Wimpy Nodes with 10GbE: Leveraging One-Sided Operations in Soft-RDMA to Boost Memcached

Patrick Stuedi,
Animesh Trivedi, Bernard Metzler
IBM Research Zurich
Introduction

- Known: key/value stores and low-power CPUs/cores go well together
  - Workload typically not compute-heavy
  - Slower CPU clock sufficient
  - Easy to parallelize, distribute load over many low-power cores
  - Examples: FAWN, Facebook/Tilera
- What are the implications of attaching 10 GbE NICs to the low-power key/value storage nodes?
  - Improved latency
  - What about the CPU load?
Memcached/GET: 1GbE vs 10GbE

- Setup: 100K GET requests, 6 clients
- High CPU usage limits performance (ops/sec)
Does multicore help?

• Throughput performance
  • Yes, considering the scaling limitations of Memcached

• Efficiency
  • More cores consume more energy

• Efficient processing on one-core will hopefully translate to efficient multi-core setup.
How is the CPU being used?

- Depends on the size of the value
  - **Small values (~1K):** 60% of CPU usage scattered across many OS functions (e.g. context switching, etc.)
  - **Larger values (~100K):** 60% of CPU usage inside network stack

<table>
<thead>
<tr>
<th>Value Size</th>
<th>1K</th>
<th>10K</th>
<th>100K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total CPU cycles</td>
<td>46K</td>
<td>84K</td>
<td>289K</td>
</tr>
<tr>
<td>Networking</td>
<td>35%</td>
<td>42.8%</td>
<td>58%</td>
</tr>
<tr>
<td>User Space</td>
<td>5%</td>
<td>3.2%</td>
<td>1.1%</td>
</tr>
<tr>
<td>Remaining</td>
<td>60%</td>
<td>54%</td>
<td>40.9%</td>
</tr>
</tbody>
</table>

CPU at 1.1 GHz
Using RDMA

- RDMA: Remote Direct Memory Access
  - Efficient remote memory access
  - Zero-copy (inside the end hosts), low latency, low CPU usage
  - Great! 🙄
Using RDMA

- RDMA: Remote Direct Memory Access
  - Efficient remote memory access
  - Zero-copy (inside the end hosts), low latency, low CPU usage
  - Great!

- Who has RDMA capable NICs deployed?
Using RDMA

- RDMA: Remote Direct Memory Access
  - Efficient remote memory access
  - Zero-copy (inside the end hosts), low latency, low CPU usage
  - Great!

- Who has RDMA capable NICs deployed?
  - HPC: Yes
  - Commodity data centers? 😐
RDMA in Software

- No hardware acceleration, runs on Ethernet
- But still RDMA semantics
- Example: One-sided RDMA read in SoftiWARP
  - Zero copy, no context switching, low CPU footprint, etc.
Memcached/RDMA

Modified Memcached Architecture:

1) Memory management: register server-side memory chunks with RDMA

2) SET operation:
   - New value: get new chunk, store key/value pair, return stag
   - Update value: get new chunk, store store key/value pair, swap stags with old chunk

3) GET operation: client uses one-sided RDMA read to retrieve entire chunk
   - Zero copy
   - No context switch
   - Move parts of server processing to the client (e.g., request parsing)
Memcached/RDMA (2)

- single process
- chained hash
- table
- thread 1
- thread N
- event dispatcher
- soft-RDMA provider
- user space
- kernel space
- kernel network processing
Memcached/RDMA (2)

- single process
- chained hash table
- thread 1
- thread N
- event dispatcher
- Register memory chunks
- user space
- kernel space
- soft-RDMA provider
- kernel network processing
Memcached/RDMA (2)

- Single process
- Chained hash table
- Thread 1
- Thread N
- Event dispatcher
- Register memory chunks
- User space
- Kernel space
- Soft-RDMA provider
- Kernel network processing
- Memory chunk
- GET request
Implementation & Benchmarks

• Implementation Memcached/RDMA
  • Standalone prototype: server/client
  • Uses Memcached data types (e.g., item for storing key/value pairs)

• Benchmark Setup
  • 1 Server, 6 Clients
  • 4 core Intel Xeon E5345, 10GbE
  • Server CPU clock frequency: 1.1 Ghz
  • 1000 pre-insterted key/value pairs
  • OProfile to measure CPU load
CPU Efficiency

- Memcached/RDMA consumes less CPU
  - For **small packets**: less OS overhead (excluding network stack)
  - For **large packets**: less network stack overhead
Multicore Performance

- Memcached/RDMA with one core performs like Memcached with 4 cores
Conclusion

- Memcached/RDMA is a more CPU efficient Memcached based on Software RDMA
  - Zero copy, zero context switching for re-occurring GET requests, memory chunk parsing moved to client side
  - No special hardware required
- Architecture also suitable for SSD based key/value stores
  - Any combination of high bandwidth storage and fast network will put pressure on the CPU
- Outlook: multicore, latency, etc.
Thanks! Questions?

http://gitorious.org/softiwp