Reducing Network Contention with Mixed Workloads on Modern Multicore Clusters

Matthew Koop
mattkoop@nasa.gov
{luom, panda}@cse.ohio-state.edu

1NASA Center for Computational Sciences
Goddard Space Flight Center / CSC
Greenbelt, MD USA

Miao Luo

D. K. Panda

Network-Based Computing Lab
The Ohio State University
Columbus, OH USA
Introduction

- Multi-core systems are now extremely common
- There can be problems with *contention* in multicore systems unless other components are scaled appropriately
  - Memory Speed / Capacity
  - Network?
- This type of contention needs to be evaluated and addressed

Intel Core i7 (Nehalem) - courtesy Intel
Network Contention

We are specifically interested in looking at network contention on a single node.

More cores, but usually only one network device per node.
Network Contention

How can we determine if there is network contention?

MPI is the most popular programming model for scientific computing
- How can we determine how much traffic is being contributed by each core?
- Particularly more important for increasing numbers of cores per node
Introduction:

### InfiniBand Overview

- InfiniBand is a popular high-speed interconnect
  - Minimum Latency: ~1-2us
  - Peak Bandwidth: ~1500-2500MB/s
- ~30% of Top500 now uses InfiniBand as the primary interconnect
- We will use it as our testbed
Userspace Interconnects

- If MPI is run over typical implementations of TCP/IP then the kernel knows the traffic

- InfiniBand is different….
  - The Host Channel Adapter (HCA) can be directly accessed from userspace applications
  - Cannot have kernel capture statistics!
Queue Pair (QP) Model

- Each QP consists of two queues:
  - Send Queue (SQ)
  - Receive Queue (RQ)
- A QP must be linked to a Completion Queue (CQ)
  - Gives notification of operation completion from QPs
Memory and Channel Semantics

- Memory: Remote Data Memory Access (RDMA)
  - No CPU interaction in copy (below)

- Channel: Receive buffers are posted to the QP Receive Queue (or SRQ) and consumed in order

RDMA Transfer with no CPU interaction
Problem Statement

- Is the network a bottleneck for systems of increasing numbers of cores? How can we see the traffic?

- If so, what can be done to reduce this contention and increase overall cluster throughput?
Presentation Outline

- Introduction
- Problem Statement
- Measuring Network Access
- Mixed Workloads & Evaluation
- Conclusions and Future Work
Measuring Network Traffic

We cannot measure network traffic per process directly on the host without additional hardware assistance.

Instead, can we instrument or time MPI calls to give network access information?
Measuring Network Traffic:

**MPI vs. Network**

MPI has functionality through PMPI interface to allow tools to intercept MPI calls and get timing information

- Used in many MPI profiling tools – Vampir, mpiP, etc.

Message completion in MPI in general does not imply network access has started or completed at that time

- Only means that buffer is available
- e.g. MPI_Send finishing only means that the send buffer can now be reused
Measuring Network Traffic:

**MPI vs. Network (contd.)**

MPI Time:

Network Usage Time:

*Timing for MPI and the network need not match*
The MPI library can know when message sends are initiated.

Unfortunately due to asynchronous nature of InfiniBand communication we must estimate duration of network access.

Use performance characteristics based on number of concurrent accesses by number of processes on the same node:

- e.g. If there are ‘n’ processes all sending large messages each will take roughly ‘n’ times longer to finish than if it were only one process sending.
- Use specially designed benchmarks to determine characteristics.
Presentation Outline

- Introduction
- Problem Statement
- Measuring Network Access
- **Mixed Workloads & Evaluation**
- Conclusions and Future Work
Mixed Workloads & Evaluation: Experimental Setup

Cluster Configuration:
- 128-core InfiniBand Cluster
- Quad Socket, Quad-Core Opteron 2GHz
- Mellanox DDR ConnectX HCA
- OpenFabrics Enterprise Edition (OFED) 1.3

Implementation
- Based off of MVAPICH2 1.2 (http://mvapich.cse.ohio-state.edu)
- MPI library used by over 960 organizations worldwide
- Offline analysis of performance data stored per node

Benchmarks
- We use the NAS Parallel Benchmarks
- Fluid dynamics kernels / mini apps
NAS FT (and many other applications) have a very structured pattern where network communication is done concurrently.
Mixed Workloads & Evaluation

Network Concurrency

Concurrent network access is very inefficient. The network is idle most of the time and then under significant contention.

What if we could reduce this contention?
- How much would the speedup be?
- What if half the number of cores were using the same network?
By running on half the cores we can see the benefit of having ½ as many cores per network adapter.

Using ‘Half Cores’ means twice as many nodes are needed.
Network Concurrency

We ran each NAS benchmark with ‘All Cores’ and ‘Half Cores’ configurations.

<table>
<thead>
<tr>
<th></th>
<th>BT</th>
<th>CG</th>
<th>EP</th>
<th>FT</th>
<th>IS</th>
<th>LU</th>
<th>MG</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Cores</td>
<td>198.22</td>
<td>34.82</td>
<td>28.53</td>
<td>44.19</td>
<td>3.10</td>
<td>180.46</td>
<td>156.14</td>
<td>196.29</td>
</tr>
<tr>
<td>Half Cores</td>
<td>196.71</td>
<td>26.17</td>
<td>29.07</td>
<td>38.89</td>
<td>2.29</td>
<td>179.86</td>
<td>14.78</td>
<td>188.84</td>
</tr>
<tr>
<td>(on double number of nodes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Half/All</td>
<td>99%</td>
<td>75.2%</td>
<td>99%</td>
<td>88.0%</td>
<td>73.9%</td>
<td>99.6%</td>
<td>97.6%</td>
<td>96.2%</td>
</tr>
</tbody>
</table>

Runtimes in seconds for different benchmarks.
Mixed Workloads & Evaluation

Mixed Workloads

β Others have previously suggested running multiple jobs on the same node since they have different requirements
  – Generally this has been done in the context of file I/O access, memory access, or cache usage

β We propose adding the network as a key component of deciding job co-location
  – NUMA already significantly segments nodes
  – We believe network is the main shared resource for contention
We evaluate each of the NAS Parallel Benchmarks with each other to determine what patterns can be scheduled together and achieve higher throughput.
Running FT together with itself shows how communication becomes offset.

Overall runtime is 89% of Exclusive run.
### Mixed Workloads & Evaluation

#### Shared Mode Evaluation

<table>
<thead>
<tr>
<th>Background Application</th>
<th>BT</th>
<th>CG</th>
<th>EP</th>
<th>FT</th>
<th>IS</th>
<th>LU</th>
<th>MG</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>BT</td>
<td>99.1%</td>
<td>77.3%</td>
<td>100.2</td>
<td>91.9%</td>
<td>81.6%</td>
<td>99.7%</td>
<td>98.4%</td>
<td>96.6%</td>
</tr>
<tr>
<td>CG</td>
<td>100.5%</td>
<td>101.8%</td>
<td>100.5</td>
<td>96.0%</td>
<td>90.2%</td>
<td>100.9</td>
<td>100.8%</td>
<td>102.0%</td>
</tr>
<tr>
<td>EP</td>
<td>98.8%</td>
<td>75.2%</td>
<td>99.6%</td>
<td>93.8%</td>
<td>80.1%</td>
<td>100.1</td>
<td>97.9%</td>
<td>97.2%</td>
</tr>
<tr>
<td>FT</td>
<td>99.4%</td>
<td>84.3%</td>
<td>99.9%</td>
<td>89.6%</td>
<td>87.6%</td>
<td>100.5</td>
<td>99.5%</td>
<td>98.9%</td>
</tr>
<tr>
<td>IS</td>
<td>100.2%</td>
<td>79.0%</td>
<td>99.1%</td>
<td>91.0%</td>
<td>84.4%</td>
<td>99.6%</td>
<td>98.8%</td>
<td>96.2%</td>
</tr>
<tr>
<td>LU</td>
<td>99.2%</td>
<td>76.2%</td>
<td>100.0</td>
<td>88.0%</td>
<td>80.7%</td>
<td>100.4</td>
<td>98.9%</td>
<td>97.0%</td>
</tr>
<tr>
<td>MG</td>
<td>99.0%</td>
<td>77.3%</td>
<td>100.4%</td>
<td>89.4%</td>
<td>73.9%</td>
<td>99.6%</td>
<td>98.1%</td>
<td>100.5%</td>
</tr>
<tr>
<td>SP</td>
<td>99.6%</td>
<td>79.2%</td>
<td>100.4%</td>
<td>93.2%</td>
<td>86.7%</td>
<td>100.3</td>
<td>97.6%</td>
<td>99.5%</td>
</tr>
</tbody>
</table>

### Measured Application

<table>
<thead>
<tr>
<th></th>
<th>BT</th>
<th>CG</th>
<th>EP</th>
<th>FT</th>
<th>IS</th>
<th>LU</th>
<th>MG</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>99%</td>
<td>75.2%</td>
<td>99%</td>
<td>88.0%</td>
<td>73.9%</td>
<td>99.6%</td>
<td>97.6%</td>
<td>96.2%</td>
</tr>
</tbody>
</table>
Mixed Workloads & Evaluation

Shared Mode: FT & LU

- FT is extremely bandwidth sensitive
- Less communication-intensive applications, such as LU don’t improve, but offer a perfect companion
- Why does CG/CG not work if FT/FT shows improvement?
Mixed Workloads & Evaluation

Shared Mode: CG & EP

- CG has a near-constant communication profile
- Few gaps – less possibility for improvement in other applications
Conclusion and Future Work

- Network contention is an increasing concern as additional cores are added
  - Many applications exhibit very synchronized communication
- Developed a method to profile network access for MPI applications on InfiniBand
- Showed 20% improvement when using a shared method of scheduling

Future:
- Better determination of symbiotic job scheduling / automated
- Studying the effect of QDR on this study
http://mvapich.cse.ohio-state.edu
Questions?

matthew.koop@nasa.gov
{l uom, panda}@cse.ohio-state.edu