Network Contention Aware HPC job scheduling with Workload Precongnition

Xu Yang*#, John Jenkins+, Misbah Mubarak+, Robert Ross+, Zhiling Lan*

*Illinois Institute of Technology, Chicago, IL, USA
+Argonne National Laboratory, Argonne, IL, USA
# xyang56@hawk.iit.edu
Background

- Scale of supercomputers increases

<table>
<thead>
<tr>
<th></th>
<th>NO. of Nodes (core/node)</th>
<th>CPU Type</th>
<th>Peak Tflop/s (Gflop/s/core)</th>
<th>Network Topology</th>
<th>Link bw (GB/s)</th>
<th>Injection (GB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Intrepid</strong></td>
<td>40960 (04)</td>
<td>PowerPC450 (0.85)</td>
<td>557 (3.5)</td>
<td>3D torus</td>
<td>0.425</td>
<td>2.55</td>
</tr>
<tr>
<td><strong>Mira</strong></td>
<td>49152 (16)</td>
<td>PowerPC A2 (1.6)</td>
<td>10066 (12.8)</td>
<td>5D torus</td>
<td>2.000</td>
<td>20.00</td>
</tr>
</tbody>
</table>

- Concurrently running applications
  1. Network Contention
     - allocation shape
     - communication pattern
  2. Performance variability (up to 70%)
Background

Traditional batch scheduler

- **Scheduling Policies**
  1. FCFS—first come first serve
  2. STF—shortest runtime first
  3. FCFS/ Easy backfilling
  4. others based on different priorities

All these policies are priorities based. Priorities could be arrival time, runtime, size, user, etc
Background

Two types of allocation schemes on Torus system:

1. Non-contiguous (Cray XT)
   - no fragmentation
   - inter-process communication less efficient
   - network contention

2. Partition based scheduling system (IBM BlueGene/P, Q)
   - preserve locality
   - reduce network contention
   - fragmentation and low network utilization

Both have limited knowledge about Job and System:
- job information, such as arrival time, expected runtime, size
- system information, such as network topology, node availability
Observation

- **Job Communication Pattern**
  1. Nearest Neighbor
  2. P2P
  3. All to All
  4. Collectives

- **Repetitive Workload**
  1. stable user group
  2. limited application sets

The repetitive job submissions on Mira in 2014 (March-September). The percentage of repetitive submissions from certain user can be as much as 90%.
Motivation

• Compact allocation is not necessary for every application, depends on their communication pattern

• Job Scheduler should take jobs’ communication information for making scheduling decisions
  1. preferable allocation shape
  2. better task mapping

• Quantified analysis is needed for
  1. bandwidth requirement for job with specific communication pattern
  2. network contention between concurrently running jobs
  3. performance loss, i.e. extended job run time, lower system throughput

This is where CODES come to rescue!
CODES Job Allocation

• Things that CODES extension capable of
  1. Concurrently running jobs with given network topology
  2. Locate network contention
  3. Performance variability analysis

• Workload
  • DOE full apps and associate min-apps

http://portal.nersc.gov/project/CAL/designforward.htm

• Selected Apps
  1. AMG
  2. MiniFE
  3. Crystal Router
CODES Job Allocation

- Network Model
  2D torus

<table>
<thead>
<tr>
<th></th>
<th>AMG8</th>
<th>AMG27</th>
<th>MiniFE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO. of ranks</td>
<td>8</td>
<td>27</td>
<td>18</td>
</tr>
</tbody>
</table>
CODES Job Allocation

Mapping API

- num_apps
- ranks of app
- network topology
- node availability
- mapping decision

Hilbert Curve allocation

Or

Partition Based allocation
Communication Pattern V.S Alloc Shape

• All to All
  compact allocation with most bandwidth resource

• Nearest Neighbor
  if no compact allocation available, contiguous allocation will be adequate

• P2P and Collectives
  without heavy communication, can be flexible

Job with different communication patterns should get different allocation shapes
New Scheduler Design

Scheduler

- prioritizing & allocating control
- monitoring & workflow control
- optimized resource utilization
- record & report

job queue

job size
estimated runtime
communication pattern
repetitive workload

network topology
resource availability
scheduling decision
Future Work

- Provide more mapping strategies with different network topologies
- Explore the characteristics of different workloads
- Collective traces from other representative HPC applications