A Methodology for Coping with Heterogeneity of Modern Accelerators on a Massive Supercomputing Scale

Toshio Endo and Satoshi Matsuoka

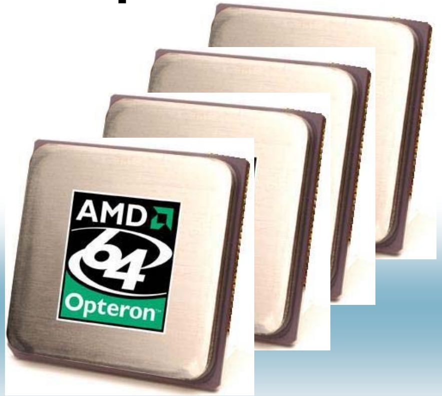
Tokyo Institute of Technology, Japan

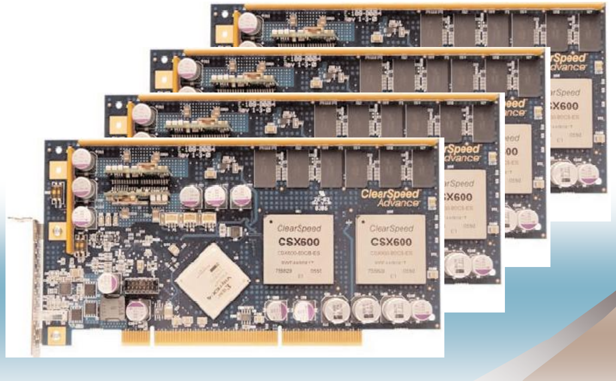
TOKYO TECH

Pursuing Excellence

Overview

- Combined use of >10,000 Opteron cores and >600 ClearSpeed SIMD accelerators for a tightly-coupled program (Linpack)
- 56.43TFlops: The world's highest Linpack performance on heterogeneous supercomputers





Heterogeneous Supercomputers

Heterogeneous architectures are getting popular for

- Generality by general purpose CPUs
- Higher performance / power ratio by accelerators
  - SIMD accelerators, GPGPUs, CellBE…

Questions:

- Are they scalable up to supercomputing scale?
- How can we utilize or program different resources effectively?

NEC/Sun/ClearSpeed/Voltaire Tokyo Tech TSUBAME Cluster

Voltaire ISR9288
Infiniband 10Gbps

SunFire X4600
16 Opteron core/node
x 655nodes

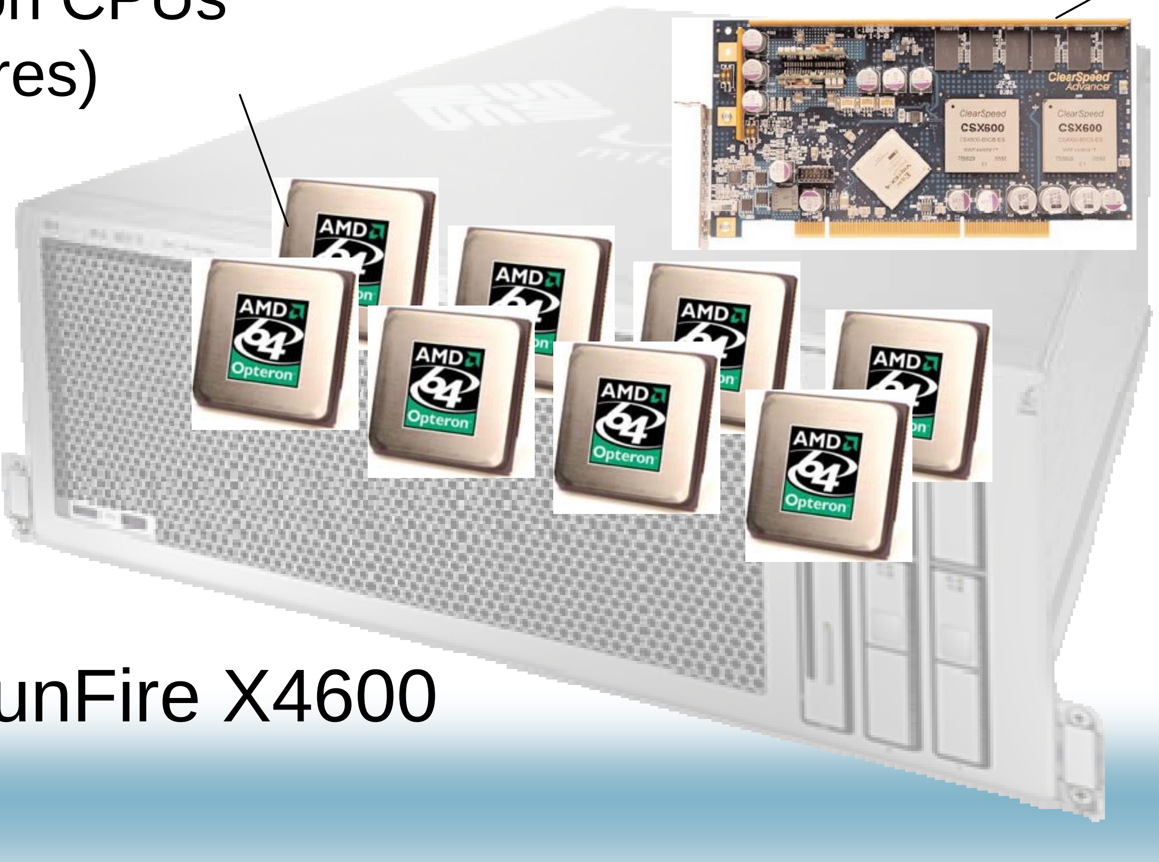
ClearSpeed CSX600
SIMD accelerator
x ~~360~~ PCI-X boards
648

102TFlops peak
= Opteron 49.8TF +
ClearSpeed 52.2TF

Structure of TSUBAME Node

8 dual-core
Opteron CPUs
(16 cores)

ClearSpeed
Accelerator

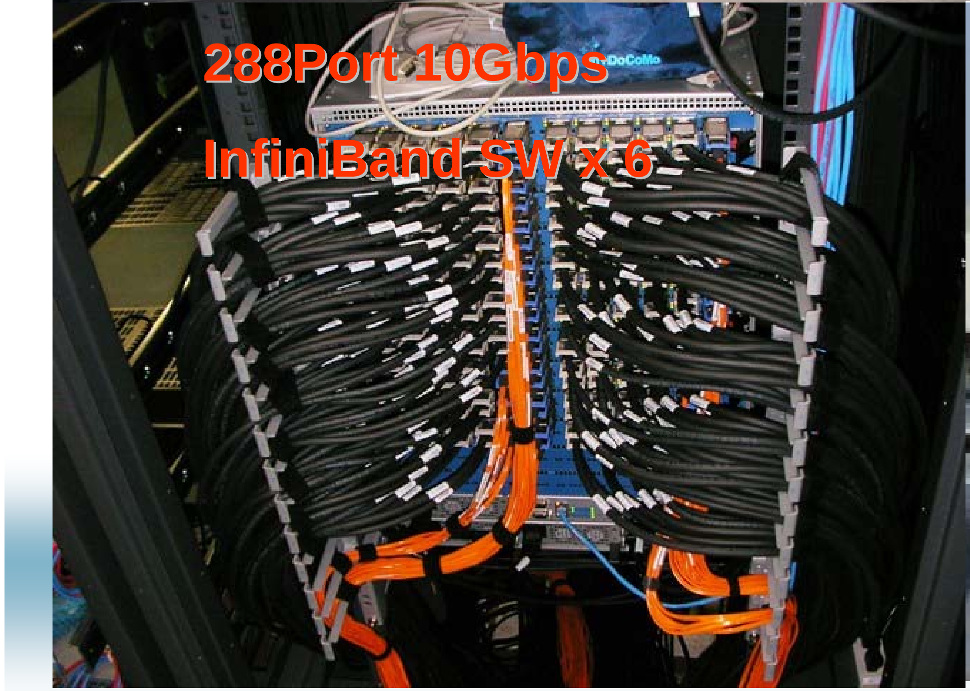


SunFire X4600

16 Opteron cores x

655 Compute nodes

1.6PByte storage



288Port 10Gbps

InfiniBand SW x 6

Cooling Towers (~20 units)

ClearSpeed Accelerator

- PCI-X accelerator boards
  - CSX600 SIMD processor x 2 + 1GB DRAM on board
  - 210MHz x 2FP x 96SIMD x 2 = 80.6GFlops peak
    - Configurable up to 250MHz
  - Power: 25W/board

Provided software:

- C^n programming language
- CSXL BLAS library <= Used by this work
- CSFFT library

Linpack: Our Target Application Benchmark

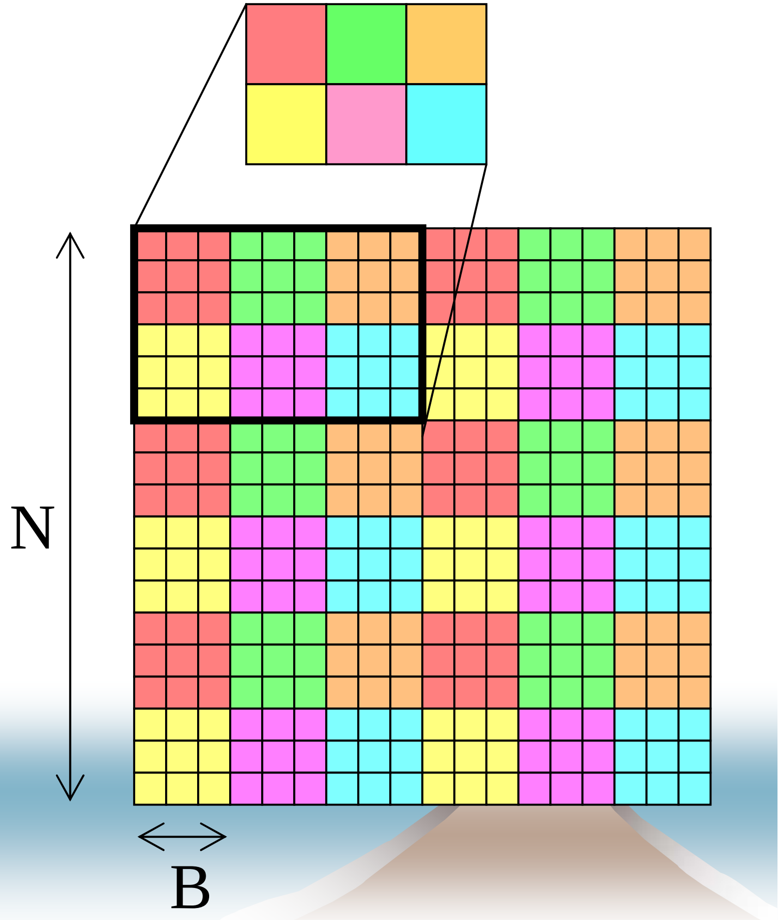
- Linpack is a numerical benchmark used in Top500
  - Solve N x N dense linear equations
- HPL (High-performance Linpack) by A. Petitet
  - A well-known MPI parallel implementation
  - Matrix multiply (DGEMM) computation is dominant

Data Decomposition in HPL

- Matrix is uniformly distributed with 2D Block-Cyclic distribution

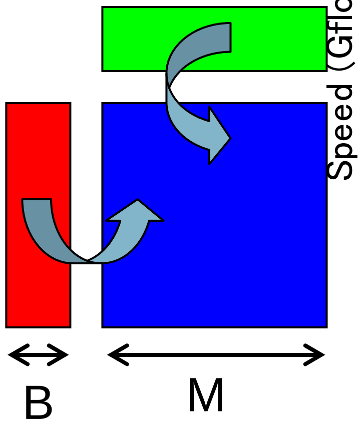
- Since HPL is designed for uniform systems, how can we run it efficiently on heterogeneous systems?

Matrix distribution on 6 (=2x3) processes

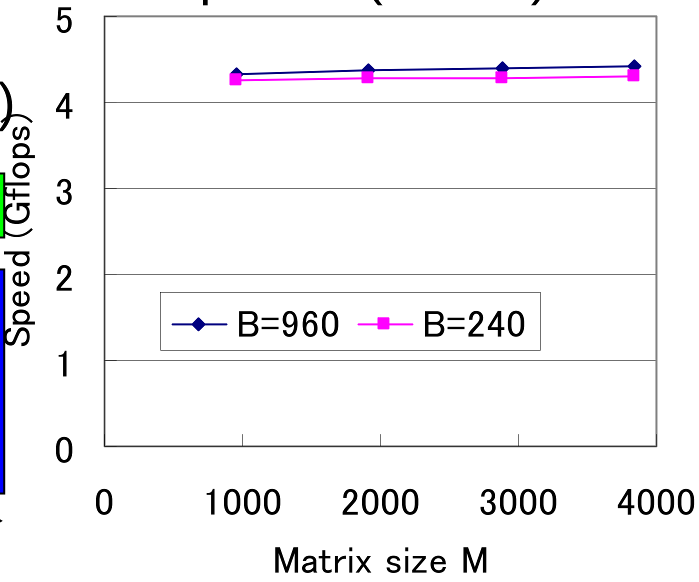


DGEMM Performance of Opteron and ClearSpeed

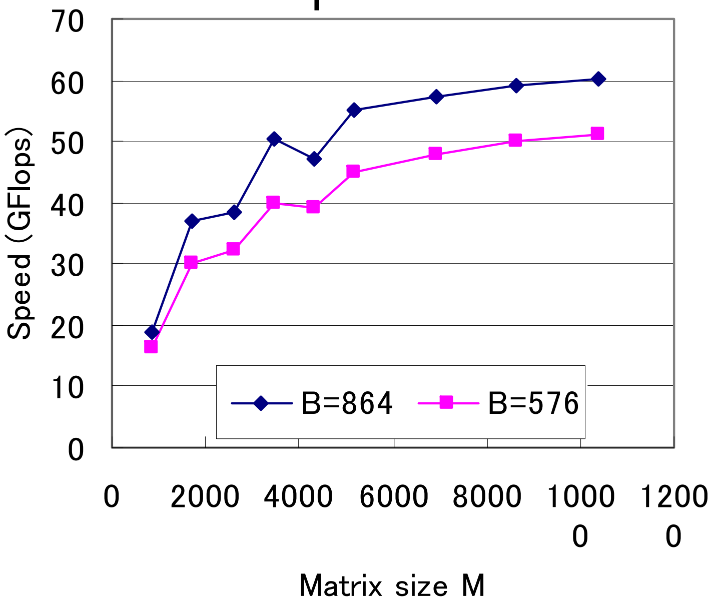
Multiply of
(MxB) x (BxM)



GOTO BLAS
on Opteron (1 core)



CSXL BLAS beta 2.50
on ClearSpeed



- An accelerator is equivalent to 14 CPU cores at peak
- ClearSpeed Performance is much more sensitive to matrix size!

- GOTO BLAS is by Kazushige Goto, U. Texas

Issues on Heterogeneous TSUBAME

- Intra-node heterogeneity: A node has both general purpose CPUs and SIMD accelerators
- Inter-node heterogeneity: About half the nodes have accelerators, while others not --- until October 2007
- It is desirable to keep modification to HPL small
  - HPL is designed for uniform systems!

Our Basic Policy (1/2)

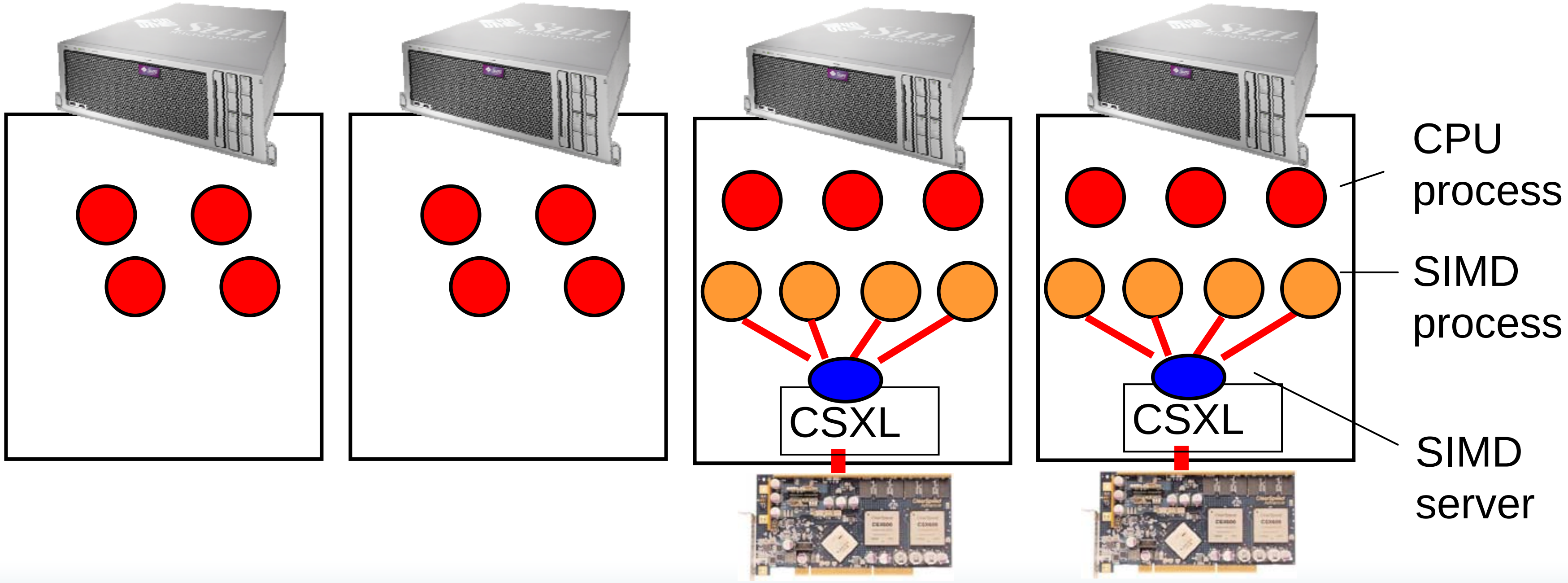
- For intra-node heterogeneity, we ‘virtualize’ heterogeneous processors at the BLAS layer
- For inter-node heterogeneity, we control the number of processes among nodes
  - cf. CHARM++, AMPI from UIUC

- We can keep kernel workload of each process uniform (good for HPL ☺), while maintaining heterogeneity

Our Basic Policy (2/2)

Two types of HPL processes are introduced:

- CPU processes use GOTO BLAS’s DGEMM
- SIMD processes throw DGEMM requests to accelerator



Additional SIMD server is introduced as multiplexer

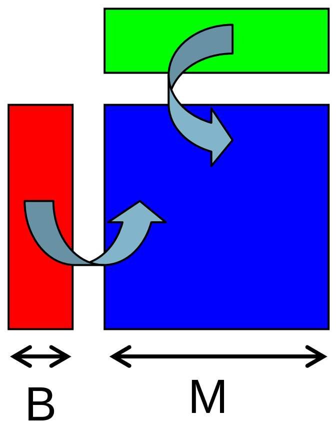
Careful Tuning is Necessary for Performance

Since SIMD accelerators are sensitive to many HPL parameters, careful tuning is necessary

- Step 1: Granularity tuning
  - Block size
  - Process granularity
- Step 2: Load balancing
  - Mapping between processes and nodes
  - Mapping between processes and processors

Tuning of Block Size

- When block size B is small, CSXL performance is heavily degraded
- When B is too large, HPL suffers from large overhead for panel factorization



- We set B=864 (sometimes 1152)

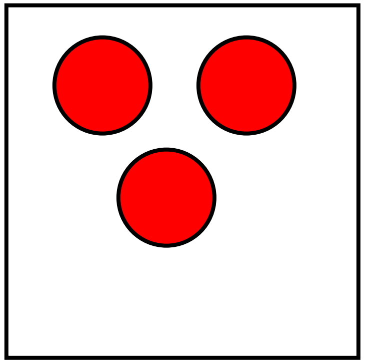
Tuning of Process Granularity

We can tune 'process granularity' as GOTO BLAS is multi-threaded

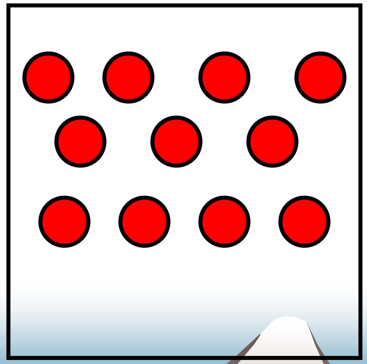
- If processes are coarse (a process uses many threads), it is more difficult to balance among nodes
- If too fine, HPL suffers from duplicated computation

- A CPU process = 4 threads

Coarse



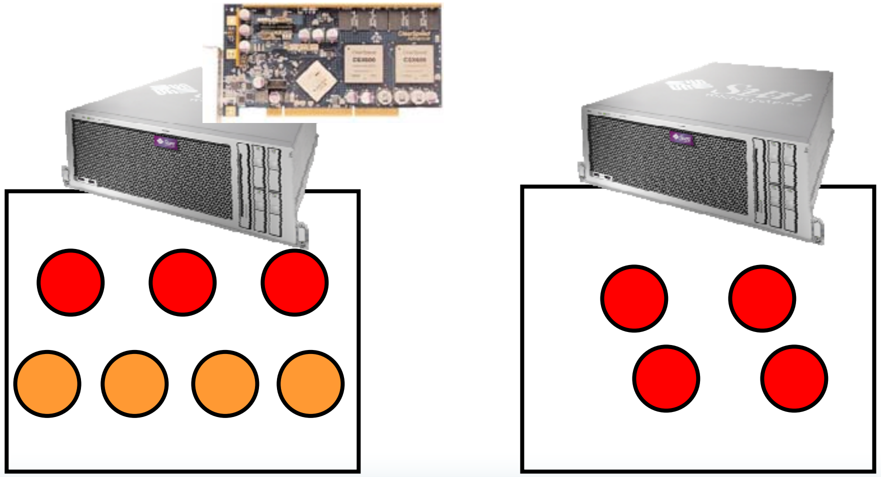
Fine



Mapping between Processes and Nodes

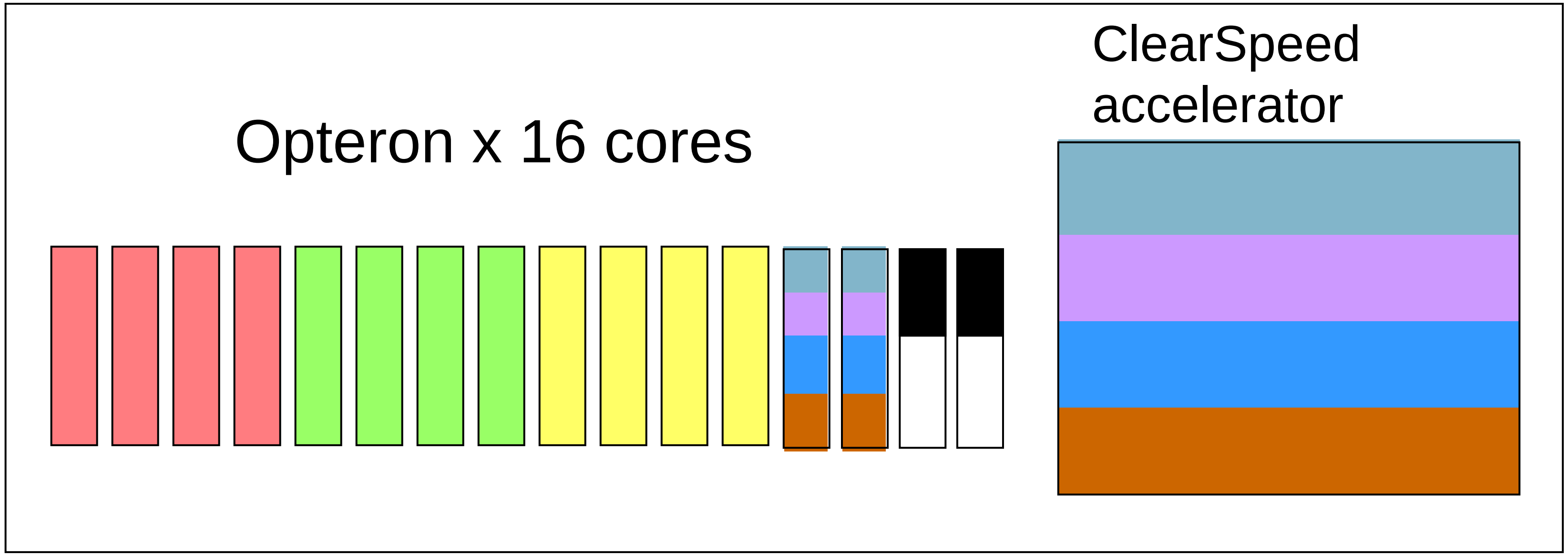
- Peak DGEMM performance per node is
  - ~ 120 GFlops with accelerator
  - ~ 70 GFlops w/o accelerator

Roughly 7:4



Mapping between Processes and Processors

Processors are divided into seven processes

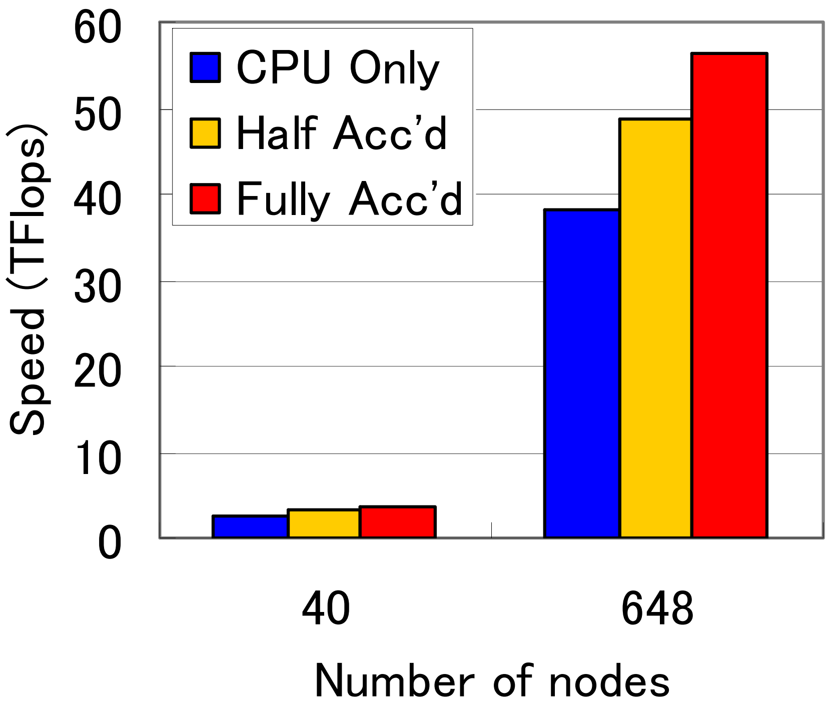


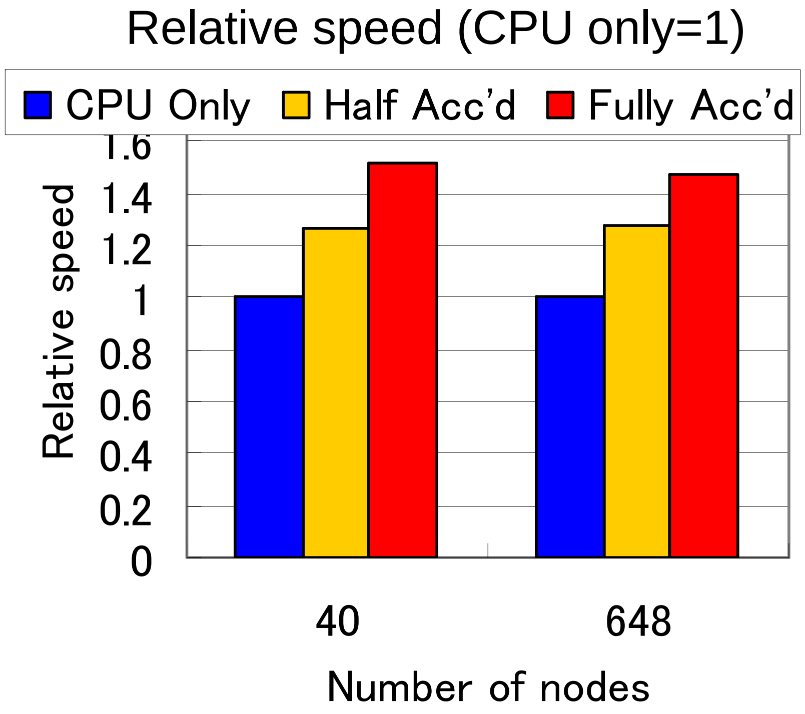
- We need consider CPU usage for communication with accelerator (black region)
- Remaining idle CPU cores are used to help accelerator

Experimentation

- 648 SunFire X4600 nodes in TSUBAME
- Modified HPL + Voltaire MPI + GOTO BLAS + CSXL BLAS
- Three measurements:
    - Full Acc: ClearSpeed boards on all nodes are used
    - Half Acc: # of ClearSpeed boards is the half of nodes (Inter-node Heterogeneous case)
    - CPU Only: Only Opteron CPUs are used

Experimental Results




Relative speed (CPU only=1)

- 48.88TF when 55% nodes are accelerated
  - +28% over CPU Only (38.18TF)
- 56.43TF when all nodes are accelerated
  - +48% over CPU Only

Summary

- Scalability of heterogeneous supercomputers with SIMD accelerators is proved
- 56.43TFlops Linpack performance is achieved
- Our method works efficiently even when nodes are partially accelerated

Future work:

- From hand-tuning to automatic tuning
- Other useful applications!