Adaptive FPGA-based Database Accelerators – Achievements, Possibilities, and Challenges

Daniel Ziener and Jürgen Teich
Database Acceleration – Overview

**Idea:** Translate each SQL query into an FPGA-based accelerator circuit through run-time assembly of dynamically reconfigurable hardware modules.

SQL query:
```
SELECT Price, Volume
FROM Trades
WHERE Symbol=“UBSN”
INTO UBSTrades
```

- Trades → a: Symbol = USBN → WHERE a → SELECT Price, Vol. → UBSTrades
- Hardware Module Library
- FPGA DynSoC
- UBSTrades
Database Acceleration – Architecture

SELECT * FROM table WHERE salary > 10000 AND year < 1990
# Database Acceleration – Overview Module Library

<table>
<thead>
<tr>
<th>Module</th>
<th>Operator Coverage</th>
<th>Number of Slots</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restriction</td>
<td>Arithmetic (+,-,*)&lt;br&gt;Comparators (&lt;,&gt;,=,≠)&lt;br&gt;Bitwise functions (AND, OR, NOT, XOR, ...)&lt;br&gt;</td>
<td>2</td>
<td>1 Sample/Cycle</td>
</tr>
<tr>
<td>Aggregation</td>
<td>SUM(), MIN(), MAX(), COUNT()</td>
<td>2</td>
<td>1 Sample/Cycle</td>
</tr>
<tr>
<td>Reorder</td>
<td>Reorder Attributes of a tuple</td>
<td>4</td>
<td>1 Sample/Cycle</td>
</tr>
<tr>
<td>Join</td>
<td>Hash and Merge Join</td>
<td>-</td>
<td>1 Sample/Cycle</td>
</tr>
<tr>
<td>Sort line</td>
<td>for sorting 2 KB (64 KB) data</td>
<td>16</td>
<td>1 Sample/Cycle</td>
</tr>
<tr>
<td>Sort tree</td>
<td>merges sorted block</td>
<td>-</td>
<td>1 Sample/Cycle</td>
</tr>
</tbody>
</table>

- Each reconfigurable area consists of 16 slots
- 4 reconfigurable areas available on our prototype
Database Acceleration – Lessons Learned

- **High processing throughput achievable**
  - Pipelined modules have a throughput of **2 GByte/s per reconfigurable area (125 MHz x 16 Bytes)**
  - The throughput is **independent** of the number of concatenated modules

- **I/O turns out to define the bottleneck**
  - PCIe Gen2 x4: **≈ 1.7 GByte/s**
  - Only one interface to feed all reconfigurable areas

- **Flexibility is the key feature**
  - For each query different decisions can be taken at run-time
  - All processing alternatives can be executed on the same static system
Database Acceleration – New High-Performance Architecture

Database Tables

FPGA

Reconfigurable Area

BLOOM

Alignment Unit

Hash Join + Aggr.

Conf. Manager

Host

Incoming queries

Query analysis + filter configuration

Host

FPGA

Data processing

Data processing

Data processing

Data processing

FPGA

Reconfigurable Area

Time
Database Acceleration – Results (FPT’15)

- Comparing Energy/Power consumption of an Intel Core i7 with our approach based on an embedded Xilinx Zynq-SoC
- Analysis of example query based on the TPC-DS benchmark (1 GB scale), including restrictions, aggregations, and joins

<table>
<thead>
<tr>
<th></th>
<th>Accl@ Zynq</th>
<th>ARM – MySQL</th>
<th>Intel i7 – MySQL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Execution time</strong></td>
<td>44.2 ms</td>
<td>6900 ms</td>
<td>420 ms</td>
</tr>
<tr>
<td><strong>Overall energy</strong></td>
<td>190 mJ</td>
<td>1.47 J</td>
<td>5.33 J</td>
</tr>
<tr>
<td><strong>Improvement t_{exe}</strong></td>
<td>156</td>
<td></td>
<td>9.5</td>
</tr>
<tr>
<td><strong>Improvement Energy</strong></td>
<td>7.72</td>
<td></td>
<td>27.97</td>
</tr>
</tbody>
</table>
Database Acceleration – Results (FPT’15)

- Comparing Energy/Power consumption of an Intel Core i7 with our approach based on an embedded Xilinx Zynq-SoC

More Information:

Current Database Management Systems

- **Database management systems are multi-user systems**
  - Different queries with different complexity have to be processed on different data at the same time
  - Response time is very important

- **Bunch of different operations**
  - Query processing
  - Sorting
  - Data analytics
  - Data update

- **Changing load scenarios over time**
  - E.g., day: query processing; night: data analytics
HW Accelerators for Big Data Applications

- **Current software solutions**
  - Multi-Core server systems with many nodes
  - On each core, data processing is done with data or time slices
  - **Advantages:**
    - OS support (task switching, mapping onto processing places)
    - Easy to extend with new operators or analytic functions

**Question:** How can we achieve such a flexibility for HW-based accelerators?