Rackscale- the things that matter

GUSTAVO ALONSO
SYSTEMS GROUP
DEPT. OF COMPUTER SCIENCE
ETH ZURICH

Systems@ETH zürich

HTDC 2014

Systems Group = www.systems.ethz.ch

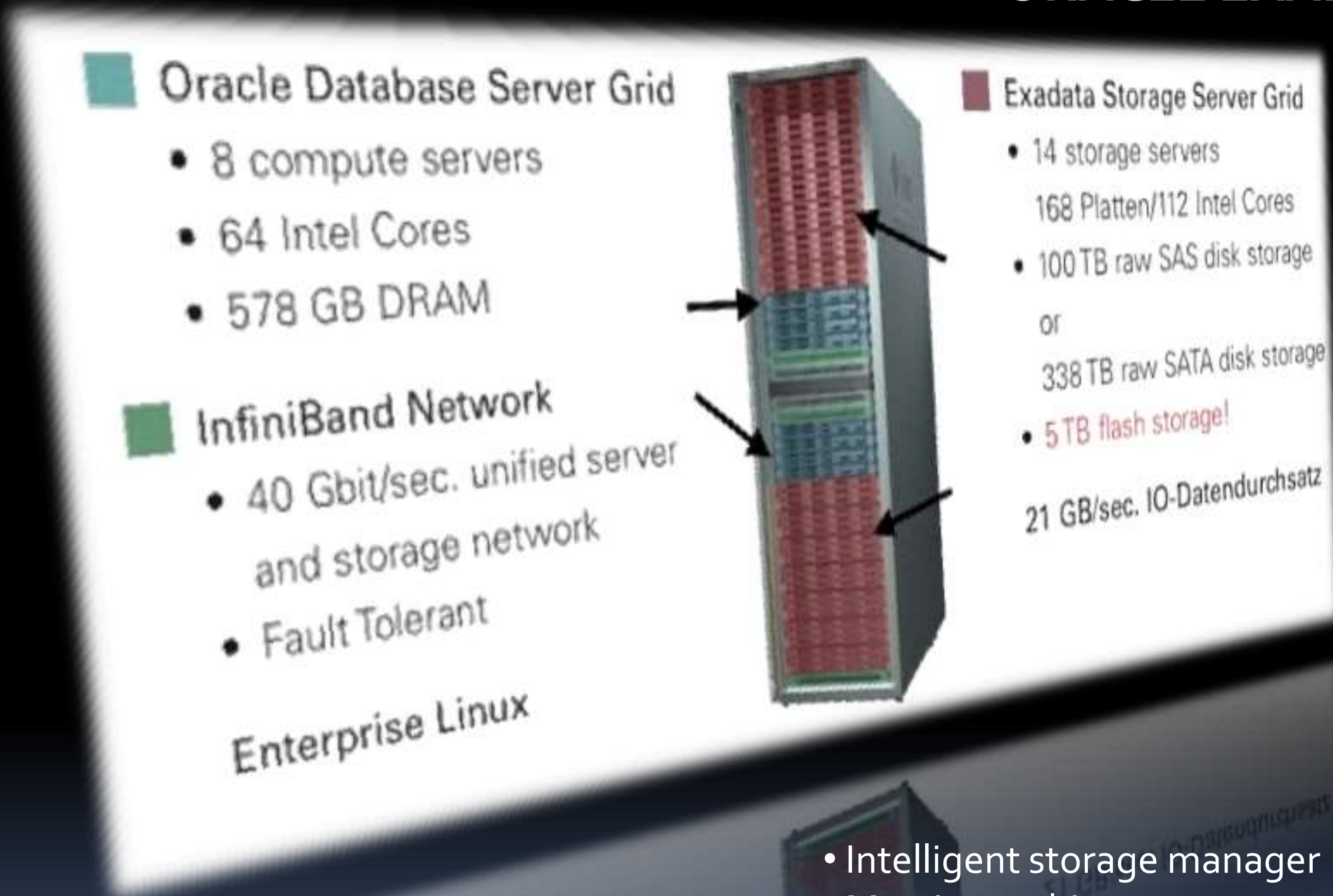
Enterprise Computing Center = www.ecc.ethz.ch

On the way to VU, this morning …

One size does not fit all ?

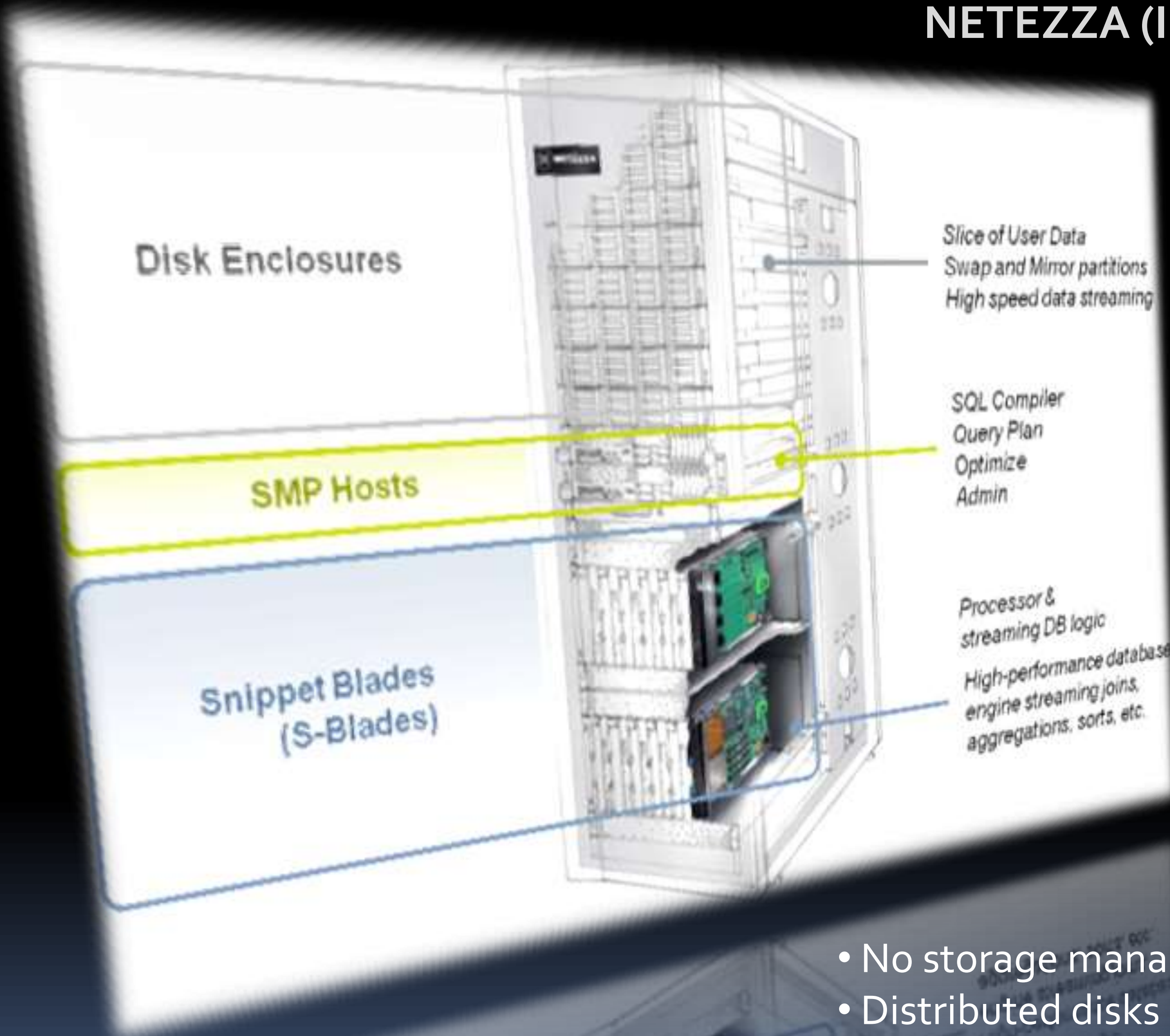
Systems@ETH Zürich

ORACLE EXADATA



- Intelligent storage manager
- Massive caching
- RAC based architecture
- Fast network interconnect

NETEZZA (IBM) TWINFIN



- No storage manager
- Distributed disks (per node)
- FPGA processing
- No indexing

Systems@ETH Zürich

Hardware rules

- Multicore, Many core
- Transactional Memory
- SIMD, AVX, vectorization
- SSDs, persistent memory
- Infiniband, RDMA
- GPUs, FPGAs (hardware acceleration)
- Intelligent storage engines, main memory
- Database appliances

- Reacting to changes we do not control

What does it mean?

- Homogeneous inside
  - The components will still be mostly general purpose
  - Economies of scale
- Heterogeneous outside
  - Systems tailored to the application
  - Performance through customization

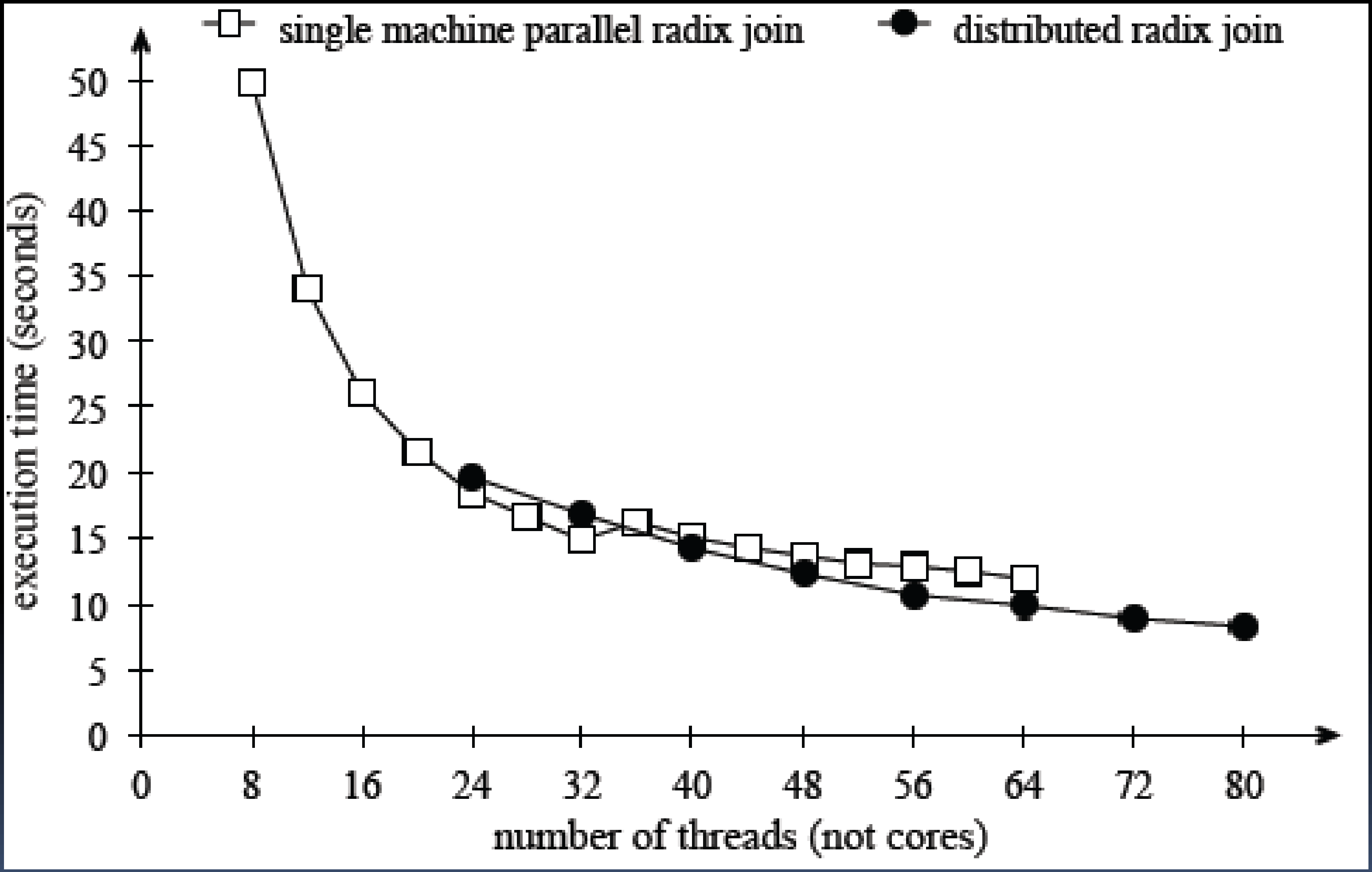
Multicore is great: avoid distribution

Nobody ever got fired for using Hadoop on a Cluster

A. Rowstron, D. Narayanan, A. Donnely, G. O'Shea, A. Douglas
HotCDP 2012, Bern, Switzerland

- Analysis of MapReduce workloads:
  - Microsoft: median job size < 14 GB
  - Yahoo: median job size < 12.5 GB
  - Facebook: 90% of jobs less than 100 GB
- Fit in main memory
- One server more efficient than a cluster
- Adding memory to a big server better than using a cluster

Where is the heterogeneity?



The take away message

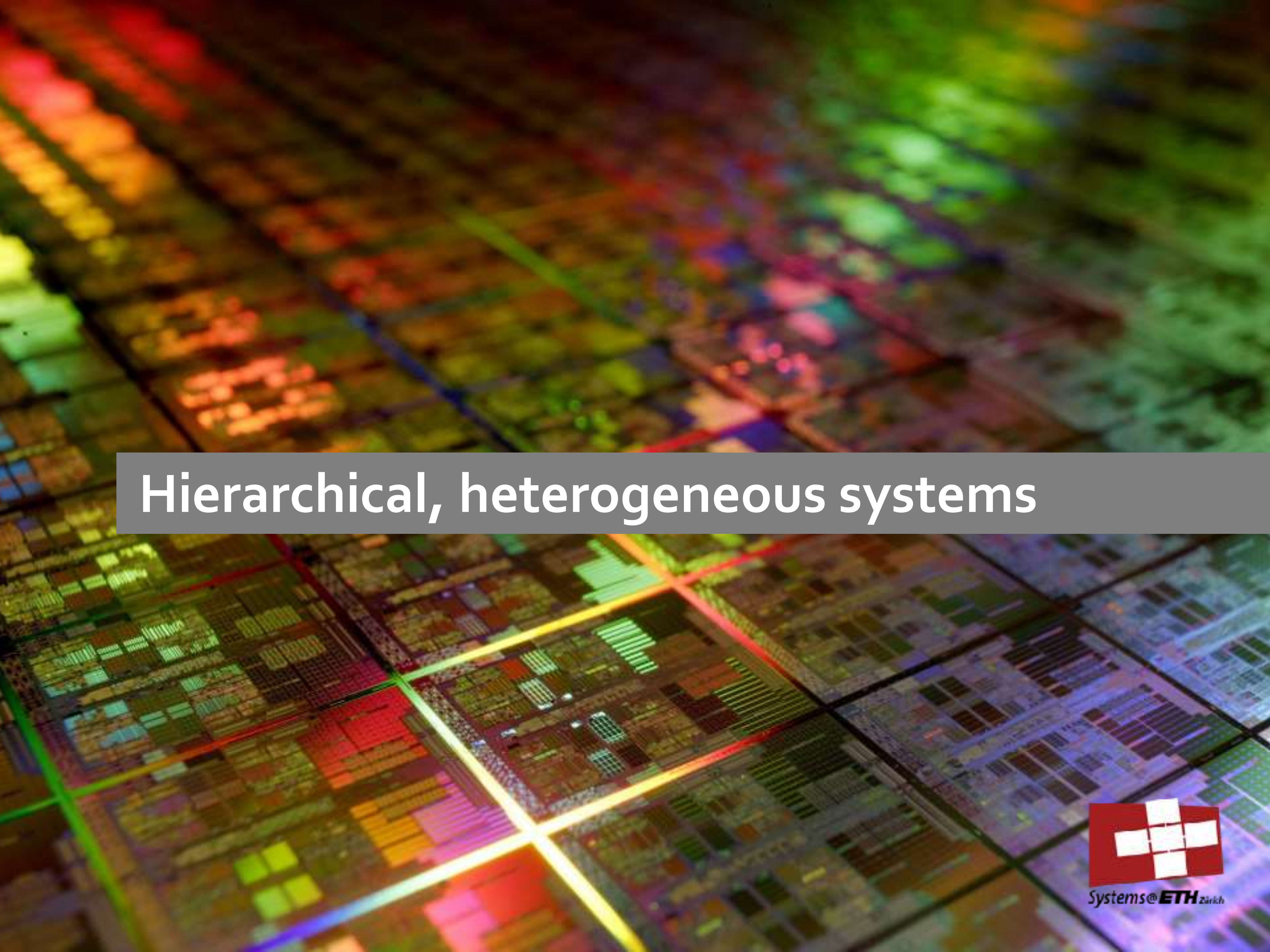
- Easy to build a customized system addressing one use case
  - Less and less interesting
- Difficult to design techniques and tools for developing customized systems
  - Increasingly relevant

Systems@ETH zürich

What matters

- Hierarchical, heterogeneous processors
  - Processing at all levels
- Using the hardware, knowing the load
  - Determining what to run where
- The case for sharing
  - Batch processing rather than single jobs
- It is the data, stupid
  - What a system can do and what it cannot do

Hierarchical, heterogeneous systems

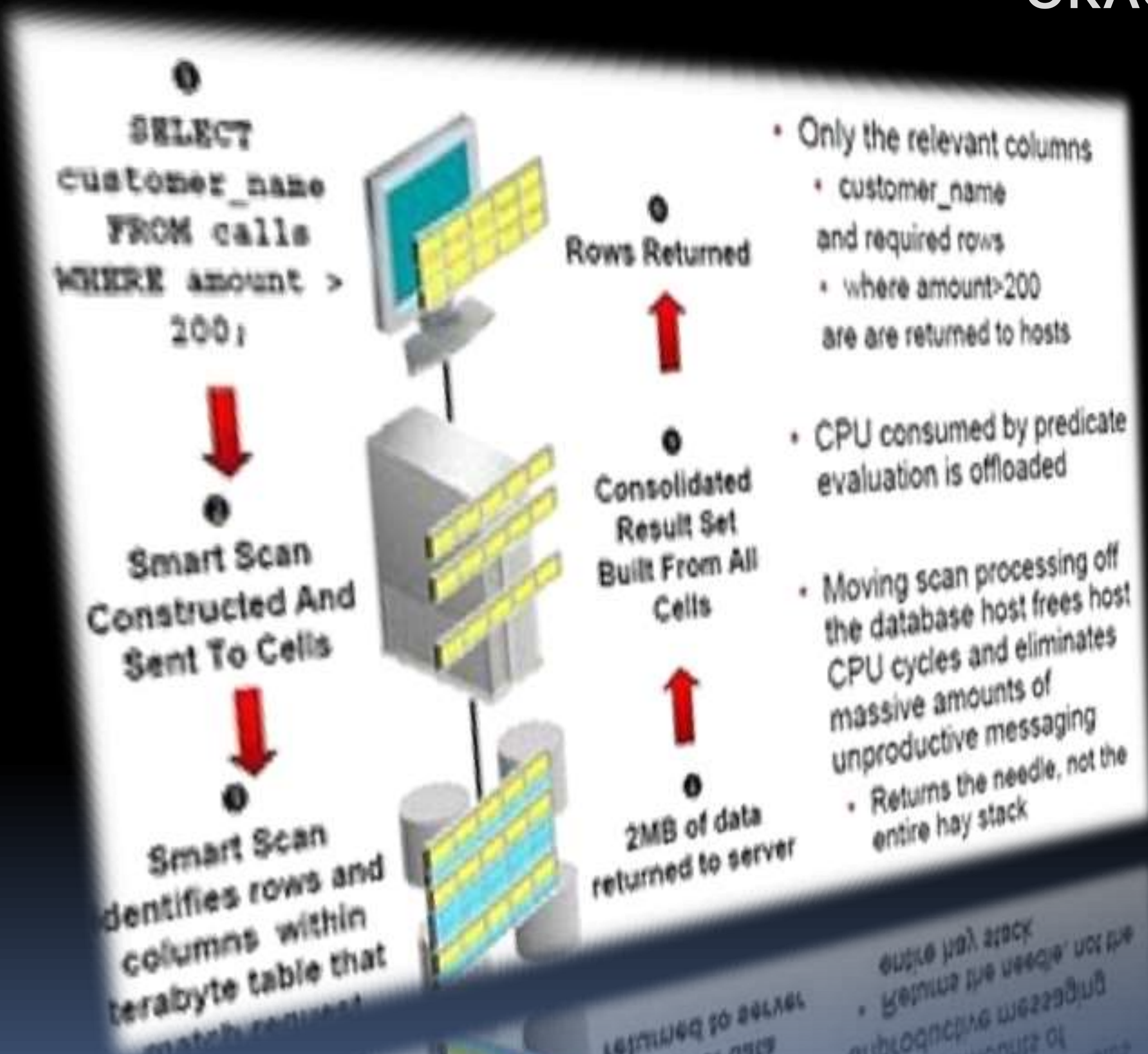


In the future …

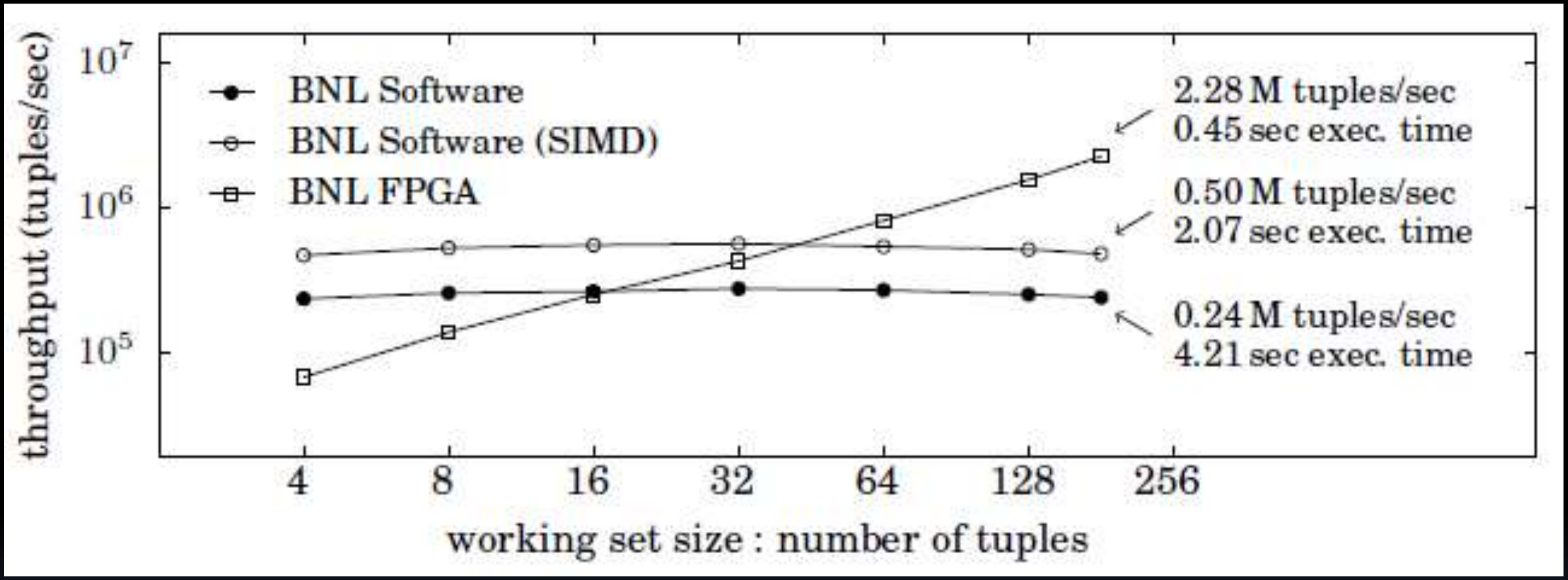
- Expect hardware acceleration everywhere:
  - Co-processors
  - Intelligent storage
  - Intelligent (active) memory
  - In-network data processing
  - Hierarchical configurations to manage complexity

Systems@ETH Zürich

ORACLE EXADATA



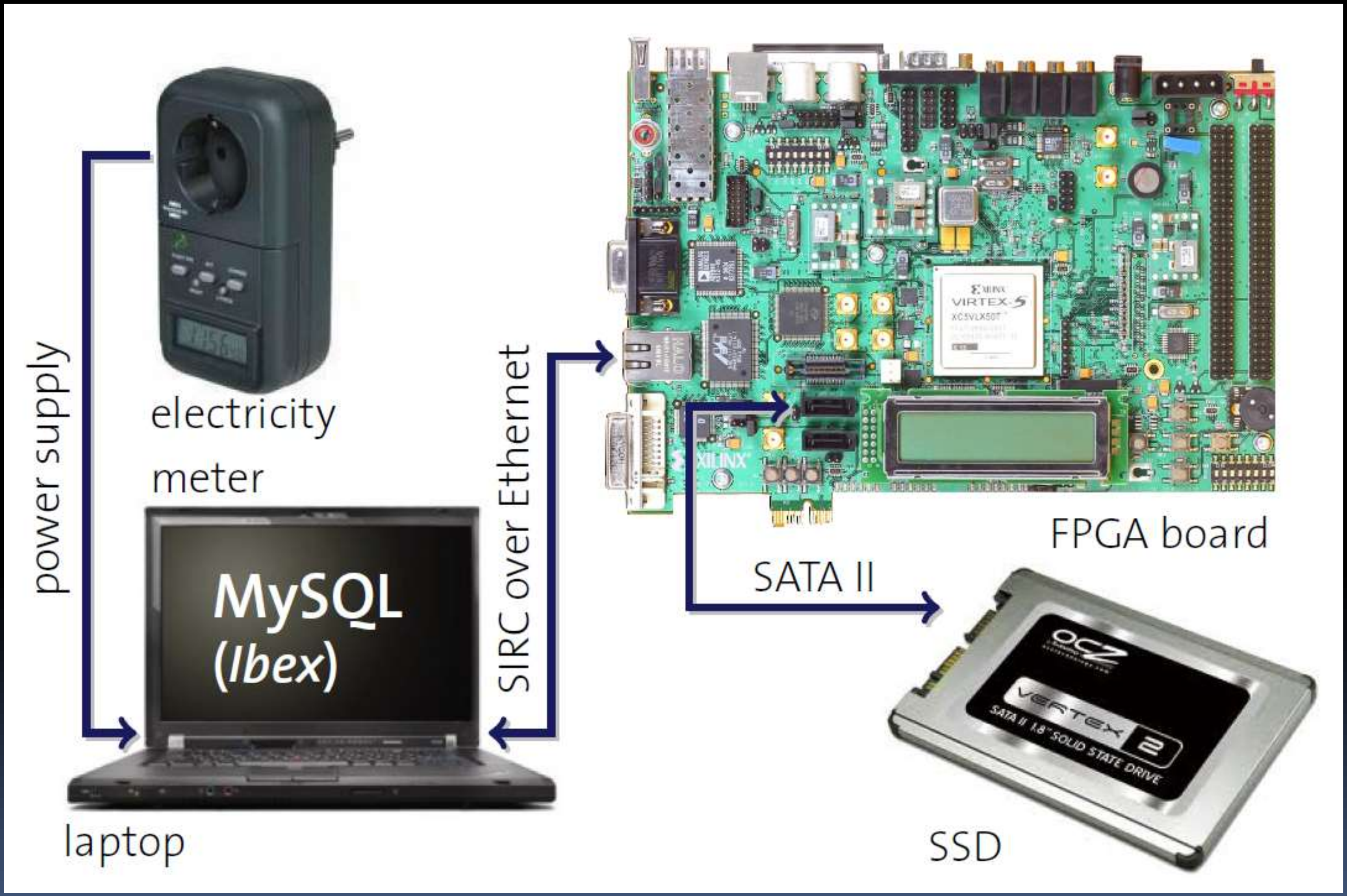
Hardware might solve your problem



Louis Woods, Gustavo Alonso, Jens Teubner:
Parallel Computation of Skyline Queries. FCCM 2013

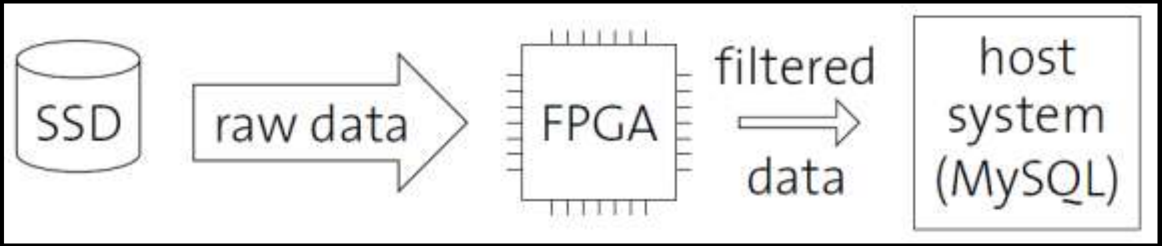
Systems@ETH Zürich

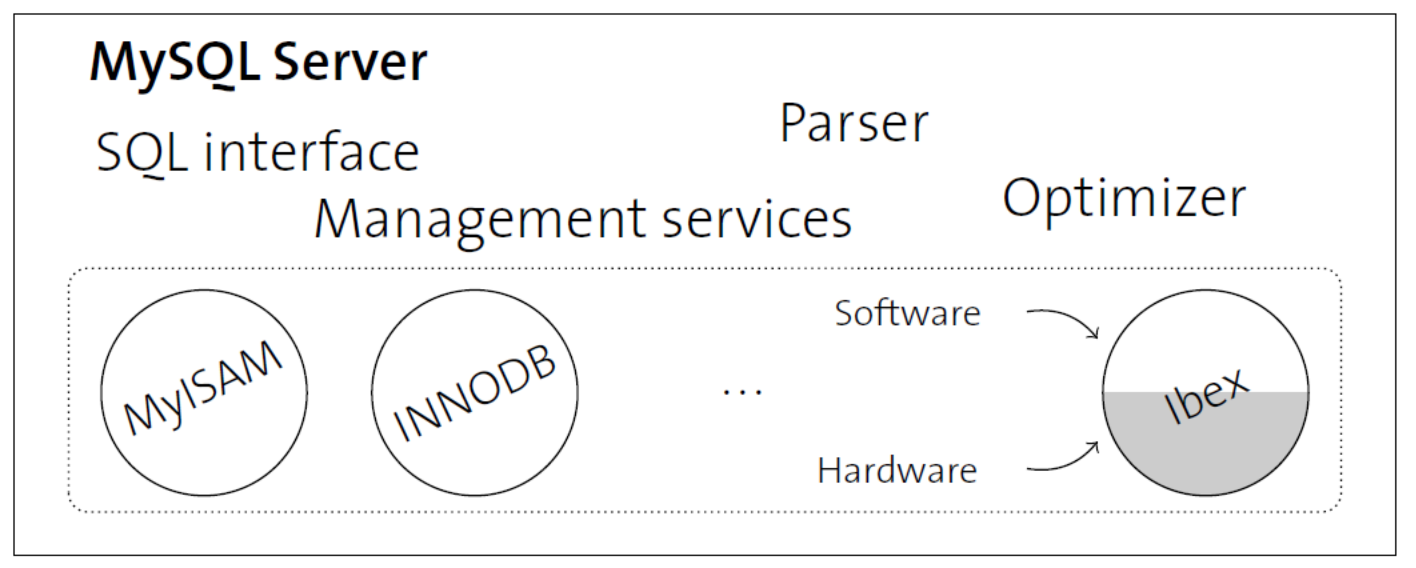
Ibex = Intelligent storage engine



Systems@ETH zürich

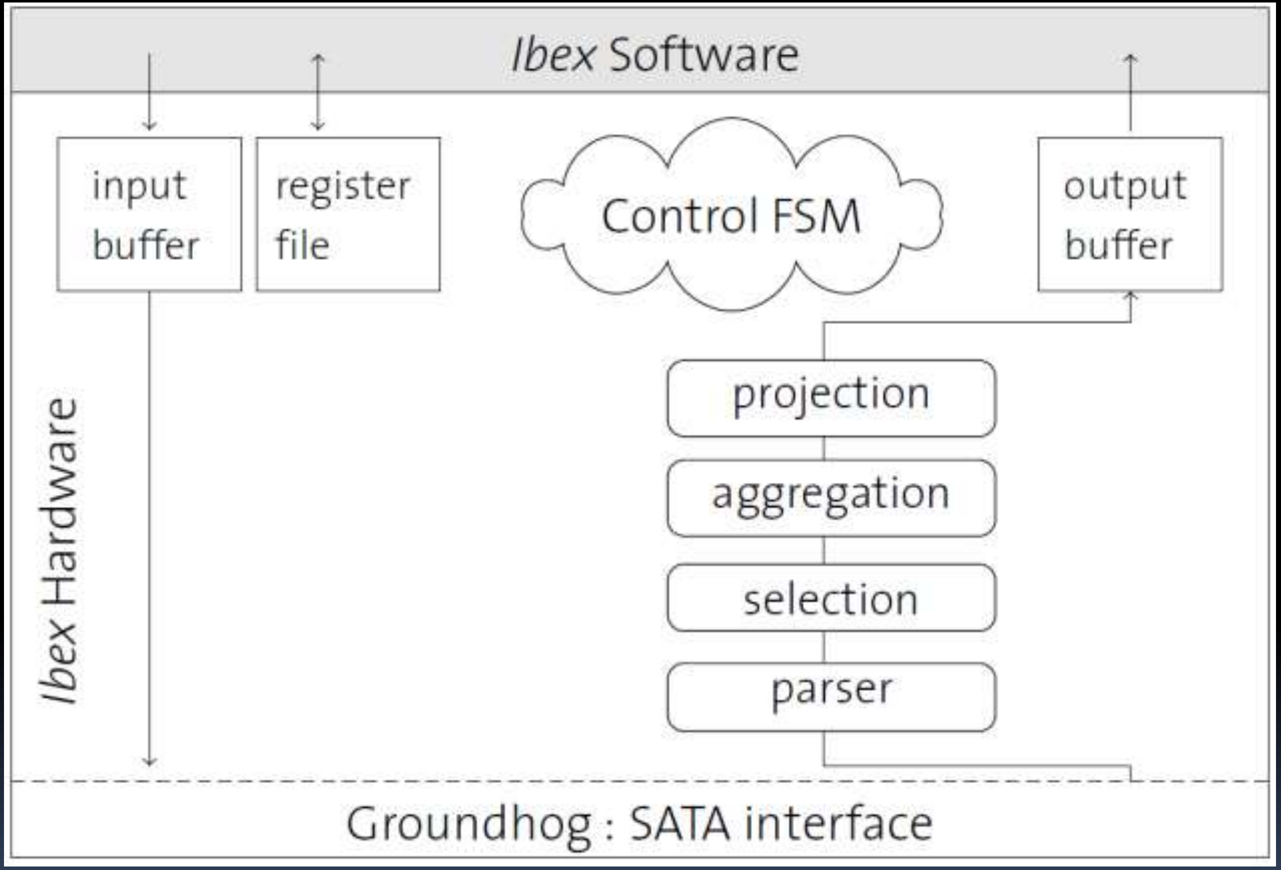
Inserting the FPGA in the data path



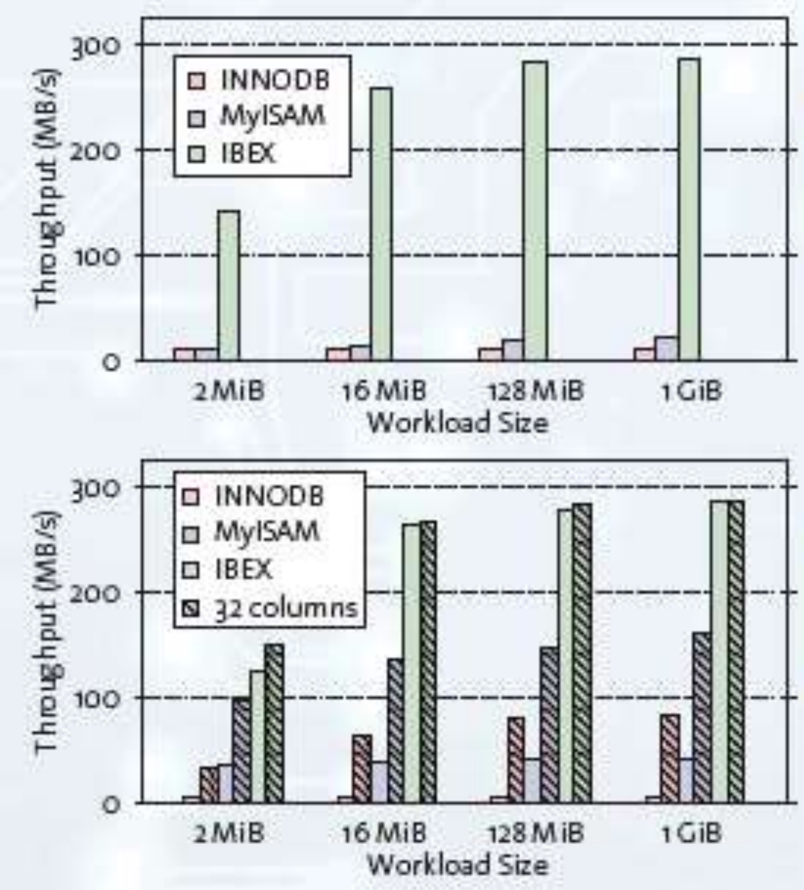


Systems@ETH Zürich

Engine design



So far so good




```
SELECT   a, COUNT (*)
FROM     table AS t
GROUP BY a
```

```
SELECT *
FROM   table AS t
WHERE  a = const
```

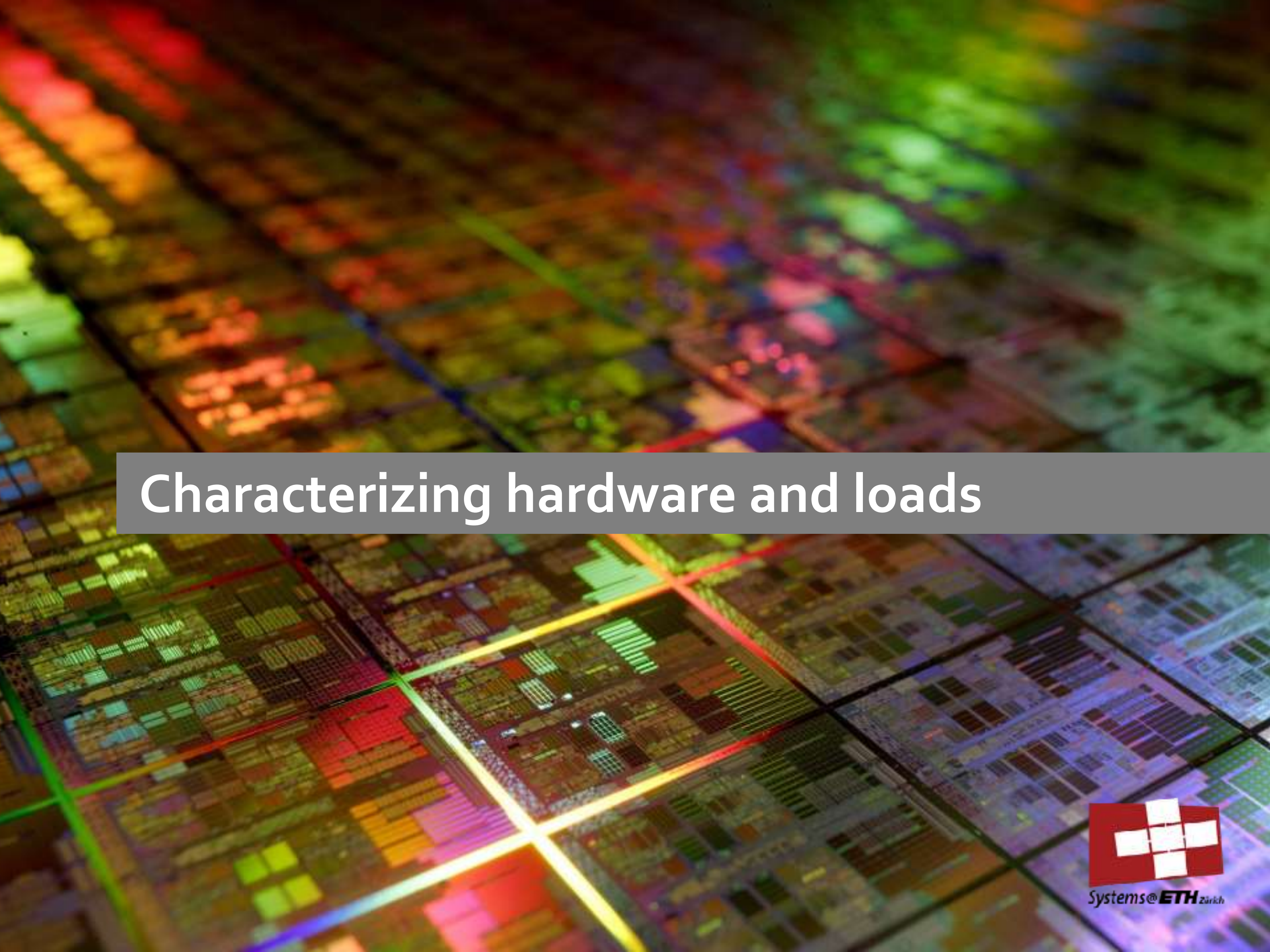
Points of interest

Query/Storage Engine	Δ-Power	Energy Consumption
Point Query / MyISAM	22 watts	864 joules
Point Query / INNODB	24 watts	7380 joules
Point Query / *Ibex*	3 watts	216 joules
Hybrid Join / MyISAM	22 watts	864 joules
Hybrid Join / INNODB	24 watts	7380 joules
Hybrid Join / *Ibex*	3 watts	216 joules
Group By / MyISAM	22 watts	864 joules
Group By / INNODB	24 watts	7380 joules
Group By / *Ibex*	3 watts	216 joules

CPU usage when executing GROUP BY

INNODB

Ibex



Characterizing hardware and loads

Systems@ETH zürich

Deployment and scheduling

- The times of over provisioning are over:
  - Too expensive
  - No longer politically correct
  - No switch on and off (too expensive)
- Dynamic deployment and scheduling
  - More complex loads
  - More data movement
  - More heterogeneous hardware

Systems@ETH Zürich

Heterogeneity is a mess

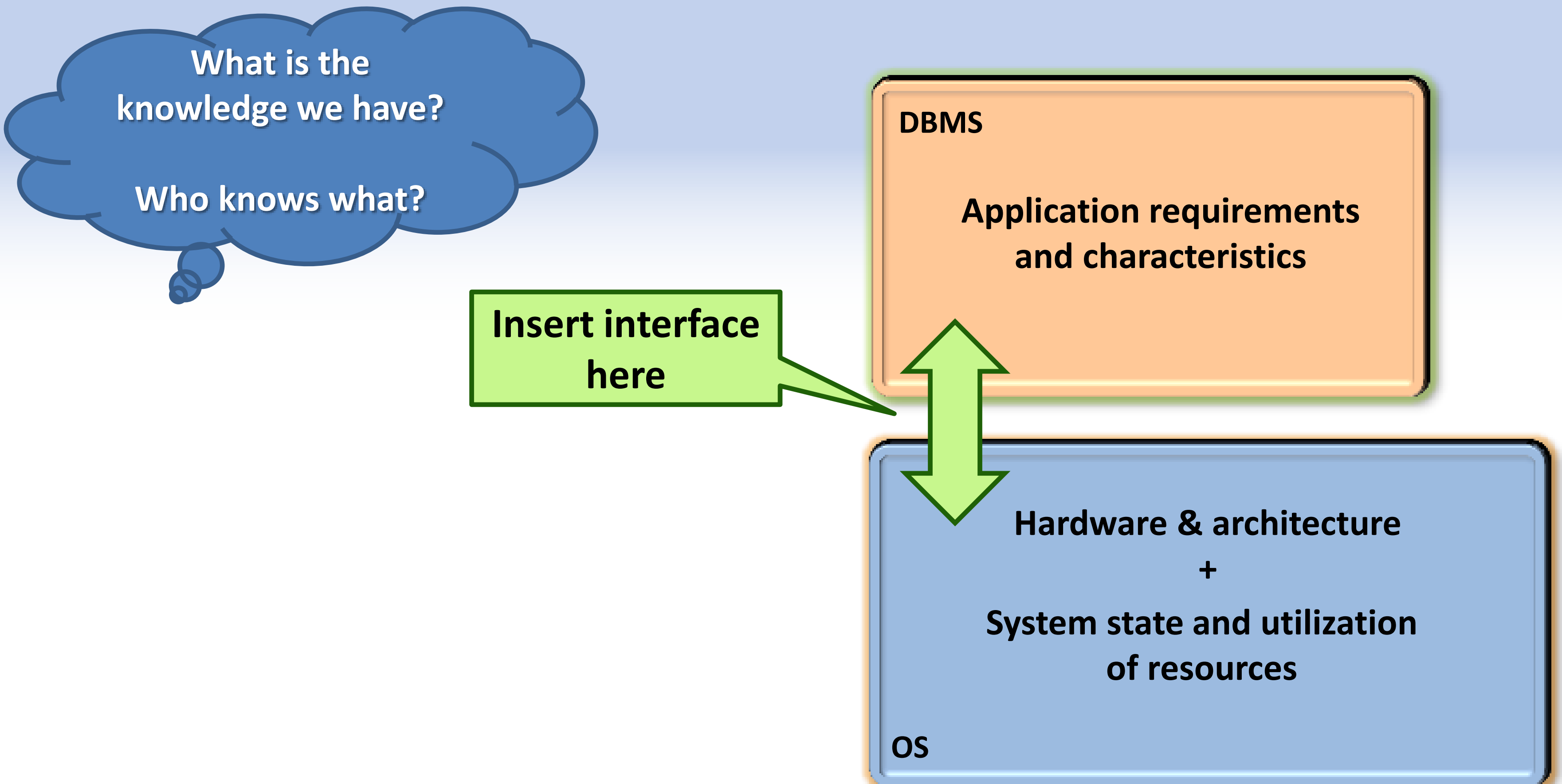
Example: deployment on multicores

Experiment setup
- 8GB datastore size
- SLA latency requirement 8s
- 4 different machines

	Min Cores	Partition Size [GB]	RT [s]
Intel Nehalem	2	4	6.54
AMD Barcelona	5	1.6	3.55
AMD Shanghai	3	2.6	4.33
AMD MagnyCours	2	2	7.37

Jana Giceva, Tudor-Ioan Salomie, Adrian Schüpbach, Gustavo Alonso, Timothy Roscoe: COD: Database / Operating System Co-Design. CIDR 2013

COD : Overview



Cod's Interface

supports

DB storage engine

OS

Policy Engine

Push application-specific facts:

- #Requests (in a batch)
- Datastore size (#Tuples, and TupleSize)
- SLA response time requirement

Needed for:

- cost / utility functions:

$$RT[ms] = c \times \frac{\#\,tuples}{\#\,cores} \times (a \times \#queries + b)$$

Application-specific

DB-specific Facts & properties

Cost functions

COD's key features

Declarative interface

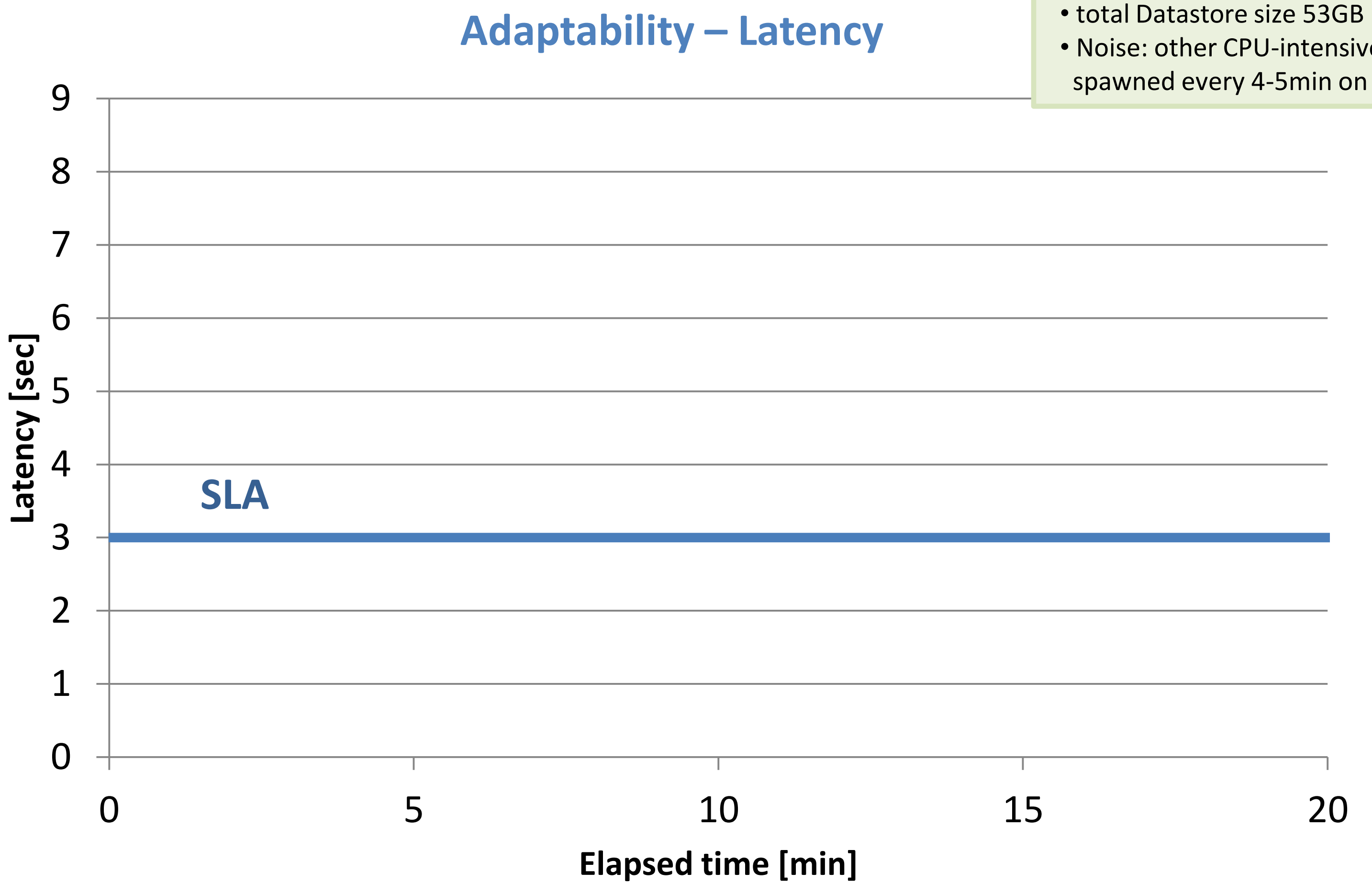
- Resource allocation for imperative requests
- Resource allocation based on cost functions

Proactive interface

- Inform of system state
- Request releasing of resources
- Recommend reallocation of resources

Experimental results

Adaptability to dynamic system state

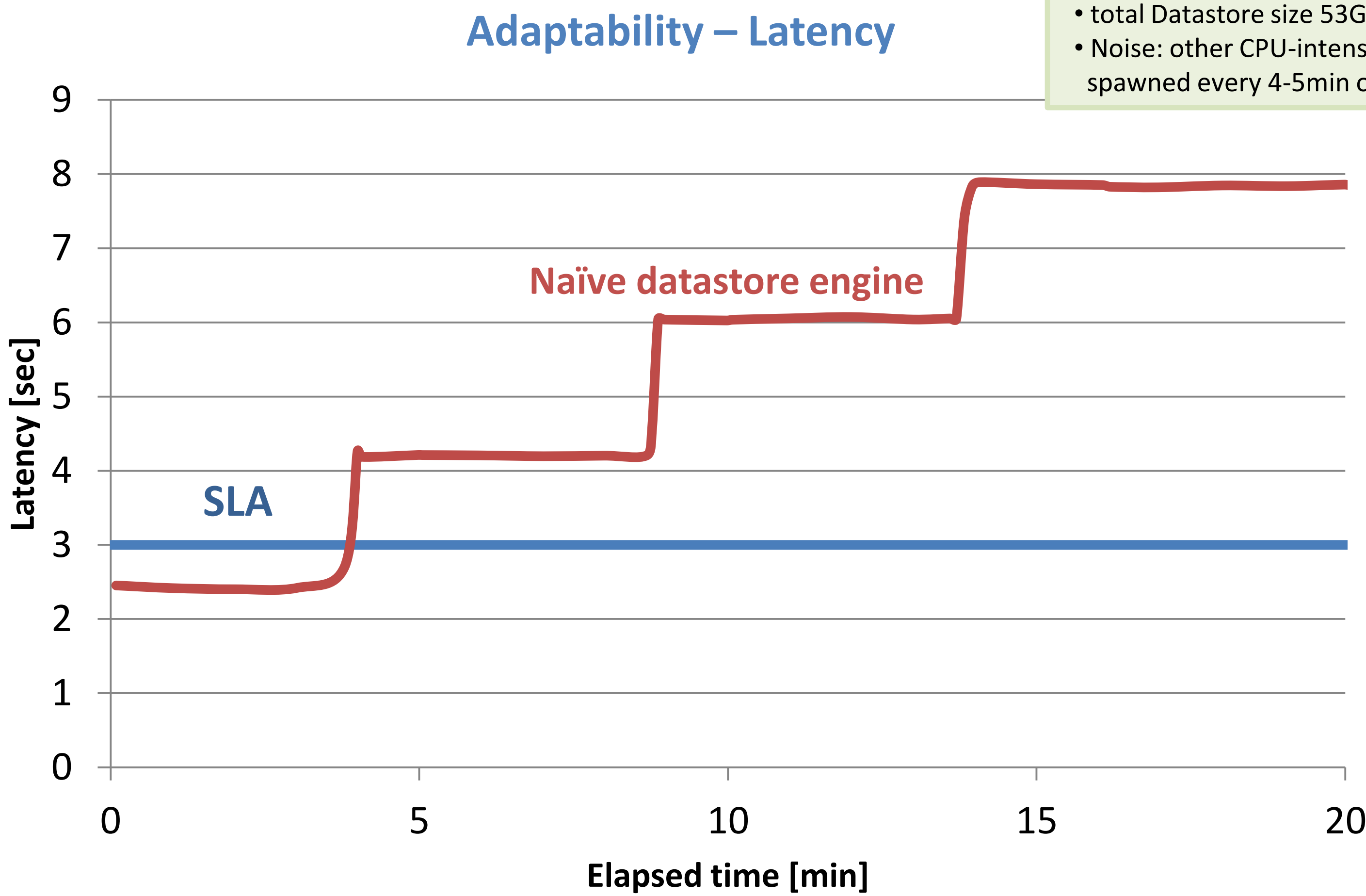


Experiment setup

- AMD MagnyCours
- 4 x 2.2GHz AMD Opteron 6174 processors
- total Datastore size 53GB
- Noise: other CPU-intensive threads spawned every 4-5min on core 0

Experimental results

Adaptability to dynamic system state

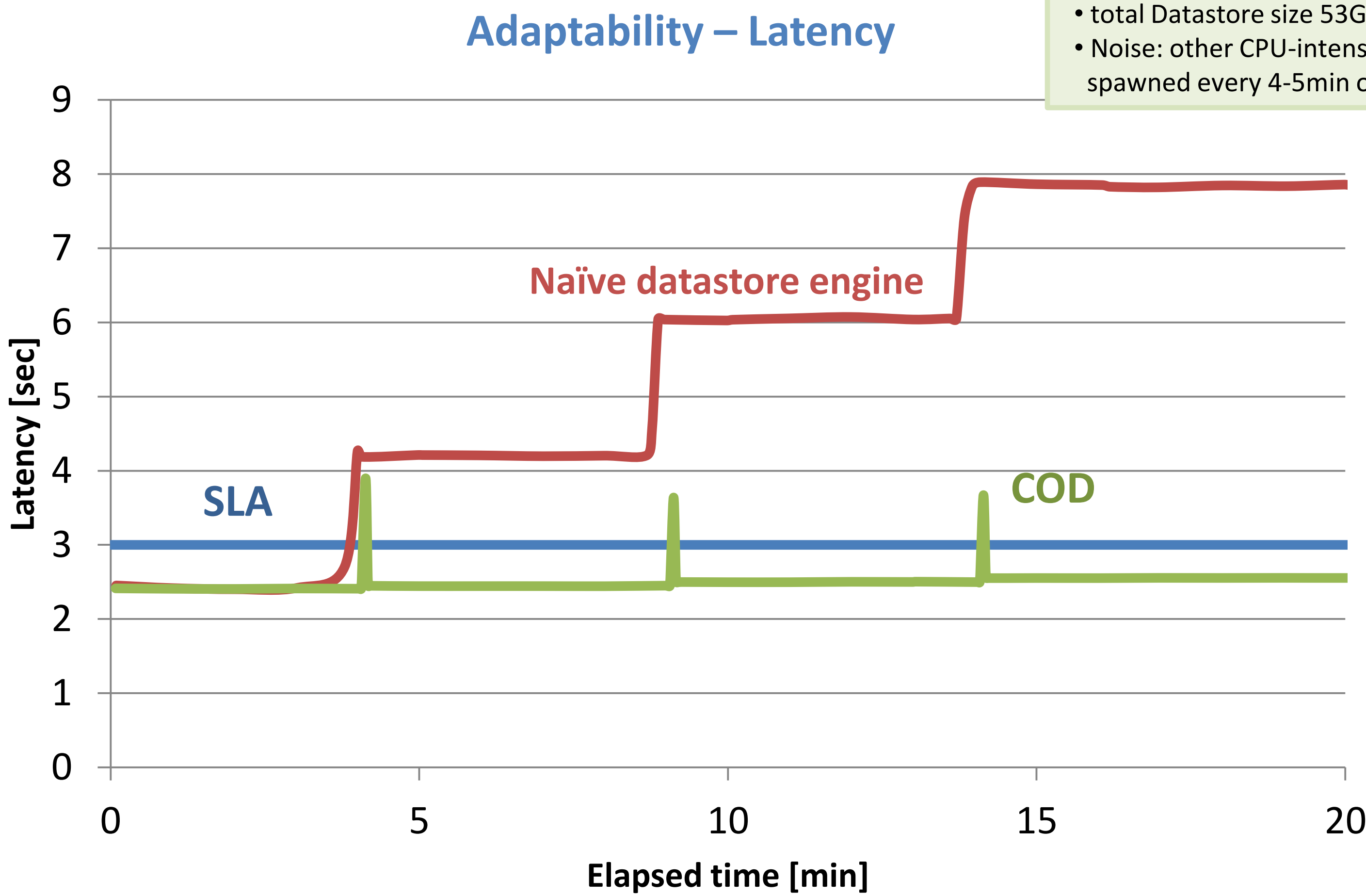


Experiment setup

- AMD MagnyCours
- 4 x 2.2GHz AMD Opteron 6174 processors
- total Datastore size 53GB
- Noise: other CPU-intensive threads spawned every 4-5min on core 0

Experimental results

Adaptability to dynamic system state



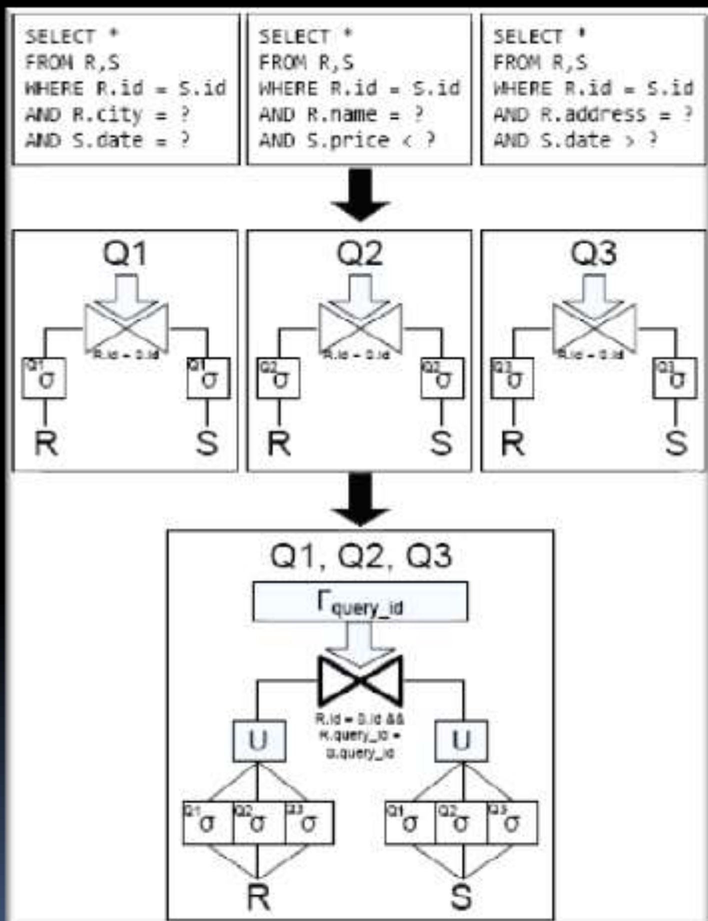
Experiment setup

- AMD MagnyCours
- 4 x 2.2GHz AMD Opteron 6174 processors
- total Datastore size 53GB
- Noise: other CPU-intensive threads spawned every 4-5min on core 0

The case for sharing

Pipeline parallelism

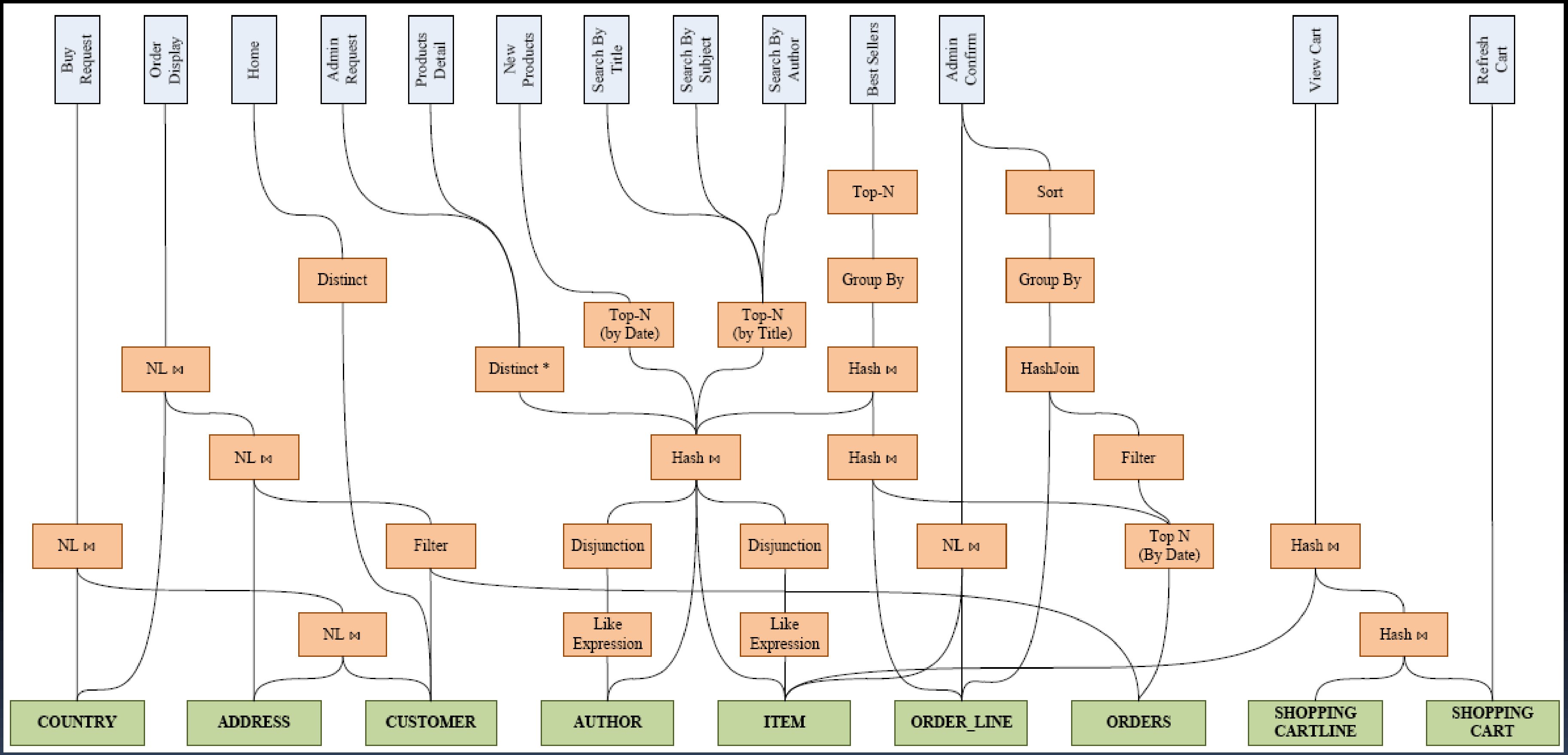
Georgios Giannikis, Gustavo Alonso, Donald Kossmann: SharedDB: Killing One Thousand Queries With One Stone. PVLDB 5(6): 526-537 (2012)



- SharedDB does not run queries individually (each one in one thread). Instead, it runs operators that process queries in batches thousands of queries at a time

Systems@ETH zürich

Shared DB can run TPC-W!

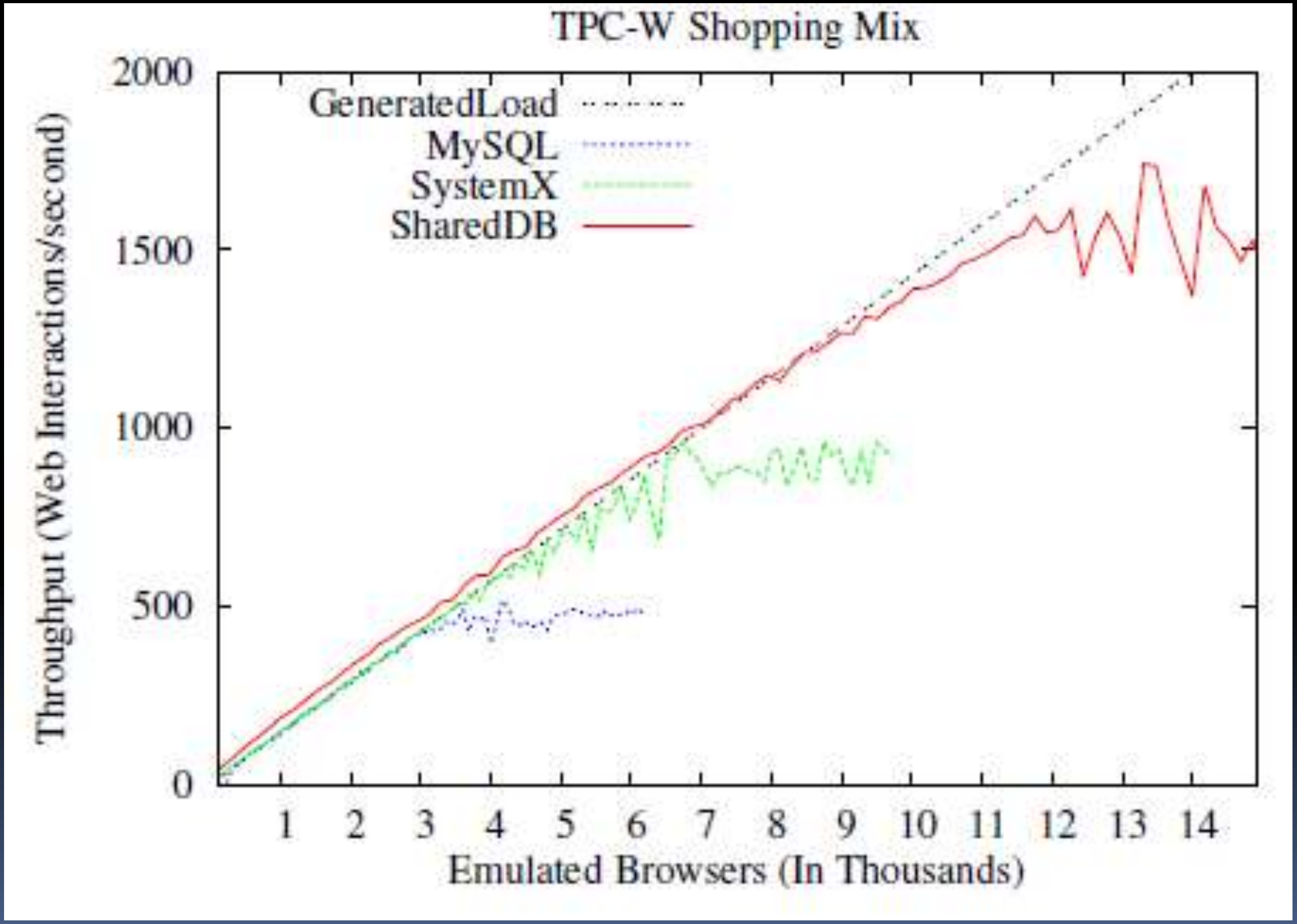


Systems@ETH Zürich

For the non-db people

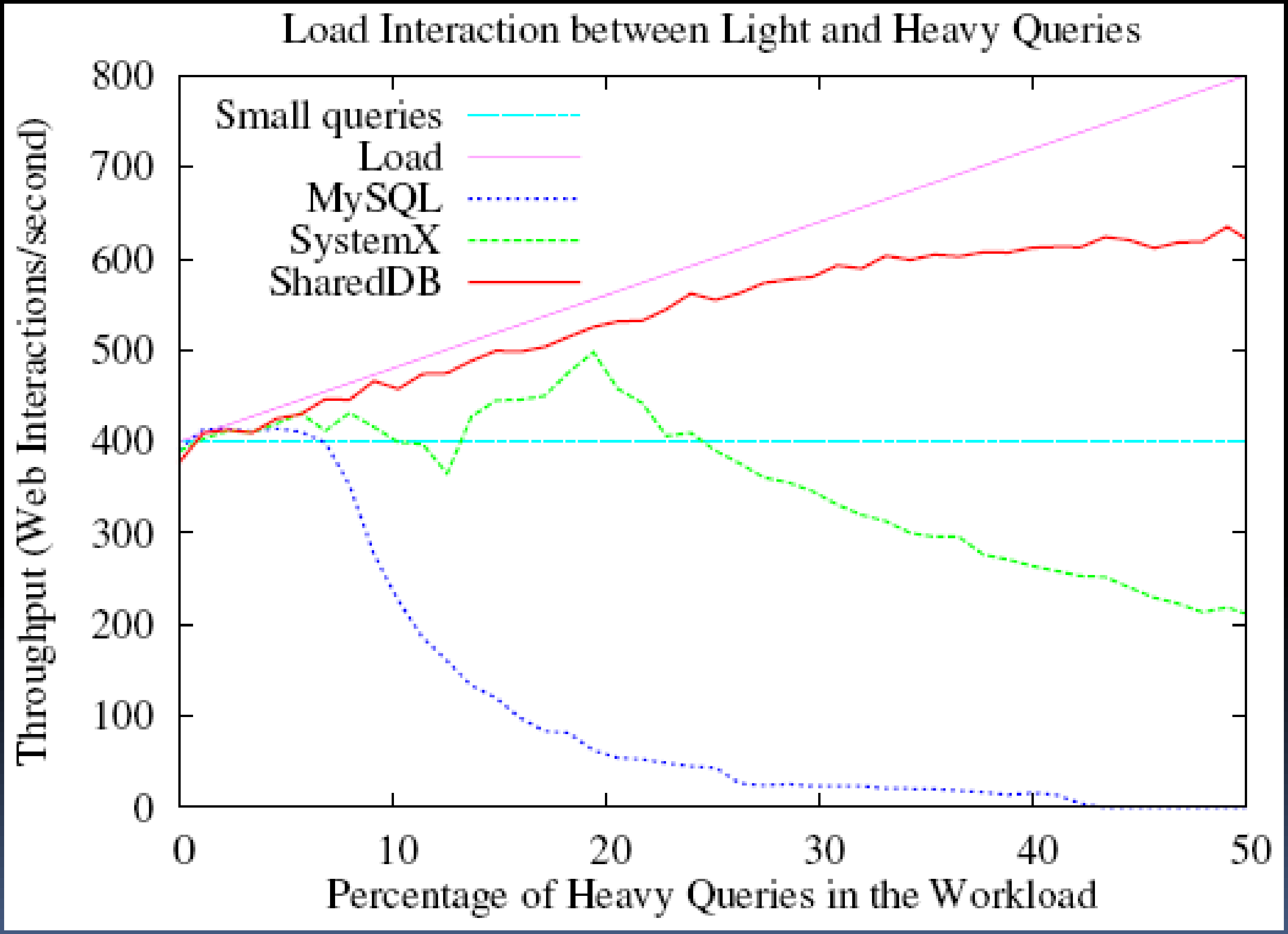
- TPC-W has updates!!!
- Full consistency without conventional transaction manager
- Transactions are no longer what you read in textbooks …
  - Sequential execution
  - Memory CoW (Hyder, TU Munich)
  - Snapshot isolation

Raw performance



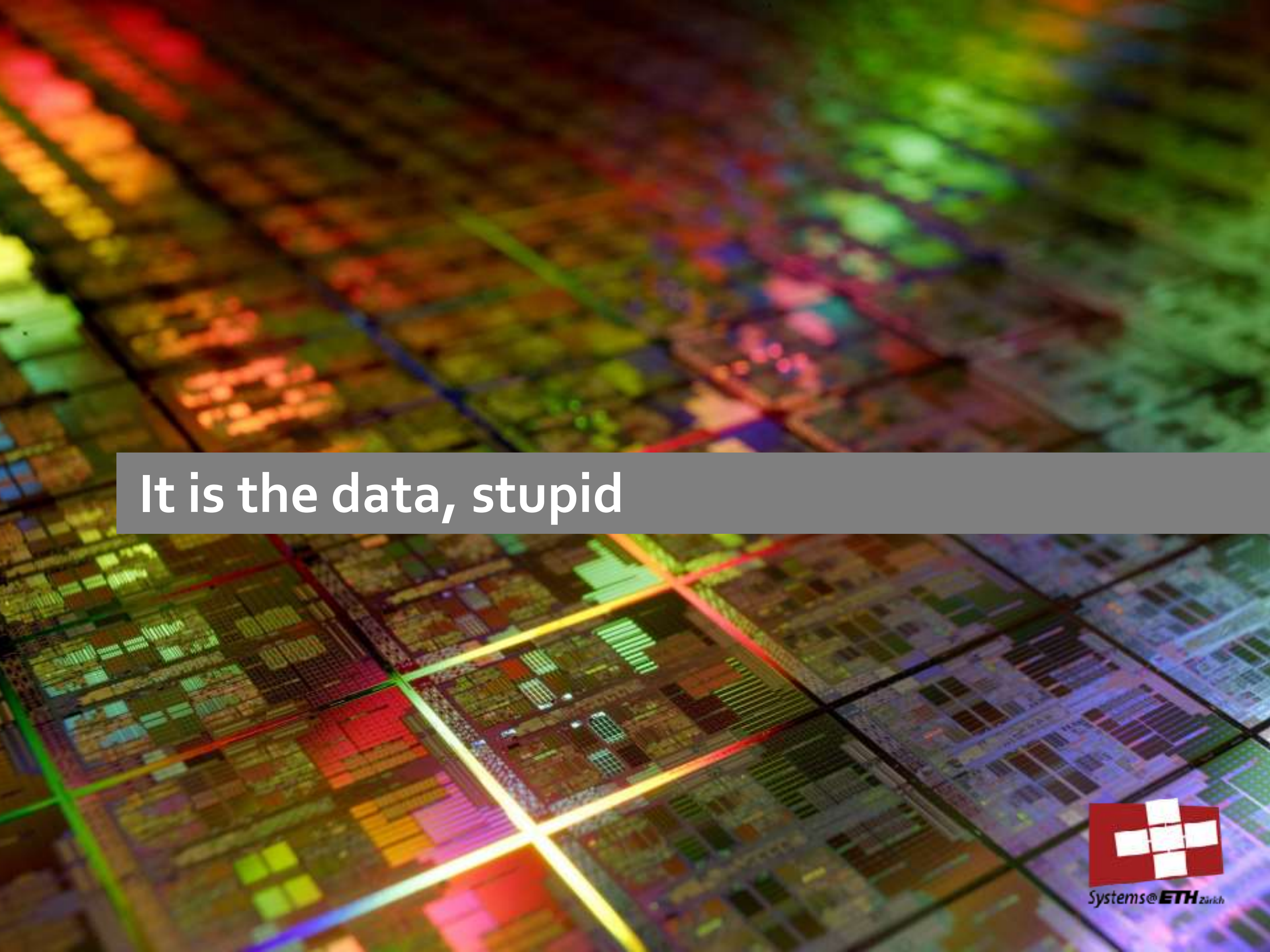
Systems@ETH zürich

Predictability, robustness



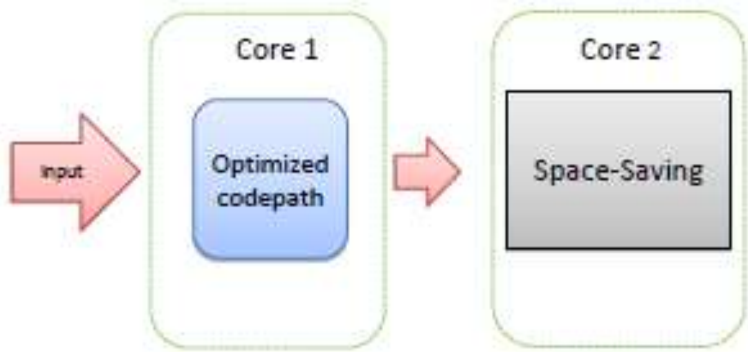
Systems@ETH Zürich

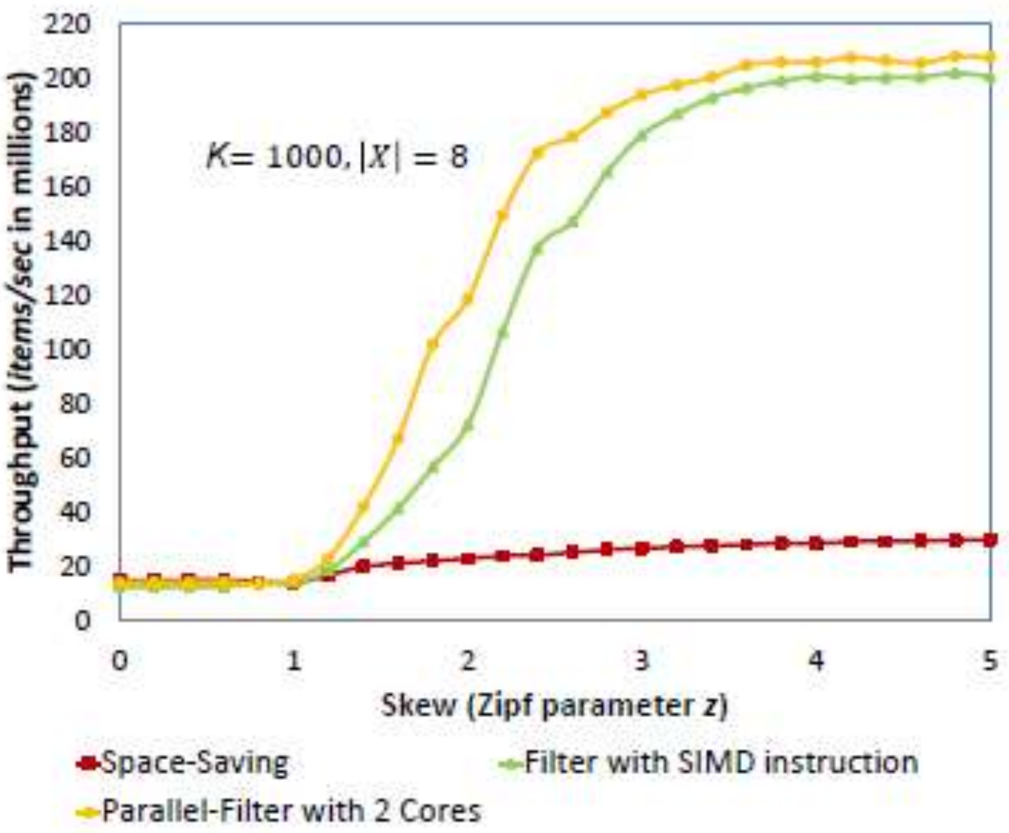
It is the data, stupid



Systems@ETH zürich

Not everything is parallel





P. Roy, J. Teubner, G. Alonso
Efficient Frequent Item Counting in Multi-Core Hardware, KDD 2012

The data ties it all together

- The previous example makes a case for all the ideas described
  - Hardware acceleration on the data path
  - Knowing where to do what
  - In network data filtering
  - On the fly statistics
  - Characterizing the hardware and the load

Conclusions

The opportunity is now

- Consensus on major crisis in hardware (from the sw perspective)
- Hardware not really improving, responsibility passed on to software
- Business models and IT systems moving towards specialization
  - Room for customized systems
  - Need for general solutions

Systems@ETH Zürich