Nothing is for granted: Making wise decisions using real-time intelligence

Anastasia Ailamaki
Data management faces critical challenges

Change means trouble
Preparation kills discovery

1. Load data
2. Clean Data
3. Tune Database
4. Ask analytical question
5. Plan execution
6. (finally) run & get answer

Cost grows with *owned* – not used! – data
Planning is expensive, often even wrong
“pay as you go”

[generate code using algebraic definitions]

Useful data

data-to-insight time = 0
prepare data only when needed
Query-driven data sanitization

FD: Zip → City

<table>
<thead>
<tr>
<th>Name</th>
<th>Zip</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jon</td>
<td>9001</td>
<td>Los Angeles</td>
</tr>
<tr>
<td>Jim</td>
<td>9001</td>
<td>San Francisco</td>
</tr>
<tr>
<td>Mary</td>
<td>10001</td>
<td>New York</td>
</tr>
<tr>
<td>Jane</td>
<td>10002</td>
<td>New York</td>
</tr>
</tbody>
</table>

Clean the useful subset of the data with probabilistic fixes
Evolving indexes

Data skipping
Fine-grained access path selection
Choose what to build & when
- Value-Existence (i.e., Bloom filters)
- Value-Position (i.e., B+ Trees)
Build / drop based on budget

Max gain amortized by cost to build
Scaling sharing-aware optimization

Q1: select * from R, S, T where R.a=S.a and R.b=T.b
Q2: select * from R, S, U, V where R.a=S.a and S.e=U.e and S.f=V.f
Q3: select * from R, S, U, W where R.a=S.a and S.e=U.e and U.h=W.h

Timely planning through adaptation

Remove optimization from critical path
Adapt planning heuristic
Virtualization layers

- Format impacts access & caching
- HW impacts processing

Virtualize format and hardware
Virtualization layers

- Format impacts access & caching
- HW impacts processing
Detecting active spambots

[Symantec data]

**Flexibility**
Ad-hoc queries over diverse data formats

**Performance**
Fast queries regardless of data format
Customizing data access layer

Traditional DBMS: Data adapts to engine

Proteus
Plug-in per data source
Build auxiliary structures

Treat each source as native storage format
How Proteus builds a just-in-time data base

SELECT bot, country, ...
FROM SpamEmail e, SpamCategories c
WHERE e.id == c.id AND
e.lang = 'English' AND ...

- Code Generate the Access Paths
- Code Generate the Query
- Build Position and Data Caches

Tame heterogeneity; Cache only useful data
Virtualization layers

- Format impacts access & caching
- HW impacts processing

Virtualize format and hardware
Harness Accelerator-Level Parallelism

Intra-operator
- Fast algorithms close to μ-arch

Intra-device
- Hardware Oblivious: Portability through online specialization
- Hardware Conscious

Inter-device
- Execution models that encapsulate heterogeneity

JIT decide the device to optimize for
Execution on CPU+GPU

- Decouple data- from control-flow
- Trait conversions

Operators encapsulate device heterogeneity
GPU Accesses Fresh Data from CPU Memory

- OLTP generates fresh data on CPU Memory
- Data access protected by concurrency control
- OLAP needs to access fresh data over interconnect

Provide snapshot isolation for GPUs w/o CC overheads
Use shared main-memory bus efficiently
Existing designs statically trades performance for isolation.

Traverse HTAP spectrum based on amount of fresh data.

*Fresh Data Access Bandwidth*
Data virtualization and JIT engines

Hardware

Data

Transactions

Analytics

HTAP

CPU

GPU

aggregate
router

gpu2cpu
aggregate

filter

unpack
cpu2gpu

mem-move

router

segmenter

Workload

</>

{;

};

.JSON

0101

0100100

0101001

.CSV

.RAW

0101001

0100100

0101001

.XML

.JSON

{;

};

0101001

0100100

0101001

.CSV

.RAW

0101001

0100100

0101001

.XML

.JSON

{;

};
five old friends revisited

- **Data variety** → Operational environment variety
  - Unpredictable application requirements
- **Data veracity** → Inter-component veracity
  - Heterogeneous data & variable importance
- **Data volume** → Structural volume
  - Multi-layered system architectures
- **Data value** → Resource value
  - Broader, multi-featured analytics
- **Data velocity** → Technological velocity
  - Hardware heterogeneity & volatility

Intelligent systems to catch-up with an evolving landscape
Incorporate change into native design.
Anticipate change and react, learning from errors.

A solution is only as efficient as its least adaptive component.
Reactive and optimistic
Imaginative and inspiring