Optimizing Memory Layout of Hyperplane Ordering for Vector Supercomputer SX-Aurora TSUBASA

MCHPC’19: Workshop on Memory Centric High Performance Computing

Osamu Watanabe\textsuperscript{1,2},
Yuta Hougi\textsuperscript{1}
Kazuhiko Komatsu\textsuperscript{1}
Masayuki Sato\textsuperscript{1}
Akihiro Musa\textsuperscript{1,2}
Hiroaki Kobayashi\textsuperscript{1}

\textsuperscript{1}Tohoku University, \textsuperscript{2}NEC Corporation

18th November, 2019
CONTENTS

• INTRODUCTION

• NUMERICAL TURBINE CODE ON SX-Aurora TSUBASA

• OPTIMIZING MEMORY-ACCESS PATTERN OF 3D HYPERPLANE ORDERING METHOD

• PERFORMANCE EVALUATION

• CONCLUSIONS
INTRODUCTION
Various phenomena can be reproduced more realistically by large-scale simulations using supercomputer systems.

There are still many issues to be addressed, and the impact of problems with social infrastructures (ex. gas and steam turbines) on our society is immeasurable.
Demand for High Performance Turbine Simulation

Necessary to conduct a **numerical simulation** of a turbine using a **supercomputer** to simulate various phenomena occurring in the turbine in order to predict failures in advance.

- Complex wet-steam flow field analysis is required, but
- the cost for experiments is expensive.

Numerical Turbine code using NEC SX-Aurora TSUBASA is the best solution.
NUMERICAL TURBINE CODE ON SX-Aurora TSUBASA
Numerical Turbine
High Performance Turbine Design Code on SX Systems

Numerical Turbine developed by Prof. Yamamoto of Tohoku University
- is a simulation code realizing High-performance and High-reliable Future Turbines and
- has been accelerated on the SX series of Cyberscience Center at Tohoku University.

Only Numerical Turbine has achieved the following simulations in the world.
- Unsteady flows with wetness and shocks in actual gas turbines and steam turbines
- Full annulus simulation for resolving unsteady wet-steam and moist-air flows in actual
turbines and compressors

Resolving such complex flows is crucial for developing high-performance and high-
reliable turbines

Unsteady shocks generated in turbine stages
Unsteady wetness in full annuuals turbine stages
Unsteady wet-steam flow in turbine stages
Numerical Schemes

Flowchart of the Iteration Loop

ITERATION START

• Space Difference of Convective and Pressure Terms
  • Roe’s Approximate Riemann Solver
  • Compact MUSCL
    (S.Yamamoto and H.Daiguji, Computer & Fluids, 1993)

• Viscous and Diffusion Terms
  • Second-order Central Difference

• Time Integration
  • LU-SGS Scheme
    (S.Yoon and A.Jameson, AIAA Jurnal, 1988)

• Turbulence Model
  • SST Turbulence Model
    (F.R.Menter, AIAA Journal, 1994)

Space difference scheme
Physical models
Time integration
Calculation of unknown variables
Boundary conditions (Data communications)
Updating of unknown variables

Done?

Yes

No
**Hardware Specification of NEC SX-Aurora TSUBASA**

**Vector Engine (VE)**

- **Type 10B**
  - **Vector Cores**: 8
  - **Frequency**: 1.4 GHz
  - **Performance/core**: 537.6 GF (SP), 268.8 GF (DP)
  - **Performance/processor**: 4.30 TF (SP), 2.15 TF (DP)
  - **Cache Capacity**: 16 MB (shared)
  - **Mem. Bandwidth**: 1.2 TB/s
  - **Mem. Capacity**: 48 GB (HBM2 8GB x 6)

**x86 Processor (Xeon)**

- **Vector Cores**: 8
- **Frequency**: 1.4 GHz
- **Performance/core**: 537.6 GF (SP), 268.8 GF (DP)
- **Performance/processor**: 4.30 TF (SP), 2.15 TF (DP)
- **Cache Capacity**: 16 MB (shared)
- **Mem. Bandwidth**: 1.2 TB/s
- **Mem. Capacity**: 48 GB (HBM2 8GB x 6)

**Long vector loop length** is exploit the high performance of SX-Aurora TSUBASA.
The time integration routine (LU-SGS) is the most dominant.
Data Dependency on the LU-SGS Scheme

- Numerical Turbine adopts the LU-SGS scheme for time-integration
- Data dependency of $q(i,j,k)$ on $q(i-1,j,k)$, $q(i,j-1,k)$ and $q(i,j,k-1)$

Multi-dimensional hyperplane method resolves this data dependency.
The sum of i, j, and k (in case of 2D-HP, the sum of I and j) of each grid point on a hyperplane is the same value.

<table>
<thead>
<tr>
<th></th>
<th>Vector Length</th>
<th>Memory Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Hyperplane</td>
<td>Short</td>
<td>Direct Access</td>
</tr>
<tr>
<td>3D Hyperplane</td>
<td>Long</td>
<td>Indirect Access</td>
</tr>
</tbody>
</table>
3D hyperplane provides long vector lengths. However...
The method is accompanied by a high memory load with indirect memory accesses.

- Latency of the indirect accesses is high.
- Some blips downward in performance occurs due to bank-conflict.

Need to reduce indirect memory accesses in order to extract more performance from SX-Aurora TSUBASA.
OPTIMIZING MEMORY-ACCESS PATTERN OF 3D HYPERPLANE ORDERING METHOD
Changing 3D Data Layout to 1D Data Layout

Original 3D Data Layout

Proposed 1D Data Layout

Indirect memory accesses

Sequential memory accesses
When storing grid points in each 1D array, it is necessary to store the grid points on the boundary area on each hyperplane.
Source Code Image of Changing the Data Layout

Code of Original 3D Data Layout

```
DO LHP=1,LIST_3DC(M,L)
  I = LIST_3DI(LHP,M,L)
  J = LIST_3DJ(LHP,M,L)
  K = LIST_3DK(LHP,M,L)
  ...
  AIM = (A(I-1,J,K) + A(I,J,K)) * 0.5D0
  ...
ENDDO
```

Code of Proposed 1D Data Layout

```
DO LHP=1,LIST_3DC(M,L)
  I = LIST_3DI(LHP,M,L)
  J = LIST_3DJ(LHP,M,L)
  K = LIST_3DK(LHP,M,L)
  IM = LIST_3DA(I-1,J,K,L)
  ...
  AIM = (A(M-1)%ARRAY(IM) * A(M)%ARRAY(LHP)) * 0.5D0
  ...
ENDDO
```

Number of Gather/Scatter Operations in Calculation Part

<table>
<thead>
<tr>
<th></th>
<th>Original 3D Hyperplane</th>
<th>Proposed 3D Hyperplane</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector Gather op.</td>
<td>228</td>
<td>162</td>
</tr>
<tr>
<td>Vector Scatter op.</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

Reducing indirect memory accesses
PERFORMANCE EVALUATION
Performance of The Proposed Method

Performance evaluation using Time-integration kernel codes.

Average: 1.27x
Max: 2.77x faster than original 3D Hyperplane
Some blips downward in performance are reduced.
Cost Distribution of Data-copy and Calculation

Calculation part performance: 123 ~ 146 Gflop/s

The proposed method involves data-copies (3D layout ⇔ 1D layout). Reducing the data-copy time can further improve the performance.
Performance Improvement of Full App.

- Numerical Turbine execution with 636 mpi
- 91x91x181 grid points for each stator and rotor passage

![Graph showing execution time improvements]

- 1.66x faster
- 1.32x faster
- 1.22x faster
- 1.12x faster

**Key Improvements**:
- Expanding vector length
- Reducing indirect accesses
CONCLUSIONS
Conclusions and Future Work

• The 3D Hyperplane ordering method is suitable for SX-Aurora TSUBASA because the method can provide long vector lengths.
• Our proposed 3D Hyperplane ordering method further improves the performance by reducing indirect memory accesses.
• Future work: examine the effect of the method on other applications and major SIMD architectures.
Thank you for your kind attention!