How to pull a cart...
...using a horse
...using four horses
...using a thousand chickens
...using four horses and a thousand chickens
...using four horses, a thousand chickens, two power drills, a hair dryer and a toaster we found along the way.
...without knowing the size of the cart
...or even the number of carts
Data Management in the Age of Heterogeneous Hardware

Holger Pirk <holger@cwi.nl>
A Different Problem...
A Different Problem...
A Different Problem...

• A moving objects database of vehicles in NW Europe
  • 240M Tuples (latitude, longitude, device, time)
• Lot’s of spatial range queries
  • Applications like traffic monitoring, forecasting or planing
...Different Means
...Different Means
...Different Means

<table>
<thead>
<tr>
<th>Device</th>
<th>Bandwidth: GB per sec and $</th>
<th>Capacity: MB per $</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU</td>
<td>1.000</td>
<td>10</td>
</tr>
<tr>
<td>CPU</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>SSD</td>
<td>5</td>
<td>1.000</td>
</tr>
<tr>
<td>Hard Disk</td>
<td>1</td>
<td>10.000</td>
</tr>
</tbody>
</table>
Approach
Remember gifs through a modem connection?
The Approach
The Approach
Approximate
Refine
Bitwise Decomposition/Distribution (BWD)
Implementation

• Vertical Partitioning and Distribution

• But with finer slices: **Bitwise Decomposition & Distribution**

```
00000000 00111010 01101110 01011101
```

Low Resolution Approximation   Residual

• Queries are evaluated in phases, tuples reconstructed lazily

• Every device does it’s best with the available data
Query Performance

Evaluation Time in 100 us

- CPU
- GPU Streaming
- GPU/CPU BWD

- 1 Query
- 256 Queries
- 2048 Queries

Values:
- CPU: 5,703,336
- GPU Streaming: 329,713
- GPU/CPU BWD: 933

CWI
Load Balancing

- Percentage of Time Spend
- Number Of Queries
- GPU Phase
- CPU Phase

100%
75%
50%
25%
0%
Conclusion

• Specialized hardware has good value for money
  • *If* the application requirements are known

• We seek a generic strategy to use all available devices efficiently
  • **Bitwise Decomposition** might be that strategy
But What about “generic”...
Outlook

- Other Applications:
  - Relational Queries (Joins, Aggregations, ...)

- Other Data: Categorical, Float, ...

- Other Setups: SSD-HDD, Mobile - Server, Hierarchical Setup
Thank you!
Questions?