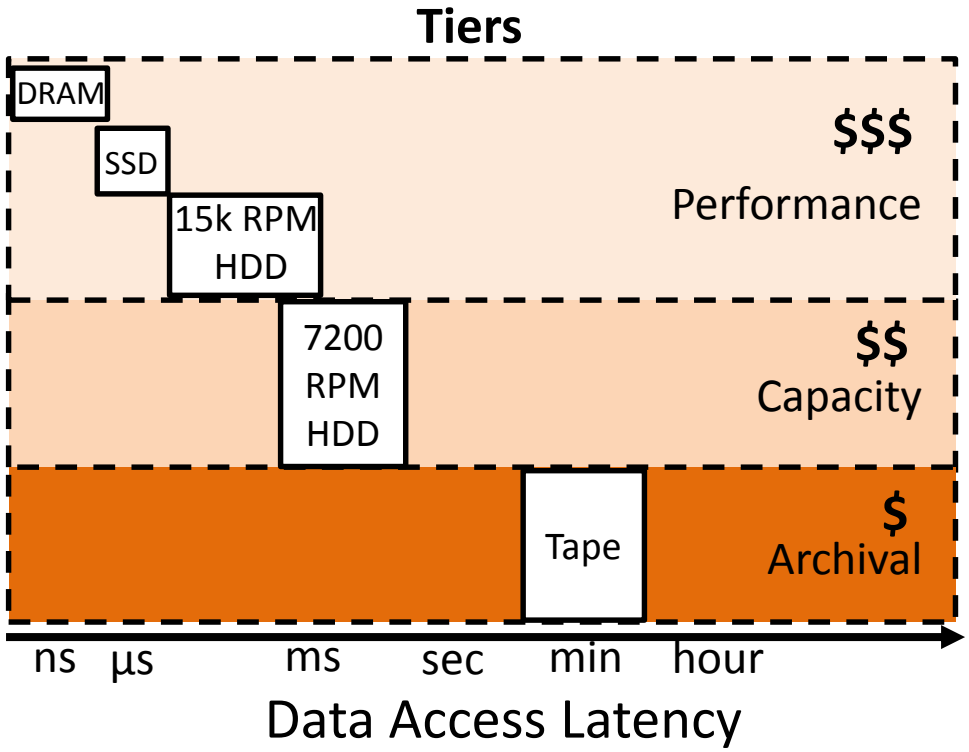
*"80% enterprise data is **cold** with 60% CAGR" [Horison, 2015]*
"cold data: incredibly valuable for analysis" [Intel, 2013]

Cold Storage Devices (CSD) to the rescue

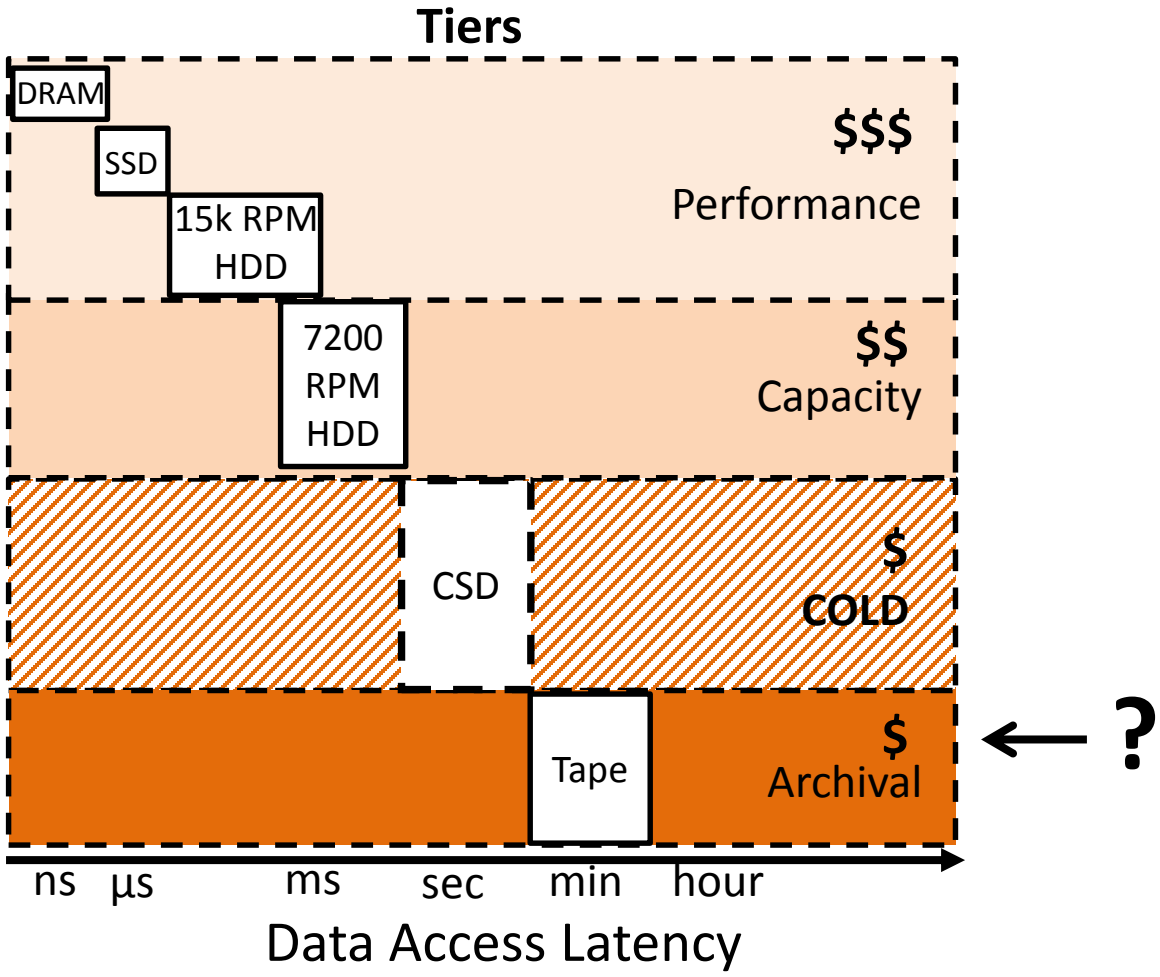


PB-size storage at cost ~ tape and latency ~ disks

CSD in the storage tiering hierarchy

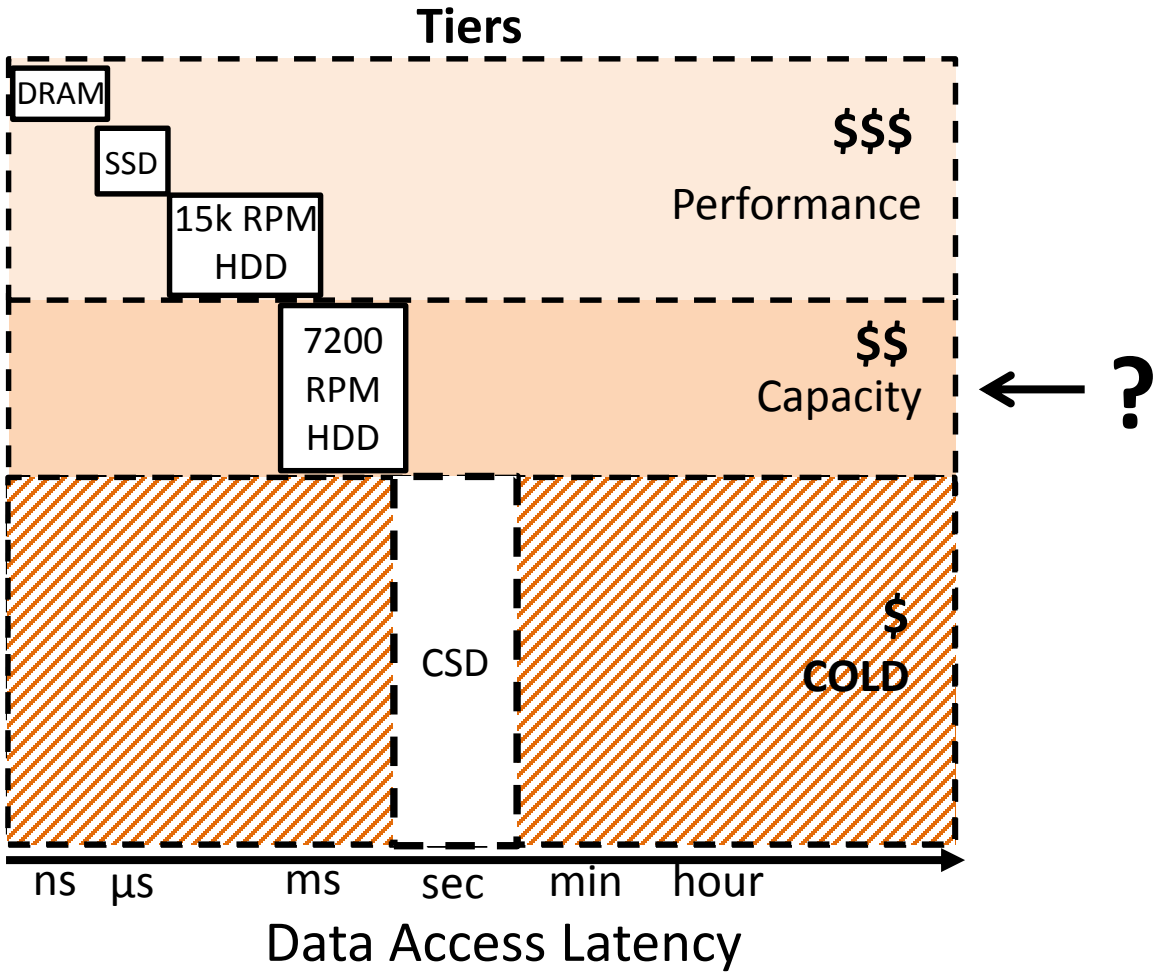


CSD in the storage tiering hierarchy



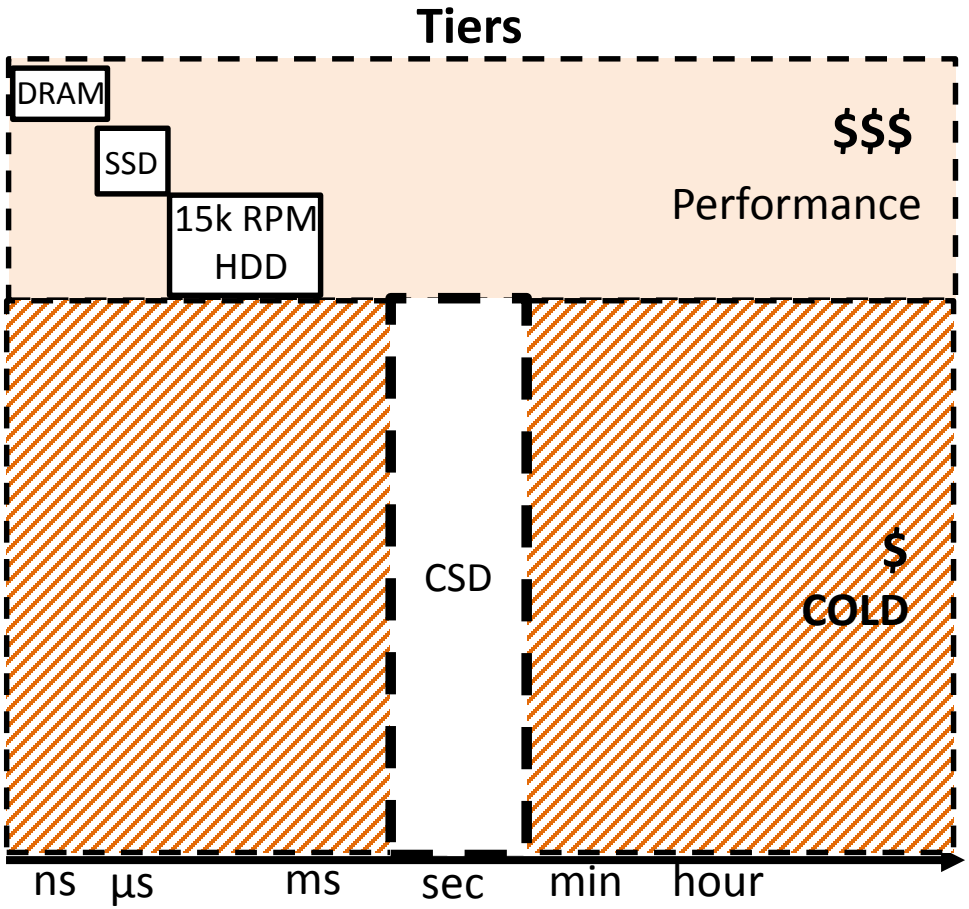
Can we shrink tiers to reduce cost?

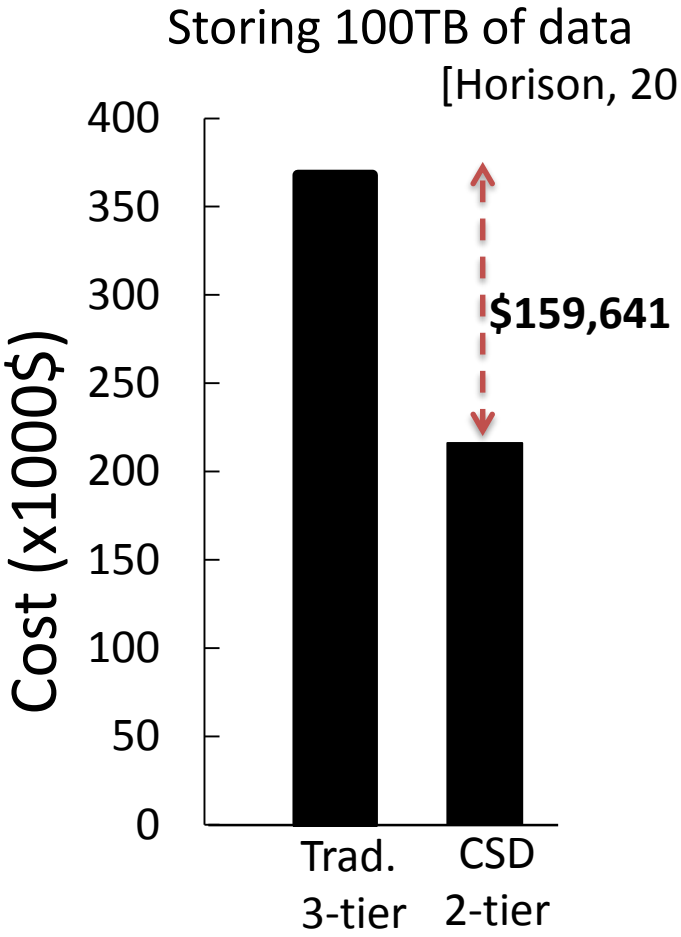
CSD in the storage tiering hierarchy



Can we shrink tiers to reduce cost?

CSD in the storage tiering hierarchy



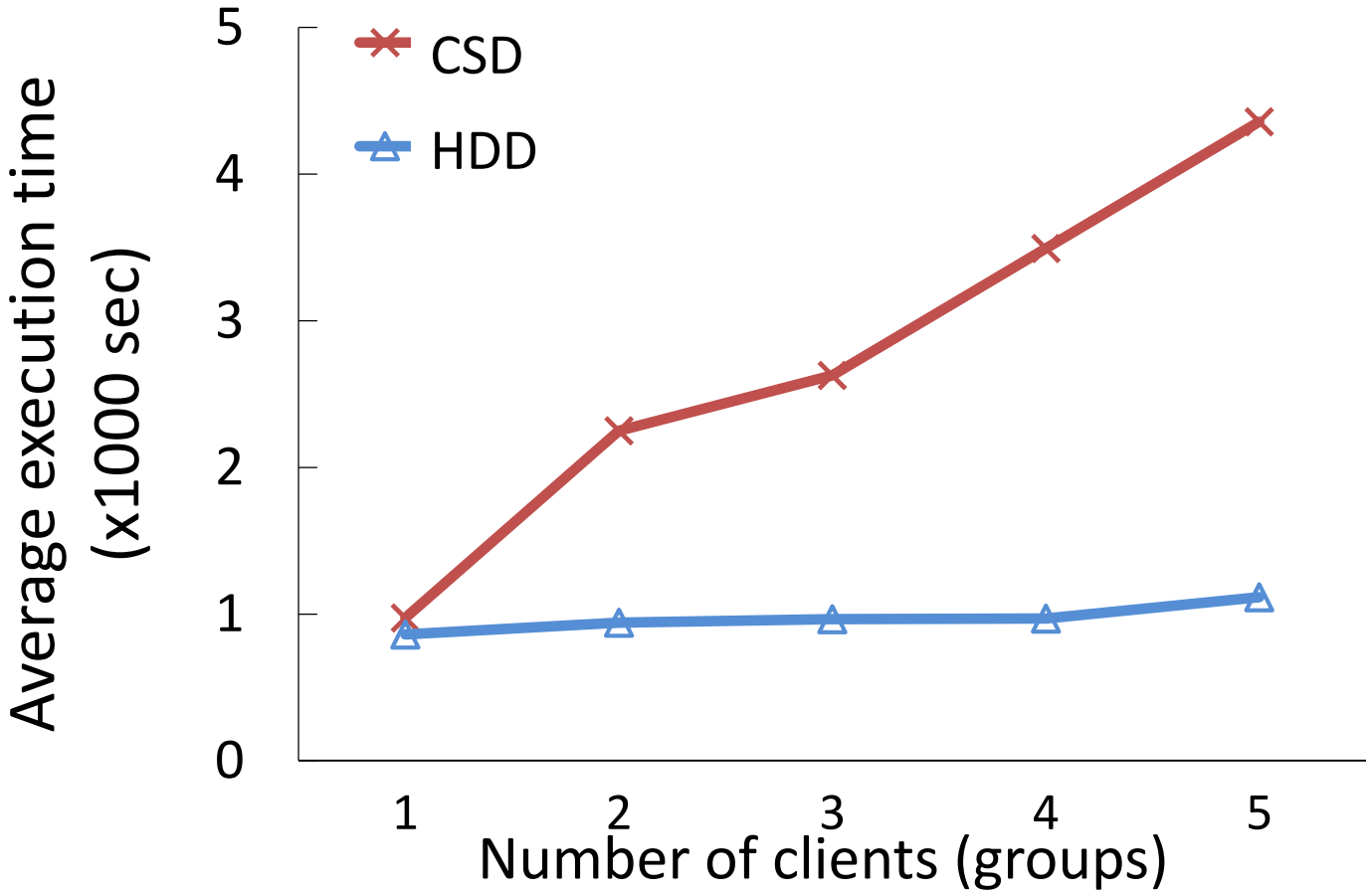


CSD offer significant cost savings (40%)

But … can we run queries over CSD?

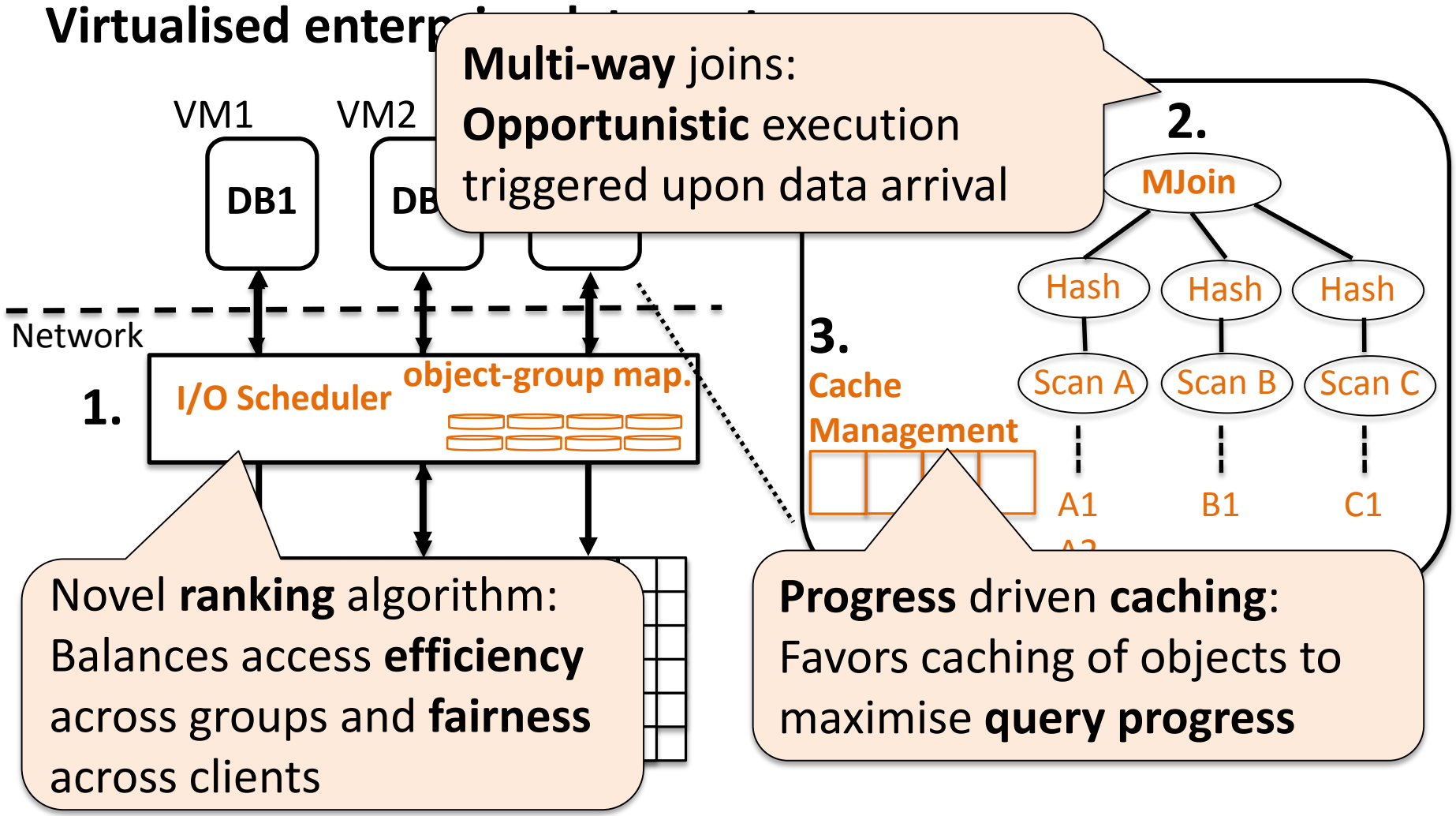
Query execution over CSD

Setting: virtualised enterprise datacenter, clients: PostgreSQL , TPCH 50, Q12,
CSD: shared, layout: one client per group



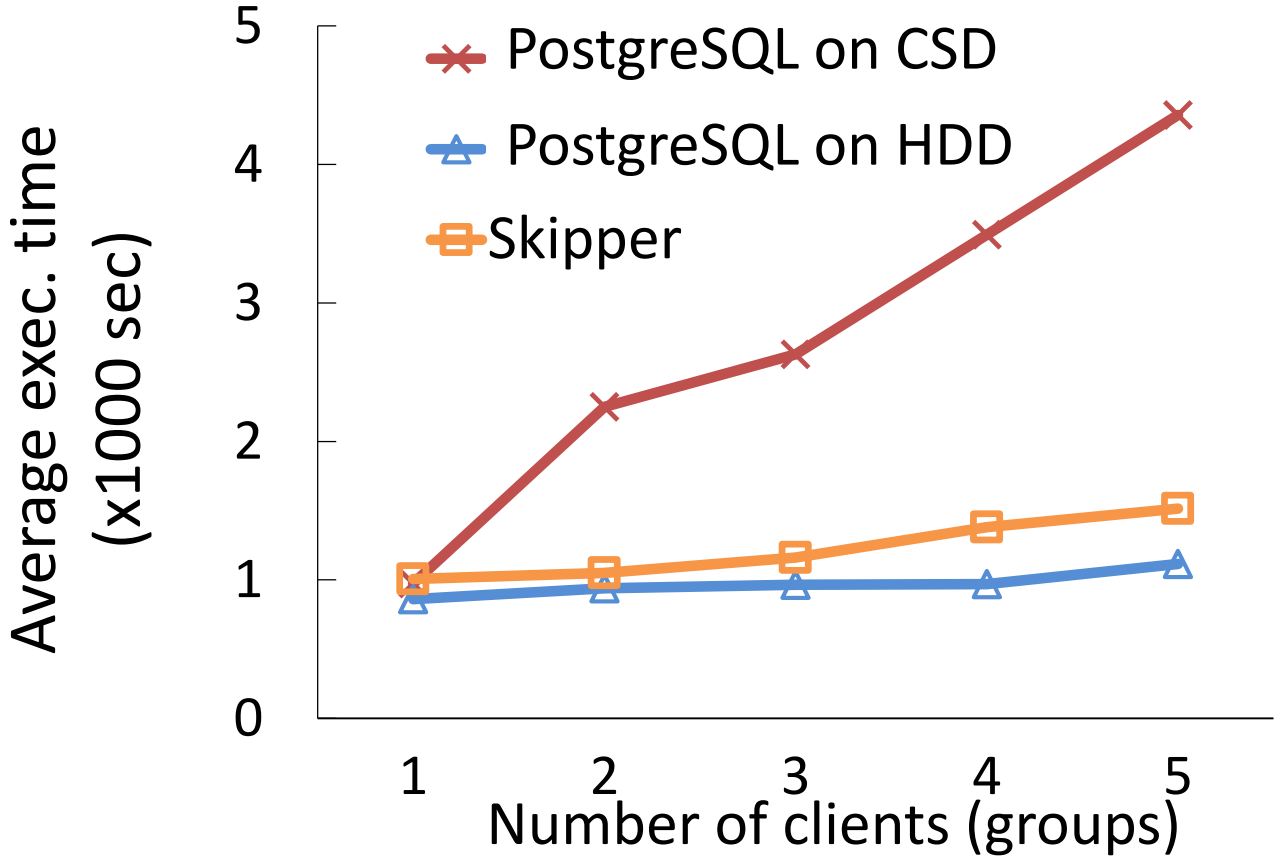
Lost opportunity: CSD relegated to archival storage

Skipper to the rescue



Skipper in action

Setting: multitenant enterprise datacenter, clients: TPCH 50, Q12,
CSD: shared, layout: one client per group



Approximates HDD-based capacity tier by 20% avg.

Summary of Skipper

- Efficient query execution over CSD with:
  1. **Rank-based** I/O **scheduling**
  2. Out-of-order execution based on **multi-way joins**
  3. **Progress based caching** policy
- Approximates **performance** of **HDD-based storage** tier

IMPACT

- Cold storage **can reduce TCO** by **shrinking** storage hierarchy
- Skipper enables data analytics-over-CSD-as-a-service

Contributions

- **Minimise data-to-insight time**
  - Workload-driven adaptation
  - **Skip loading, tune as a byproduct of query execution**
- **Improve predictability of response time**
  - Data-driven adaptation
  - **Remove access decisions a priori, transform gradually**
- **Reduce analytics cost**
  - Cold storage & hardware-driven adaptation
  - **From plan pull-based to hardware push-based execution**
- **Uncertainty cured with adaptivity**

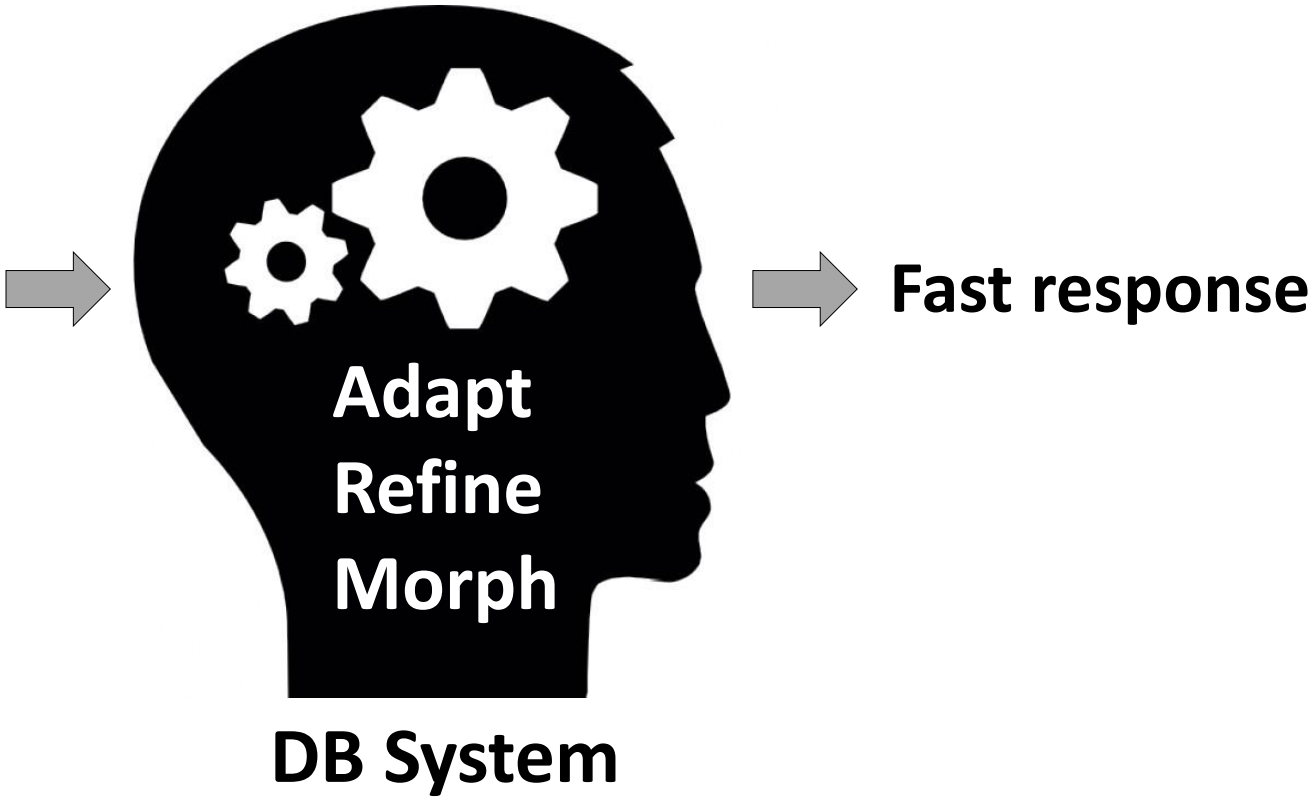
Self-designing systems for data analysis

"It is not the strongest species that survive, nor the most intelligent, but the ones most responsive to change." Charles Darwin

Queries
[SIGMOD'12]
[VLDB'12]
[CACM'15]

Data
[DBTest'12]
[ICDE'15]

Hardware
[VLDB'16]



Everyone can be a data scientist with intelligent DB

Looking ahead

Business analyst

Source: *

Source: ₸

Data analysis for the masses

Data classification

Dynamic query plans

Approximate answers

Storage layouts

HW-SW co-design

* http://reportlogix.com/reporting.html ₸ www.tableausoftware.com

My collaborators




Anastasia Ailamaki, EPFL & Raw Labs
Ioannis Alagiannis, EPFL
Miguel Branco, EPFL & Raw Labs
Raja Appuswamy, EPFL

Stratos Idreos,
Harvard University

HARVARD School of Engineering and Applied Sciences

Thomas Heinis,
EPFL & Imperial

Imperial College London

Satya Valluri,
EPFL & Oracle

ORACLE

Farhan Tauheed,
EPFL & Oracle

ORACLE
Oracle Labs

Campbell Fraser,
Google

Google

Marcin Zukowski,
Snowflake

snowflake

Thank you!

Questions?

THANK YOU

Related publications

- **[VLDB'16]** R. Borovica-Gajic, R. Appuswamy and A. Ailamaki. *Cheap Data Analytics Using Cold Storage Devices*. VLDB, 2016.
- **[CACM'15]** I. Alagiannis, R. Borovica-Gajic, M. Branco, S. Idreos and A. Ailamaki. *NoDB: Efficient Query Execution on Raw Data Files*. Communications of the ACM, Research Highlights, 2015.
- **[ICDE'15]** R. Borovica-Gajic, S. Idreos, A. Ailamaki, M. Zukowski, and C. Fraser. *Smooth Scan: Statistics-Oblivious Access Paths.* ICDE, 2015.
- **[SIGMOD'12]** I. Alagiannis, R. Borovica, M. Branco, S. Idreos and A. Ailamaki. *NoDB: Efficient Query Execution on Raw Data Files.* SIGMOD, 2012.
- **[VLDB'12]** I. Alagiannis, R. Borovica, M. Branco, S. Idreos and A. Ailamaki. *NoDB in Action: Adaptive Query Processing on Raw Data.* VLDB, 2012. (demo)
- **[DBTest'12]** R. Borovica, I. Alagiannis and A. Ailamaki. *Automated Physical Designers: What You See is (Not) What You Get.* DBTest, 2012.
- **[IEEE'11]** T. Heinis, M. Branco, I. Alagiannis, R. Borovica and F. Tauheed et al. *Challenges and Opportunities in Self-Managing Scientific Databases.* IEEE Data Eng. Bull, 2011.