Throughput Oriented Runtimes for Manycore Clusters

Costin Iancu, Khaled Ibrahim

Tsukuba – March 19
We **know** how to build runtimes that scale with the node count (**Latency**) 

**BUT**

We have **little experience** building runtimes that scale with **cores per node** (and nodes) (**Throughput**)
32 cores

<10 buses, NICs, wires

2012  2020
32 cores < 10 buses, NICs, wires

1000 cores < 10 buses, NICs, wires

2012 → 2020
Outline

- Network performance in manycore clusters
- Designing a throughput oriented runtime
- Future work
Network Performance
InfiniBand Performance

InfiniBand - 8 byte Msg Throughput

Bandwidth (MBs)

Cores Active

Proc
Hyb
Pth
InfiniBand Performance

InfiniBand - 8 byte Msg Throughput

<table>
<thead>
<tr>
<th>Cores Active</th>
<th>Bandwidth (MBs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
</tr>
</tbody>
</table>

- Proc
- Hyb
- Pth

Pthreads
Qthreads, Chapel, SoftXMT, Parallex
InfiniBand Performance

InfiniBand - 8 byte Msg Throughput

Bandwidth (MBs)

Cores Active

Proc
Hyb
Pth

Hybrid
Hierarchical, locales, X10, Chapel

Pthreads
Qthreads, Chapel, SoftXMT, Parallex
InfiniBand - 8 byte Msg Throughput

- **Proc**: Processes
- **Hyb**: Hybrid
- **Pth**: Pthreads

**Process Types**:
- Optimal, BUPO
- Hierarchical, locales, X10, Chapel
- Qthreads, Chapel, SoftXMT, Parallex

**Cores Active**:
- 4
- 8
- 12
- 16
- 20
- 24
- 28
- 32

**Bandwidth (MBs)**:
- 0
- 5
- 10
- 15
- 20
- 25
- 30
- 35
InfiniBand Performance

InfiniBand - 8 byte Msg Throughput

Bandwidth (MBs)

Software overhead increases with cores per node

Cores Active

Proc  Hyb  Pth

3X  5X
InfiniBand Performance

InfiniBand - 8 byte Msg Throughput

Bandwidth (MBs)

Software overhead increases with cores per node

Cores Active

Proc

Hyb

Pth

Lawrence Berkeley National Laboratory
InfiniBand Performance

InfiniBand - 8 byte Msg Throughput

- Software overhead increases with cores per node
- Non-scalable Throughput

Bandwidth (MBs)

Cores Active

Proc | Hyb | Pth

2X

3X

5X
BUPC/GASNet on InfiniBand

Throughput Improvement when Restricting Active Cores: InfiniBand

Active Cores

Size (B)
BUPC/GASNet on InfiniBand

Serializing communication using 16 cores 40% faster than using 32 cores
Cray MPI on Gemini

Throughput Improvement when Restricting Active Cores: MPI on Gemini
Cray MPI on Gemini

Serializing communication using 12 cores is 6X faster than using 48 cores.
Modern NICs are saturated by a small number of messages (cores*messages = K)

Low latency is not enough
- OpenIB Verbs and Cray DMAPP throughput worse than GASNet

Common behavior across software and hardware
- InfiniBand (Mellanox), Cray Gemini have similar behavior
- MPI, OpenIB and DMAPP have similar behavior

Behavior is likely to get worse
Future large scale systems are manycore

- Homogeneous or heterogeneous hardware
- Asymmetric software architecture or performance

Multithreading proven to work for irregular applications

Many research projects advocate “fine grained” multithreading (SoftXMT, Qthreads, ParalleX, SWARM)

- Energy efficiency
- Data movement
- Latency hiding
- Load balancing
- Performance and productivity

Fine grained multithreading requires network throughput
We do not know that we do not know how to build integrated “threading” and networking runtimes that provide good throughput

(building user level threading mapped on pthreads)
Throughput Oriented Runtime for Large Scale Manycore Systems (THOR)
What is optimal?
What is optimal?
What is optimal?

How to reach optimum in implementation independent manner?
What is optimal?

How to reach optimum in implementation independent manner?
What is optimal?

How to reach optimum in implementation independent manner?

How to achieve scalability with cores?
What is optimal?

How to achieve scalability with cores?

How to reach optimum in implementation independent manner?

Is it hardware or software?
Goals

Efficiency and productivity layer for heavily threaded/asynchronous runtimes

- Decouple application/runtime level concurrency from runtime concurrency
- Manage asynchrony for clients
- Provide optimal throughput for
  - Any implementation (pthreads, procs …)
  - Any hardware architecture (asymmetric, heterogeneous)
  - Any message mix
  - Any source, target
THOR Architecture

- Programming Models
  (SPMD, task and data parallel) (UPC, Chapel)

- Runtimes: SoftXMT, BUPC, Qthreads

- Admission Control Layer

- Optimization Layer

- Scheduling Layer

- GASNet/MPI

Driven by runtime analysis and performance models
Thor Layers

- **Admission Control Layer**
  - Congestion Avoidance
  - Flow Control
  - Memory Consistency/Ordering
  - Dispatch to Optimization Services

- **Optimization Layer**
  - Coalescing
  - Aggregation

- **Scheduling Layer**
  - Reordering
  - Instantiate and Retire Communication to Network
Scalable/Portable Design

- **Multiple implementations for asymmetric/heterogeneous hardware**
  - Inline: mechanisms implemented in a distributed manner, e.g. GASNet/MPI
  - Proxy: servers acting on behalf of clients

- **Open loop control**
  - With as little “global” state as possible

- **Declarative behavior**
  - Intuitive (human descriptions) – e.g. train of messages
  - Annotated by compilers/optimizers
Scalability with Cores

- Congestion avoidance runtime prototyped
  - BUPC/GASNet/InfiniBand
  - Cray UPC/DMAPP/Gemini
- Admission Control + Scheduling Layer
  - Not well tuned yet
- Results:
  - 4X performance improvement for all-to-all
  - 70% improvement on GUPS/HPCC RA
  - 17% on NAS Parallel Benchmarks

To appear as “Congestion Avoidance on Manycore Clusters” in ICS 2012
All-to-all InfiniBand 1024 Cores

Speedup over GASNet tuned all-to-all - 2x
Performance Portable – single implementation
All-to-all Gemini 768 Cores

Speedup over Cray UPC all-to-all - 4x
Performance Portable – single implementation
Future Work
Why don’t we get maximal throughput?

How do we get maximal throughput?
Thank You!
THOR Timeline

- **Year 1**: Performance study and prototype implementation
- **Year 2**: Admission Control Layer + Scheduling Layer
- **Year 3-4**: Optimization Layer and tuning on target systems

**Demonstration:**
- Languages: UPC, Chapel
- Runtimes: BUPC, SoftXMT (Qthreads?)
- Networks: InfiniBand, Cray
- CPUs: Intel/AMD, Knights Corner/Landing (other suggestions)