Advanced Topics in Databases

Introduction

Sebastian Breß, David Broneske, Gabriel Campero, Andreas Meister, Marcus Pinnecke, Gunter Saake
Otto-von-Guericke University Magdeburg
Summer Term 2017
Motivation for the Lecture

- Familiarize students with current developments in database research

- Topics chosen:
  - First solutions currently making their way into database management systems and applications → practical relevance
  - Solutions not yet fully developed and open problems exist → research relevance

- Possible starting points for scientific work, e.g. master thesis, position in academia, Ph.D. thesis, industry R&D, etc.
Yesterday’s DBMS Landscape

Application

Application

„Banking, SAP, …“

„Server“

„Disk“

Database

DBMS

ORACLE

IBM DB2

Microsoft SQL Server

TERADATA
Yesterday’s DBMS Hardware

Small main memory

Disk-based systems
Assumptions of yesterday’s DBMSs

- Capacity of main memory < 1% of the stored data
- Fixed block size based on the transfer unit between disks and main memory
- Central scheduler to schedule transactions
- No redundant data storage in main memory
- Pipelining is always beneficial (no storage of intermediate results)
- Compiling of SQL for one processor architecture $\rightarrow$ Reuse of compiled plan
Today’s DBMS Topics

- Genomic Data
- In Memory
- Data In Memory
- Cloud
- Data Provenance
- Co Processing
- New Storage Devices
- Social Networks
- Graph Database
- Security
- Geographical Databases
- Parallel Processing
- Stream Processing
- Data Warehouse
- Big Data
- Internet of Things
- Scientific Data
Today’s DBMS Hardware

- Large main memory
- Multi-core CPUs
- Solid state disks
- Co-processors
Future DBMSs

- Capacity of main memory <1% of the stored data
  - DB in main memory
- Fixed block size based on the transfer unit
  - Direct access of data on all devices
- Central scheduler to schedule transactions
  - Which processor should do the job?
- No redundant data storage in main memory
  - Redundant data at co-processors
- Pipelining is always beneficial
  - Co-processors like GPU support massive parallelism
- Reuse of compiled plan
  - Load-balancing between co-processors requires different plans
The Goals of a "Databaser"

- Performance

Picture taken from [6]
The Goals of a "Databaser"

- Performance
- Performance

Picture taken from [6]
The Goals of a "Databaser"

- Performance
- Performance
- Performance

Picture taken from [6]
The Goals of a "Databaser"

• Performance
• Performance
• Performance

How can we achieve more performance?
Trends for DBMS’s

• Use main memory as primary storage → Speed up data access

• Exploit all hardware capabilities such as co-processors → Speed up database operators
Are DBMSs written for yesterdays hardware efficient on todays hardware as well?
Are DBMSs written for yesterdays hardware efficient on todays hardware as well?

”30 years of Moore’s law has antiquated the disk-oriented relational architecture for OLTP applications”  [Stonebraker et al., 2007]
Data Access – Yesterday’s Bottleneck

Bottleneck

Bottleneck
Data Access – Today’s Bottleneck

Bottleneck
The World of Co-Processors

PCI Express Bus

Picture taken from [7]
What do we have to change in DBMSs’ architecture to **exploit new hardware capabilities** and to **meet tomorrow’s challenges**?
Overview of Topics

1. Main-Memory DBMSs (Meister, 2 lectures)
2. Advanced Query Optimization (Meister, 2 lectures)
3. Hardware-sensitive DB Algorithms (Broneske, 2 lectures)
4. HTAP and Self-Driving Systems (Pinnecke, 2 lectures)
5. New Data Models in MM DBMSs (Campero, 1-2 lectures)
6. GPU-accelerated Data Management (Breß, 1-2 lectures)
Main-Memory DBMSs

Address main-memory access bottleneck using cache-conscious

• data layouts (e.g., column-stores [Abadi et al., 2008]) and
• data processing [Manegold et al., 2000]

Query processing in row-stores

<table>
<thead>
<tr>
<th>Location</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD</td>
<td>1</td>
</tr>
<tr>
<td>EF</td>
<td>5</td>
</tr>
<tr>
<td>MD</td>
<td>7</td>
</tr>
<tr>
<td>EF</td>
<td>4</td>
</tr>
</tbody>
</table>

\[\text{SELECT SUM(Sales)}\]
\[\text{FROM Shop}\]
\[\text{WHERE Location = 'MD'}\]

\[\text{\sigma}_{\text{Location = 'MD'}}\]

\[\text{\pi}_{\text{Sales}}\]

\[\text{SUM(Sales)}\]

Adapted from [Köppen et al., 2012]

Query processing in column-stores

<table>
<thead>
<tr>
<th>Location</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD</td>
<td>1</td>
</tr>
<tr>
<td>EF</td>
<td>5</td>
</tr>
<tr>
<td>MD</td>
<td>7</td>
</tr>
<tr>
<td>EF</td>
<td>4</td>
</tr>
</tbody>
</table>

\[\text{\sigma}_{\text{Location = 'MD'}}\]

<table>
<thead>
<tr>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

\[\text{SUM}\]

<table>
<thead>
<tr>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
</tr>
<tr>
<td>2009</td>
</tr>
<tr>
<td>2011</td>
</tr>
<tr>
<td>2010</td>
</tr>
</tbody>
</table>
Advanced Query Optimization

\[ r(\text{ORDER}) \bowtie r(\text{PRODUCT}) \bowtie r(\text{CUSTOMER}) \bowtie r(\text{SUPPLIER}) \]

Equivalent plans for one query \( \implies \) Large search space

Search space increases in modern systems
\( \implies \) Advanced optimization algorithms needed:
- Parallelization
- Usage of co-processors

Taken from [Saake et al., 2012]
Hardware-sensitive DB Algorithms

CPU "smaller than"-selection

```c
int pos = 0;
for (int i=0; i < array_size; ++i){
    if (array[i] < comp_val)
        result[pos++] = i;
}
```

GPU "smaller than"-selection

```c
int tid = threadIdx.x + blockIdx.x * blockDim.x;
while (tid < array_size){
    bitmask[tid] = (array[tid] < comparison_value);
    tid += blockDim.x * gridDim.x;
}
```

Optimized Code differs between processing devices w.r.t.

- Access pattern
- Code optimizations (e.g., branch-free code)
- Parallelization capabilities
The HTAP Challenge

Analytical (OLAP) and Transactional processing (OLTP):
- Different access patterns
- Different performance goals (i.e., latency vs. throughput)
⇒ Different optimizations required

Hybrid Transactional/Analytical Processing (HTAP):
⇒ a challenge for DBMSs
Self-Managing Solutions to the HTAP Challenge

- Advanced storage engine designs exist for HTAP\(^1\).
- Self-managing features are important in these designs, enabling the DBMS to adapt to changing workloads.

---

\(^{1}\)For examples, see: [Kemper and Neumann, 2011], [Dittrich and Jindal, 2011], [Arulraj et al., 2016]
New Data Models in Main-Memory DBMSs

New data models have been proposed\(^2\) to help users make the most out of Variety (one of the 3V’s of Big Data).

- Graphs (e.g. RDF, property graphs), documents, column families, key-value models, etc.

- How are these models supported in state-of-the-art MM DBMSs?
- Does this support outperform model-specific databases?
- What are some challenges and opportunities?

E.g.: Graph support in SAP HANA [Rudolf et al., 2013]

\(^2\) For an overview, see [Bugiotti and Cabibbo, 2013]
GPU-accelerated Data Management

- Specialized GPU operators
- Predicting the benefit of GPU acceleration
- Data placement strategies
- Increased complexity of query optimization
- Query compilation for modern processors
Organization

- **Lecture:** Friday, 13:15 - 14:45, Raum G10-460
  - Lecturer: Sebastian Breß, David Broneske, Gabriel Campero, Andreas Meister, Marcus Pinnecke, Gunter Saake
  - Each lecture will provide insights about his own research topics
    - Office hours: upon consultation
    - Email: firstname.lastname@ovgu.de
- **Exercise:** Wednesday, 15:15 - 16:45, Raum G22A-112
  - Starting 26.04.2017
  - Tutor: Andreas Meister
    - Office hours: upon consultation
    - Room: G29-105
    - Email: andreas.meister@ovgu.de
Organization

- Information (time, rooms) & slides available on www.dbse.ovgu.de/Lehre/ATDB.html
- Examination
  - Likely oral examinations
  - 20-30 minutes
  - Questions about lecture and exercise
- Requirements:
  - Registration for exercises via LSF until 23.04.2017
  - Assigned minimal 60% of all exercise tasks
  - Presentation of at least 3 tasks in the exercise
Exercise

- Task based on lecture content
- Available at our website
- Requirements:
  - Preparation
  - Understanding
  - Presentations
- Plagiarism not allowed → Tasks will not be accepted
References I

Column-Stores vs. Row-Stores: How different are they really?
In *SIGMOD*, pages 967–980.

Bridging the archipelago between row-stores and column-stores for hybrid workloads.
In *Proceedings of the 2016 International Conference on Management of Data*, pages 583–598. ACM.

A comparison of data models and aps of nosql datastores.
*Dipartamento di Ingegneria della Università di Roma*.

Towards a one size fits all database architecture.

Hyper: A hybrid oltp&olap main memory database system based on virtual memory snapshots.
In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 195–206. IEEE.

*Data Warehouse Technologien*.
mitp-Verlag.
References II


Web Resources


