Fast Barrier Synchronization for InfiniBand™

Torsten Höfler
Department of Computer Science
TU Chemnitz

(Talk held at the Technical University of Munich)

29.09.2005
Outline

1. Introduction
   - Terms
   - Motivation

2. Modelling
   - The LogP Model
   - The LoP Model

3. Barrier Algorithms
   - Well known Algorithms
   - InfiniBand™ specific Barrier

4. Summary
Outline

1 Introduction
   - Terms
   - Motivation

2 Modelling
   - The LogP Model
   - The LoP Model

3 Barrier Algorithms
   - Well known Algorithms
   - InfiniBand™ specific Barrier

4 Summary
InfiniBand™ Cluster

What is a Cluster?
- some computers with an interconnection network
- e.g. InfiniBand™ cluster @ TUM

...MPI?
- Message Passing Interface
- portable abstraction layer

...InfiniBand™?
- high-speed interconnection technology
- RDMA Read/Write features
- scalable bandwidth
- user level access
- reliable and unreliable transport
- multicast capability
"MPI_Barrier blocks the caller until all group members have called it. The call returns at any process only after all group members have entered the call" (MPI-1.1 Standard)
Outline

1 Introduction
   - Terms
   - Motivation

2 Modelling
   - The LogP Model
   - The LoP Model

3 Barrier Algorithms
   - Well known Algorithms
   - InfiniBand™ specific Barrier

4 Summary
Motivation

- barrier performance is crucial
- e.g.
  - access to shared resources
  - collective operations with hardware support
- existing implementations deny architectural specialities
- no constant-time barrier for InfiniBand™
Outline

1. Introduction
   - Terms
   - Motivation

2. Modelling
   - The LogP Model
   - The LoP Model

3. Barrier Algorithms
   - Well known Algorithms
   - InfiniBand™ specific Barrier

4. Summary
network model proposed by Culler et. al.
several system parameters
- L - Latency
- o - overhead
- g - gap
- P - number of processors
network model proposed by Culler et. al.
several system parameters
- $L$ - Latency
- $o$ - overhead
- $g$ - gap
- $P$ - number of processors
network model proposed by Culler et. al.
several system parameters
- L - Latency
- o - overhead
- g - gap
- P - number of processors

**The LogP Model**

The LogP model is a network model proposed by Culler et al. It involves several system parameters:
- **L**: Latency
- **o**: Overhead
- **g**: Gap
- **P**: Number of processors

**The LoP Model**

This model extends the LogP model by introducing additional parameters and considerations.
network model proposed by Culler et. al.
several system parameters
- L - Latency
- o - overhead
- g - gap
- P - number of processors
Outline

1. Introduction
   - Terms
   - Motivation

2. Modelling
   - The LogP Model
   - The LoP Model

3. Barrier Algorithms
   - Well known Algorithms
   - InfiniBand™ specific Barrier

4. Summary
- considers all architectural details
- LogPh not parametrizable
- LoP as more abstract approach
- considers all architectural details
- LogPh not parametrizable
- LoP as more abstract approach

![Diagram showing CPU, HCA, and Network levels over time]

---

Torsten Höfler Department of Computer Science TU Chemnitz

MPI Barrier Synchronization
• considers all architectural details
• LogPh not parametrizable
• LoP as more abstract approach
LoP Parameters

- $o(P) = \lambda_1 + \frac{\lambda_2}{\lambda_3 + P}$
- $RTT(P) = \frac{\lambda_1}{\lambda_2 + P} + \lambda_3 + \lambda_4 \cdot (1 - e^{\lambda_5 \cdot (P - \lambda_6)})$

![Graph showing $t$ vs $P$ with $t_{\text{min}}$ and $t$ axis labeled]
LoP Parameters

- \( o(P) = \lambda_1 + \frac{\lambda_2}{\lambda_3 + P} \)
- \( RTT(P) = \frac{\lambda_1}{\lambda_2 + P} + \lambda_3 + \lambda_4 \cdot (1 - e^{\lambda_5 \cdot (P - \lambda_6)}) \)
InfiniBand™ Benchmark

Sender

```c
take_time(t0);
p * VAPI_post_rr();
take_time(t1);
p * VAPI_post_sr();
take_time(t2);
p * VAPI_poll_cq();
take_time(t3);
p * VAPI_poll_cq();
take_time(t4);
```

IBA

Receiver

```c
VAPI_post_rr();
VAPI_poll_cq();
VAPI_post_sr();
VAPI_poll_cq();
```
LoP Measurements

\[ t_{srov}^{rdmaw,i}(P) = 0.54 + \frac{0.50}{-0.49 + P} \]

\[ t_{rtt,min}^{rdmaw,i}(P) = 3.01 + \frac{14.96}{0.04 + P} + 6.19 \cdot (1 - e^{-0.05 \cdot (P - 8.27)}) \]
LoP Measurements

\[ t_{srov}^{rdmaw,i}(P) = 0.54 + \frac{0.50}{-0.49 + P} \]
\[ t_{rtt,\text{min}}^{rdmaw,i}(P) = 3.01 + \frac{14.96}{0.04 + P} + 6.19 \cdot (1 - e^{-0.05 \cdot (P - 8.27)}) \]
Outline

1. Introduction
   - Terms
   - Motivation

2. Modelling
   - The LogP Model
   - The LoP Model

3. Barrier Algorithms
   - Well known Algorithms
     - InfiniBand™ specific Barrier

4. Summary
Central Counter

- easiest implementation / scales worst - $O(P)$
- a single counter counts arrived nodes
Central Counter

- easiest implementation / scales worst - $O(P)$
- a single counter counts arrived nodes

0
1
2

0
0
0

normal calculation
Central Counter

- easiest implementation / scales worst - $O(P)$
- a single counter counts arrived nodes

0 \rightarrow 1

node 1 reached barrier
Central Counter

- easiest implementation / scales worst - $O(P)$
- a single counter counts arrived nodes

node 2 reached barrier
Central Counter

- easiest implementation / scales worst - $O(P)$
- a single counter counts arrived nodes

node 0 reached barrier
Central Counter

- easiest implementation / scales worst - $O(P)$
- a single counter counts arrived nodes

Node 0 notifies all group members
Central Counter

- easiest implementation / scales worst - $O(P)$
- a single counter counts arrived nodes

![](image)

barrier done (all continue)
Dissemination Barrier

Step 1 [stage 0]:

```
0 1 2 3 4 5
```

Communication Diagram

```
0 1 2 3 4 5
```

Step 2 [stage 1]:

```
0 1 2 3 4 5
```

Communication Diagram

```
0 1 2 3 4 5
```

Step 3 [stage 2]:

```
0 1 2 3 4 5
```

Communication Diagram

```
0 1 2 3 4 5
```

3 competing network transactions
Overview

- Central Counter - $O(P)$
- Combining Tree - $O(n \cdot \log_n P)$
- Tournament - $O(\log_2 P)$
- f-way Tournament - $O(f \cdot \log_f P)$
- MCS - $O(n \cdot \log_n P)$
- BST - $O(\log_2 P)$
- Butterfly - $O(\log_2 P)$
- Pairwise Exchange - $O(\log_2 P)$
- Dissemination - $O(\log_2 P)$
Outline

1. Introduction
   - Terms
   - Motivation

2. Modelling
   - The LogP Model
   - The LoP Model

3. Barrier Algorithms
   - Well known Algorithms
   - InfiniBand™ specific Barrier

4. Summary
n-way Dissemination

- new algorithm
- uses the dissemination principle
- each node sends to $n$ nodes per round
- exploits hardware parallelism (LoP)
- 1-way dissemination = normal dissemination
2-way Dissemination Barrier

Round 0:

Round 1:
InfiniBand™ Implementation - Results

The diagram shows the average runtime in microseconds for different algorithms as a function of the number of processors. The algorithms compared include LAM-MPI, SHIBA, MVAPICH, IBBARR-1, IBBARR-2, IBBARR-3, IBBARR-4, and IBBARR-6. The x-axis represents the number of processors (P), and the y-axis represents the average runtime in microseconds (rt).
InfiniBand™ Implementation - Results

The graph shows the average runtime in microseconds (rt) vs. the number of processors (P) for different implementations of MPI Barrier Synchronization. The implementations are:

- MVAPICH
- IBBARR-1
- IBBARR-2
- IBBARR-3

The results indicate that as the number of processors increases, the average runtime also increases for all implementations. MVAPICH consistently shows the highest average runtime, followed by IBBARR-1, IBBARR-2, and IBBARR-3 in that order.
InfiniBand™ Implementation - Results

The graph shows the speedup of InfiniBand™ specific Barrier Algorithms (IBBARR-1, IBBARR-2, IBBARR-3) as a function of the number of processors (P). The graph indicates that the speedup generally increases with the number of processors, with IBBARR-3 showing the most consistent improvement as the number of processors increases.
Results

- proof of optimality (LogP) of the Dissemination Barrier
- new communication model for InfiniBand™ (LoP)
- extensive InfiniBand™ benchmark to parametrize LoP
- best barrier solution for InfiniBand™ (ibbarr - 40%)
- design of barrier for high speed interconnect switches
Combining Tree

Step 1-4 (one fetch and increment for each node per group):

Step 5-6:

Step 6-11:

Communication Diagram

9 competing network transactions
Appendix

**Tournament Barrier**

Step 1 (pairwise games - the node with the lower id wins) [stage 0]:

0 1 2 3 4 5

Step 2 [stage 1]:

0 1 2 3 4 5

Step 3 [stage 2]:

0 1 2 3 4 5

Step 4-8 [stage 3]:

0 1 2 3 4 5

Communication Diagram:

0 1 2 3 4 5

8 competing network transactions

Time

0 1 2 3 4 5

MPI Barrier Synchronization