1. Introduction

In this study, mSMS [1], which is page-based distributed shared memory system is used for multi-node implementation of FDTD(2,4) method, and the performance evaluation is presented. For efficient large-scale acoustic analysis, it is important to employ high-order FDTD method such as FDTD(2,4) method [2], which has second-order accuracy in time and fourth-order accuracy in space. To improve the programming productivity in multi-node environments, the PGAS language that provides a global view of programming can be used.

2. Performance Evaluation

![Graphs showing weak scaling results and barrier time](image)

**Experimental Environment and Conditions**

TSUBAME 3.0 (Intel Xeon E5-2680 v4, 14 core, 2.4GHz × 2 / node, Intel Omni-Path 100Gb/s × 4, Intel MPI 2018.1.163)

ITO Subsystem A (Intel Xeon Gold 6154, 18 core, 3.0 GHz × 2 / node, InfiniBand EDR 4x 100Gb/s, MVAPICH2-X 2.2)

mSMS version: r121, Calculation Precision: double, Calculation Mesh Size: 1024^3/node, OpenMP Threads Number: 24

**Remote data preload API**

Efficient explicit data prefetching (preloading continuous pages with specified size) is possible by using preload API.

3. Application Example

**Helmholtz Resonator Model with PML (Perfectly Matched Layer) [3]**

![Graph showing pressure distribution](image)

**Even in the complex boundary problem, mSMS can achieve decent results with few code modifications. The below results are without using Preload API (no explicit code optimization for data transfer).**

4. Conclusion and Future Work

**Conclusion**

We demonstrated the nearly ideal weak scaling performance of the FDTD(2,4) method parallelized with mSMS and OpenMP, in spite of using a simple implementation.

**Future Work**

- Overlapping computation/communication by using a non-blocking preload API.
- Implementation and performance evaluation of FDTD(2,4) method with mSMS-GPGPU.
- Incorporating spatial and temporal blocking (non-redundant) into the FDTD(2,4) method.

5. References

