A practical Approach to the Rating of Barrier Algorithms using the LogP Model and Open MPI

Torsten Höfler, Wolfgang Rehm
TU Chemnitz, Germany

24.05.2005
Outline

• Motivation

1 LogP Predictions

2 Implementation

3 Conclusions
Motivation

- optimal solution for the barrier problem
- barrier time complexity studies
- exhaustive comparison of different algorithms
- framework for general comparison studies
- Open MPI is easily extensible
- Question: is LogP accurate enough?
Problems

- unlimited number of architectures
  - generic optimal solution = holy grail?
- definition of several constraints for a given architecture
  - Fast Ethernet, Extreme Black Diamond Switch, 512 nodes
- new architectures have to be added by hand
- several models available -> LogP should be accurate enough
Principles

- one architecture as example
- easy testing of new architectures
- framework to implement and test new algorithms
Architectural Assumptions

- full bisectional bandwidth
- full duplex operation
- unlimited switch forwarding rate
- constant latency
- overhead bigger than gap
- overhead is constant ($o_s = o_r$)
Base Equations

several basic equations and variables:

\[ f_r = \max\{o_r, g\} \]
\[ f_s = \max\{o_s, g\} \]
\[ t_r = \max\{f_r, o_s + L + o_r\} \]
\[ = \max\{\max\{g, o_r\}, o_s + L + o_r\} \]
\[ = \max\{g, o_s + L + o_r\} \]

simplifying assumptions:

\[ f_r = f_s = o \]
\[ t_r = t_s = 2o + L \]
algorithms are divided into different complexity classes

- $O(P) \Rightarrow \textbf{Central Counter}$
- $O(n \cdot \log n P) \Rightarrow \textbf{Combining Tree}$, f-way Tournament, MCS
- $O(\log_2 P) + \text{Bcast} \Rightarrow \textbf{Tournament}$, BST
- $O(\log_2 P) \Rightarrow \textbf{Butterfly}$, Pairwise Exchange, \textbf{Dissemination}

- $O(\log_2 P)$ within the LogP is an optimal solution
- prove is trivial
- Assumption: Dissemination Barrier should perform best
Example - Dissemination Barrier

Step 1 [stage 0]:

Step 2 [stage 1]:

Step 3 [stage 2]:
Example - Dissemination Barrier

LogP Predictions
Implementation
Conclusions

Torsten Höfler, Wolfgang Rehm TU Chemnitz, Germany
Barrier Rating
Example - Dissemination Barrier

\[ rt = \max\{t_r, t_s\} \cdot \lceil \log_2 P \rceil \]
\[ (t_r = \max\{g, o_s + L + o_r\}) \]
Example - Dissemination Barrier

\[ \text{assume: } o > g \]

\[ rt = (2o + L) \cdot \left[ \log_2 P \right] \]
Benchmark Results

Dissemination Barrier

runtime in microseconds (rt)

# processors (P)

Dissemination barrier rating
Benchmark Results

Tournament Barrier

The graph shows the runtime in microseconds (rt) against the number of processors (P). The data points indicate the performance of the Tournament Barrier algorithm across different processor counts. The trend line suggests an increasing runtime as the number of processors increases.
Benchmark Results

Algorithm Comparison

- Central Counter
- Combining Tree (n=4)
- Tournament Barrier
- Dissemination
- Open MPI

runtime in microseconds (µs) vs. # processors (P)
### Benchmark Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>128 nodes</th>
<th>256 nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central Counter</td>
<td>4594.50µs</td>
<td>4909.67µs</td>
</tr>
<tr>
<td>Combining Tree</td>
<td>4009.79µs</td>
<td>4343.63µs</td>
</tr>
<tr>
<td>Tournament</td>
<td>3642.54µs</td>
<td>4378.77µs</td>
</tr>
<tr>
<td>Dissemination</td>
<td>1904.57µs</td>
<td>1977.12µs</td>
</tr>
<tr>
<td>Open MPI</td>
<td>3559.88µs</td>
<td>4226.88µs</td>
</tr>
</tbody>
</table>
Open MPI

- also useable for production environments
- ⇒ Open MPI as MPI framework
Open MPI

- also useable for production environments
- $\Rightarrow$ Open MPI as MPI framework

![Diagram]

User Application

MPI API

Run Time Environment (RTE)

Modular Component Architecture (MCA)

- TOPO
- COLL
- PtP Mgmt. Layer (PML)
- PTL
- PTL
- PTL
Component Implementation

- Initialization returns user-defined priority
- Algorithm selection:
  - 0: automatic benchmark
  - 1: Central Counter
  - 2: Combining Tree
  - 3: Tournament
  - 4: Dissemination
  - 5: Binomial Tree
  - 6: n-way Dissemination
- Checkpoint/Restart is handled by lower layers
Conclusions

- taken assumptions are valid
- LogP model is accurate
- Dissemination is optimal for given scenario
- different networks exhibit different behavior
- derivation of new algorithms for different hardware (e.g. offloading based HW) could require detailed models
  $\Rightarrow$ general methodology for developing optimal barrier algorithms has been shown
Future Work

- new model for small messages for offloading based NICs (LoP)
- new barrier algorithms to support hardware parallelism
- simplification of the LoP model (non linear, >6 parameters)