Efficient Data Compression by Efficient Use of HDF5 Format

Katsumi Hagita
National Defense Academy
1-10-20, Hashirimizu,
Yokosuka, 239-8686, JAPAN
+81-46-841-3810
hagita@nda.ac.jp

Manabu Omiya
Hokkaido University
Kita 11, Nishi 5, Kita-ku,
Sapporo 060-0811, JAPAN
+81-11-706-2946
omiya@iic.hokudai.ac.jp

Takashi Honda
ZEON Corporation
1-2-1, Yako, Kawasaki-ku,
Kawasaki, 210-9507, JAPAN
+81-44-276-4154
T5.Honda@zeon.co.jp

Masao Ogino
Nagoya University
Furo-cho, Chikusa-ku,
Nagoya, 464-8601, JAPAN
+81- 52-789-5899
masao.ogino@cc.nagoya-u.ac.jp

ABSTRACT
In supercomputing, transferring data is a significant problem. We have thus proposed JHPCN-DF (Jointed Hierarchical Precision Compression Number - Data Format), an efficient data compression method, which does not require change to common application programming interfaces (APIs) such as HDF5. This compression method uses bit segmentation and Huffman coding. For visualizations and analyses, the lower bits of the IEEE 754 format are considered to be unnecessary. The data size of the HDF5 is thus reduced by setting these lower bits to 0. The required bits can be determined by direct numerical computations and their corresponding resultant visualizations and analyses. In this paper, we evaluated the data size of HDF5 files in the JPHCN-DF framework for various types of simulations such as those using the plasma PIC (electromagnetic particle-in-cell) model, the finite element method (FEM), and for phase separated polymer materials. We confirmed that visualization software work well with HDF5 files coordinated by the JPHCN-DF framework. In visualization, the data size of the HDF5 file appears to be reduced to 1/5 of the original HDF5 file size. JPHCN-DF is useful for a wide range of applications in data science and simulation.

Categories and Subject Descriptors
I.3.6 [**COMPUTER GRAPHICS**]: Methodology and Techniques – *Graphics data structures and data types.*

General Terms
Experimentation

Keywords
Data compression, visualization, data format

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SC'14, Nov. 16–21, 2014, New Orleans, LA, USA.
Copyright is held by the authors.

1. INTRODUCTION
Recently, supercomputing is producing huge amounts of data which is difficult to store and transfer between different storage devices. Depending on the various purposes, visualization and analyses of scientific data require systematic effort when recording the data. It is considered that different numbers of bits in the IEEE 754 format is used to represent data for different purposes, as shown in Figure 1. In scientific simulations using the laws of physics, higher bits are correlated in space and time. However, sometimes the lower bits do not contribute to visualization and analyses at all. For these cases, the lower bits seem to present random noise-like behavior which negatively affects data compression. A higher compression ratio using only the higher bits can be observed if the lower bits can be neglected. Therefore, segmented recording based on the required numerical precision may provide a good solution to the need for compression. For a record containing a variable-length field, an application programming interface (API) needs to be implemented. A combination of zero padding and a common data format, such as the Hierarchical Data Format 5 (HDF5) [1], which can support Huffman coding, can be effective because the size of information of the zero padded bits becomes negligibly small after Huffman coding. We have thus proposed JHPCN-DF (Jointed Hierarchical Precision Compression Number - Data Format), which is a hierarchical treatment of this concept.

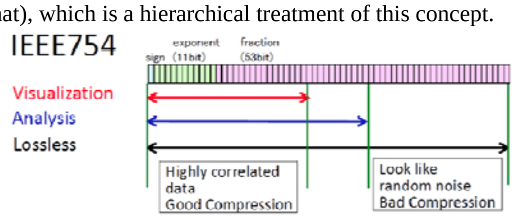

Figure 1 Required bits for visualization and analysis.

2. CONCEPT of JHPCN-DF
Basically, lossless compression is proposed for the framework of JHPCN-DF. For efficiency, HDF5 files with higher bits can be used in many applications that support HDF5 without changing the API. In our implementation, we checked the required bit-length of floating point data for user-specified precisions. For the case of IEEE 754 single-precision floating point data, the stored

value of the original one-variable requires zero padding and a 32-bit integer to record the separated bits necessary to reconstruct higher precision data and the original data (lossless). The recording in the separated HDF5 files using the JHPCN-DF framework are presented in Figure 2.

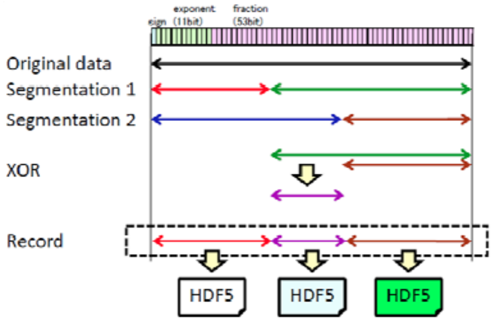

Figure 2 Separated HDF5 files created within the JHPCN-DF.

3. RESULTS AND DISCUSSION

In the present work, compression performance was studied in the practical visualization of different simulations such as the plasma PIC (electromagnetic particle-in-cell) model, finite element method applications, and phase separated polymer materials.

3.1 Plasma PIC simulation

As an example of a particle system, we studied the trajectories (2000 time steps) of 150,000 ions and 150,000 electrons, data available as a subset in related studies [2].The size of 20 HDF5 files with 100 time steps was 6,336 MB in the HDF5 format, where the ID and index (4-byte integer) and 3d position data (4 byte/float) were recorded for each particle. Snapshots captured the allowed errors 0.01, 0.02, and 0.05 respectively (Figure 3). The results of visualization indicated that the required precision of the 3d positions should be 0.02. As a test, we separated the HDF5 files into two parts in the JHPCN-DF framework. When the allowed error was set at 0.02, the size of the first section (left side of HDF5 in Figure 2) was 1,413 MB and the recorded size for the other bits was 5,586 MB. For the case of 0.01 error, 1,940 MB and 5,298 MB respectively were obtained. At 0.05 error, the values of 883.1 MB and 5,712 MB respectively were attained. We established that the data size required for visualization purposes can be reduced to about 1/5 of the original HDF5 size.

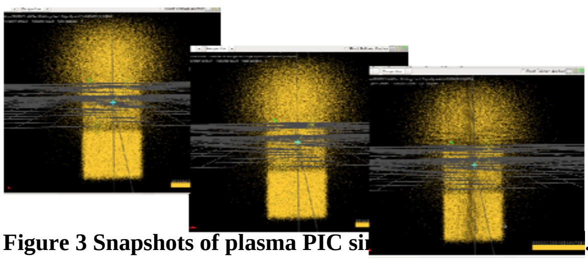

Figure 3 Snapshots of plasma PIC si .
(Allowed errors: left: 0.01, center: 0.02, and right: 0.05)

3.2 Finite Element Method (FEM)

As an example of grid-based methods, finite element analysis (FEA) of 0.24 million elements and 0.38 million nodes was carried out. To perform FEA, the ADVENTURE system was used, which is an open source software for solving practical engineering problems [3]. The size of FEA data of an elastostatic analysis was 8.6 MB in the HDF5 format, where 3d displacements with double precision (8-bytes) were recorded for each node. We measured the size of HDF5 (within JHPCN-DF) for two precisions such as 10^{-3} and 10^{-6} corresponding to the required precisions for visualization and evaluation, respectively. An allowed error for evaluation was decided by a convergence criterion for a linear solver. The data were successfully reduced to 3.3 MB and 1.3 MB in the cases of evaluation and visualization, respectively.

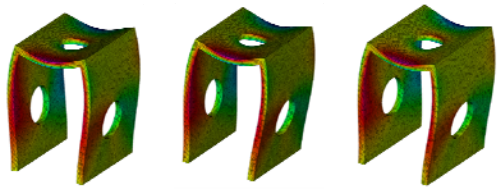

Figure 4 Snapshots of FEM simulation visualization.
(left: original precision, center: 10^{-6}, and right: 10^{-3})

3.3 Phase separated polymer materials.

As an example of density field data, we examined a phase separated polymer material, consisting of one large domain and many small spherical droplets. For industrial applications, we performed large-scale simulations with an 512^3 mesh using OCTA/SUSHI [4]. The size of the original ascii output was 1182 MB. On the other hand, the size of HDF5 in AVS/Express format was 42 MB and 422 MB with and without the JHPCN-DF framework, respectively. The allowed error was 0.05.

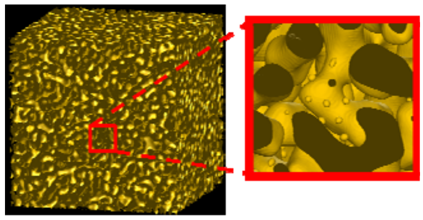

Figure 5 Isosurface of the phase separated polymer material.

4. CONCLUSIONS

We tested the performance of JHPCN-DF for data of plasma PIC simulations, finite element method (FEM) analysis, and phase separated polymer materials. JHPCN-DF was found to be effective in a wide range of data science and simulations. For further improvement, HDF5 filters to implement compression and cooperative functions with the system software are considered desirable. Studies in this direction are in progress.

5. ACKNOWLEDGMENTS

This work was supported by "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures", “High Performance Computing Infrastructure” in Japan, and the HPC project of Nagoya University. (Project ID: 14-jh140004-NA02, 14-NA28, hp140191) One of the authors (K.H.) thanks Dr. H. Ohtani, Dr. T. Kato, Dr. A. M. Ito, Dr. T. Saito, Dr. T. Takeda, Prof. S. Ishiguro, and Prof. Y. Ishikawa for useful discussion.

6. REFERENCES

[1] http://www.hdfgroup.org/

[2] K. Hagita, H. Ohtani, T. Kato and S. Ishiguro: Plasma and Fusion Research Vol.9 (2014) 3401083

[3] http://adventure.sys.t.u-tokyo.ac.jp/

[4] http://www.octa.jp/