OLTP on Hardware Islands

Danica Porobic, Ippokratis Pandis*, Miguel Branco, Pınar Tözün, Anastasia Ailamaki

Data-Intensive Application and Systems Lab, EPFL

*IBM Research - Almaden
Hardware topologies have changed

Core
Hardware topologies have changed

<table>
<thead>
<tr>
<th>Topology</th>
<th>Core to core latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMP</td>
<td>high (~100ns)</td>
</tr>
</tbody>
</table>
Hardware topologies have changed

<table>
<thead>
<tr>
<th>Topology</th>
<th>Core to core latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMP</td>
<td>high (~100ns)</td>
</tr>
<tr>
<td>CMP</td>
<td>low (~10ns)</td>
</tr>
</tbody>
</table>
Hardware topologies have changed

<table>
<thead>
<tr>
<th>Topology</th>
<th>Core to core latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMP</td>
<td>high (~100ns)</td>
</tr>
<tr>
<td>CMP</td>
<td>low (~10ns)</td>
</tr>
<tr>
<td>SMP of CMP</td>
<td>variable (10-100ns)</td>
</tr>
</tbody>
</table>

Variable latencies affect performance & predictability
Deploying OLTP on Hardware Islands

Performance

Throughput

% Multisite Transactions in Workload

Shared-Nothing

Shared-Everything

Thread to core assignment

Best

Worst

HW + Workload -> Optimal OLTP configuration
Outline

• Introduction
• Hardware Islands
• OLTP on Hardware Islands
• Conclusions and future work
Multisocket multicores

Communication latency varies significantly
Placement of application threads

Counter microbenchmark
8socket x 10cores

Throughput (Mtps)

Unpredictable

47%

40%

OS | Spread | Island

TPC-C Payment
4socket x 6cores

Throughput (Ktps)

39%

OS | Spread | Island

Islands-aware placement matters
Impact of sharing data among threads

Counter microbenchmark

Throughput (Mtps)

Log scale

8socket x 10cores

Counter per core

10000

Counter per socket

18.7x

Single counter

160

TPC-C Payment – local-only

Throughput (Ktps)

8socket x 10cores

4.5x

4socket x 6cores

Shared nothing

Shared everything

Fewer sharers lead to higher performance
Outline

• Introduction
• Hardware Islands

• OLTP on Hardware Islands
  – Experimental setup
  – Read-only workloads
  – Update workloads
  – Impact of skew

• Conclusions and future work
Experimental setup

• Shore-MT
  – Top-of-the-line open source storage manager
  – Enabled shared-nothing capability

• Multisocket servers
  – 4-socket, 6-core Intel Xeon E7530, 64GB RAM
  – 8-socket, 10-core Intel Xeon E7-L8867, 192GB RAM

• Disabled hyper-threading

• OS: Red Hat Enterprise Linux 6.2, kernel 2.6.32

• Profiler: Intel VTune Amplifier XE 2011

• Workloads: TPC-C, microbenchmarks
Microbenchmark workload

• Singlesite version
  – Probe/update \( N \) rows from the local partition

• Multisite version
  – Probe/update 1 row from the local partition
  – Probe/update \( N-1 \) rows uniformly from any partition
  – Partitions may reside on the same instance

• Input size: 10 000 rows/core
Software System Configurations

1 Island
Shared-everything

24 Islands
Shared-nothing

4 Islands
Increasing % of multisite xcts: reads

Throughput (KTPs)

% Multisite transactions

Finer grained configurations are more sensitive to distributed transactions

No locks or latches

Messaging overhead

Contention for shared data

Fewer messages for 1 transaction

1 Island

24 Islands

4 Islands
Where are the bottlenecks? Read case

Communication overhead dominates

4 Islands
10 rows
Increasing size of multisite xct: read case

Communication costs rise until all instances are involved in every transaction.
Increasing % of multisite xcts: updates

Distributed update transactions are more expensive
Where are the bottlenecks? Update case

4 Islands
10 rows

Communication overhead dominates
Increasing size of multisite xct: update case

- More instances per transaction
- Increased contention
- Efficient logging with Aether*

Shared everything exposes constructive interference
Effects of skewed input

Throughput (KTps)

Skew factor

Local only

Few instances are highly loaded

24 Islands

Contention for hot data

4 Islands

1 Island

50% multisite

Larger instances can balance load

Still few hot instances

4 Islands effectively balance skew and contention
OLTP systems on Hardware Islands

• Shared-everything: stable, but non-optimal
• Shared-nothing: fast, but sensitive to workload
• OLTP Islands: a robust, middle-ground
  – Runs on close cores
  – Small instances limits contention between threads
  – Few instances simplify partitioning

• Future work:
  – Automatically choose and setup optimal configuration
  – Dynamically adjust to workload changes

Thank you!