A Network Contention Model for the Extreme-scale Simulator

Christian Engelmann and Thomas Naughton

Computer Science and Mathematics Division
Oak Ridge National Laboratory, USA

34th IASTED International Conference on Modelling, Identification and Control (MIC) 2015

Innsbruck, Austria, February 17-18, 2015.
Scientific Computing and Simulation at ORNL
Motivation

• At the forefront of extreme-scale scientific computing
  • Titan at ORNL: Currently 2\textsuperscript{nd} fastest supercomputer in the world
  • 560,640 cores (AMD Opteron + NVIDIA Kepler GPUs, 17.6 PFlops)

• We are on road to exascale computing: 1,000 Pflop/s by 2023
  • Billions of cores (see next slides)

• There are several major challenges:
  • \textit{Power consumption}: Envelope of \textasciitilde{}20-40 MW (drives everything else)
  • \textit{Programmability}: Accelerators and PIM-like architectures
  • \textit{Performance}: Extreme-scale parallelism (up to 1B hardware threads)
  • \textit{Data movement}: Complex memory hierarchy and locality
  • \textit{Data management}: Too much data to track and store
  • \textit{Resilience}: Faults will occur continuously

# Discussed Exascale Road Map

Many design factors are driven by the power ceiling (op. costs)

<table>
<thead>
<tr>
<th>Systems</th>
<th>2009</th>
<th>2012</th>
<th>2017</th>
<th>2023</th>
</tr>
</thead>
<tbody>
<tr>
<td>System peak</td>
<td>2 Peta</td>
<td>20 Peta</td>
<td>100-200 Peta</td>
<td>1 Exa</td>
</tr>
<tr>
<td>System memory</td>
<td>0.3 PB</td>
<td>1.6 PB</td>
<td>5 PB</td>
<td>10 PB</td>
</tr>
<tr>
<td>Node performance</td>
<td>125 GF</td>
<td>200GF</td>
<td>200-400 GF</td>
<td>1-10TF</td>
</tr>
<tr>
<td>Node memory BW</td>
<td>25 GB/s</td>
<td>40 GB/s</td>
<td>100 GB/s</td>
<td>200-400 GB/s</td>
</tr>
<tr>
<td>Node concurrency</td>
<td>12</td>
<td>32</td>
<td>O(100)</td>
<td>O(1000)</td>
</tr>
<tr>
<td>Interconnect BW</td>
<td>1.5 GB/s</td>
<td>22 GB/s</td>
<td>25 GB/s</td>
<td>50 GB/s</td>
</tr>
<tr>
<td>System size (nodes)</td>
<td>18,700</td>
<td>100,000</td>
<td>500,000</td>
<td>O(million)</td>
</tr>
<tr>
<td>Total concurrency</td>
<td>225,000</td>
<td>3,200,000</td>
<td>O(50,000,000)</td>
<td>O(billion)</td>
</tr>
<tr>
<td>Storage</td>
<td>15 PB</td>
<td>30 PB</td>
<td>150 PB</td>
<td>300 PB</td>
</tr>
<tr>
<td>IO</td>
<td>0.2 TB/s</td>
<td>2 TB/s</td>
<td>10 TB/s</td>
<td>20 TB/s</td>
</tr>
<tr>
<td>MTTI</td>
<td>1-4 days</td>
<td>5-19 hours</td>
<td>50-230 min</td>
<td>22-120 min</td>
</tr>
<tr>
<td>Power</td>
<td>6 MW</td>
<td>~10 MW</td>
<td>~10 MW</td>
<td>~20 MW</td>
</tr>
</tbody>
</table>

HPC Hardware/Software Co-Design

• Helps closing the system-peak vs. application performance gap

• Develops HPC systems and applications jointly to deal with architecture limitations and application needs

• Employs hardware prototypes of future HPC architectures for performance evaluation at small scale

• Utilizes software simulation of future HPC architectures for performance evaluation at small and large scale

• Simulations investigate the impact of different architectural parameters on parallel application performance

• Parallel discrete event simulation (PDES) is often used with cycle accuracy at small scale and less accuracy at large scale
Overall Approach

• Execution of real applications, algorithms, or their models atop a simulated HPC environment at scale for:
  – Performance evaluation, including identification of resource contention and underutilization issues
  – Investigation at extreme scale, beyond the capabilities of existing simulation efforts

• xSim: A highly scalable solution that trades off accuracy
Technical Approach

• Combining highly oversubscribed execution, a virtual MPI, & a time-accurate PDES

• PDES uses the native MPI and simulates virtual procs.

• The virtual procs. expose a virtual MPI to applications

• Applications run within the context of virtual processors:
  – Global and local virtual time
  – Execution on native processor
  – Processor and network model

Simulator Design

- The simulator is a library
- Utilizes PMPI to intercept MPI calls and to hide the PDES
- Implemented in C with 2 threads per native process
- Support for C/Fortran MPI
- Easy to use:
  - Compile with xSim header
  - Link with the xSim library
  - Execute: `mpirun -np <np> <application> -xsim-<vp>`
Processor and Network Models

- Scaling processor model
  - Relative to native execution

- Configurable network model
  - Link latency & bandwidth
  - NIC contention and routing
  - Star, ring, mesh, torus, twisted torus, and tree
  - Hierarchical combinations, e.g., on-chip, on-node, & off-node
  - Simulated rendezvous protocol

- Example: NAS MG in a dual-core 3D mesh or twisted torus

Scaling a Monte Carlo Solver to $2^{24}$ Cores

**Shortcomings and Challenges**

- Configurable network model
  - Link latency & bandwidth only
  - NIC contention only
  - No full network contention modeling at the NIC or router
  - Multiple cores sharing a NIC not properly simulated:
    - IS program of the NAS Parallel Benchmark suite with a class C problem size on a 16-node Linux cluster with 8 cores per node.
  - Risk of high simulation overhead with full network contention modeling

---

New Network Modeling Features (1/2)

• Access contention modeling
  • Pre-computes a message’s path through networks and updates the injection list for each network based on flow direction: up/out and down/in
  • A message’s timestamps are compared with a network’s injection list, unused network time is allocated and the timestamps are updated to reflect injection contention
  • The injection lists are distributed for scalability, which limits accuracy

New Network Modeling Features (2/2)

- Capacity contention modeling
  - Maintains an additional usage list per network to track and limit transfer traffic
  - A message’s timestamps are compared with a network’s transfer list, unused network time is allocated and the timestamps are updated to reflect transfer contention
  - The transfer lists are distributed for scalability, which limits accuracy

Results (1/2)

- 128-core Linux cluster computer with 16 nodes, two processors per node, 4 cores per processor, 1 G Ethernet

- NAS Parallel Benchmark:
  - CG, a conjugate gradient solver (class B)
  - IS, an integer sort (class C)

- Simulation error reduced:
  - From 87% to 24.3% for CG
  - From 87.5% to 17.5% for IS

Results (2/2)

- Simulation overhead changed:
  - From 228% and 506% for CG
  - From 283% to 187% for IS
Conclusion

• The Extreme-scale Simulator (xSim) is a performance investigation toolkit that utilizes a PDES and oversubscription

• It supports a basic processor model and an advanced network model to simulate a future-generation HPC system

• The newly developed network modeling features increase accuracy while only slightly increasing simulation overhead
Questions