# An Approach to the Highest Efficiency of the HPCG Benchmark on the SX-ACE Supercomputer

Kazuhiko Komatsu[1)4)], Ryusuke Egawa[1)4)], Yoko Isobe[1)3)], Ryusei Ogata[3)], Hiroyuki Takizawa[2)4)], Hiroaki Kobayashi[1)]

1) Cyberscience Center, Tohoku University 2) GSIS, Tohoku University 3) NEC Corporation 4) JST CREST

## Introduction

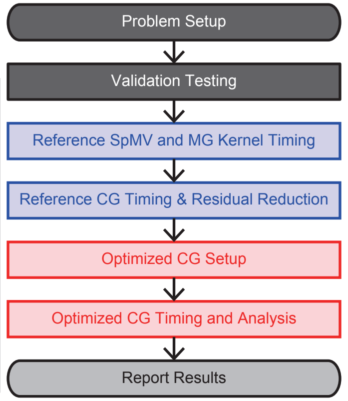
HPCG (High Performance Conjugate Gradient) has been developed to narrow the large performance gap between real applications and the HPL benchmark. The major features of HPCG are;

- **Including major communication and computational patterns of real applications.**
- **Easy to understand, optimize, and run.**
- **Able to examine memory and network performances.**
  ex) indirect memory accesses, collective ops, p2p messages

HPCG solves a linear system of a sparse matrix.

$$Ax=b$$

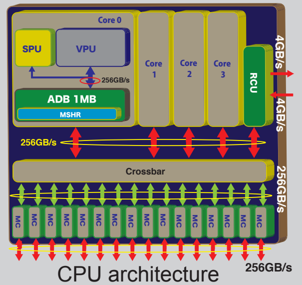
$A$ is a large sparse matrix discretized by the finite element method.
The linear system is solved by multigrid preconditioned conjugate gradient with the symmetric Gauss-Seidel smoother.



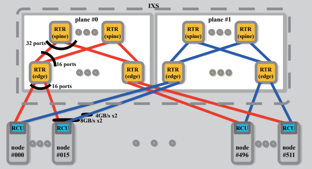

## Overview of the SX-ACE Supercomputer

The SX-ACE supercomputer consists of 512 nodes. Each node is equipped with an SX-ACE processor, which can provide a high memory bandwidth for practical HPC applications.

- **High vector computational performance by a 4-core vector processor**
- **High sustained memory bandwidth by a strong memory subsystem**
- **ADB(Assignable Data Buffer) to keep a high sustained memory bandwidth**
- **Scalable multinode system by a two-stage fat-tree custom network**




CPU architecture



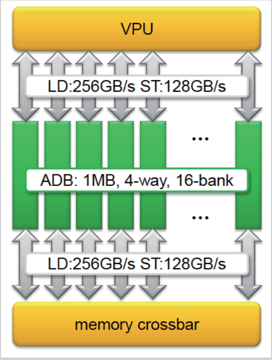
IXS network by two-stage fat-tree

**ADB of SX-ACE**

- Private on-chip memory
- 1 MB, 4-way, 16-bank
- 256 GB/s bandwidth
- Customized for fast random accesses

**Software controllable function**

- Compiler and user can specify the use of the ADB
- A bypass mechanism for memory instructions
- Avoiding cache pollution
- Enhancement of indirect memory access




## Preliminary Evaluation before Optimization

To decide the optimization plan, the reference HPCG is executed.

**Preliminary Evaluation Environments**

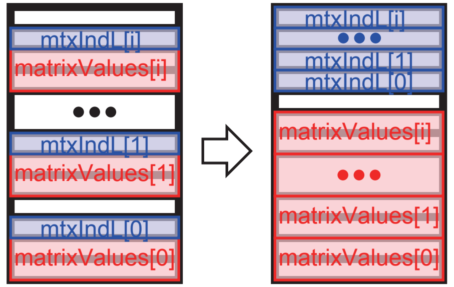
| | |
|---|---|
| Supercomputer | : SX-ACE |
| # of nodes | : 1 |
| Compiler | : NEC SX C/C++ Compiler Version 1.0 |
| HPCG version | : Release 2.4 |
| Problem size | : 104 x 104 x 104 (default) |
| Parallelization | : Flat-MPI |

<snip>
GFLOP/s Summary: HPCG-Benchmark-2.4.yaml
Raw DDOT: 28.2595
Raw WAXPBY: 17.8771
**Raw SpMV: 0.441609**
**Raw MG: 0.586906**
Raw Total: 0.577329
Total with convergence overhead: 0.577329
<snip>

These low performances are caused by quite low vectorization rates and inefficient memory accesses. **Optimizations for efficient vector calculations and memory accesses** are essential.

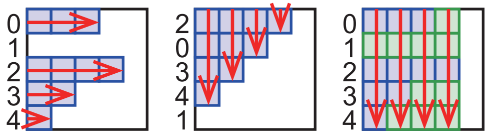
## Optimizations of HPCG for SX-ACE

**Data rearrangement for continuous memory accesses**




* Since an overhead of memory arrangement is too large for the SX-ACE processor, the row data rearrangement was not applied.

**Data packing for vector-friendly matrix memory allocation of sparse matrices**

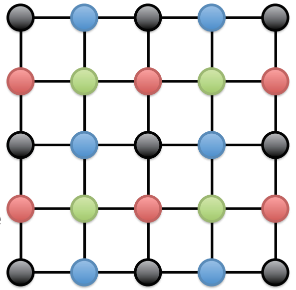


**CSR** Default matrix data storage format of HPCG2.4

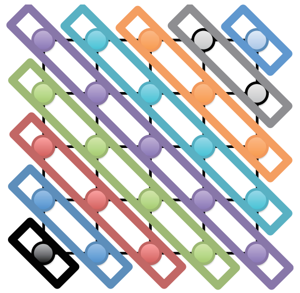
**JAD** SpMV function in a highly optimized numerical library

**ELL** Well known as suitable for the vector architecture

**Parallelization approaches to eliminate data dependencies**



Eight-coloring



Hyperplane

**Selective data store into the on-chip memory ADB**

- **Only highly-reusable data are selectively stored in ADB**

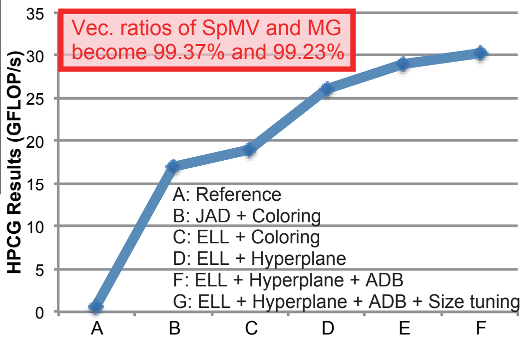
By exploiting programmer's knowledge about the reusability, data can selectively be stored in the on-chip memory ADB of the SX-ACE processor. All data that are considered reusable by the compiler are not always necessary.

**Problem size tuning for efficient use of ADB**

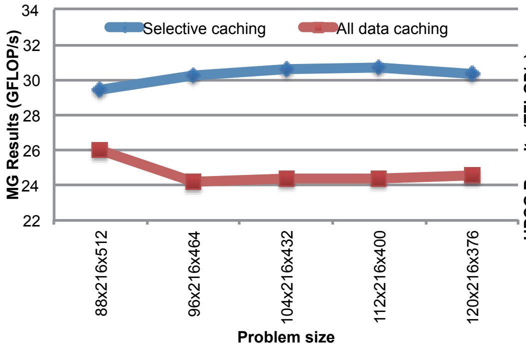
To avoid evicting highly-reusable data from ADB, **the problem size of HPCG has been tuned under consideration of both the capacity of ADB and the size of hyperplanes.** Especially, the size of each dimension becomes important for the hyperplane method.

## Performance Evaluation of HPCG on SX-ACE

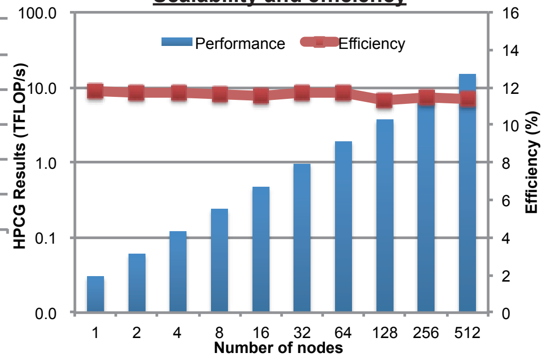
**Performance improvements of the optimizations**




**Effect of selective caching and size tuning**




**Scalability and efficiency**




## Conclusions

To exploit high potential of the SX-ACE supercomputer on the HPCG benchmark, this poster discusses the optimization techniques;

- Data rearrangement for efficient continuous memory accesses
- Various sparse matrix memory allocation
- the eight-coloring method and the hyperplane method for parallelization
- Selectively data stored in ADB, and the problem size tuning.

As a result, **the SX-ACE supercomputer successfully achieves the highest efficiency of 11.4% in the latest HPCG ranking.**