

Применение средств анализа больших данных в проблеме визуализации результатов решения задач газодинамики большой размерности

В.И. Решетников¹, Е.А. Голубчиков², А.В. Пятлин³, А.К. Кузин⁴, В.А. Киев⁵,
Н.Н. Шабров⁶, А.С. Журавлёв⁷, Е.К. Гусева⁸

Санкт-Петербургский Политехнический университет Петра Великого

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

Аннотация

Работа посвящена созданию программной среды для быстрой визуализации результатов решения задач гидрогазодинамики на больших сетках. В качестве графического интерфейса пользователя выбран популярный в инженерных и научных кругах пакет визуализации Kitware ParaView. В рамках проекта применяется взаимодействие клиентской и серверной версий ParaView. Ключевым моментом является использование систем Apache Hadoop и Apache Spark для распределенного извлечения результатов моделирования из файлов. Результаты хранятся на кластере в файловой системе Hadoop Distributed File System (HDFS) под управлением Apache Hadoop и предоставляются серверной версии ParaView средствами обработки данных Spark.

Ключевые слова: быстрая визуализация, Apache Spark, ParaView, анализ больших данных.

1. Введение

Возможности современных высокопроизводительных суперкомпьютеров и уровень развития специализированных проблемно-ориентированных программных комплексов предсказательного моделирования позволяют при необходимости увеличить разрешающую способность вычислительных сеток до величин порядка миллиардов узлов и более. Файлы результатов моделирования на таких больших сетках представляют собой огромные массивы данных, особенно в случае нестационарных процессов. Объем полученных данных порождает проблему низкой скорости научной визуализации и анализа результатов.

В то время как программные и аппаратные средства предсказательного моделирования, доступные широкому кругу ученых, позволяют производить газодинамические расчеты на сетках объемом до 10 млрд. ячеек, средства визуализации на таких сетках пока не обеспечивают желаемой производительности. Одной из основных проблем остается низкая скорость визуализации, причем это относится не только непосредственно к процедуре рендеринга. В [1] показано, что время ввода и вывода данных может быть на два порядка больше, чем рендеринг и вычисление. Вывод результатов вы-

числений на конечно-элементной сетке из 5,5 млрд. узлов на персональном среднестатистическом компьютере занимает около 30 секунд.

Возникает необходимость в создании программного комплекса, обеспечивающего быстрое чтение больших данных. Его использование было бы актуально, например, при анализе результатов предсказательного моделирования задач внешней и внутренней аэродинамики современных летательных аппаратов на сверхбольших сетках. Это могут быть такие задачи механики жидкости и газа, как исследование турбулентных течений в кольцевых и иных соплах ракетных двигателей или исследование структуры течения при обтекании твердого тела газовой средой. Данная статья посвящена разработке программной среды, обеспечивающей быстрое чтение результатов расчета с их дальнейшим отображением пользователю. Под быстротой здесь понимается способность предоставить скорость чтения данных выше, чем при традиционном последовательном подходе.

Ключевой идеей, лежащей в основе разрабатываемого программного комплекса, является использование программных средств распределенного анализа больших данных, а именно Apache Hadoop в связке с Apache Spark. Эти средства обеспечивают распределенное получение данных с узлов кластера, что может существенно сэкономить время доступа к данным в сравнении с традиционным последовательным подходом. Hadoop, в первую очередь, используется для поддержки распределенной файловой системы Hadoop Distributed File System (HDFS), а построенный на основе фреймворка Spark сервер обеспечивает распределенную обработку запросов на получение требуемого набора данных с кластера. Клиентом сервера Spark выступает создаваемый авторами плагин ParaView, встроенный в серверный ParaView и осуществляющий формирование запросов для сервера Spark. Компьютер же конечного пользователя оснащен клиентской версией ParaView, получающей готовые результаты рендеринга с сервера ParaView.

Проблема применения Apache Hadoop в создании комплекса визуализации результатов конечно-элементного моделирования развивается в проекте, описанном в [2]. Схожая постановка была рассмотрена в [3] с тем отличием, что в этой работе использовался Apache Hive вместо Apache Spark. Также следует отметить статью [4], в которой предложен гибридный подход, предполагающий использование файловой системы HDFS и Kitware ParaView в качестве пользовательского интерфейса. В работах [5, 6] фреймворки Hadoop и Spark применены для анализа и визуализации результатов моделирования атмосферных явлений в задачах исследования климата. Основное внимание здесь уделено, в первую очередь, задачам статистического анализа.

Решение поставленной научной проблемы — программный комплекс — должен удовлетворять следующим требованиям:

1. Программный комплекс должен предоставлять результаты в удобном для восприятия и анализа виде, так как предполагается, что это решение может использоваться инженерами, не имеющими специальных знаний в информационных технологиях.

2. Программный комплекс должен обеспечивать чтение результатов анализа механической проблемы с большей скоростью по сравнению с традиционным подходом чтения данных с локальной машины. В качестве критерия ставится следующее условие: время чтения данных при помощи программного комплекса должно быть хотя бы в 2 раза меньше, чем время чтения исходных данных с локальной машины.

Помимо этого, ставится задача исследовать изменение скорости чтения данных при увеличении числа задействованных узлов.

2. Архитектура

В этом разделе описывается архитектура программного комплекса и рассказывается, как отдельные его части взаимодействуют между собой в процессе выполнения типичной задачи визуализации.

Программная среда построена по клиент-серверной схеме и имеет структуру, приведенную на рисунке 1. Основные элементы схемы таковы:

- **Клиентский ParaView.** Клиентская версия известного пакета научной визуализации Kitware ParaView, установленная на локальной машине и предназначенная для непосредственного взаимодействия с пользователем. Визуализирует готовые результаты рендеринга, полученные с серверного ParaView. Единственный компонент программной среды, работающий на локальной машине.

- **Серверный ParaView.** Серверная версия ParaView, установленная на кластере. Обеспечивает эффективный параллельный рендеринг на основе данных, полученных средствами разработанного авторами плагина к ParaView.

- **Плагин к ParaView.** Созданный авторами плагин, встроенный в серверный ParaView и предназначенный для эффективного чтения данных. Чтение осуществляется в виде ответов на клиентские SQL-запросы к серверу данных, также запущенному на кластере, этом же, или отдельном.

- **Сервер данных.** Написанный на Python с использованием фреймворка Apache Thrift сервер, получающий запросы плагина ParaView и возвращающий блоки данных в ответах. Сервер перенаправляет запросы системе Spark, которая и выполняет распределенное извлечение данных из файловой системы HDFS.

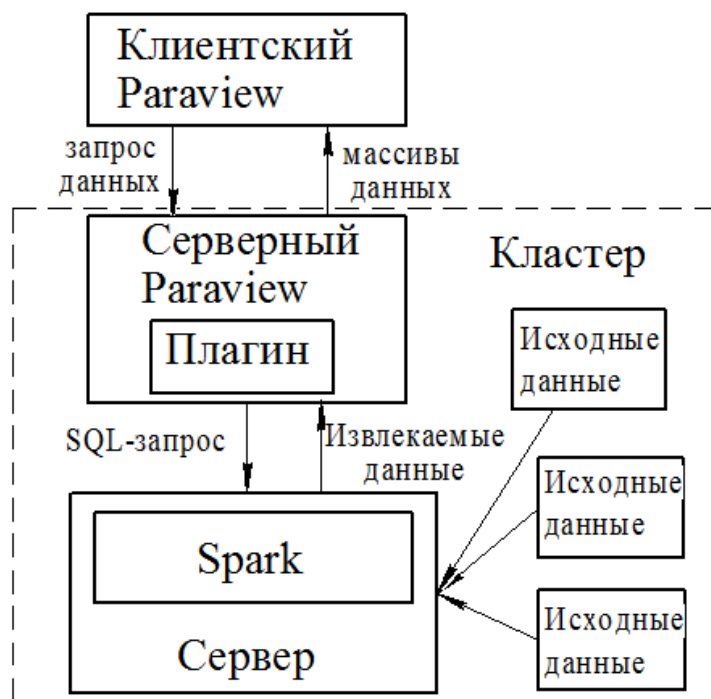


Рисунок 1. Схема взаимодействия программных компонентов.

Взаимодействие клиентской и серверной частей ParaView — вполне традиционная схема использования ParaView, обеспечивающая параллельный рендеринг модели [7]. Основным направлением деятельности авторов является разработка плагина ParaView и сервера данных. Плагин с точки зрения ParaView есть обычный плагин чтения модели, представляющий данные на структурированной сетке. В настоящее время он предоставляет клиентскому ParaView VTK-объект `vtkMultiBlockDataset`. Однако вместо непосредственного чтения файла данных плагин формирует SQL-запрос к серверу данных и получает данные в ответе. Сервер передает полученный запрос в Apache Spark. Spark выполняет распределенное чтение, сборку (операция `collect`) и отправку данных в виде ответа на SQL-запрос. Преимущество такой схемы перед непосредственным чтением из файла состоит в том, что чтение выполняется параллельно на нескольких узлах кластера, что может дать выигрыш в скорости, особенно на больших файлах.

Скорость чтения существенно зависит от используемого формата файла. Формат должен удовлетворять следующим требованиям: во-первых, размер файла должен быть как можно меньше, во-вторых, он должен обеспечивать относительно быстрый доступ к отдельным блокам данных, что важно в случаях, когда нет необходимости читать файл целиком, в-третьих, должен поддерживать распределенное хранение и читаться Spark. По сумме этих требований в качестве формата данных был выбран Apache Parquet. С одной стороны, он удовлетворяет приведенным требованиям, с другой — обладает достаточной гибкостью для формирования необходимой блочной структуры хранения данных. Непосредственно формат представляет собой набор столбцов, организованных в иерархическую структуру под управлением специальной схемы. Выбор схемы остается за тем, кто записывает файл. Сама схема является частью формата, записывается в метаданные файла и может быть восстановлена при чтении. Таким образом, файл Parquet может содержать иерархическую структуру высокой сложности. В качестве способа хранения данных на структурированной сетке была выбрана блочная структура, получающаяся из исходного индексного параллелепипеда элементов сетки его делением параллельными индексными плоскостями в трех направлениях. Каждый полученный таким образом блок пространственно занимает область в виде, в общем случае, криволинейного гексаэдра и представляет собой структурированную сетку. Хранение такого блока внутри Parquet-файла организовано в виде набора отдельных Parquet-столбцов для каждой из узловых координат и компонент поля. В силу специфики формата Parquet адреса столбцов хранятся в метаданных файла и доступ к каждому столбцу может быть осуществлен непосредственно без необходимости чтения всего файла, что решает задачу выборочного чтения только необходимых блоков сетки.

3. Пример использования

В данной работе рассматривается пример визуализации результатов нестационарного газодинамического расчета обтекания газовой средой твердого тела на структурированной сетке гексаэдров (см. рис. 2). Задача, рассматриваемая в примере, моделировалась в трехмерной постановке. На рисунке представлена картина распределения кинематического коэффициента турбулентной вязкости. Исходные данные записаны в виде временных слоев, каждый слой изначально сохранен в отдельном файле в формате Tecplot. Модель содержит около $5 \cdot 10^6$ узлов, каждый Tecplot-файл имеет размер 1,27 Гб.

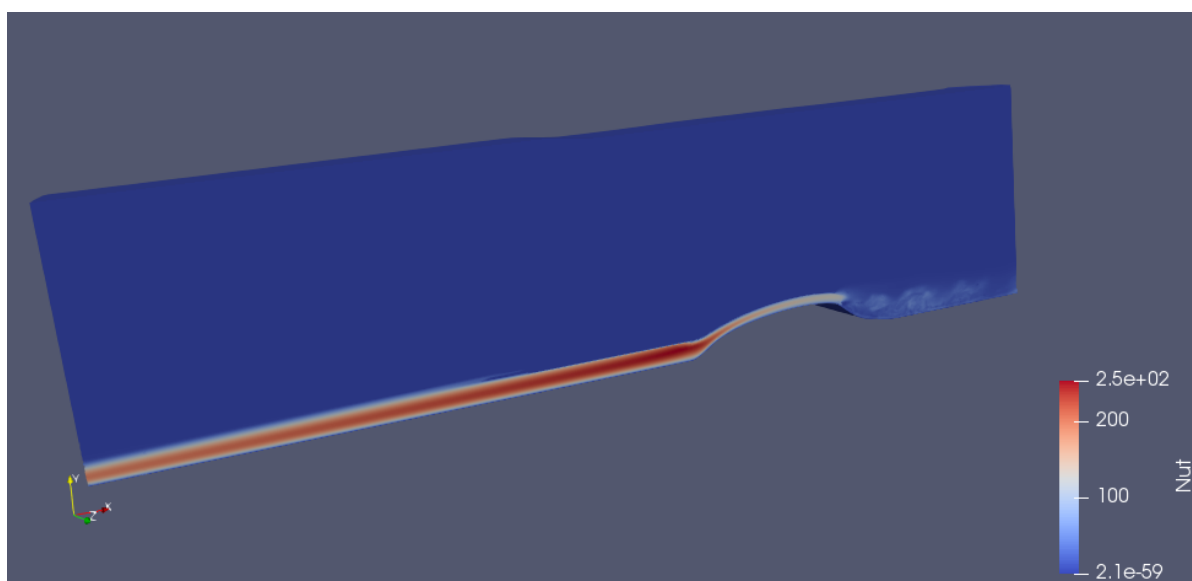


Рисунок 2. Кадр визуализации результатов трехмерной газодинамической задачи обтекания твердого тела, рассматриваемой в качестве примера. Представлена картина распределения кинематического коэффициента турбулентной вязкости.

При непосредственной визуализации такой последовательности кадров на персональном компьютере в ParaView загрузка одного кадра занимает около 30 секунд, что выступает большим неудобством для анализа результатов симуляции. Основная часть времени тратится именно на чтение файла.

Эти же данные визуализируются с помощью разрабатываемой среды. Все элементы среды, за исключением клиентского ParaView установлены на кластере. В качестве кластера использованы узлы суперкомпьютера "РСК Торнадо" Суперкомпьютерного Центра СПбПУ. Каждый узел состоит из двух CPU Intel Xeon E5-2697 v3 (14 ядер, 2,6 ГГц) и 64 ГБ оперативной памяти DDR4.

Файлы данных для визуализации в разрабатываемой среде были предварительно конвертированы в формат Parquet, при этом размер одного кадра уменьшился примерно до 400 Мб. Данные при записи в Parquet были преобразованы так, как было описано выше: исходный индексный параллелепипед структурированной сетки был разрезан взаимно ортогональными индексными плоскостями на параллелепипеды малого размера. Внутри каждого параллелепипеда каждая координата и каждая компонента поля — отдельный столбец Parquet. Такая структура эффективна при извлечении только выбранных блоков, так как не требует чтения всего файла.

Apache Spark запускается на кластере под управлением системы Slurm. Чтение Parquet-файлов в Spark выполняется с помощью библиотеки ruarrow, так как она обеспечивает большую скорость и требует существенно меньше памяти, чем встроенные в Spark средства чтения Parquet.

Таблица 1. Время чтения и отображения отдельных кадров на 8 узлах кластера "РСК Торнадо"

Кадр	Время чтения, с	Общее время, с	Кадров в секунду
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

Время отображения первых пяти кадров при использовании 8 узлов кластера представлено в таблице 1. Во втором столбце приведено время непосредственного чтения Parquet-файла в Spark с использованием ruarrow. Третий столбец — общее время отображения кадра. Последний столбец — это число кадров в секунду — величина, обратная общему времени. Основная часть времени тратится на передачу данных от сервера к клиентскому ParaView, из-за чего общее время отклика очень большое по сравнению с временем чтения. Это говорит о необходимости использования более быстрой сети. Увеличенное время на чтение и передачу первого кадра вызвано, в первую очередь, необходимостью чтения сетки. Дело в том, что сетка остается неизменной при переходе от кадра к кадру, поэтому читается только один раз. Начиная со второго кадра, система читает только значения отображаемого поля.

Несмотря на достаточно большое общее время отображения кадра, достигнуто значительное уменьшение времени чтения. Так при использовании разработанного программного комплекса время чтения первого кадра примерно в 6 раз меньше, чем время чтения с локальной машины. Для следующих кадров это отношение примерно равно 10.

Увеличение числа задействованных узлов не привело к ожидаемому уменьшению времени чтения данных в Spark. Поиск причины этого явления — одна из задач дальнейших исследований.

4. Выводы

Разрабатывается программная среда для обеспечения быстрой визуализации результатов моделирования задач гидроаэродинамики на больших сетках. Среда состоит из клиентского ParaView, серверного ParaView, сервера данных, переправляющего SQL запросы в Apache Spark. Последний используется для повышения скорости чтения данных большого размера за счет обеспечения к ним распределенного доступа.

Проведенные эксперименты показали эффективность использования данных в формате Parquet. По сравнению с текстовым форматом Tecplot этот формат обеспечивает меньший размер файла, непосредственно читается Apache Spark и предоставляет возможность извлечения отдельных блоков без необходимости чтения всего файла.

Разработанная программная среда удовлетворяет выдвинутым к ней требованиям. Она не требует от пользователя специализированных навыков в информационных технологиях. Все, что он должен сделать при использовании, — подключить плагин к ParaView и конвертировать исходные данные с помощью вспомогательных средств в формат Parquet. Также, использование программной среды значительно уменьшает время чтения файлов данных. При чтении первого кадра с использованием разработанного программного комплекса отношение затраченного времени в 6 раз ниже, чем при чтении данных с локальной машины. Для последующих кадров этот показатель примерно равен 10. Это различие для первого и остальных кадров объясняется тем, что при обработке первого кадра считываются расчетная сетка и отображаемое поле, тогда как при чтении остальных кадров читается только поле.

Также эксперименты показали отсутствие масштабируемости скорости чтения данных Spark при увеличении числа узлов и высокие накладные расходы на передачу данных от Spark к серверному ParaView, порождающие большое время отклика системы на действие пользователя. Эти проблемы — задачи для дальнейшей разработки.

Другая задача состоит в том, чтобы использовать более сложные SQL-запросы. Например, это могут быть запросы на получение данных, соответствующих видимой части модели. Кроме того, мог бы существовать запрос на получение данных, распределенных по слоям. Слой, данные которого следовало бы извлекать, зависит от позиции камеры. Если она соответствует более высокой детализации, то слой должен содержать большее количество узлов.

Благодарность

Авторы благодарят Российский Научный Фонд за поддержку, грант №18-11-00245.

Список литературы

1. Childs H., Brugger E., Bonnell K., Meredith J., Miller M., Whitlock B., Maxin 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., Marin 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>.

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

V.I. Reshetnikov¹, E.A. Golubchikov², A.V. Pyatlin³, A.K. Kuzin⁴, V.A. Kiev⁵,
N.N. Shabrov⁶, A.S. Zhuravlev⁷, E.K. Guseva⁸

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.

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., Cucchiatti F. M., Montes C. T., Vicente D., Calmet H., Mariin 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>.