Scientific Visualization, 2020, volume 12, number 1, pages 83 - 89, DOI: 10.26583/sv.12.1.07

Using Big Data Analysis Instruments to Tackle the Problem of Simulation Results Visualization in Large Scale Gas Dynamics

V.I. Reshetnikov[1], E.A. Golubchikov[2], A.V. Pyatlin[3], A.K. Kuzin[4], V.A. Kiev[5],
N.N. Shabrov[6], A.S. Zhuravlev[7], E.K. Guseva[8]

Peter the Great St. Petersburg Polytechnic University

[1] ORCID: 0000-0002-3953-3440, skemobob@gmail.com
[2] ORCID: 0000-0002-0366-1424, golubchea@mail.ru
[3] ORCID: 0000-0002-1851-702X, andrey@pyatlin.com
[4] ORCID: 0000-0002-3715-5829, kuzin_aleksei@mail.ru
[5] ORCID: 0000-0002-3790-2994, vlad@rwwws.ru
[6] ORCID: 0000-0001-7721-1824, shabrov@rwwws.ru
[7] ORCID: 0000-0002-9973-1705, zhurus@mail.ru
[8] ORCID: 0000-0002-7117-2454, katia.guseva@inbox.ru

Abstract

The article is dedicated to the development of the software for fast visualization of simulation results of gas dynamics problems on large meshes. Kitware ParaView, which is popular software among engineers and scientists, was selected as a graphical frontend. Within the framework of the project, an interaction between ParaView client and server is used. The key feature of the work is the usage of Apache Hadoop and Apache Spark systems for distributed loading of the data into ParaView. The results are stored on the cluster in Hadoop Distributed File System (HDFS) and are provided to ParaView server by means of data processing tool Apache Spark.

Keywords: efficient visualization, Apache Spark, ParaView, big data analysis.

1. Introduction

The computing power of modern high-efficient supercomputers and the level of development of specialized problem-oriented software of predictive modeling allow to increase the resolution of computational meshes up to the order of more than billions of nodes. The simulation results on such large meshes are big data arrays, especially in case of unsteady processes. The amount of obtained data raises the problem of low visualization and analysis speed.

Software and hardware tools of predictive modeling that are available for the wide community of engineers and scientists allow to process gas-dynamic simulations on the meshes up to 10 billions elements. But the visualization tools do not provide desired productivity. One of the main problems is the low visualization speed. Moreover, it is not only due to processing. It is revealed [1] that input/output time can be much bigger than rendering and evaluation. For example, reading of simulation results on the mesh with 5.5 billions nodes takes about 30 seconds on the regular PC.

The necessity to develop the software that provides high speed of input takes place. Its usage could be relevant with problems of gas dynamics on extra-large meshes. For example, such software may be demanded for visualization of simulation results by exploring the turbulent flows in the ring nozzles or around the rigid body. The current paper is dedicated to the software development providing the fast reading of the simulation results to present them for the final user. Hereinafter, quickness in reading is meant to be the ability to provide a higher reading speed than the traditional sequential approach.

The software developed in this work is based on the idea of using programming tools for big data distributed analysis such as Apache Hadoop and Apache Spark. These components give distributed access to the data from the cluster nodes that can significantly reduce the time of data access in comparison with the traditional sequential way. Hadoop Distributed File System (HDFS) is used, and the server based on Spark framework gives an ability to process the queries for retrieving the required data set from the cluster. A plugin for ParaView, which is developed by the authors, attached to ParaView server, composes queries to Spark and plays the role of client for Spark. The user on the client side has a client version of ParaView, which receives results of rendering from ParaView server.

The similar problem of Apache Hadoop usage in development of visualization system for finite elements modeling results was considered in [2]. One more problem linked with Hadoop usage was investigated in [3] with the difference that this work was based on the usage of Apache Hive instead of Apache Spark. The hybrid approach assuming usage of HDFS and Kitware ParaView as a user interface was the key idea in [4]. In papers [5, 6] Hadoop and Spark are applied for visualization and analysis of atmospheric phenomena modeling results by climate investigation. Main attention in these two works is payed to the task of statistical analysis.

The solution of considered scientific problem must meet following requirements:

1. The software must represent results in a convenient form for understanding and analysis, so that it can be used by engineers, who do not have advanced skills in IT.
2. The software must provide reading of simulation results on higher speed than in the case of traditional sequential approach. The criterion is the following condition: the time of the data reading by the developed software must be at least twice less than the time of reading in traditional way from the local computer.

In addition, the authors performed the exploratory analysis of dependency between node count and reading speed.

2. Architecture

In this section, an architecture of the developed software and communications between its components are described.

The environment is built on the client-server scheme and has the structure shown in the figure 1. There are the following main components:

- **ParaView Client**. The client version of ParaView. It is installed on the local computer and is developed to interact with the user. It visualizes rendering results prepared by the ParaView server.
- **ParaView Server.** The Server version of ParaView installed on the cluster. It provides efficient parallel rendering based on the data, which are sent from the ParaView plugin designed by the authors.
- **Plugin to ParaView.** Developed by the authors plugin which is integrated with ParaView server and intended for efficient data reading. Reading is processed as a response to SQL-queries to the data server that is launched on the same or another cluster.
- **Data server.** Server is developed using Python with the usage of Apache Thrift framework. The server receives queries from the ParaView plugin and returns data blocks in response. It transfers queries to the Spark system, which processes data fetching from HDFS.

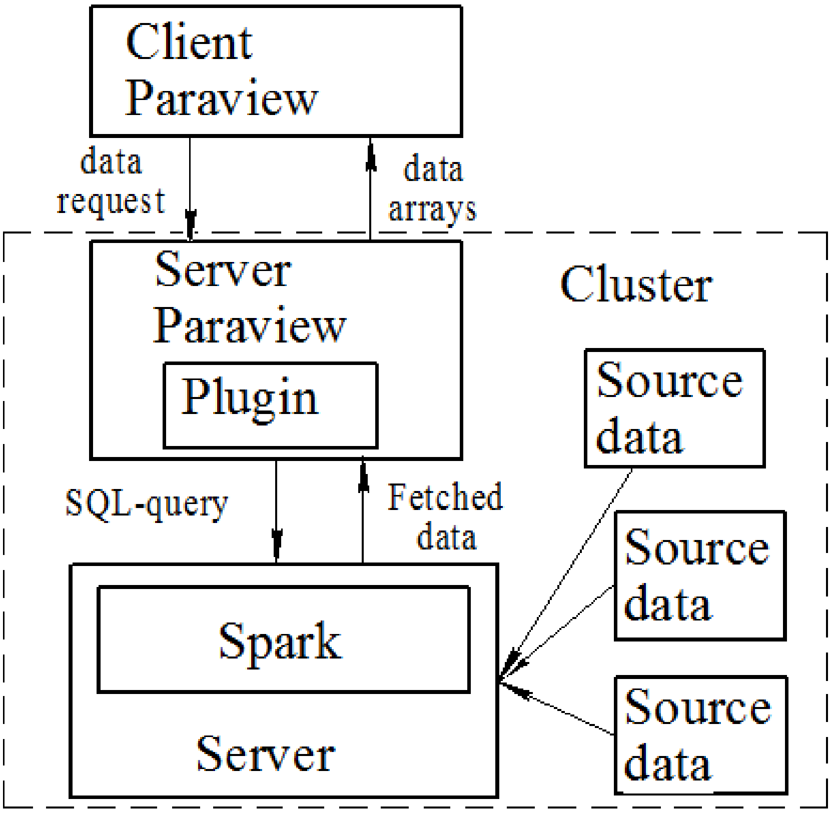

Figure 1. The scheme of the interaction between main parts of software.

The interaction between client and server parts of ParaView is a quite traditional way of using ParaView providing parallel model rendering [7]. The main purpose of the authors' work is development of the plugin to ParaView and data server. Currently the plugin sends the VTK-object vtkMultiBlockDataset to the ParaView client. But instead of direct data file reading it forms SQL-query to data server and receives data in response. Server transfers received query to Apache Spark. Spark performs distributed reading, collection and sending the data as a response to SQL-query. The advantage of such a scheme over the direct reading from a file is that the reading is performed in parallel on several nodes of the cluster, which can give a gain in speed, especially on large files.

The speed of reading significantly depends on the file format. The format must meet some requirements. Firstly, the size of the file must be as small as possible. Secondly, it must give the fast access to the particular data blocks, which is important in cases when there is no need to read the whole file. Thirdly, it must support distributed storing and be readable by Spark. Apache Parquet meets all of these points. Moreover, it has enough flexibility to form necessary block structure of data storing. The format itself presents the set of columns organized in hierarchical way under the rules of the special scheme. The choice of the scheme stays for the person who develops the file recording. The scheme itself is a part of format and is recorded to the file metadata and can be recovered from the file. Thus, Parquet file can store hierarchical structure of high complexity.

Block structure was chosen as a way of structured mesh data storage. It is obtained from the original index parallelepiped of grid elements by dividing it by parallel index planes in three directions. Each obtained block occupies the space in the way of curvilinear hexahedron and also forms structured mesh. The storage of such block in Parquet file is organized as a set of Parquet columns for each coordinate and field. Due to the specifics of the Parquet format, column addresses are stored in the metadata of the file and each column can be accessed directly without having to read the entire file, which solves the problem of selective reading of only the necessary blocks of the grid.

3. Usage example

The example of the simulation results visualization is considered. Simulation represents the flow around the rigid body evaluated on the structured mesh of hexahedrons (see Fig. 2). Considered model is three-dimensional. Original data are presented as a set of time frames where each frame is stored in Tecplot file. The model consists of approximately $5 \cdot 10^6$ nodes. Each file has a size of 1.27 GB.

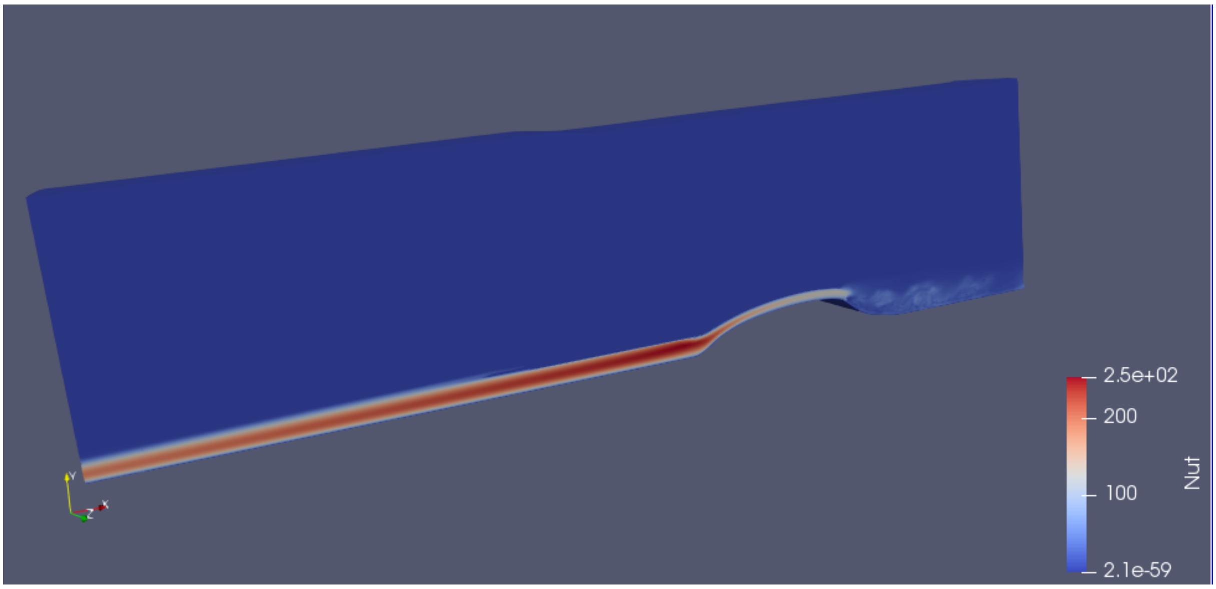

Figure 2. The time frame of the simulation results visualization. Simulation represents the turbulent flow around the rigid body. Visualization of the field of turbulent viscosity kinematic coefficient is presented.

When such a sequence of frames is directly visualized on a personal computer in ParaView, it takes about 30 seconds to load one frame, which is a serious inconvenience for analyzing the simulation results. Most of the time is spent on reading of the file.

The same data are visualized with the usage of developed software. All components except ParaView client are installed on the cluster. Nodes of "RSC Tornado" of Saint-Petersburg Ploytechnic University Supercomputer Center are used as the cluster. Each node consists of two CPU Intel Xeon E5-2697 v3 (14 cores, 2.6 GHz) and 64 GB RAM DDR4.

Data files were initially converted to Parquet format for visualization in developed environment so that the size of one file was reduced approximately to 400 MB. Within the recording in Parquet data was transformed as it was described earlier. Original index parallelepiped of structured mesh was cut with mutually orthogonal index surfaces into parallelepipeds of smaller size. Inside of each parallelepiped each coordinate and each field became separate Parquet column. Such structure is efficient in retrieving of particular blocks as it does not require the whole data reading.

Apache Spark is launched on the cluster under the management of the Slurm system. Reading of the Parquet files is performed with a help of pyarrow library as it provides high speed and requires much less memory than built-in Spark tools.

Table 1. Time of reading and presentation of particular frames on 8 cluster nodes "RSC Tornado"

Frame	Reading time, s	Total time, s	FPS
1	4.655	49.314	0.020
2	2.794	14.142	0.071
3	2.723	11.862	0.084
4	2.790	10.895	0.092
5	2.956	15.596	0.064

The time of the whole processing is presented in the table 1. The second column shows the time of reading of Parquet file by Spark with a help of pyarrow. The third column presents total time of processing including the request for the data, its reading and sending the reading results to client. The last column is the quantity, which shows the inverse of the total time. The biggest part of time is spent to transfer from the ParaView server to client, because the time of response is quite bigger than the reading time. It illustrates the necessity of faster network. Increased time of reading and transfer the first time frame is caused primarily by spending resources on mesh reading. The main point is that mesh remains the same frame by frame. Beginning from the second frame the system reads only the values of the displayed field.

Although the total frame display time is quite long, a significant reduction in reading time has been achieved. Thus, when using the developed software package the time of reading the first frame is about 6 times less than the time of reading from the local machine. For the following frames this ratio approximately equals to 10.

The increase in the number of nodes involved did not result in the expected decrease in data read time in Spark. Finding the cause of this phenomenon is one of the tasks of further research.

4. Conclusions

The environment for the fast visualization of the results of the simulations in gas dynamics evaluated on the meshes of the large size has been developed. The software consists of ParaView client, ParaView server, data server transferring SQL-queries to Apache Spark. Apache Spark is used to get the increase in big data reading speed due to providing the distributed access.

Experiments done show the efficiency of using Parquet format to store the data. Comparing with text format Tecplot, the size of Parquet file is less, it is directly readable by Apache Spark and provides the ability to retrieve particular blocks avoiding the reading of the whole file.

Developed software meets the requirements put forward to it. It does not require the user to have any special skills in IT. The only thing the user should do when working with it is to attach the plugin to ParaView and to convert the source data with the help of auxiliary tools to Parquet format. Also the use of the environment significantly reduces the time of the data reading. When reading the first frame with the use of developed software the ratio of spent time is 6 times less than the time of reading from the local machine. For the following frames this ratio equals to 10. This difference between the first and the next frames is caused by necessity of reading the mesh during the reading of the first frame. During the reading of the next frames, the system reads only the values of the fields.

In addition, experiments have shown the lack of scalability of the speed of reading Spark data with an increase in the number of nodes and the high efficiency losses on the data transfer from Spark to ParaView server causing the big response time of system on user's actions. These problems are the challenges for the further development.

Another challenge is to use more complex SQL-queries. For example, these can be requests to retrieve data that correspond to the visible part of the model. In addition, there might be a query for data that is spread across layers. The layer whose data should be retrieved depends on the camera position. If it corresponds to a higher level of detail, the layer must contain more nodes.

Gratitude

The authors thank Russian Science Foundation for support, grant No. 18-11-00245.

References

1. Childs H., Brugger E., Bonnell K., Meredith J., Miller M., Whitlock B., Maxi. N.: A contract based system for large data visualization. // In: Visualization, 2005. VIS 05. IEEE, 191–198. 2005.
2. Lange B., Nguyen T.: A Hadoop distribution for engineering simulation. [Research Report] INRIA Grenoble - Rhône-Alpes 2014
3. Artigues A., Cucchietti F. M., Montes C. T., Vicente D., Calmet H., Marıın G., Houzeaux G., Vazquez M.: Scientific Big Data Visualization: a Coupled Tools Approach. // Supercomputing Frontiers And Innovations, 1(3), 4–18. 2014.
4. Mitchell C., Ahrens J., Wang J.: VisIO: Enabling Interactive Visualization of Ultra-Scale, Time Series Data via High-Bandwidth Distributed I/O Systems. // In: Parallel & Distributed Processing Symposium (IPDPS), 2011 IEEE International, 68–79. 2011.
5. Shujia Zhou, Xi Yang, Xiaowen Li, Toshihisa Matsui, Si Liu, Xian-He Sun, Weikuo Tao. A Hadoop-Based Visualization and Diagnosis Framework for Earth Science Data. // In: IEEE International Conference on Big Data, 1911–1916. 2015.
6. Shujia Zhou, Xiaowen Li, Toshihisa Matsui, Weikuo Tao. Visualization and Diagnosis of Earth Science Data through Hadoop and Spark. // In: IEEE International Conference on Big Data, 2974–2980. 2016.
7. Paraview software. http://www.paraview.org . Paraview is developed by Kitware company. http://www.kitware.com.