Large Scale Transient CFD Simulations for Buildings using OpenFOAM on a World’s Top-class Supercomputer

©Shimizu Corporation
Fujitsu Limited
RIKEN

Pham Van PHUC
Shuichi CHIBA
Kazuo MINAMI

- The 4th Annual OpenFOAM User Conference 2016 -
Thanks from Japan

- **OpenCFD**
  - Distributed the open-source software, OpenFOAM

- **Global communities**
  - Useful information in open web-sites to help us to study OpenFOAM from the first stage

- **ESI-OpenCFD**
  - Giving an opportunity to introduce our works
Outline

- General Introduction
- Challenges on a Top-class Supercomputer
- Large-scale transient simulations and results
- Recent studies
- Concluding remarks

Key points for OpenFOAM

☆ Comprehensive applications in building engineering field
☆ Problems, abilities of large-scale simulation
  (1~100 billion cells mesh, 10~100 thousand parallels)
Outline

- General Introduction
  - Target top-class supercomputer
  - My company and applications using OpenFOAM

- Challenges on a Top-class Supercomputer
- Large-scale transient simulations and results
- Recent studies
- Concluding remarks
Target Top-class Supercomputer

World’s fastest Supercomputer in 2011 (LINPACK record)

- **Nickname:** K Computer
  - Developed by RIKEN, Fujitsu under promotion of Japan's Government
- **Design and Build Cost:** Over $1.25 billion US
- **System configuration**
  - **Node:** 88,128 CPUs (2GHz, 8cores SPARC64 VIIIfx) → a total of 705,024 cores
  - **Network:** Fujitsu's proprietary Torus fusion (Tofu) interconnect
**First team** has succeed in compiling an open-source **OpenFOAM** in the **K Computer** and doing the sophisticated simulations (2012)
Shimizu Corporation

One of the largest construction companies in Japan

- Founded: In 1804
- Employees: 10,751 (As of April 1, 2016)
- Capital: $724 million US
- Net sales: $15 billion US
- Lines of Business:
  - Construction, Architecture, Engineering and Property Services
Institute of Technology

- Established in 1944
- Researchers: 181
- Staffs: 234
- Mission: Develop new technologies to support the business
The start of using OpenFOAM

- **Early 1950s: In-house codes**
  - Poor operability and less functionality
  - Large maintenance cost

- **Early 1990s: Commercial codes**
  - Rapidly increasing of initial cost with multi-cores
  - Specific needs increase, but low cost is required

- **Early 2000s: Open-source codes**
  - 2009-: Study hard on OpenFOAM (validation & verification, practicality and numerical models)
  - 2010-: Implementations of OpenFOAM on the current world’s top-class supercomputers
Verify the accuracy of OpenFOAM in comparison with experimental results in various building applications.
An application with social impact

Succeed in clarifying the mechanism of Tsunami damages in the Great East Japan Earthquake 2011

3D-VOF Model (interFOAM)

2D Shallow Water Model (in-house)

Multiscale simulation

Estimate directly the Tsunami load on the building
Modeling a city in complex mountain

Generating model using SnappyHexMesh
Observation comparison

Point near Land

High accuracy!

2D: Shallow Water Model
3D: VOF method (interFOAM@LES)
An application for tsunami run-up and inundation in an urban city.
An application for tsunami run-up and inundation in an urban city

OpenFOAM is used widely in our building engineering applications, nowadays
CFD software performance in 2010

Nearly almost the same performance
GPU Speedup Development in 2010

TSUBAME (World’s 4th fastest Supercomputer at that time)

Speedup more 2 times using GPU

- GS: Gauss-Seidel
- ICCG: Incomplete Cholesky Conjugate Gradient
- AMG: Algebraic MultiGrid

![Graph showing speedup of different solvers using GPU](image)
Outline

- General Introduction
- Challenges on a Top-class Supercomputer
  - Motivation of using the supercomputer
  - Challenges/Problems & Solutions for OpenFOAM
- Large-scale transient CFD simulations
- Recent studies
- Concluding remarks
Motivation for using the world’s top-class/fastest supercomputer

- Explore the power of computer, outlook for problems & abilities of simulation in next future

![Graph showing the performance development of supercomputers over the years.](http://www.computerweekly.com/news/2240148760/Supercomputers-will-reach-exascale-speeds-within-decade)

- World’s fastest supercomputer
- Target computer for actual design
- Our PC

Motivation for using the world’s top-class/fastest supercomputer

- Challenging on the large-scale simulation in higher mesh resolution to obtain the higher precision

---

![Graph showing power spectrum intensity vs. dimensionless frequency with different mesh resolutions and supercomputers mentioned](attachment:image.png)

- 100G cells (Δ=1mm)
- 3M cells
- 30M cells
- 100M cells (Δ=0.1mm)
- 100B cells (Δ=0.01mm)

- Supercomputer in 1986
- Supercomputer TSUBAME in 2011

World’s Top-class Supercomputer

Exp.
Motivation for using the world’s top-class/fastest supercomputer

- Challenging to the wide-area and high-level simulation

RANS simulation

LES simulation

※10M cells

500m

500m

30km

10km

2km

※2Billion cells

WIND
Problems in the K Computer

※Newest release OpenFOAM-2.1.1 (2012)

- **Compile a huge codes (OpenFOAM)**
  - Compiler problem

- **Handle a large-scale model**
  - Integer 32bit (Int32) problem
  - Pre-post processing problem

- **Conduct a massively parallel computation**
  - Parallel communication problem
Compiler Problem

- Compiler C++ in K Computer
  - A specific compiler developed by Fujistu Limited
  - Supports ISO/IEC 14882:2003 standard (too old)

- OpenFOAM Codes
  - Based on ISO/IEC 14882:2011 (C++11) standard

Hard-work to analysis/modify the codes for compiling OpenFOAM in the K Computer
Int32 Problem

To handle a model of more than 1 billion cells

C++ Integer

<table>
<thead>
<tr>
<th>Type</th>
<th>Range</th>
<th>Int32 (default)</th>
</tr>
</thead>
<tbody>
<tr>
<td>signed char</td>
<td>SCHAR_MIN</td>
<td>-127</td>
</tr>
<tr>
<td></td>
<td>SCHAR_MAX</td>
<td>+127</td>
</tr>
<tr>
<td>unsigned</td>
<td>UCHAR_MAX</td>
<td>255</td>
</tr>
<tr>
<td>signed short int</td>
<td>SHRT_MIN</td>
<td>-32,767</td>
</tr>
<tr>
<td></td>
<td>SHRT_MAX</td>
<td>+32,767</td>
</tr>
<tr>
<td>unsigned</td>
<td>USHRT_MAX</td>
<td>65,535</td>
</tr>
<tr>
<td>signed int</td>
<td>INT_MIN</td>
<td>-32,767</td>
</tr>
<tr>
<td></td>
<td>INT_MAX</td>
<td>+32,767</td>
</tr>
<tr>
<td>unsigned</td>
<td>UINT_MAX</td>
<td>65,535</td>
</tr>
<tr>
<td>signed long int</td>
<td>LONG_MIN</td>
<td>-2,147,483,647</td>
</tr>
<tr>
<td></td>
<td>LONG_MAX</td>
<td>+2,147,483,647</td>
</tr>
<tr>
<td>unsigned</td>
<td>ULONG_MAX</td>
<td>4,294,967,295</td>
</tr>
<tr>
<td>signed long long int</td>
<td>LONGLONG_MIN</td>
<td>-9,223,372,036,854,775,807</td>
</tr>
<tr>
<td></td>
<td>LONGLONG_MAX</td>
<td>+9,223,372,036,854,775,807</td>
</tr>
<tr>
<td>unsigned</td>
<td>ULLONGLONG_MAX</td>
<td>18,446,744,073,709,551,615</td>
</tr>
</tbody>
</table>

Face number: -1,880,129,018
Error result!

Modification for Int64

2 Billions
Faces: 17,742,344,704 ~ 17.7B
Cells: 5,908,037,632 ~ 5.9B

9 Trillions
Corrected number

(Unfortunately, Int64 option wasn’t available at that time)

※ Int64 Option in the newest OpenFOAM versions should be taken care (memory usage, performance)
**Pre-Post Processing**

- **Serial processing** is impossible by memory shortage

1 billion mesh cells model

Need 1 Terabyte memory onboard of 1 node

1TB: a largest memory in server

<table>
<thead>
<tr>
<th>Model generation</th>
<th>Simulation</th>
<th>Output/Visualization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decompose</td>
<td></td>
<td>Computational domain, results in a single processor</td>
</tr>
<tr>
<td>Load balance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computational domain in a single processor</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Serial processing is impossible by memory shortage

New parallel distributed processing was developed

Model generation
- Initialization
- Load balance

Simulation
- Post process

Output/Visualization
- Data analysis

Image processing

Model of more than 1 billion cells
Massively parallel computation

- **Weak-scaling benchmark**
  - Investigate the large-scale simulation of OpenFOAM

![Tunnel section model](image)

<table>
<thead>
<tr>
<th>No</th>
<th>Decompose</th>
<th>MPI parallels</th>
<th>Cells</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32 × 6 × 4</td>
<td>768</td>
<td>2.013E+08</td>
</tr>
<tr>
<td>2</td>
<td>64 × 6 × 4</td>
<td>1,536</td>
<td>4.027E+08</td>
</tr>
<tr>
<td>3</td>
<td>64 × 12 × 4</td>
<td>3,072</td>
<td>8.053E+08</td>
</tr>
<tr>
<td>4</td>
<td>64 × 12 × 8</td>
<td>6,144</td>
<td>1.611E+09</td>
</tr>
<tr>
<td>5</td>
<td>128 × 24 × 16</td>
<td>49,152</td>
<td>1.288E+10</td>
</tr>
</tbody>
</table>

※ Fixed 262,144 cells/parallel

5e+4 parallels
12.8 billion cells

※Version: OpenFOAM-2-2.x
Massively parallel computation

※Weak-scaling for pisoFOAM

Useless parallel communications were found!
**Problem & Optimization**

**Original OpenFOAM**

```
Optimized code:
- Delete useless parallel communication
- Optimize the parallel platform with specific tuning
```

**Reduce MPI message size**

- ~$N^2$ Increase ($N$: parallel)
- Increase (Number of parallels)
Massively parallel computation

※Weak-scaling for pisoFOAM
Massively parallel computation

※Weak-scaling for pisoFOAM

Detail:
Massively parallel computation

Be surprise at the quick action of the latest version of OpenCFD

Parallel scalability tests indicate that OpenFOAM generally performs well on future systems. Following a review by the Jules and Jeanne Research Organization for significant performance benchmark improvements, OpenFOAM version 1.06 has been revised to make decisions and pass the data to the lower processor on the right. The revised version only obtains the size of each processor and sends it to the master processor, which is significantly faster than the previous version. The master processor distributes the data to all processors, and each processor receives its portion of the data and processes it. This improves the performance of the various tasks parallel to the master processor. The revised version is significantly faster than the previous version, improving the overall performance of OpenFOAM on future systems.
Latest OpenFOAM Performance

Weak-scaling for interFOAM of DamBreak example (Lastweek)

Unfortunately, OpenFOAM-4.0 & Older were stopped with Memory Shortage Error

Done for 49,152 parallels

Original OpenFOAM-v1606+

Optimized code -v1606+

Ideal line for performance

Normalized Performance

Memory Usage (Gbytes)

Number of parallels
Performance of Optimized Code

Weak-Scaling for pisoFOAM

High Performance

Optimized OpenFOAM worked well for ultra-large-scale simulation
Outline

- General Introduction
- Challenges on a Top-class Supercomputer
  - Large-scale transient CFD simulations
    - Large-eddy simulation (LES) for wind tunnel test
- Recent studies
- Concluding remarks
LES for Wind Tunnel Test

Modelling conscientiously a boundary-layer wind tunnel
Computational model

Wind tunnel
Length 104m

Mesh: 6.4Billion cells
using modified `snappyHexMesh`
with low load balancing overhead
under 12,288 parallels

Roughness blocks
Spires
Fence
A tip to generate a large model

Large-scale and high resolution of mesh was generated from the refinement method.

- 15M basic coarse cells
- Refinement 2X
- 120M cells
- 6.4 billion cells

Mesh refinement is powerful (refineMesh, refineHexMesh, redistributePar…).
Simulation result

Detail vortices around building are obtained
The critical local pressure distribution, shows good agreement with experimental result in high accuracy.
Pressure at critical point

Time series of wind pressure is similar to exp. ones.
LES for a Real Building Design

- Estimate the pressure distribution acting on a high-rise building with complicated shape in Tokyo area in comparison with experimental results.
Simulation

Computational domain & meshes

Modeling conscientiously the wind tunnel measurement section

Mesh generation using modified SnappyHexMesh
Simulation

Computational domain & meshes

Mesh A: 140M cells
Mesh B: 1.1 billion cells

Modeling conscientiously the wind tunnel measurement section

Modeling exactly the shape of building

Outlet boundary

Target Building

Roughness Blocks

Spires

No-slip wall

\[ \Delta = 0.5 \text{mm} = \frac{D}{170} \]

\[ \Delta = 0.25 \text{mm} = \frac{D}{340} \]
Computational domain animations

Outlet boundary

Target Building

Roughness Blocks

Spires

No-slip wall

30m

U = 15 m/s

Outlet boundary

Mesh A: 140M cells

Q-criterion isosurface (Q=300,000)

Pressure
Computational domain animations

Outlet boundary
2.5m
Target Building
Roughness Blocks
Spires

30m
No-slip wall

U = 15 m/s

Q-criterion isosurface
(Q = 300,000)

More detail vortices by using high resolution mesh

Mesh B: 1.1 Billion cells
Simulation for all wind directions

Generating mesh for 36 directions at 10° intervals as the same as experiment
Simulation for all wind directions

K Computer

Using 27,648 CPUs and running on 20 days

Meshes for 36 wind directions

Total meshes: ~ 5 Billion cells
Simulation for all wind directions

K Computer

Using 27,648 CPUs and running on 20 days

Parallel simulations for 36 wind directions
Obtaining the wind pressure distributions acting on building,
Then estimating the largest peak coefficients for all wind directions …
Simulation for all wind directions

K Computer

Using 27,648 CPUs and running on 20 days

Largest positive peak

Largest negative peak

Achieve the same variation as experimental ones (20%)

Largest peak coefficients for building design
Outline

■ General Introduction
■ Challenges on a Top-class Supercomputer
■ Large-scale transient CFD simulations
■ Recent studies
■ Concluding remarks
New Subgrid-scale model for LES

Half computational cost & high accuracy

Comparison of pressure coefficients

Smagorinsky Model

New Model

Wind pressure distribution

Conical vortex

Horseshoe-like vortex
Challenging on Multi-scale simulation

Multi-scale simulation from meteorology to building with several billion cell meshes
## Challenging for the Next Generation of K Computer (Post-K Computer in 2020)

### Learning from Current World’s TOP5 supercomputer (2016)

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Processors</th>
<th>Cores</th>
<th>System Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sunway TaihuLight</td>
<td>40,960</td>
<td>10,649,600</td>
<td>Sunway SW26010 (260cores)</td>
</tr>
<tr>
<td>2</td>
<td>Tianhe-2</td>
<td>32,000</td>
<td>3,120,000</td>
<td>Xeon E5–2692 + Xeon Phi 31S1P</td>
</tr>
<tr>
<td>3</td>
<td>Titan</td>
<td>35,040</td>
<td>560,640*</td>
<td>Opteron 6274 + Tesla K20X</td>
</tr>
<tr>
<td>4</td>
<td>Sequoia</td>
<td>98,304</td>
<td>1,572,864</td>
<td>PowerPC A2, Custom</td>
</tr>
<tr>
<td>5</td>
<td>K</td>
<td>88,128</td>
<td>705,024</td>
<td>SPARC64 VIIIfx (8cores)</td>
</tr>
</tbody>
</table>

Various architecture, Many-cores with low performance!
Collaboration

Studies on a new advanced compiler and hardware-software system co-design
  - OpenFOAM is a good target C++ software

Ultra-large scale simulation of several million parallels using OpenFOAM
  - Several problems are found
Parallel computation with only MPI (Flat-MPI) will be impossible!!

Full Hybrid OpenFOAM is necessary !?!!

Challenging for the Next Generation of K Computer (Post-K Computer in 2020)

※K Computer: limited by number of 165,888MPI parallels
Full-Hybrid OpenFOAM Development

Weak Scaling

New Full-Hybrid OpenFOAM (MPI+OpenMP)

Original OpenFOAM (MPI)

Efficiency (%) vs. nodes

2 Billion cells  4 Billion cells

pisoFoam with structure model
Concluding remarks

- **OpenFOAM** is a robust **CFD** platform.
  - It expedites transparency in numerical modelling and productivity for customizations and be used widely in the building engineering applications, nowadays.

- **OpenFOAM** has a large potential abilities in large-scale simulation.
  - Under specific modifications, it worked for a large scale transient CFD simulation up to 100 billion cell meshes and achieved a high performance for 100 thousand MPI parallels.
Acknowledgments

- This research used computational resources of the **K computer** provided by the RIKEN Advanced Institute for Computational Science through the **HPCI** System Research projects (Project ID:hp120050, hp140055, hp150031)

- A part of studies was supported by Research Organization for Information Science and Technology (**RIST**)
Thank you for your attention!