Topology-Aware Resource Selection

Joint work with: Emmanuel Jeannot, Guillaume Mercier
1

Context
The topology is not flat
Due to multicore processors current and future parallel machines are hierarchical

Not all the processes exchange the same amount of data

The speed of the communications, and hence performance of the application depends on the way processes are mapped to resources.
Process Placement Problem

Given:

• Parallel machine **topology**
• Process **affinity** (communication pattern)

Map processes to resources (cores) to reduce communication cost: a nice algorithmic problem:

• Graph partitioning (Scotch, Metis)
• Application tuning [Aktulga et al. Euro-Par 12]
• Topology-to-pattern matching (TreeMatch)
2

TreeMatch, a Process Placement Solution
We need affinity between processing elements: communication pattern

- Statically (thanks to compiler)
- Dynamic Monitoring
- Blank execution and tracing
- After data partitioning (e.g. Scotch)
Obtaining the topology

Abstract the topology with a tree
Assume communication always cost more when you need to reach higher levels
Putting everything together: Process Placement with TreeMatch

Communication matrix + Tree Topology = Process permutation
3

Resource selection
Selecting Resources

Model of the machine

Model of the application

TreeMatch Algorithm in the batch scheduler
Implementation

- Within SLURM 2.6.5
- Only with select/cons_res plugin
- srun
- Binding possibility with cgroup
- Resource selection and process placement at the same time
Why topology-aware resource selection could work?

<table>
<thead>
<tr>
<th></th>
<th>0-7</th>
<th>8-15</th>
<th>16-23</th>
<th>24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-7</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>8-15</td>
<td>1000</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>16-23</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>24-31</td>
<td>20</td>
<td>0</td>
<td>1000</td>
<td>0</td>
</tr>
</tbody>
</table>

SLURM Diagram:

- sw0
- sw1
- sw2
- sw3
- sw4
- sw5
- sw6
- sw7

- n0
- n1
- n2
- n3
- n4
- n5
- n6
- n7

- c0
- c1
- c2
- c3

- s0
- s1

- n0: p0, p4, p8, p12, p16, p20, p24, p28
- n1: p1, p5, p9, p13, p17, p21, p25, p29
- n2: p2, p6, p10, p14, p18, p22, p26, p30
- n3: p3, p7, p11, p15, p19, p23, p27, p31
Why topology-aware resource selection could work?
Why topology-aware resource selection could work?

SLURM and TreeMatch

<table>
<thead>
<tr>
<th>Time</th>
<th>Procs</th>
<th>0-7</th>
<th>8-15</th>
<th>16-23</th>
<th>24-31</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-7</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>8-15</td>
<td>1000</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1000</td>
</tr>
<tr>
<td>16-23</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>24-31</td>
<td>20</td>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Early experiments

• Same protocol as SLURM/Bull team.

• Simulation using real traces of the Curie CEA machine: 80640 cores.

• Model of performance gain of TreeMatch depending on the amount of communication performed by application (10%, 30%, 50%).

• Randomly generated communication matrices.

• Same starting workflow:
  – 130 running jobs
  – 26 queued jobs

• Submitted jobs from 372 (1 hour) to 14171 (100 hours).

• Evaluation on the difference of the submitted jobs.
Simulation: makespan

1 hour simulation

Ratio to SLURM

Percentage of Communication of the Application Runtime
Simulation: makespan

Percentage of communication: 30

Ratio to SLURM

Number of simulated hours

SLURM then TM
SLURM and TM

September 24, 2014
Simulation: average stretch

Percentage of communication: 30

Ratio to SLURM

SLURM then TM
SLURM and TM

Number of simulated hours
Simulation: average flow

Percentage of communication: 30

Ratio to SLURM

Number of simulated hours

SLURM then TM
SLURM and TM
• Simulation results encouraging

• Start a PhD to continue

• Future works:
  Emulation
  Complete the implementation

• Improvement ideas
  Build a usual communication matrix list
  Improve algorithmic of resource selection
Thanks!

www.inria.fr