Performance of Cluster-enabled OpenMP for the SCASH Software Distributed Shared Memory System

Yoshinori Ojima[1)]

Mitsuhisa Sato[1)]

Hiroshi Harada[2)]

Yutaka Ishikawa[3)]

1) Information Science and Electronics, University of Tsukuba

2) Hewlett-Packard Japan,Ltd

3) The University of Tokyo

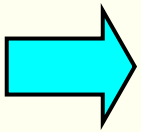
HPCS Lab.

Overview

- Background, objective
- Software DSM system SCASH
- Omni/SCASH: OpenMP implementation for Software DSM
- Comparison of the basic performance of Myrinet and Ethernet
- Performance evaluation, Discussion
- Conclusion, future work

Background

- PC Cluster became a popular parallel computing platform.
- Distributed memory programming
  - Message passing library(MPI, PVM)
  - Programming cost is large
- Shared memory programming
  - Programmers can parallelize easily with OpenMP
  - Programming cost is small



Omni/SCASH : OpenMP for Software Distributed Shared Memory(DSM) System

This is cluster-enabled OpenMP

The objectives of research

- To evaluate performance of Omni/SCASH
- To investigate the performance factor depending on the communication performance of networks
- To investigate the problem of using a commodity network (Ethernet) as well as Myrinet

Software DSM System SCASH

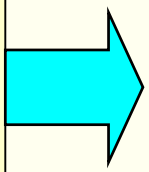
- A Software DSM System as a part of SCore cluster-system software
- Uses PM communication library, implemented as user level library
- Per page-basis (using kernel page-faults)
- Two page consistency protocols
  - invalidate and update
- Eager Release Consistency(ERC) memory model with multiple writer protocol (diff)
- Consistency maintenance communication at synchronization point

Omni/SCASH

- OpenMP Compiler for SCASH
  - It translates OpenMP programs to multi-threaded programs linked to SCASH runtime library

OpenMP

All variables are shared by default
No explicit shared memory allocation



"shmem" memory model

All variables declared statically in global scope are private.
The shared address space must be allocated by a library function at runtime.

Transformation for “shmem” model

- OpenMP compiler for “shmem” memory model
  - Detects references to a shared data object
  - Rewrite it to the references to the objects which are allocated in shared address space at runtime.
  - A global variable declaration
    ⇨ a declaration of the pointer which will point into a shared object at runtime.

```
double x;
double a[100];
…
a[10] = x;
```

```
double *_G_x;
double *_G_a;
…
(_G_a)[10] = (*_G_x);
…
static _G_DATA_INIT(){
  _shm_data_init(&_G_x,8,0);
  _shm_data_init(&_G_a,800,0);
}
```

Extension of Omni/SCASH to OpenMP

- In Software DSMs, the allocation of pages to home nodes greater affects performance
- OpenMP
  - doesn't provide facilities for specifying how data is to be arranged within the memory space
  - there are no loop scheduling methods to define the way in which data is passed between iterations

⇨ Extension to OpenMP

An example of the extension

- Data mapping directive

```
double a[100][200];
#pragma omni mapping(a[block][*])
```

- Loop scheduling clause, "affinity"

```
#pragma omp for schedule(affinity,a[i][*])
for(i = 1; i < 99; i++)
    for(j = 0; j < 200; j++)
        a[i][j] = ...;
```

Comparison of basic performance of network

Basic performance of network

- Comparison of basic communication performance of Ethernet and Myrinet
  - Page transmission cost
  - Overhead of barrier operation
- Measurement condition
  - programs are parallelized with OpenMP
  - 1 processor per node is used

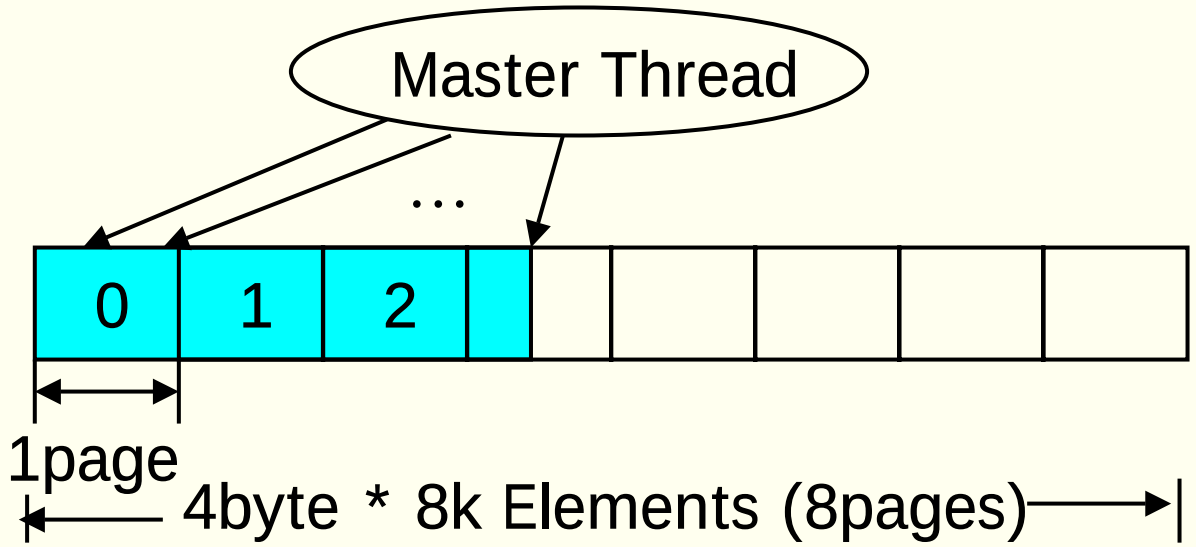
Evaluation platform

- PC cluster “COSMO”

CPU	PentiumII Xeon 450MHz(4-way SMP)
L2 Cache	1MB
Memory	2GB
Nodes	8
Network	800Mbps Myrinet, 100base-TX Ethernet
OS	Linux Kernel 2.4.18
SCore	version 5.0.1
Compiler	gcc 2.96(Optimize option –O4)

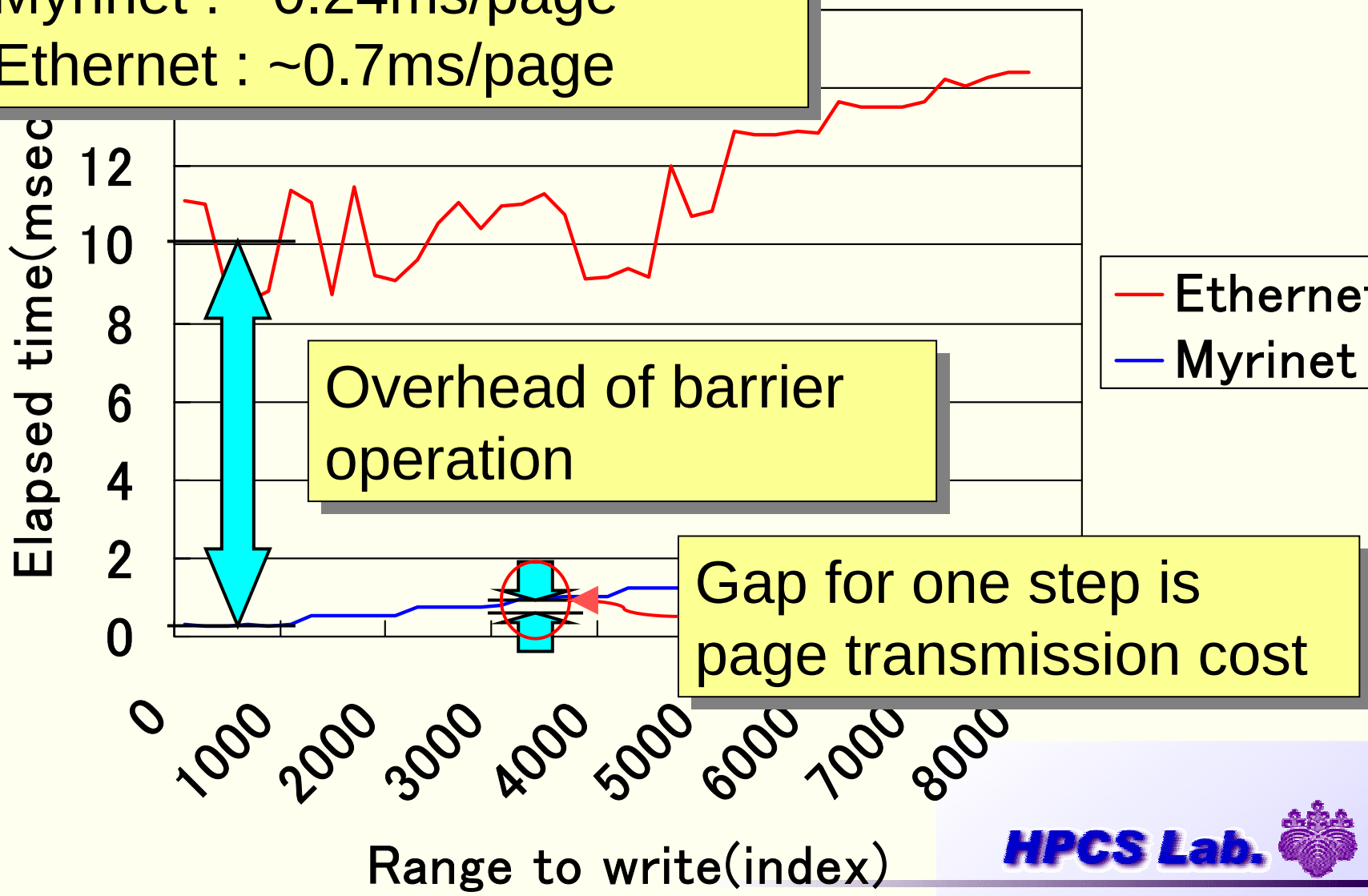
Page transmission cost

- One dimensional array of 32KB(8pages)
- Master thread writes to every element in an array which is followed by a barrier point
- At the barrier point, modified data is copied back to their home nodes.
- Execution time from the beginning of the array write operation to completion of barrier operation



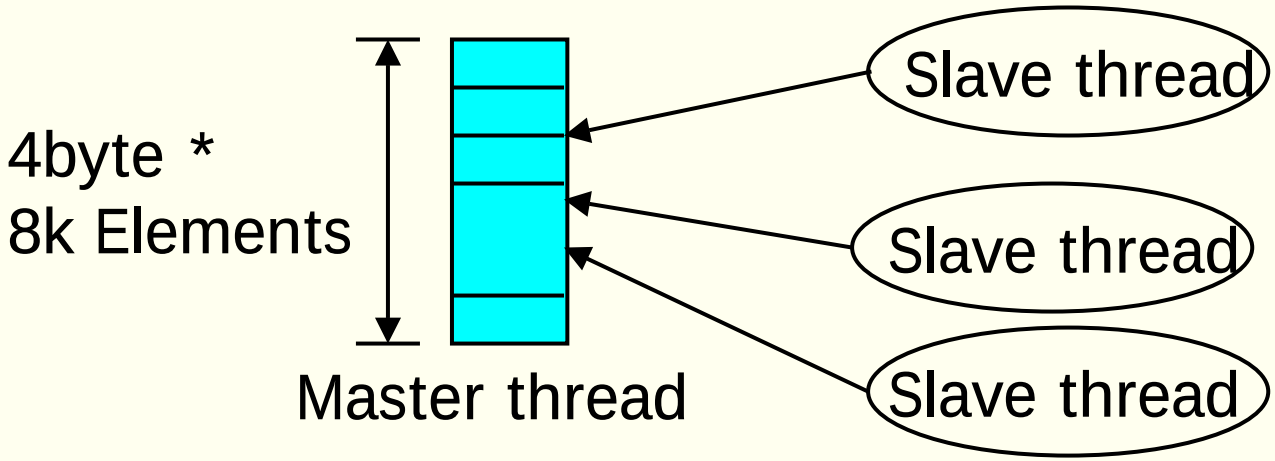
Page transmission cost

Myrinet : ~0.24ms/page
Ethernet : ~0.7ms/page



Overhead of barrier operation

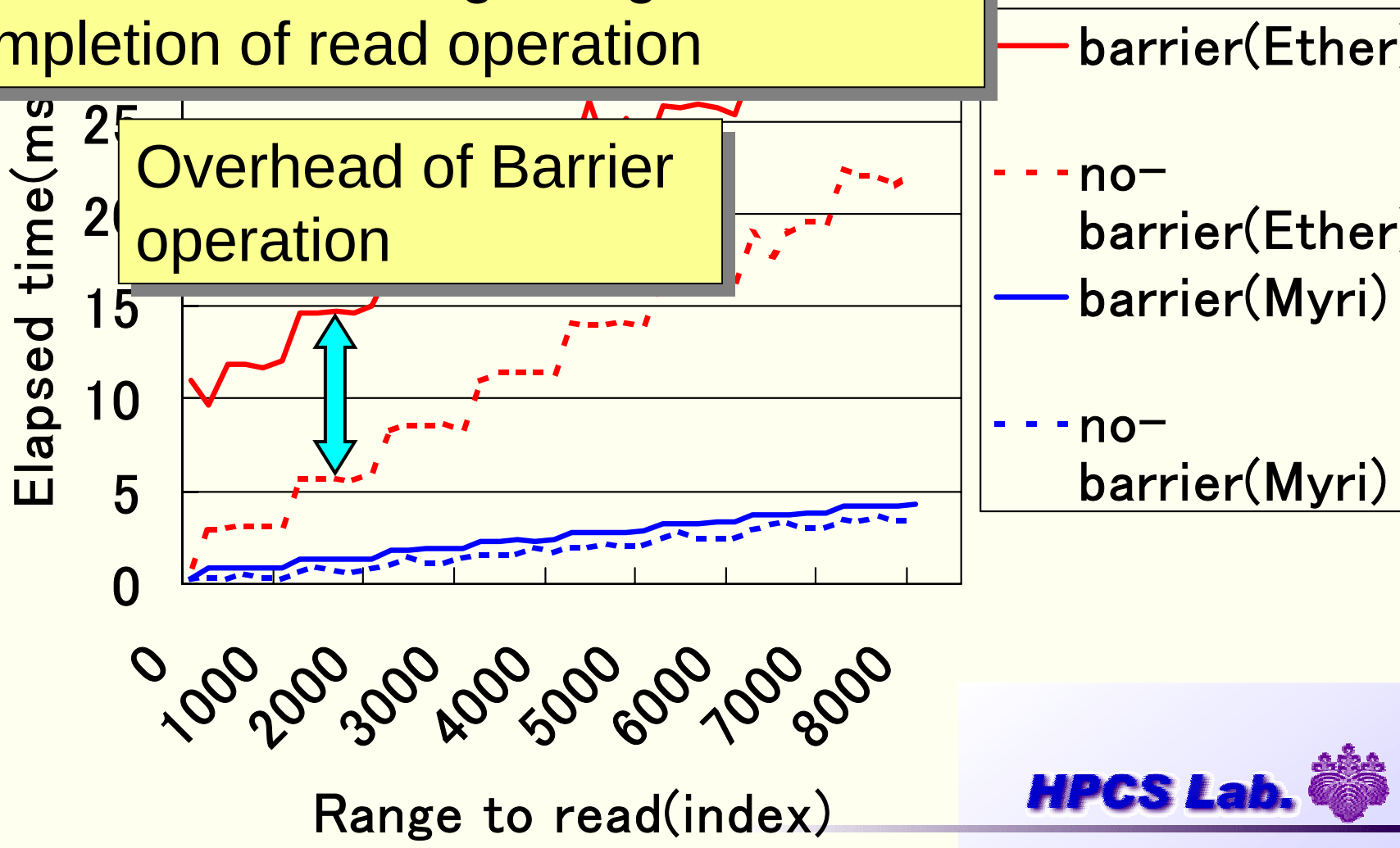
- Two-dimensional array of 256KB(8pages * 8) is allocated in shared memory space
- It is mapped with block distribution
- Slave threads read every element mapped to master thread which is followed by a barrier point
- At the barrier point, no consistency maintenance communication occurs



Barrier : from beginning to completion of barrier operation

No-barrier : from beginning to completion of read operation

ation



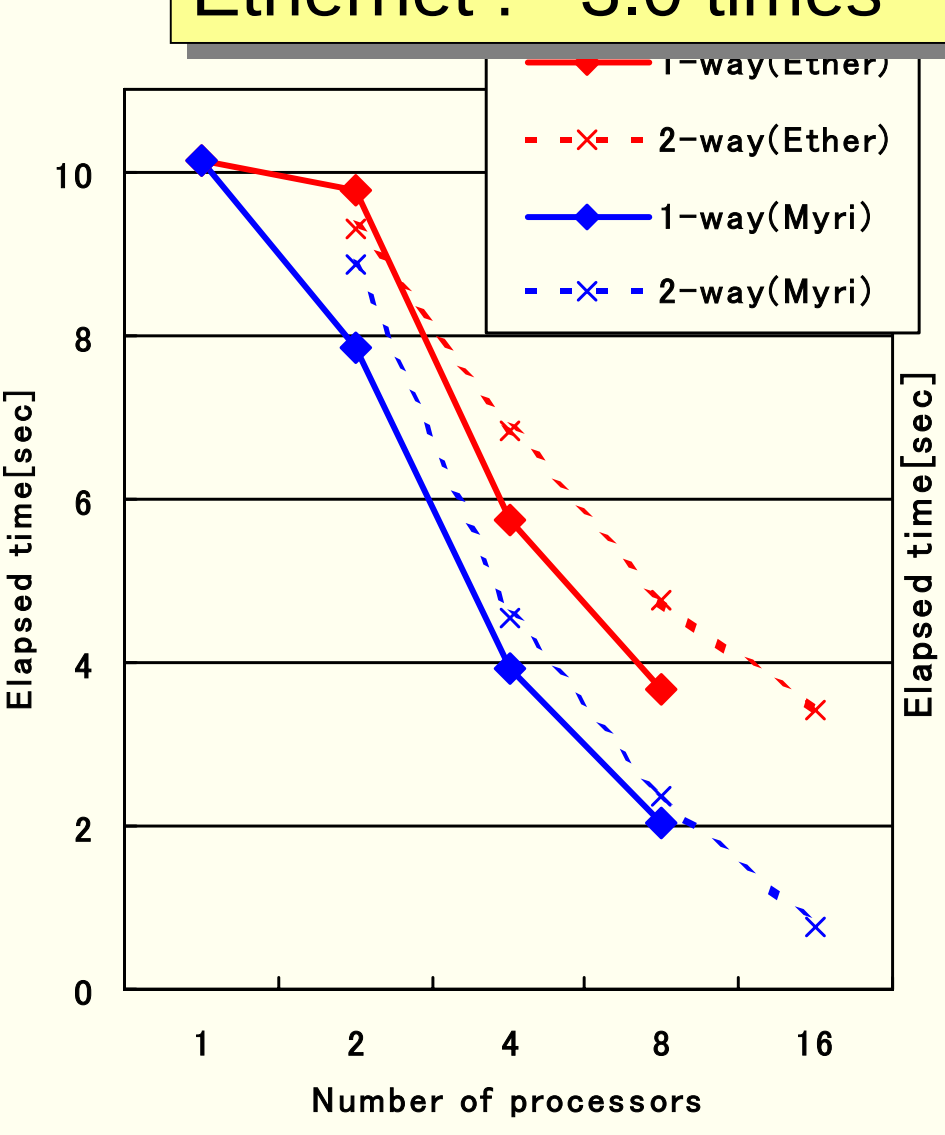
Performance Evaluation

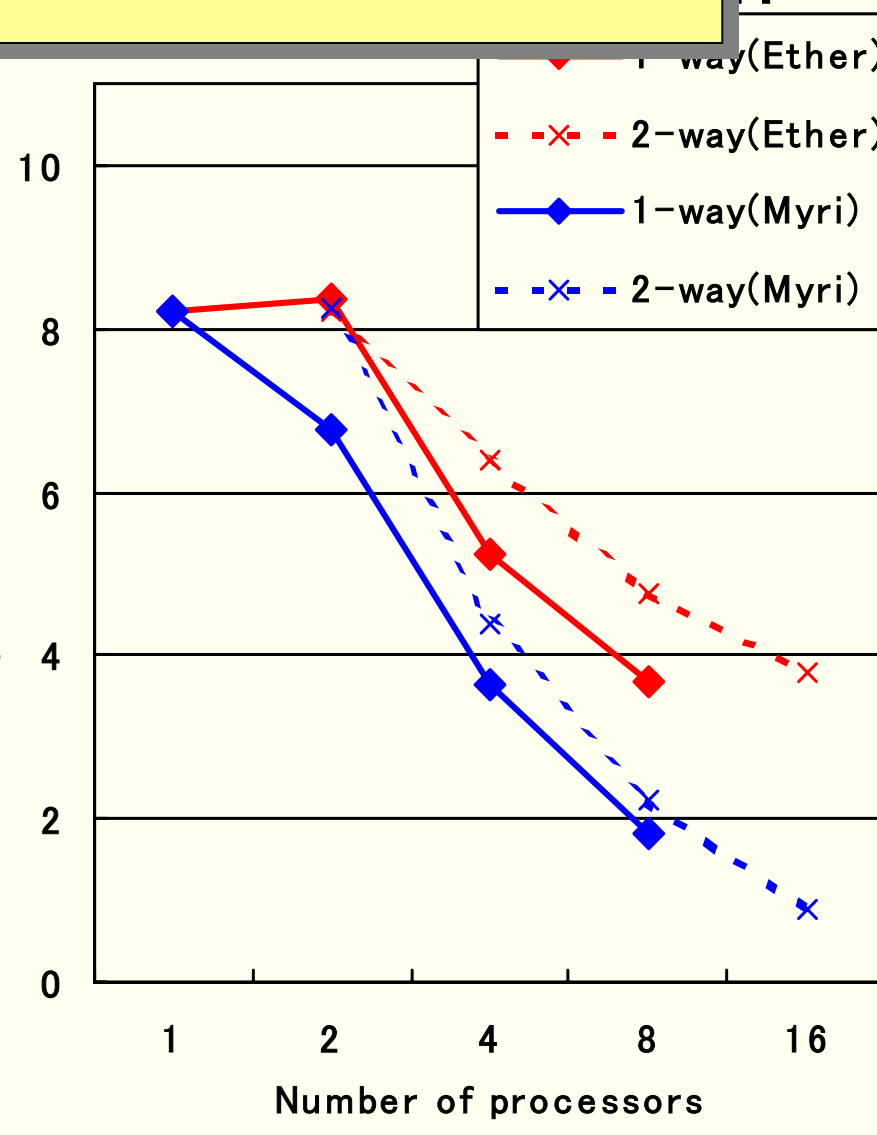
Performance Evaluation

- laplace
  - A simple Laplace equation solver using a Jacobi 5-point stencil operation
  - Written in C
  - Two versions
    - Parallelized with OpenMP(OpenMP version)
    - Written using SCASH library(SCASH version)
  - The array size is 1024 × 1024(double precision)
  - The number of iteration is 50
- NPB EP
  - Fortran, parallelized with OpenMP(by RWC), Class A
- NPB BT, SP
  - Fortran, parallelized with OpenMP, Class A
  - Affinity scheduling

Myrinet : ~13.3 times with 16 processors

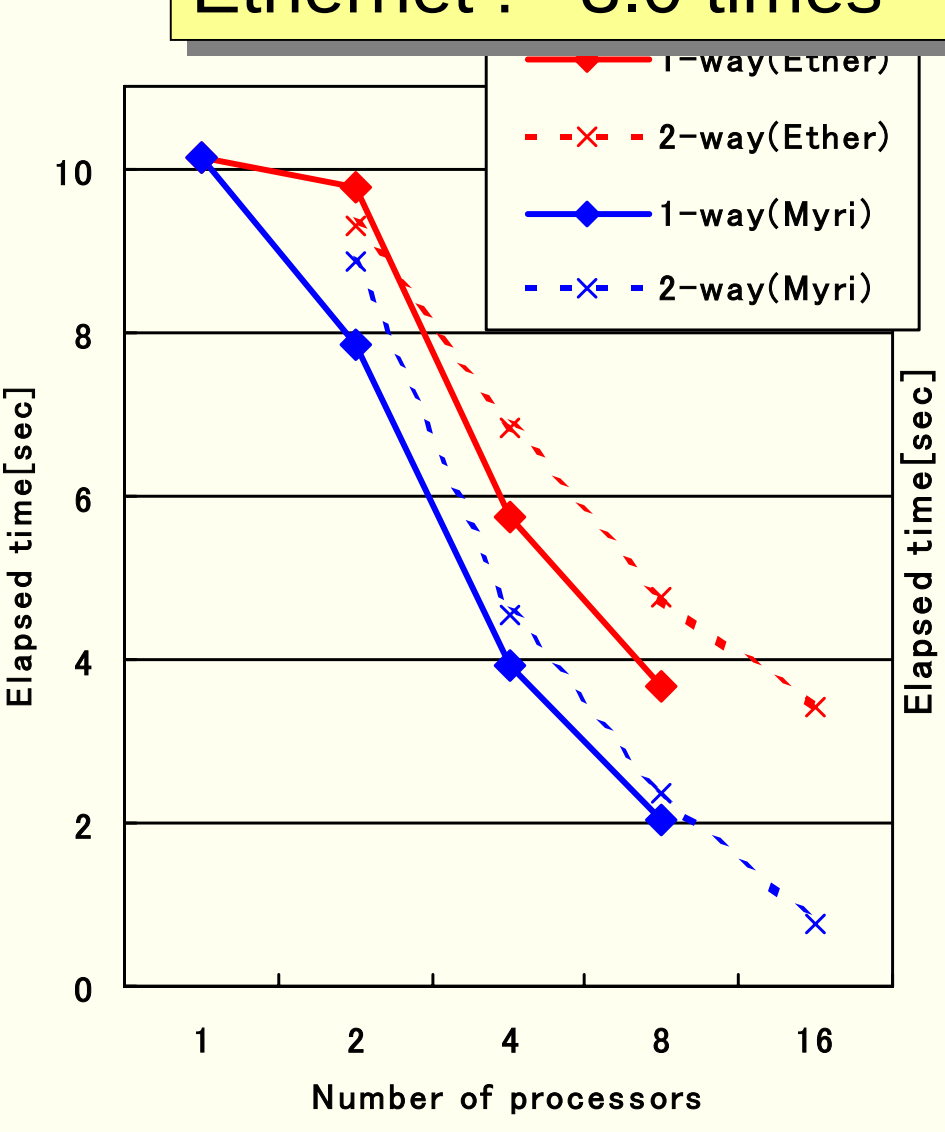
Ethernet : ~3.0 times

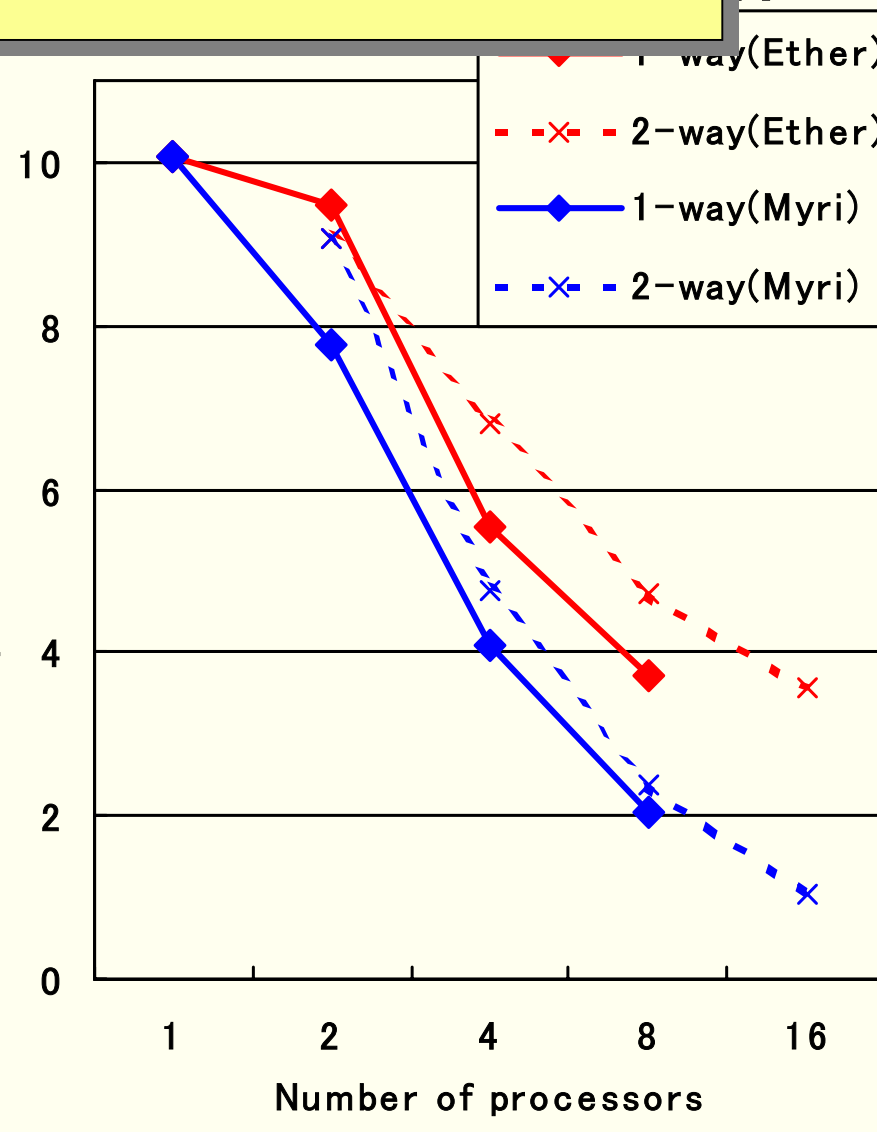




Myrinet : ~13.3 times with 16 processors

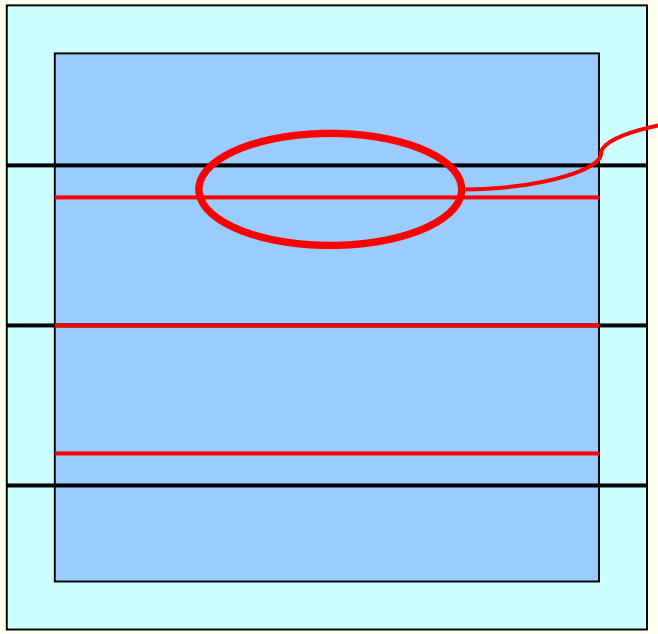
Ethernet : ~3.0 times





Application of affinity scheduling

- Laplace(OpenMP version)

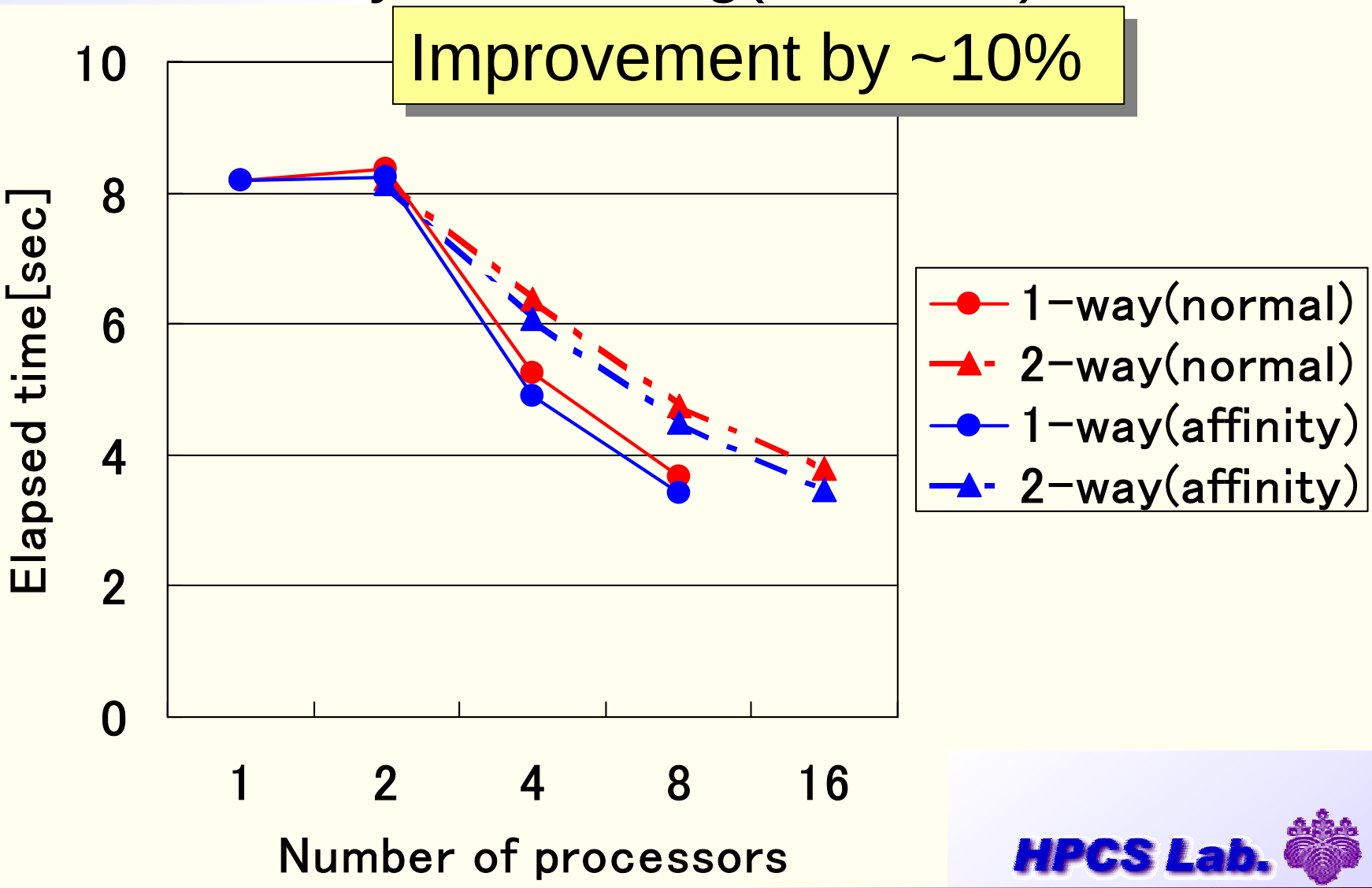


A mismatch exists between the alignment of data and loops

The number of page faults increases

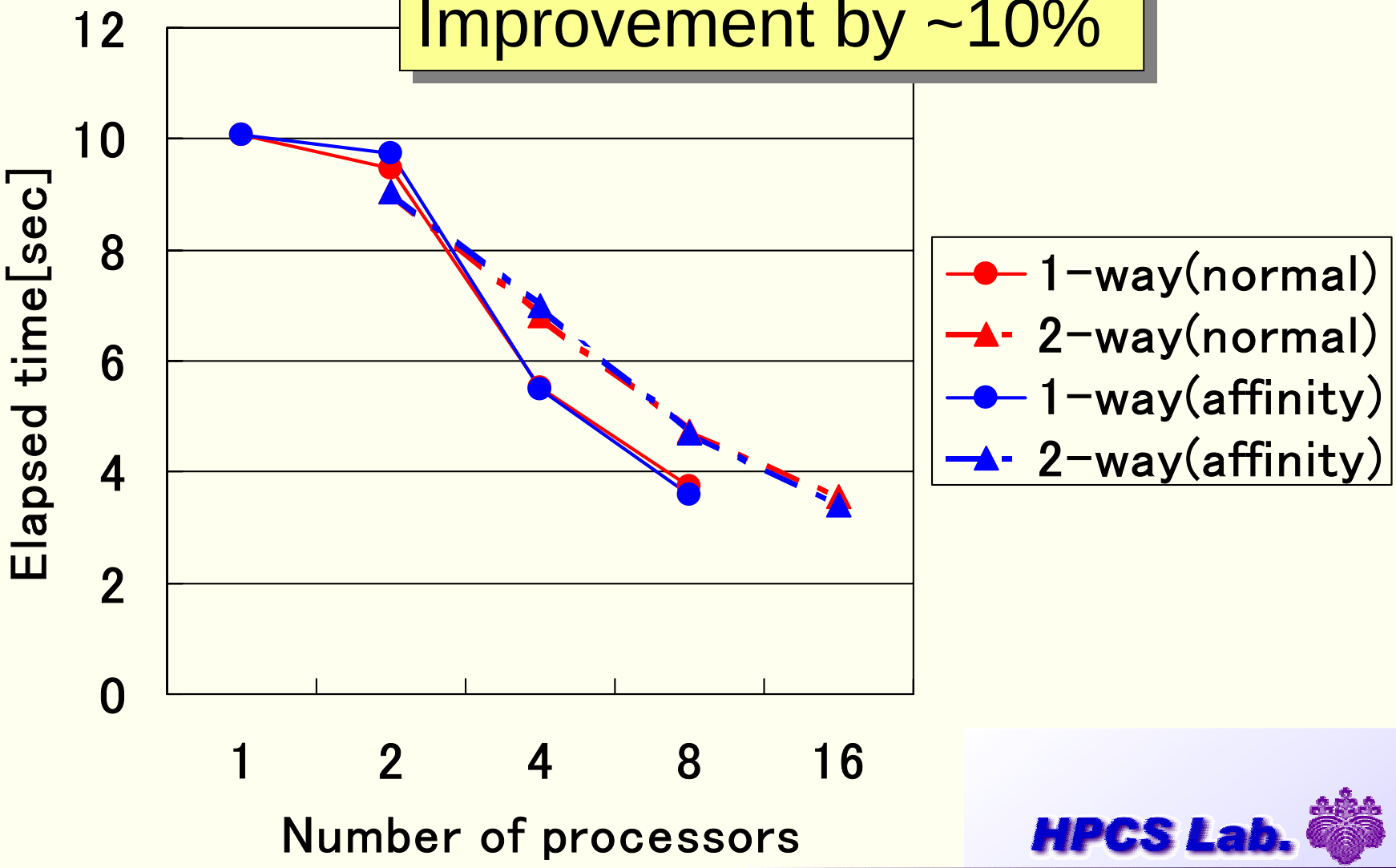
affinity scheduling to match alignment of data and loops

Peformance improvement with affinity scheduling(Ethernet)



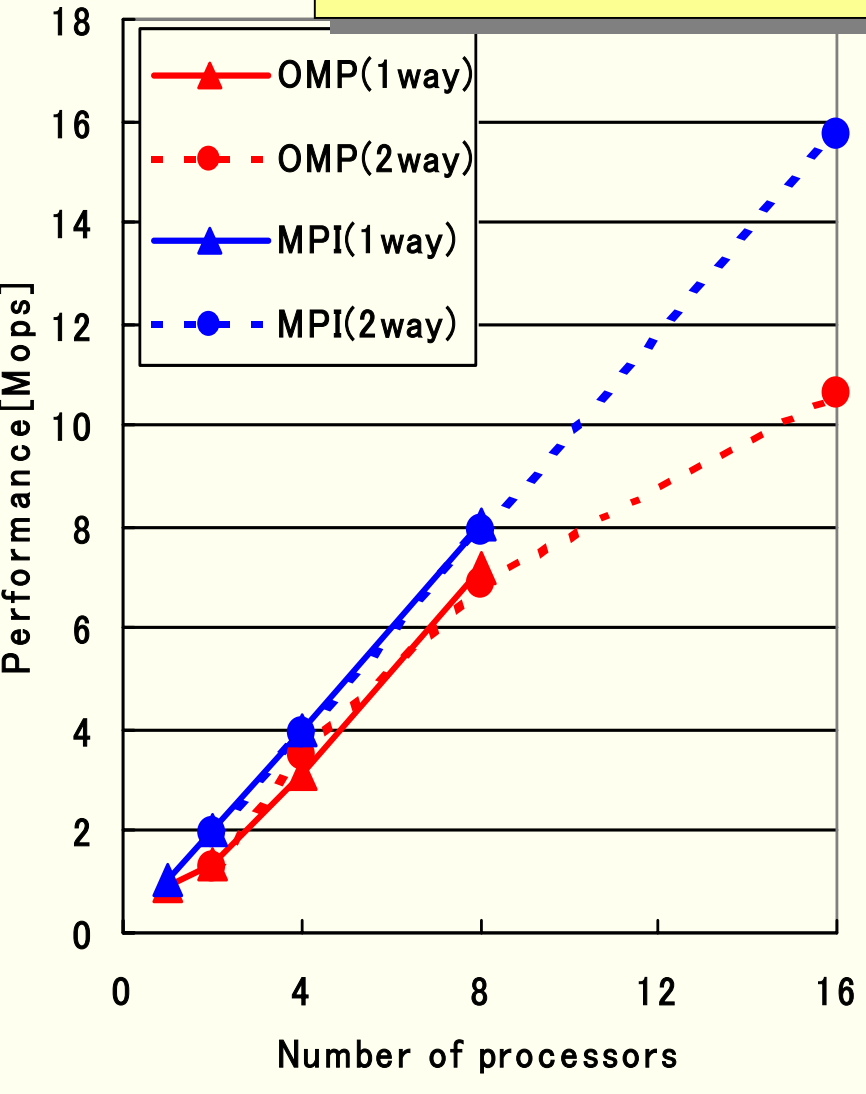
Peformance improvement with affinity scheduling(Ethernet)

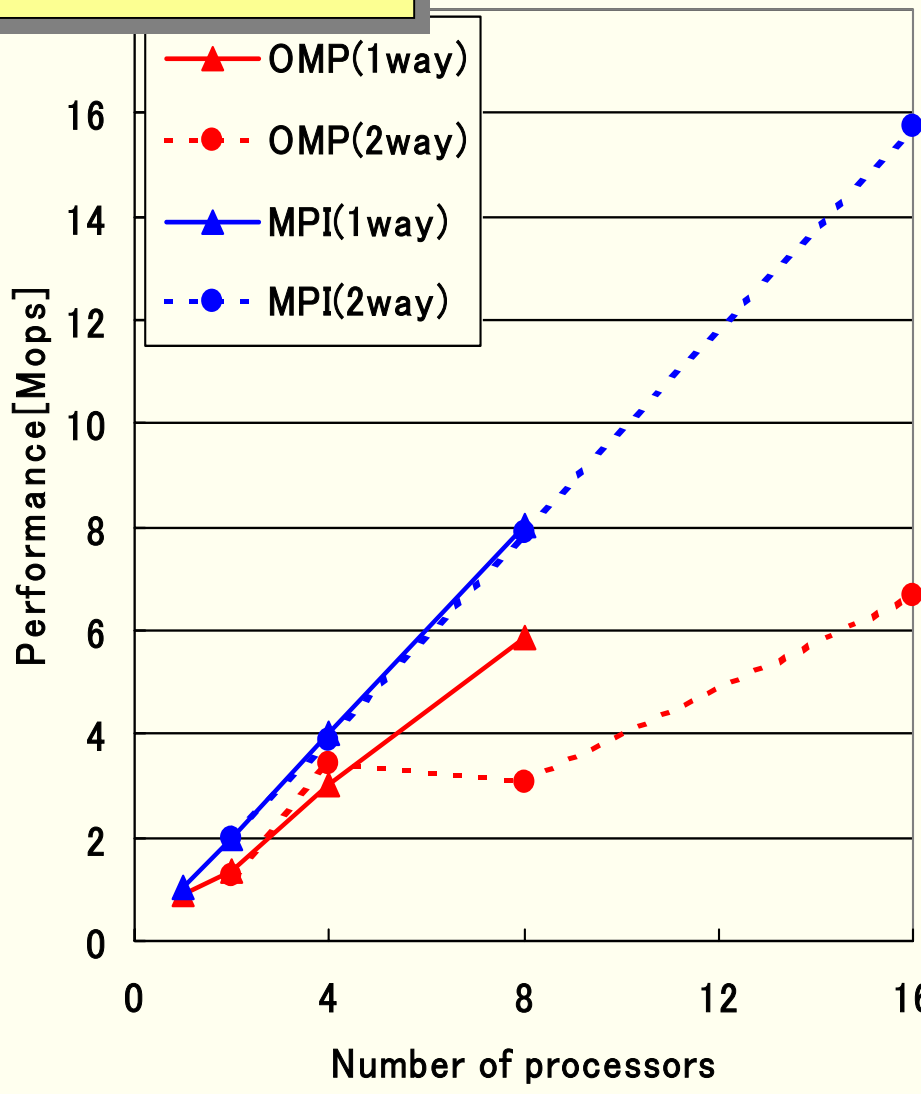
Improvement by ~10%



Myrinet : ~11.6 times

Ethernet : ~7.2 times

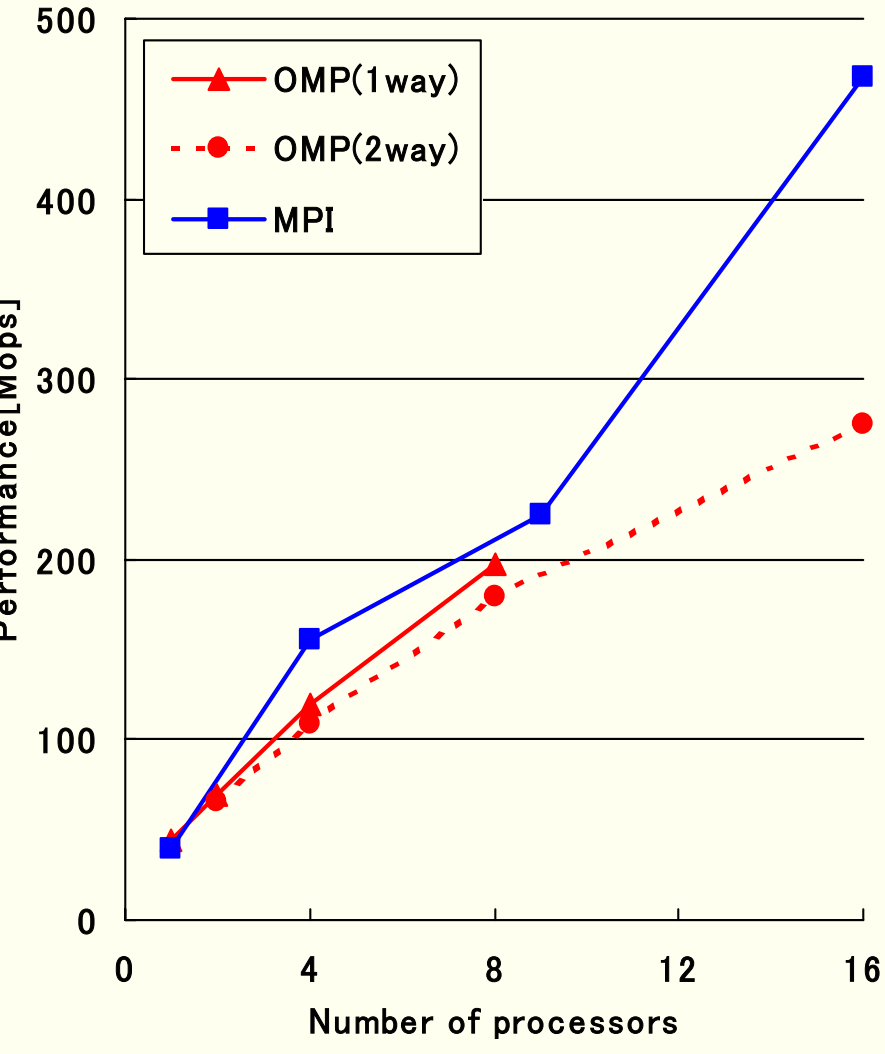


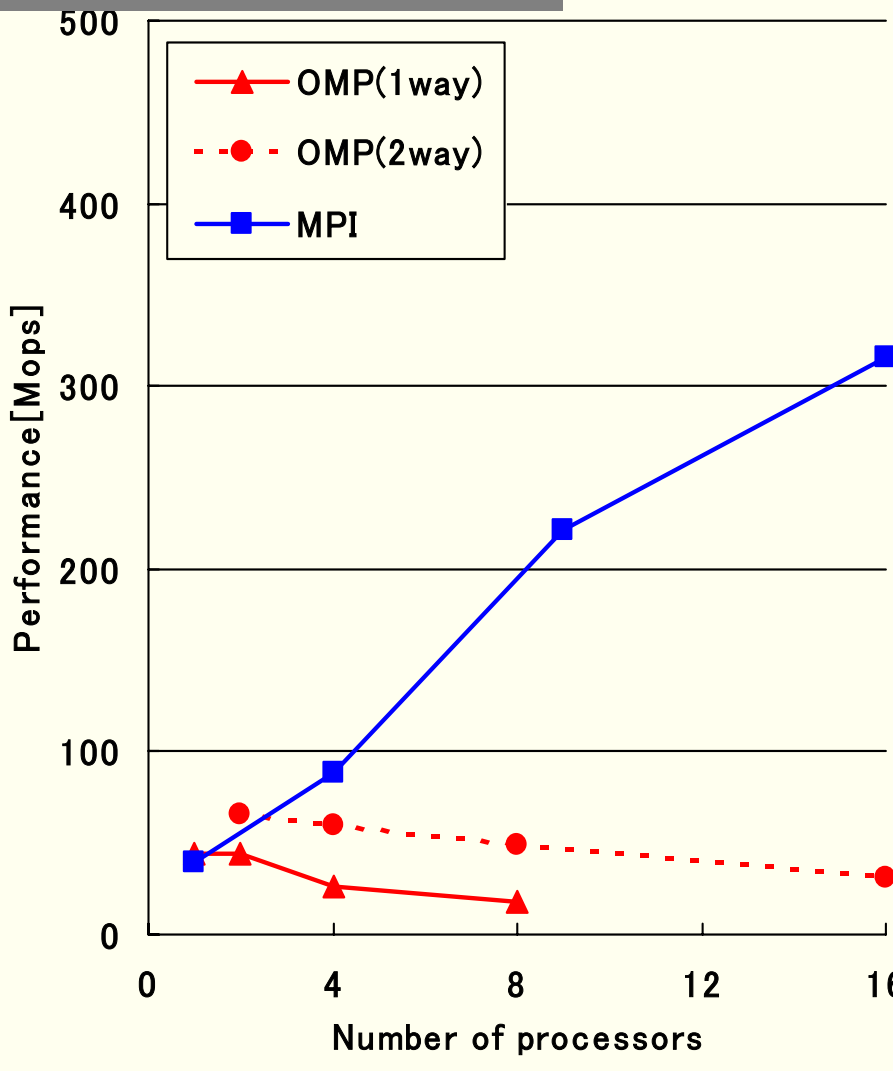


Myrinet : ~6.2 times

Ethernet : no speedup

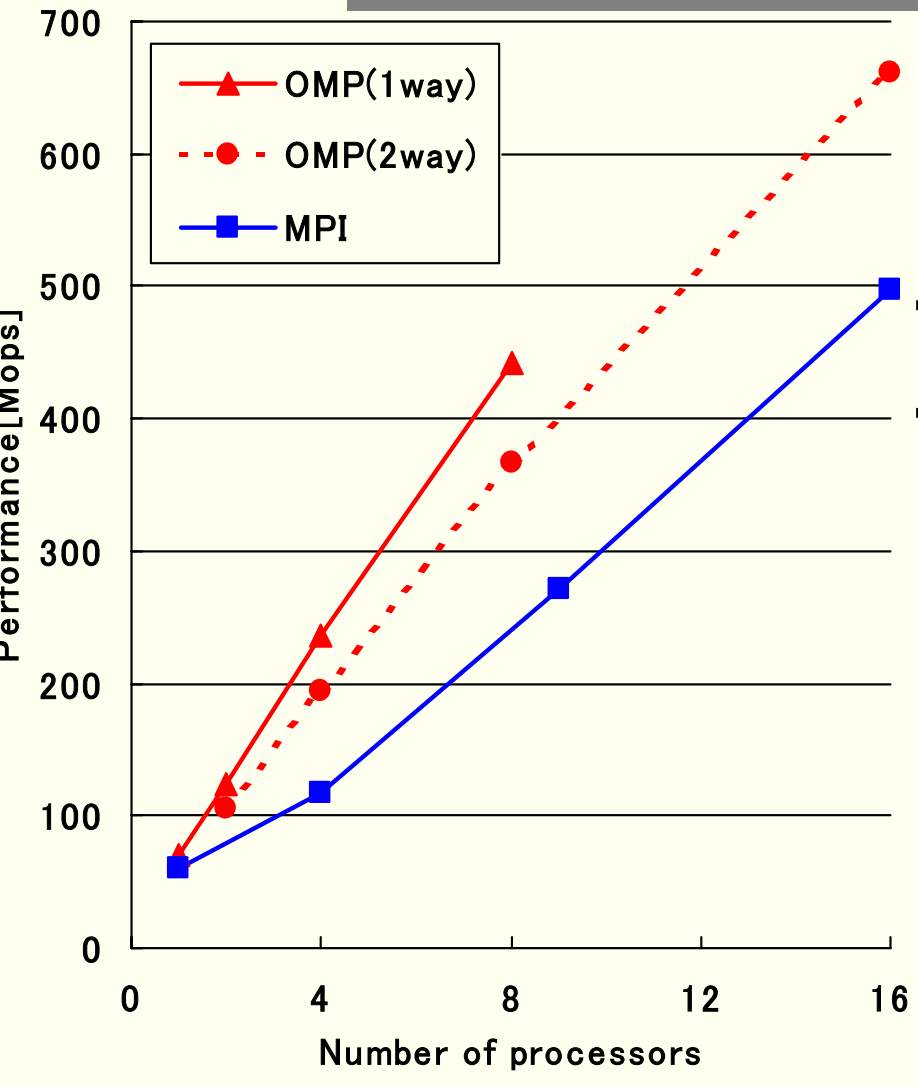
net

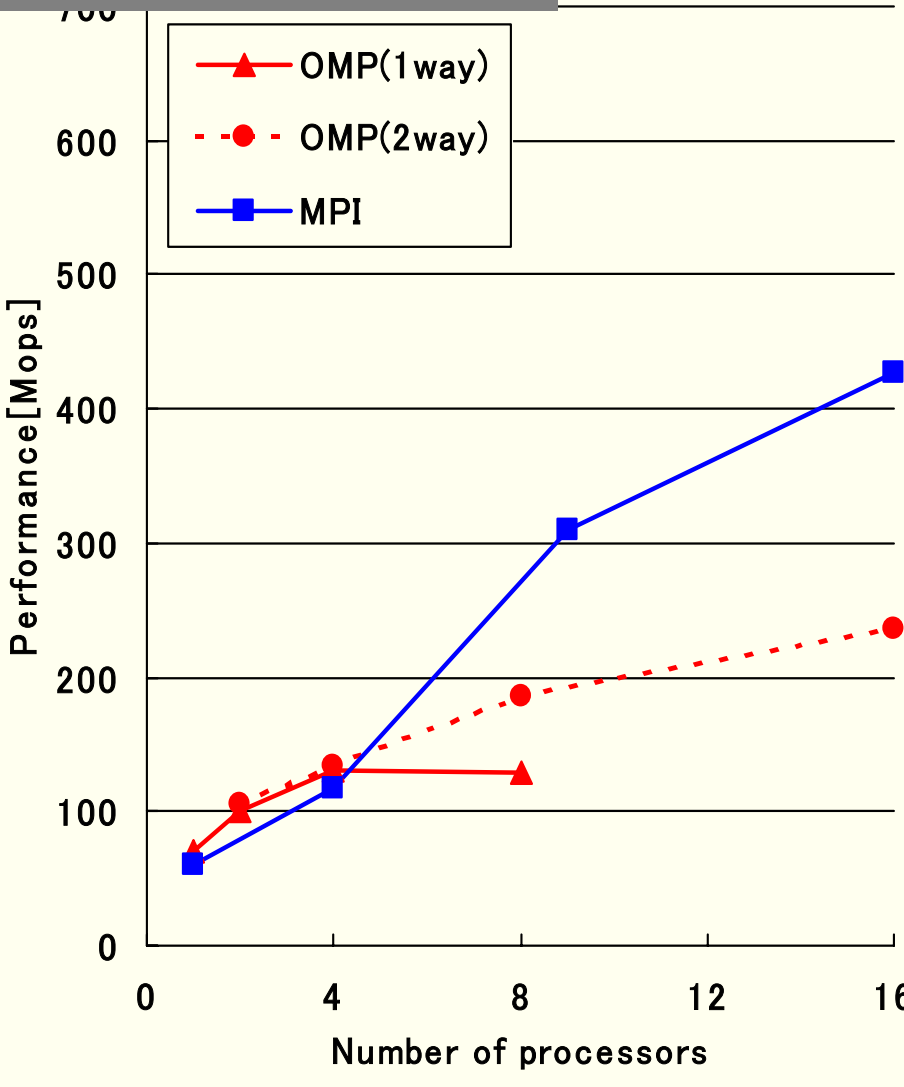




Myrinet : ~9.5 times

Ethernet : ~3.4 times





Summary

- We evaluated performance of Omni/SCASH
  - Good performance was achieved with Myrinet
  - With Ethernet, the overhead of barrier operation was very large
- For ethernet-based cluster:
  - Applications with small communication are suitable for Ethernet
  - When using Omni/SCASH, we have to carefully optimize communication in applications

Future work

- More detailed analysis on:
  - performance difference of Ethernet and Myrinet
  - overhead of barrier operation with Ethernet
- Re-design Omni/SCASH for Ethernet
- To improve locality, we are currently designing first touch page allocation facility
- Performance tuning tools to make performance tuning easier
  - Performance counters, profiler