

Performance Estimation of High Performance Computing Systems with Energy Efficient Ethernet Technology

Shinobu Miwa

Sho Aita

Hiroshi Nakamura

The University of Tokyo

{miwa, aita, nakamura}@hal.ipc.i.u-tokyo.ac.jp



Executive Summary

- ▶ Energy Efficient Ethernet (EEE) is a technique to lower power consumption of networks
 - ▶ Pros: significant power saving of network links
 - ▶ Cons: slight performance penalty caused by link-on/off
- ▶ To support system developers, we propose a perf. estimation method of HPC systems with EEE
 - ▶ Using novel performance models with network profiles
- ▶ The experimental results show that our method has significant accuracy in the most cases
 - ▶ 2.63% on average and 20.0% in worst case



Agenda

- ▶ Introduction
- ▶ Energy Efficient Ethernet (EEE) technology
- ▶ Performance estimation of HPC systems with EEE
- ▶ Experimental result
- ▶ Summary & future work



Power of Interconnection Networks

- ▶ Power consumption of interconnection networks is not negligible in modern HPC systems
 - ▶ It may achieve up to 33% of total system power*
 - ▶ The reason is that interconnection networks have widened bandwidth and increased redundancy
 - ▶ Ex.) Tofu network has ten links per node, each with 6.25GB/s
- ▶ PHYs (physical layer devices) are dominant modules in networks in terms of power consumption
 - ▶ Around 70% of network device power
 - ▶ Always activated to maintain link connection

* P. M. Kogge, Architectural Challenges at the Exascale Frontier, Simulating the Future: Using One Million Cores and Beyond (invited talk), 2008



Energy Efficient Ethernet (EEE)

- ▶ Ethernet standard for saving power of PHYs
 - ▶ Standardized as IEEE802.3az in 2010
 - ▶ Change into a low power mode during low network loads
 - ▶ Save PHYs' power by up to 70%*

▶ Devices comp

ase gradually



[PowerConnect 5548 (Dell)]



[GEU-0820 (Le



[8-port Gigabit GREENnet (Trend



[Catalyst 3560CG (Cisco)]

* <http://www.broadcom.com/press/release.php?id=s430231>

Situation of EEE for HPC Use

- ▶ Few studies about EEE for HPC use have been done
 - ▶ There only exists a study of power evaluation of EEE-supported devices for a ping-pong test*
 - ▶ Power and performance of EEE-supported devices for HPC applications are still unknown

- ▶ Why?
 - ▶ No hardware for HPC systems
 - ▶ Quite new technology

* P. Reviriego et al., An Energy Consumption Model for Energy Efficient Ethernet Switches, HPCS, 2012

Requirements for the Spread of EEE in HPC

- ▶ Development of EEE-supported devices for HPC systems
 - ▶ Each task force of a high-performance network (e.g. InfiniBand) should standardize EEE-technology rapidly
 - ▶ EEE-supported devices should be developed immediately
- ▶ Power/performance estimation when using EEE
 - ▶ Although there does not exist EEE-supported HPC systems yet, we want to know the impact of EEE on existing systems
 - ▶ If it is small, it would motivate system developers to use EEE
- ▶ Establishment of power management scheme
 - ▶ Optimal power management scheme may be different between Internet and interconnection networks



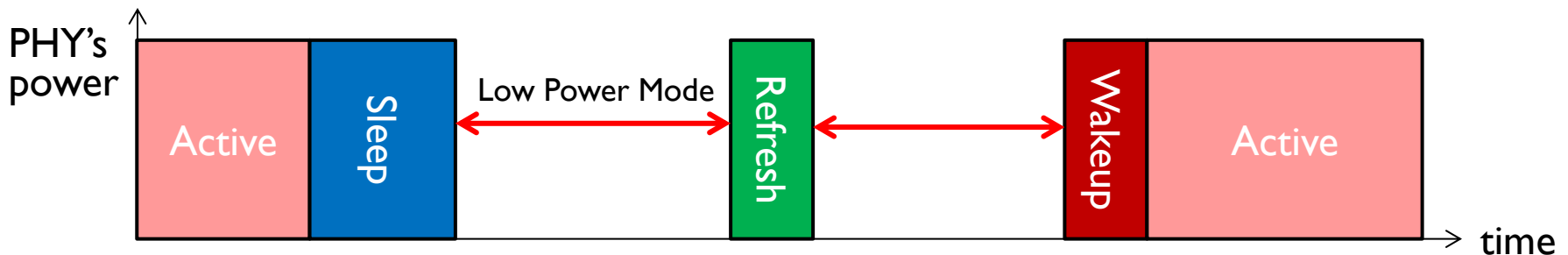
Mission of This Work

- ▶ Our goal
 - ▶ To develop a performance estimation method of EEE-supported HPC systems
 - ▶ Power model of EEE already exists, but performance one does not
 - ▶ We can start the discussion about power management schemes without EEE-supported hardware
- ▶ Prerequisite for estimation
 - ▶ We do not have any EEE-supported devices for HPC
- ▶ Our approach
 - ▶ Using performance models with network profiles



EEE

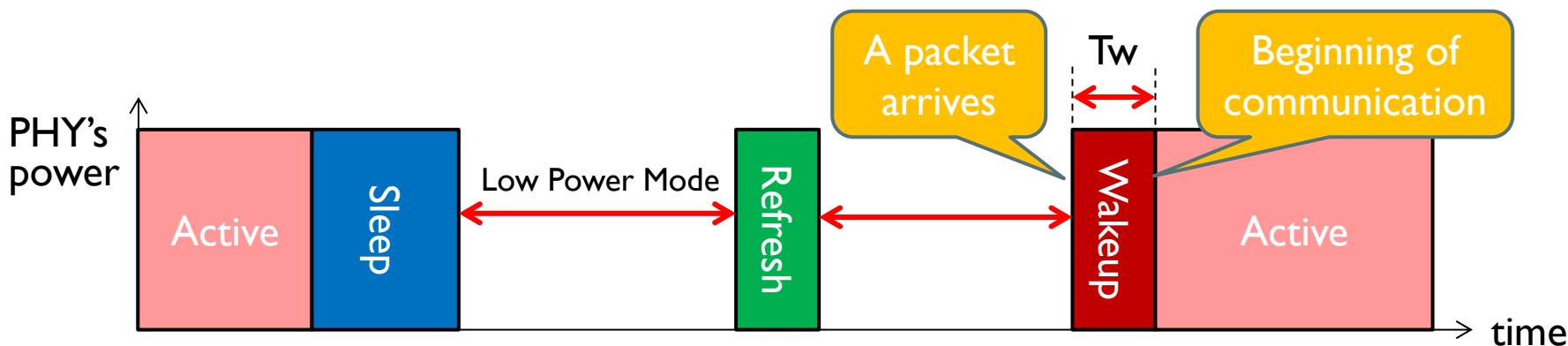
- ▶ Technique to lower power consumption of PHYs during low network loads
 - ▶ Start to power a PHY off when detecting an idle state
 - ▶ Periodically get up for confirmation of link connectivity
 - ▶ Start to power the PHY on when a packet arrives



- ▶ Although the detailed power management of EEE is not published, the most devices seem to use time-out control and on-demand wake-up

Performance Penalty of EEE

- ▶ Packets arrived during a low power mode are delayed



- ▶ The wake-up delay is at least 16 microseconds in 1000BASE-T networks

We must model this penalty!

Protocol	Min Tw (usec)
100BASE-TX	30
1000BASE-T	16
10GBASE-T	4.48

Proposed Performance Model

- ▶ Suppose that an application i runs with j threads on an EEE-supported HPC system
- ▶ Elapsed time T^{ij} can be described below

$$T^{ij} = T_{base}^{ij} + T_{overhead}^{ij}$$

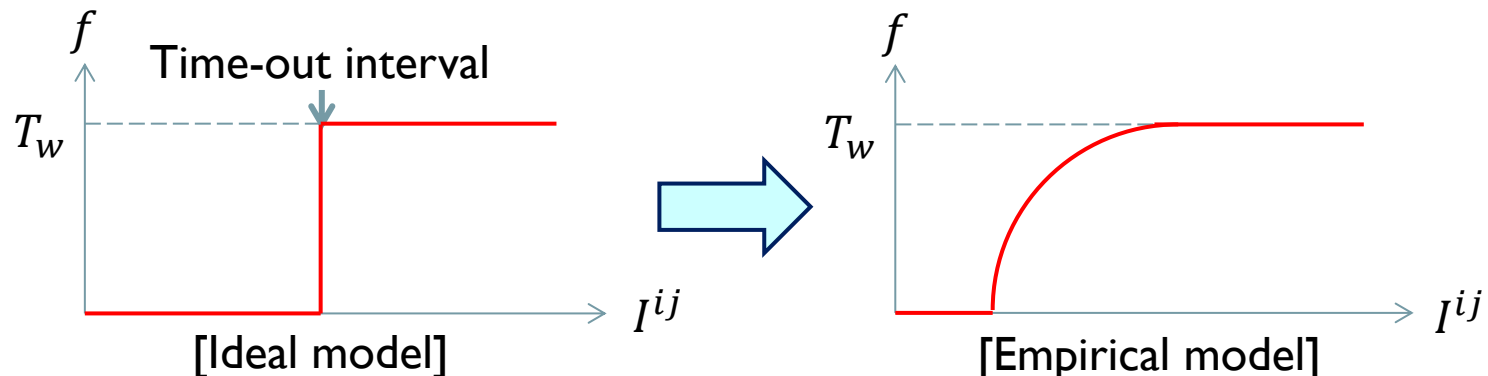
- ▶ T_{base}^{ij} : Elapsed time when the application i runs with j threads on an EEE-unsupported system
- ▶ $T_{overhead}^{ij}$: Time overhead caused by EEE

Model of $T_{overhead}^{ij}$

- ▶ We assume that $T_{overhead}^{ij}$ is written as follows

$$T_{overhead}^{ij} = n^{ij} \times f(I^{ij})$$

- ▶ n^{ij} : Communication count per node
- ▶ I^{ij} : Average idle interval of network links
- ▶ f : Performance penalty per communication
 - ▶ The function f forms a step function Ideally, but performance penalty actually shows gradual increase because of I^{ij} variation

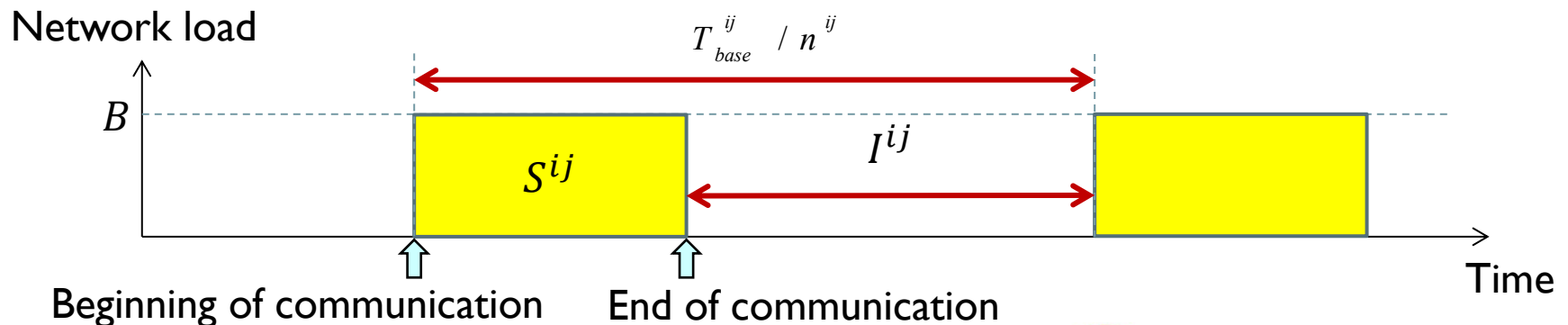


Model of I^{ij}

- ▶ We suppose that all communication occurs periodically and transmits the same size of data
- ▶ Under the above assumption, I^{ij} can be written below

$$I^{ij} = (T_{base}^{ij} / n^{ij}) - (S^{ij} / B)$$

- ▶ S^{ij} : Average communication data size per node
- ▶ B : Network bandwidth per node







List of Proposed Models

▶ $T^{ij} = T_{base}^{ij} + T_{overhead}^{ij}$

▶ $T_{overhead}^{ij} = n^{ij} \times f(I^{ij})$

▶ $I^{ij} = (T_{base}^{ij} / n^{ij}) - (S^{ij} / B)$

- ▶ T_{base}^{ij} : Elapsed time on EEE-unsupported systems  By measurement
 - ▶ n^{ij} : Communication count per node
 - ▶ I^{ij} : Average idle interval of network links
 - ▶ S^{ij} : Average communication data size per node
 - ▶ f : Performance penalty per communication  By assumed power management scheme
 - ▶ B : Network bandwidth per node  Already known
-  By network profiles

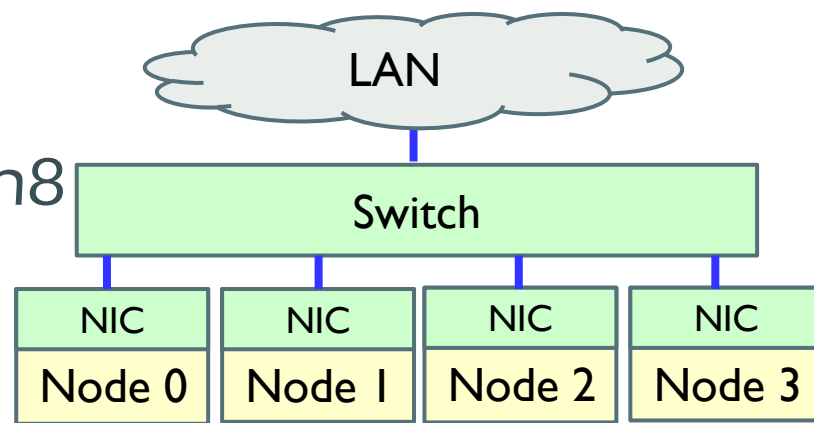


Evaluation Methodology

- ▶ Evaluation item
 - ▶ Accuracy of the proposed models
- ▶ Evaluation method
 - ▶ Estimate the performance under the following situation
 - ▶ EEE-disabled system \Rightarrow EEE-unsupported HPC system
 - ▶ EEE-enabled system \Rightarrow future EEE-supported HPC system
- ▶ Benchmark programs
 - ▶ Synthetic application
 - ▶ HPC applications (NAS Parallel Benchmark)

System Configuration

- ▶ Switch: Dell PowerConnect 5548
 - ▶ 48-port Gigabit Ethernet
 - ▶ Compliant with EEE
 - ▶ Time-out interval: 1 msec
- ▶ Node: HP ProLiant DL360p Gen8
 - ▶ 4 nodes
 - ▶ CPU: Xeon E5-2680, 2-socket
 - ▶ 8C16T, 2.7GHz, 130W TDP
 - ▶ Memory: 64GB (8GB x8)
 - ▶ NIC: HP FlexibleLOM 1Gb 4-port 331FLR Ethernet adapter
 - ▶ Compliant with EEE
 - ▶ Disable Turbo Boost and cpuspeed



Evaluation with Synthetic Application

- ▶ Synthetic application that all processes repeat concurrent communication
 - ▶ Repeat all-to-all 100,000 times for given array
 - ▶ Insert *usleep* function to adjust communication intervals
- ▶ Parameters used for experiment
 - ▶ # of Rank: 4 (1 rank/node),
16 (4 rank/node)
 - ▶ Array size: 256-131,072 Byte
 - ▶ Sleep time: 0, 100, 500, 1,000 usec

```
#include <mpi.h>
#define ITERATION 100000

int main(int argc, char** argv) {
    /* initialization */
    for (i = 0 ; i < ITERATION ; ++i) {
        MPI_Alltoall(&data, numdata, MPI_INT, &out,
                    numdata, MPI_INT, MPI_COMM_WORLD);
        usleep(stime);
    }
    /* finalization */
}
```

[Pseudo code of synthetic application]

- ▶ Execute 5 times in each parameter and then average the results

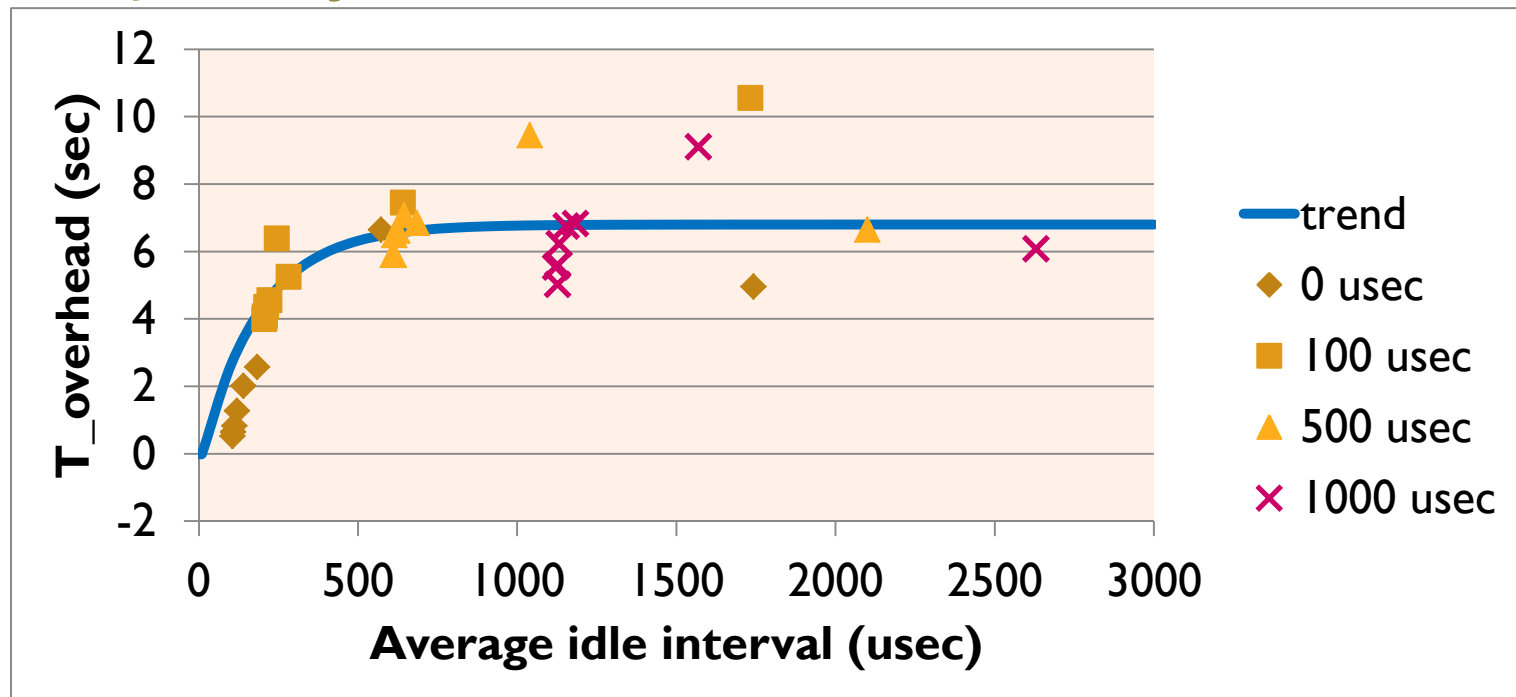
Evaluation with NAS Parallel Benchmark

- ▶ Version: 3.3.1
- ▶ Compile options: -O2 -funroll-loops
- ▶ Parameters used for experiment
 - ▶ # of Rank: 4 (1 rank/node), 16 (4 rank/node)
 - ▶ Class: A, B, C

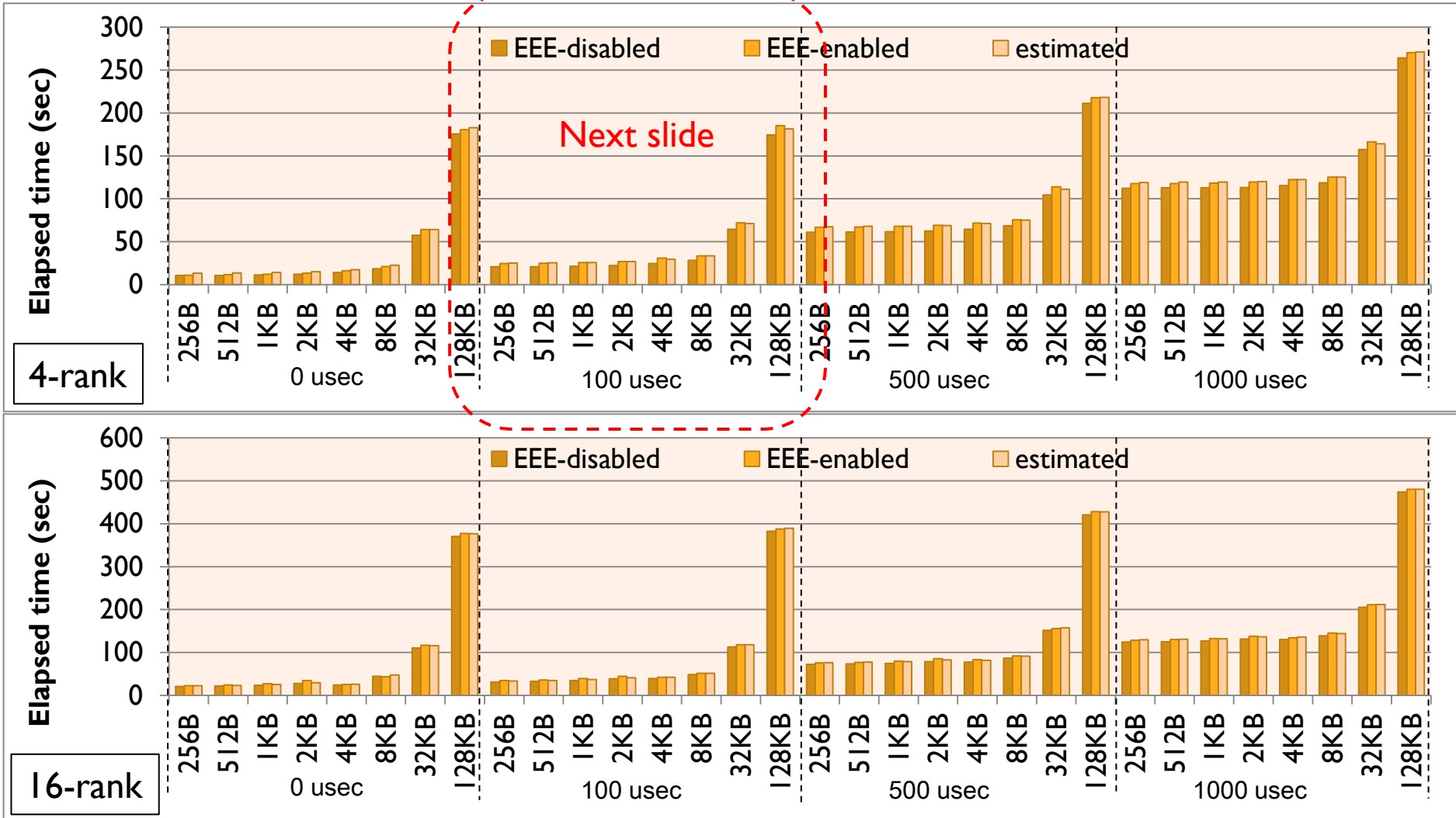
(However, we will only show the result of 16-rank Class-B)
- ▶ Execute 10 times in each parameter and average the results

Evaluation Result of $T_{overhead}^{ij}$ (Synthetic, 4-rank)

- ▶ We can model many cases correctly
- ▶ There exists a few points that show large errors
 - ▶ This is because the firmware changes CPU frequency unexpectedly

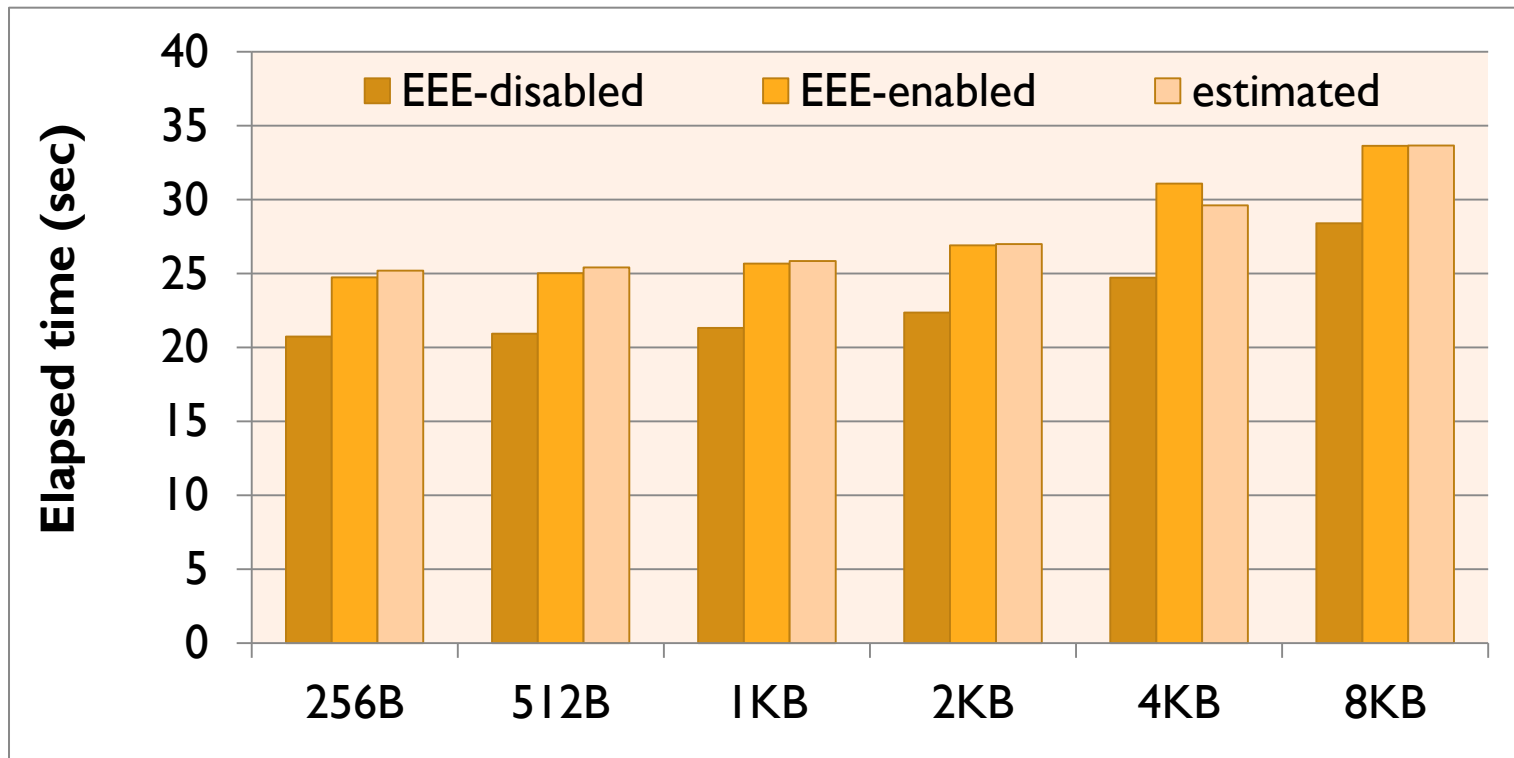


Accuracy of Performance Estimation (Synthetic)



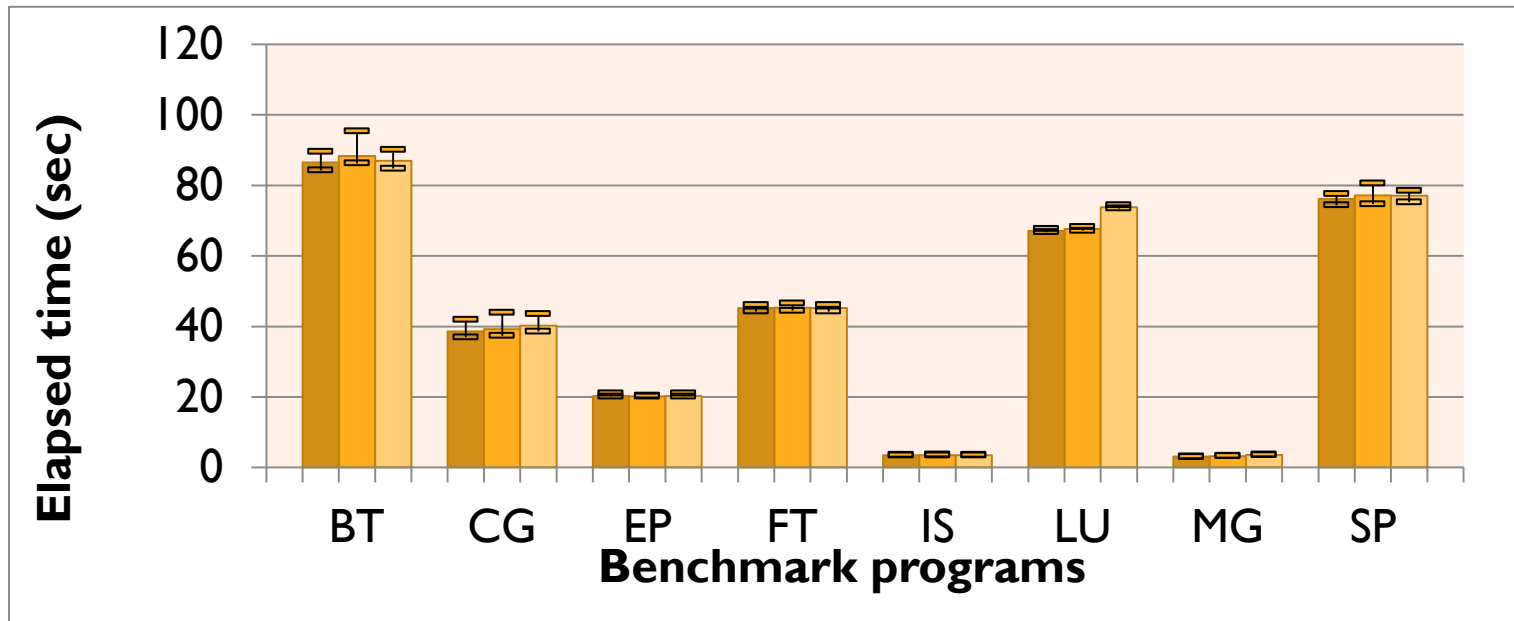
Accuracy of Performance Estimation (Synthetic, 4-rank, 100 usec sleep)

- ▶ Performance degradation by EEE: up to 25.8% (4KB)
- ▶ Estimation error: 2.63% (on average)
20.0% (in the worst case)



Accuracy of Performance Estimation (NPB, 16-rank, class B)

- ▶ Since the most applications have a little communication, EEE hardly degrades the performance
- ▶ Only LU (which communicates frequently) shows a large error because of inaccuracy of model of average idle interval



Summary and Future Work

▶ Summary

- ▶ Summarize requirements for the spread of EEE in HPC
- ▶ Propose a novel performance estimation method for EEE-enabled HPC systems
- ▶ The most cases show good accuracy but some cases do not
 - ▶ Accurate profile-based estimation is hard because of many impractical assumption

▶ Future work

- ▶ Develop trace-based estimation
- ▶ Evaluate other situation (other applications and topologies)

Any Questions?

