Accelerating Aggregation using Intra-cycle Parallelism

Ziqiang Feng, Eric Lo
Department of Computing, The Hong Kong Polytechnic University, Hong Kong

What is Intra-cycle Parallelism?

<table>
<thead>
<tr>
<th>Memory residency + column store + compression</th>
<th>Column values are compressed in to <strong>short</strong> codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor words are so <strong>wide</strong> that they can process 64 bits information per cycle.</td>
<td></td>
</tr>
</tbody>
</table>

![Figure 1](image1.png)

**Fig 1.** Wasted intra-cycle parallelism: loading an 8-bit column value into a 64-bit processor register

HBP and VBP: Two special data layouts

- Proposed in [1]
- **Horizontal** Bit Packing (HBP)
- **Vertical** Bit Packing (VBP)
- Load multiple values packed together into the processor register
- Algorithms can (really) 64 bits information per cycle

![Figure 2](image2.png)

**Fig 2.** An example of HBP: Loading 8 values simultaneously into the register

Aggregation in HBP/VBP?

Non-bit-parallel approach:
- Lookup – recover the individual values in its plain form
- Then perform the aggregation as usual

![Figure 3](image3.png)

**Fig 3.** An example of non-bit-parallel sum in HBP

**Our** Bit-parallel approach:
- No lookup is needed.
- Exploit intra-cycle parallelism in aggregation
- Dedicated algorithm for different aggregator/layout

Example: sum in HBP

![Figure 4](image4.png)

**Fig 4.** Illustration of the bit-parallel sum algorithm in HBP layout

- Implementation: use only **five** 64-bit instructions
  - **Shift** × 2
  - **Add** × 1
  - **And** × 1
  - **Multiply** × 1
- One 64-bit word contains 8 eight-bit values
- Advantage:
  - **8** values are processed at a time
  - # of instructions is low

TPC-H Results

Implemented on Intel i7-4770 with 256-bit AVX2.

<table>
<thead>
<tr>
<th></th>
<th>HBP</th>
<th>VBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation time</td>
<td>Reduced by 28.1%</td>
<td>Reduced by 55.0%</td>
</tr>
<tr>
<td>Whole-query time</td>
<td>Reduced by 20.4%</td>
<td>Reduced by 44.4%</td>
</tr>
</tbody>
</table>

Future Work

This paper is based on two storage layouts – HBP and VBP. We have devised a more performant layout called ByteSlice. ByteSlice outperforms HBP and VBP both in scans and lookups. We have published ByteSlice in SIGMOD [2]. Our future work will be based on this new superior layout.

References:


Copyright © 2015, Database Group, Department of Computing, The Hong Kong Polytechnic University.