

# Проблемы разработки средств визуализации для супервычислений

В.Л. Авербух, А.Ю. Байдалин, П.А. Васёв, Д.Ю. Горбашевский  
Д.Р. Исмагилов, А.Ю. Казанцев, Д.В. Манаков, А.Н. Шинкевич

В докладе рассматриваются проблемы, возникающие при разработке средств визуализации для задач, решаемых на современных параллельных вычислителях. Среди них - проблемы распараллеливания визуализации и параллельной фильтрации очень больших объемов данных, использования средств виртуальной реальности, разработка обеспечения on-line визуализации. Кроме того делается анализ возможностей систем визуализации программного обеспечения параллельных вычислений. Рассматриваются примеры разработок, проводимых в ИММ УрО РАН в рамках проекта системы интерактивной визуализации параллельных вычислений.

## 1. Введение

Двадцать лет назад в специальном выпуске журнала Computer Graphics был напечатан доклад «Visualization in Scientific Computing», где научная визуализация была заявлена как самостоятельная дисциплина. На тот момент источником больших объемов данных, требующих специальных усилий по визуализации, в первую очередь указывались супервычисления [1]. Сейчас супервычисления ведутся на параллельных компьютерах, производительность которых превосходит Крэи и Амдалы 80-х на много порядков. Развиваются распределенные вычисления на базе инфраструктуры Grid. Многократно возросли возможности аппаратуры компьютерной графики. В принципе вместе с возможностями техники должны измениться задачи и методы визуализации. При этом надо учитывать, что в понятие "визуализация" традиционно включаются как конкретные методики построения визуальных образов (или даже процесс построения конкретного визуального образа), так и когнитивные аспекты процесса формирования образов, представляющих модельные объекты. Кроме того, в связи с супервычислениями появляются новые задачи в области визуализации программного обеспечения, связанные с спецификацией очень сложного программного кода, также как и с отладкой правильности и производительности чрезвычайно сложных параллельных и распределенных программ.

Анализ литературы по визуализации, а также опыт нашего коллектива по разработке специализированных систем научной визуализации в задачах, решаемых на параллельных вычислителях, и средств визуализации для систем отладки и настройки производительности параллельных программ, позволяет выделить некоторые проблемы разработки средств визуализации для супервычислений.

## 2. Визуальные супервычисления

В работе группы британских исследователей «Visual Supercomputing: Technologies, Applications and Challenges» [2] дан обзор состояния дел в научной визуализации для супервычислений. В этой работе вводится понятие визуальных супервычислений, которое определяются следующим образом:

*Термин «Визуальные Супервычисления» относится к описанию инфраструктурных технологий для поддержки в сложных сетевых вычислительных средах визуальных и интерактивных вычислений вообще, и визуализации в частности.*

Отметим, что, хотя и сам термин и его определение имеют определенные недостатки, но в тексте статьи понятия, связанные с визуализацией супервычислений, рассматриваются

достаточно подробно и четко. Так что их вполне можно использовать в качестве рабочих.

Среди заявленных к изучению в рамках этого нового направления вопросов мы обратим внимание на следующие, связанные как с технологиями визуализации, так и с когнитивными проблемами:

- Распараллеливание процесса формирования и генерации изображений;
- Интерактивная визуализация;
- Визуализация программного обеспечения, когнитивные аспекты визуализации параллельных вычислений;
- Виртуальный испытательный стенд, как интегрированная система on-line визуализации параллельных вычислений.

### 3. Подходы к распараллеливанию визуализации

Визуализационный конвейер рассматривается как последовательность, включающая следующие процессы:

- фильтрация, обработка и восполнение «сырых» данных, полученных в результате моделирования,
- геометрическая обработка (мэппирование), то есть отображение данных в геометрические примитивы различного уровня;
- рендеринг, то есть непосредственное получение двумерного изображения на базе имеющихся геометрических описаний.

Со времен создания первых графических систем стала традиционной схема «большая ЭВМ - графическая рабочая станция». При этой схеме счет осуществлялся на достаточно мощном (естественно по тем временам) компьютере, а так или иначе преобразованные в графическую форму данные передавались на мини-ЭВМ, к которой и подключались графические устройства. При создании параллельных вычислителей эта схема во многих случаях также имеет место. Вот как описана в [3] организация работы на параллельном вычислителе ASCII Red, установленном в Sandia National Laboratories. Обработка состоит из следующих этапов:

1. Модельный счет на параллельном мультипроцессорном вычислителе;
2. Передача данных на сервер визуализации;
3. Извлечение изоповерхностей и полигональная децимация;
4. Объединение файлов, содержащих полигоны;
5. Изучение изоповерхностей в реальном времени.

Правда, в данном случае (а это вторая половина 90-х) сервер визуализации представляет собой не маломощную мини-ЭВМ, а 16 процессорную рабочую станцию ONYX II производства Silicon Graphics. Сам параллельный вычислитель ASCII Red состоит из примерно 9000 процессоров, а его пиковая производительность - триллион операций с плавающей запятой в секунду. Таким образом, здесь все фазы визуализационного конвейера были переданы на сервер визуализации. Однако, возможны и другие подходы к реализации конвейера на параллельных системах. При этом следует оценивать объемы передаваемых для визуализации данных. Так в [3] упоминается о 350 Гбайтах при трехдневном счете на 2500 процессорах вычислителя. На современном вычислителе Blue Gene/L (131072 процессоров) называются уже совсем заоблачные цифры в тысячи Тбайтов. На менее мощных системах при меньших объемах данных менее мощными оказываются и средства передачи. Поэтому правильным представляется по крайней мере часть элементов визуализационного конвейера размещать на параллельных процессорах основного вычислителя. В работах [4], [5] рассмотрено понятие параллельной фильтрации. В качестве фильтров могут использоваться различные методики отбора данных, например, сечений, построения поверхностей уровня, а также сжатия [6]. Для визуализации могут передаваться отфильтрованные математические данные, связанные с видом отображения, а не графические примитивы или растровое изображение. Это позволяет воспользоваться аппаратными возможностями видеокарт и в ряде случаев сократить объем передаваемых данных. Преимущества применения параллельной

фильтрации данных на счетных узлах вычислителя сейчас для многих очевидны (см., например, [7]). Среди них:

- обрабатываемые для визуализации данные уже находятся на узлах вычислителя, что позволяет избежать дополнительных обменов;
- при визуализации можно учесть характерные особенности задачи;
- возможна on-line визуализация большого объема данных;
- увеличивается скорость визуализации.

Вместе с тем имеется ряд технических трудностей при реализации этого подхода. Кроме того, среди недостатков может оказаться необходимость приостановки основной счетной задачи на время предобработки. Решение в этом случае может быть связано с выделением части процессоров вычислителя для организации графической обработки по ходу работы или использованием только лишь «посмертной» визуализации большого счета. Тогда естественно использовать алгоритмы распараллеливания рендеринга, примеров которых существует огромное множество. (См., в частности, доступный через интернет сравнительно старый учебный курс [8] или более свежую монографию [9].) Одной из проблем может стать передача больших объемов графических данных на графические устройства. Чтобы уменьшить нагрузку на каналы связи можно генерировать не растр картинка, а некоторый графический метафайл, разбор которого занимается сервер визуализации [10]. Это решение позволяет также использовать богатые возможности современной аппаратуры графики по скоростной генерации изображений.

В рамках среды визуализации ParaView [11], которая способна перерабатывать огромные объемы данных, предусмотрено несколько режимов рендеринга с различным распределением этой задачи по параллельному вычислителю, серверам визуализации и графическим устройствам, включая среды виртуальной реальности. Таким образом в полной мере достигается использование особенностей имеющейся в распоряжении пользователей аппаратуры.

## 4. Интерактивная визуализация

Вывод графики по ходу счета задачи с возможностью вмешиваться в вычислительный процесс всегда был мощным инструментом моделирования, особенно на начальных этапах разработки модели. Осуществление этой возможности на параллельных ЭВМ без специальных средств невозможно. Поэтому необходима разработка специальных сред интерактивной визуализации и управления ходом вычислений.

Одним из примеров таких систем является система CUMULVS [12]. Система CUMULVS, созданная в конце 90-х годов, предлагала on-line визуализацию, управление ходом вычислений и поддержку контрольных точек. В состав системы входил модуль для известной универсальной системы визуализации AVS - пользователю остается модифицировать свою параллельную программу и описать виды отображения для AVS. Для внедрения механизма on-line визуализации программист должен для визуализируемого распределенного массива данных выполнить вызов в параллельной программе, в котором указать его логическое имя, тип данных, размерности, декомпозицию по процессорам и указатель на данные текущего процессора. После этого системе AVS необходимо указать на основе каких массивов и как строить визуальное представление.

В реализованной также в 90-х годах системе VIPER [13] on-line визуализация основана на специальном механизме. Отфильтрованные данные, взятые из соответствующих структур данных по ходу параллельного процесса моделирования, передаются в модули визуализации, что осуществляется за счет точек синхронизации. Каждый объект структуры данных (скаляр, матрица, решетка и т.п.) может быть ассоциирован с точкой синхронизации посредством специальных описаний, внедренных системой в программный код. При обработке во время счета параллельных модулей точек синхронизации посылается специальный сигнал в систему, которой известно, каким модельным сущностям (например, скоростью, температурой, плотностью) соответствуют данные структуры данных. Виды отображения, реализованные в визуализационных модулях, обеспечивают конкретный графический вывод. В рамках системы

реализовано также управление ходом вычислительного процесса.

Отметим, что в чем-то схожий механизм on-line визуализации реализован в системе отладки эффективности параллельных программ AIMS [14]. Вообще в целом ряде систем отладки правильности и настройки эффективности параллельных программ реализованы те или иные средства, обеспечивающие интерактивную работу с программным кодом. Однако, чаще подобные системы используются в режиме «посмертной» визуализации данных, что вызвано проблемами реализации и функционирования этих сложных программных комплексов на параллельных вычислителях.

## **5. Когнитивные аспекты визуализации параллельных вычислений**

Относительно недавно нами были подготовлены обзоры состояния дел в визуализации программного обеспечения параллельных вычислений [15], [16]. Материалы по этой тематике находятся постоянно в сфере нашего внимания. Что же показывает изучение этого важного вопроса.

Разработки языков визуального программирования, отладчиков правильности и эффективности для параллельных вычислений начались уже в 80-х годах. За более чем три десятилетия созданы сотни систем в области визуализации программного обеспечения параллельных вычислений, ежегодно появляются новые интересные решения. Вместе с тем средств визуализации, являющихся существенно большим, чем вспомогательным инструментом, практически нет. Причина этого вряд ли заключается лишь в реальных сложностях обнаружения ошибок и неэффективности, присущих параллельным программам, или в серьезных ограничениях, которые возникают при работе с современным программным инструментарием параллельного программирования. Думается, что несоответствие между затраченными усилиями и сравнительно незначительным результатом происходит также из-за недостаточного внимания к когнитивным (то есть связанным с проблематикой мышления и восприятия пользователя) аспектам разработки систем визуализации.

Разработка видов отображения, то есть спецификации способов представления модельных объектов и методов взаимодействия с ними, является существенной частью проектирования систем визуализации.

В системах отладки правильности пользователь может воспользоваться несколькими видами отображения, как-то представляющими структуру параллельного вычислителя, и доступ к традиционной («текстовой») отладке единичного процесса. В системах отладки и настройки эффективности, как правило, все построено на анализе статистики выполнения процессов. Интересно, что даже использование средств виртуальной реальности сводится к выводу метрик производительности в виде обычных графиков. Существуют, конечно, примеры и других видов отображения, в частности, представление очередей обработки сообщений, распределенных массивов в отладчиках правильности или использование графов вызовов для анализа производительности программ.

Однако, складывается впечатление, что зачастую в конкретных системах визуализации программного обеспечения репертуар видов отображения ограничен и диктует пользователю сложные и не прямые методы анализа программных сущностей. Можно вспомнить, тот факт, что во многих системах предусмотрены возможности конструирования видов отображения самим пользователем. Такие возможности полезны, но похоже, что проектировщики таким образом просто перекалывают нерешенную задачу на плечи пользователя. Нужна специальная разработка видов отображения, базирующаяся на определенной теории визуализации [17]. Для правильного и эффективного визуального представления необходимо определение этих сущностей как в плане того, какой именно объект подлежит изучению (то есть в понятийном плане), так и в плане описания этого объекта на уровне компьютерной реализации. Необходимо также четко уяснить какие именно состояния и особенности данного объекта нас интересуют, так как представление особенностей, состояний и смены состояний и есть основная задача визуализации. Следующий вопрос - какими графическими и неграфическими средствами сам объект, его атрибуты и особенности можно и нужно представить, как следует визуально выделить элементы особого интереса данной сущности.

Можно говорить и об ограниченности репертуара в системах научной визуализации. Существующие универсальные средства поддержки визуализации не всегда могут удовлетворить потребности пользователей в изучении новых модельных сущностей, обладающих не до конца ясными свойствами, или требуют от пользователя для представления этих сущностей слишком больших усилий. Следовательно, необходима разработка специализированных систем визуализации, при проектировании которых максимально учитываются нужды конкретных пользователей, решающих конкретный класс задач. Одной из важнейших задач проектирования является обеспечение прагматических свойств систем визуализации, которые должны оцениваться с субъективных, пользовательских позиций.

Особенно важной стала эта задача при активном использовании сред виртуальной реальности, получивших сейчас широкое распространение. Многие проектировщики увлечены блестящими возможностями этих сред по представлению реалистической графики и погружению пользователей в виртуальный мир. Однако, надо помнить, что в средах, обеспечивающих глубокое погружение, пользователь должен не только (и не столько) разглядывать красивые картинки, а решать сложные задачи по анализу и интерпретации информации и взаимодействовать в реальном времени с моделью. В этой связи необходимы серьезные исследования восприятия сложной графики и поиск способов представления. Поиск новых видов отображения необходим и в тех случаях систем научной визуализации, когда традиция предлагает нам набор привычных решений.

В любом случае в этой области больше проблем, чем решений. Наши усилия в этом направлении только лишь открывают новые и новые задачи.

## **6. Виртуальный испытательный стенд**

В ИММ УрО РАН разработка системы интерактивной визуализации параллельных вычислений идет в рамках идеи виртуального испытательного стенда, служащего для поддержки вычислительных экспериментов. Современный вычислительный (компьютерный) эксперимент является составной частью исследований в различных областях науки. Его проведение предполагает реализацию математической модели, создание вычислительных алгоритмов, реализацию счета на (как правило) параллельных вычислителях, визуализацию как результатов счета, так и модельных объектов различных уровней. Виртуальный испытательный стенд может представлять собой программно-аппаратный комплекс, который должен включать как реализацию на параллельном вычислителе математической модели, так и специализированные аппаратные и программные средства визуализации, поддерживающие, в том числе, элементы виртуальной реальности. Кроме этого, в программный комплекс должны войти системные средства организации счета, запуска (и перезапуска) программ, смены характеристик эксперимента и пр.

В связи с реализацией этой идеи работа идет по нескольким направлениям.

Во-первых, потенциальные пользователи нашей системы значительное внимание уделяют визуализации больших и очень больших сеток сложной структуры. Поэтому первоочередные задачи связаны с созданием средств визуализации этих сеток. Начиная с объема сеток порядка миллиона и более точек, уже нельзя осуществить их графический вывод на единичном процессоре. Поэтому естественно встает задача распараллеливания фильтрации, предобработки визуальной информации и возможно распараллеливания непосредственного рендеринга сеток.

Во-вторых, необходима разработка системы видов отображения для системы визуализации и, конкретно, для визуализации сеток. В нашу задачу входит и проектирование видов отображения на базе средств виртуальной реальности. Причем виртуальная реальность может использоваться и для вывода графики и для организации взаимодействия с моделирующей программой и управления процессом счета.

В-третьих, в рамках проектирования виртуального испытательного стенда необходима поддержка on-line визуализации параллельных вычислений, организация взаимодействия с задачей, считающейся на параллельном вычислителе.

Кроме того, продолжают разработки средств визуализации программного обеспечения в

рамках системы параллельного программирования DVM. В этой связи ведутся исследования проблематики проектирования видов отображения для представления сущностей, связанных с отладкой правильности и настройкой производительности параллельных программ.

Параллельная фильтрация, понимаемая в частности, как предобработка данных при визуализации параллельных вычислений, нацелена на минимизацию передаваемой для визуализации информации. Исследования в области параллельной фильтрации [18], [19] позволяют проанализировать как технологии фильтрации, так и реализацию конкретных параллельных фильтров. Ясно, что параллельная фильтрация необходима, во-первых, при визуализации данных большого и очень большого объема, а, во-вторых, когда требуется активное взаимодействие с программой по ходу визуализации. Как правило, фильтр быстрая операция, и распараллеливание самих алгоритмов фильтрации может не принести ощутимой пользы. Однако при интерактивной визуализации оно оправдано для обеспечения приемлемого времени ответа.

Конкретные работы, прежде всего, связаны с распараллеливанием вывода сеток.

Общая схема системы параллельной визуализации предполагает, что сами сетки (координаты узлов, значения параметров в узлах, данные о многогранниках) естественно располагаются на процессорах параллельного вычислителя. Фильтрация узлов сетки (пространственная и по значению) частично также проводится на параллельном вычислителе, а частично вынесена на графическую станцию (возможно достаточно мощную ПЭВМ с хорошей видеокартой). Там же проходит непосредственное отображение сетки, в том числе и вывод воксельного представления сетки.

В схему распараллеливания входят такие взаимодействующие блоки как входной модуль, ядро обработки и визуализатор. Задача входного модуля заключается в считывании данных и преобразует их к единому внутреннему формату. Затем данные передаются ядру, которое распределяет их по процессорам. Визуализатор получает от вычислителей через ядро набор графических примитивов для отображения (набор отрезков в случае каркасного отображения сетки). Кроме того, он получает от пользователя информацию об отображаемом участке сетки и передает ее через ядро на конкретные процессоры.

Примитивы визуализации (элементы сетки, например, тетраэдр, куб) распределяются по процессорам равномерно по общему количеству ребер. Существует необходимость удалять продублированные ребра смежных примитивов, попавших на разные вычислители.

Пространственная фильтрация и фильтрация по значению реализована за счет набора видов отображения. (См. Рис. 2-5. в конце статьи.) Однако необходим систематический подход к их разработке, предусматривающий дополнительное исследование запросов потенциальных пользователей. Это особенно важно для разработки видов отображения с использованием технологий виртуальной реальности.

В настоящее время нами реализованы примеры визуализации сеток с использованием специальных очков. Причем простые манипуляции с изображением осуществляются за счет движений пальцев, одетых в перчатку. Данное направление весьма перспективно и требует дальнейших исследований и опытных разработок.

В плане создания средств интерактивной визуализации нами было реализовано несколько макетных систем, ([20-22]. Затем был разработан макетный вариант универсальной транспортной подсистемы, которая базируется на некоторой модели, определяющей распределение и связь. Транспортная подсистема является сохранной асинхронной системой обмена сообщениями (неделимыми единицами обмена). Данные в сообщении - произвольные. Процессы-участники посылают друг другу сообщения через очереди сообщений. Сообщения очереди гарантированно доставляются всем подписавшимся участникам в порядке отправки с учетом приоритетов сообщений и приоритетов отправителей [23].

Однако в дальнейшем основные усилия были направлены на разработку специализированных средств, обеспечивающих интерактивную визуализацию отдельных параллельных программ. Данный подход связан с необходимостью накопления опыта разработки соответствующих механизмов обмена. На базе этих средств созданы интерактивные

версии нескольких параллельных программ. За счет внедрения on-line визуализации и управления процессом счета удастся ускорить и упростить процесс разработки параллельных программных комплексов. В ближайшей перспективе намечено создание интерактивного варианта некоторых средств визуализации для системы программного программирования DVM. Предполагается, также разработка давно востребованных элементов систем отладки правильности параллельных программ таких, как трассировщик MPI-программ.

Проводятся также исследования и разработка видов отображения, служащих для представления данных о производительности параллельных систем. Одно из направлений связано с представлением сущностей на базе метафоры комнаты. В качестве объекта предметной области, представляемого метафорой комнаты при параллельном программировании, естественным образом напрашивается процесс. То есть рассматривается совокупность комнат как параллельных процессов, а передаваемые сообщения или данные описываются связями между комнатами. Возможно и применение этой метафоры для анимационного описания динамики программы. С помощью метафоры комнаты, пользуясь свойством контейнера, можно представлять составные структуры данных программы, такие как массивы, структуры, объекты. Возможно, большой эффект от визуального представления данных будет достигнут в системах с общей памятью. Здесь также напрашивается рассмотрение данных в динамике, что дает возможность изучить, как и (что также немаловажно) где меняются данные. Это упрощает, например, поиск мест программы, где обрабатывается и/или пересылается большое количество данных, на основании такой информации могут быть сделаны выводы об эффективности ее работы. С помощью метафоры комнаты можно также отобразить граф вызовов функций (см. Рис.1). По такому отображению можно судить о «слабых» в смысле эффективности работы местах программы. Можно обнаружить такие важные особенности, как часто вызываемые функции; функции, в которых программа «проводит» значительное время; функции, ответственные за передачу данных между процессами, etc. Другим примером применения этой метафоры является визуализация иерархии классов программы. Иерархия классов очень важна на этапе проектирования программы. От удачного ее построения во многом зависит удобство написания и внесения изменения в программу, что в итоге скажется и на эффективности. В качестве объекта предметной области, который представляет комната, рассмотрим класс. Содержимое класса (данные и методы) будет содержимым комнаты, а отношения наследования/агрегации - связями между комнатами. [24]. Подобные идеи высказывались в [25].

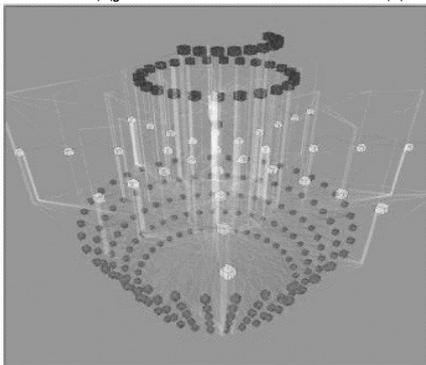


Рис 1. граф вызовов функций программы, осуществляющей рисование графа вызовов. рисунок представлен в черно-белом варианте

## 6. Заключение

Проблемы, возникающие при создании средств визуализации для супервычислений, требуют комплексного решения в связи исследованиями в области визуализации и параллельных вычислений. Разработка системы «виртуальный испытательный стенд» даже на начальном этапе потребовала значительных усилий по ряду так или иначе связанных между

собой направлений. Ясно, что большее внимание следует уделять когнитивным аспектам проектирования визуализации и учету особенностей пользователей супервычислений. Наши дальнейшие исследования направлены как на выявление теоретических аспектов проблем визуализации параллельных вычислений, так и на подбор реальных методик разработки программных сред.

## Литература

1. Visualization in Scientific Computing, Special Issue, ACM SIGGRAPH Computer Graphics, V. 21, N 6, November 1987.
2. Brodlie K., Brooke J., Chen Min, Chisnall D., Fewings A., Hughes Ch., John N.W., Jones M. W., Riding M., Roard N. Visual Supercomputing: Technologies, Applications and Challenges // COMPUTER GRAPHICS forum, Volume 24 (2005), N 2, pp. 217-245.
3. Heermann Ph. D. Production Visualization for the ASCI One TeraFLOPS Machine // Proceedings of the 9th Annual IEEE Conference on Visualization (VIS-98), Oct 18-23 1998, ACM Press, New York, 1998, pp. 459-482.
4. Manakov D., Mukhachev A., Shinkevich A. Visualization of the Distributed Data of Huge Volume. Assembly, Filtration, Sorting. // Proceedings of the 13-th International Conference on Computer Graphics and Vision Graphicon-2003 Moscow, September 5-10, 2003. pp. 198-201.
5. Горбашевский Д.Ю., Казанцев А.Ю., Манаков Д.В. Параллельная фильтрация в системе визуализации параллельных вычислений // ГрафиКон'2006, 1-5 июля 2006. Россия. Новосибирск, Академгородок. Труды Конференции. Новосибирск Институт Вычислительной математике и математической геофизике. 2006. Стр. 333-336.
6. Кринов П.С., Якобовский М.В., Муравьев С.В. Визуализация данных большого объема в распределенных многопроцессорных системах // Высокопроизводительные параллельные вычисления на кластерных системах /Материалы третьего Международного научно-практического семинара 13-15 ноября 2003 г. Издательство Нижегородского государственного университета Нижний Новгород, стр. 84-92.
7. Карпов А. Н. Визуализация данных на параллельных вычислительных комплексах // 15-я Международная конференция по компьютерной графике и ее приложениям ГрафиКон'2005 20-14 июня 2005, Россия, Новосибирск, Академгородок Труды Конференции. Новосибирск Институт Вычислительной математике и математической геофизике. Стр. 211-215.
8. Chalmers A., Reinhard E. Parallel and Distributed Photo-Realistic Rendering (course notes). ACM SIGGRAPH. July 1998 [<http://www.cs.bris.ac.uk/Publications/Papers/1000268.pdf>]
9. Chalmers A., Davis T., Reinhard E. (editors) Practical Parallel Rendering. AKPeters, 2002.
10. Manakov D., Komarovskiy I. Principles of Automatic Generation And Parsing of Device-Independent Metfiles by the Example of Object-Oriented VTK Library Adaptation // Proceedings of the 13-th International Conference on Computer Graphics and Vision Graphicon-2003. Moscow, September 5-10, 2003. pp. 196-197.
11. Cedilnik A., Geveci B., Moreland K., Ahrens J., Favre J. Remote Large Data Visualization in the ParaView // Eurographics Symposium on Parallel Graphics and Visualization 2006, pp. 163-170.
12. Geist G.A., Kohl J.A., Papadopoulos P.M. CUMULVS: Providing Fault-Tolerance, Visualization and Steering of Parallel Applications // International Journal of High Performance Computing Applications, Volume 11, Number 3, August 1997, pp. 224-236.
13. Rathmayer S., Lenke M. A Tool for On-line Visualization and Interactive Steering of Parallel HPC Applications // Proceedings of the 11th IPPS'97, 1997, pp. 181-186.

14. Yan J., Sarukkai S., Mehra P. Performance Measurement, Visualisation and Modeling of Parallel And Distributed Programs using AIMS Toolkit // Software - Practice and Experience. V. 25, N 4 (April 1995) pp. 429-461.
15. Авербух В.Л., Байдалин А.Ю. Разработка средств визуализации программного обеспечения параллельных вычислений. Визуальное программирование и визуальная отладка параллельных программ // Вопросы атомной науки и техники. Сер Математическое моделирование физических процессов. 2003. Вып. 4. С. 68--80.
16. Авербух В.Л., Байдалин А.Ю. Разработка средств визуализации программного обеспечения параллельных вычислений. Оптимизация программ // Вопросы атомной науки и техники. Сер. Мат. моделирование физических процессов. 2004. вып. 1. С. 70-80.
17. Авербух В.Л. К теории компьютерной визуализации // Вычислительные технологии Т. 10, N 4, 2005 , стр 21-51.
18. Manakov D., Mukhachev A., Shinkevich A. Visualization of the Distributed Data of Huge Volume. Assembly, Filtration, Sorting. // Proceedings of the 13-th International Conference on Computer Graphics and Vision Graphicon-2003 Moscow, September 5-10, 2003. pp. 198-201.
19. Горбашевский Д.Ю., Казанцев А.Ю., Манаков Д.В. Параллельная фильтрация в системе визуализации параллельных вычислений // ГрафиКон'2006, 1-5 июля 2006. Россия. Новосибирск, Академгородок. Труды Конференции. Новосибирск Институт Вычислительной математике и математической геофизике. 2006. Стр. 333-336.
20. Авербух В.Л., Исмагилов Т.Р., Манаков Д.В. Подходы к реализации интерактивной графики на МВС-1000 // Материалы Всероссийской научной конференции «Высокопроизводительные вычисления и их приложения». Черноголовка. Подмосковный филиал МГУ им. М.В. Ломоносова, Институт Проблем Химической Физики РАН, 2000. Стр. 241-244.
21. Манаков Д.В. Шагубаков М.Р. Адаптивный конструктор для интерактивных задач для масс-параллельных машинах // Труды 12 Международной Конференции по Компьютерной Графике и Машинному Зрению ГрафиКон'2002. Нижний Новгород, 16-21 сентября 2002 г. Нижегородский Гос. Университет. Нижний Новгород, 2002. Стр. 405-408.
22. Авербух В.Л., Манаков Д.В, Васёв П.А., Комаровский И.А., Мухачёв А.А., Шинкевич А.Н. Подходы к реализации средств on-line визуализации параллельных вычислений // «Супервычисления и математическое моделирование» Тезисы международного семинара. Саров, ВНИИЭФ-РФЯЦ, 2003. Стр. 14-16
23. Васёв П.А. Разработка механизма транспортной подсистемы для интерактивной визуализации параллельных вычислений // Тезисы Международного семинара «Супервычисления и математическое моделирование» Саров, РФЯЦ ВНИИЭФ, 2004. Стр. 22-23.
24. Байдалин А.Ю., Исмагилов Д.Р. Средства представления структур в системах визуализации программного обеспечения // ГрафиКон'2006, 1-5 июля 2006. Россия. Новосибирск, Академгородок. Труды Конференции. Новосибирск Институт Вычислительной математике и математической геофизике. 2006. Стр. 271-274.
25. Alfert K., Fronk A., Engelen F. Experiences in 3-Dimensional Visualization of Java Classes Relations // SDPS Journal of Design Process Science, V.5, № 3, pp. 91-106, September 2001.

## Рисунки

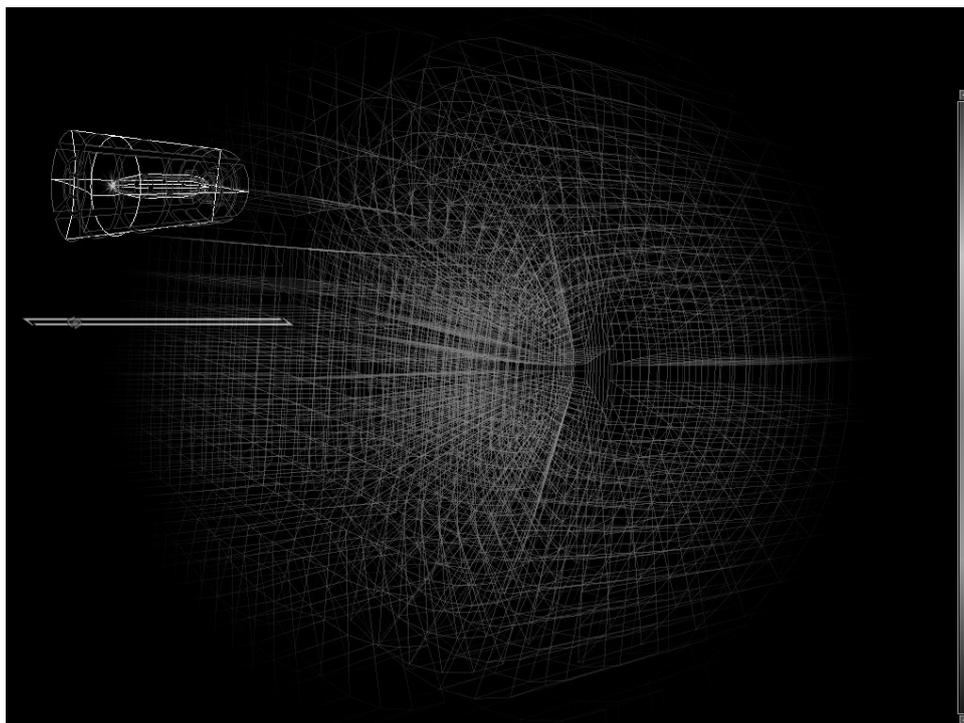


Рис 2. Визуализация сетки порядка  $10^4$  примитивов

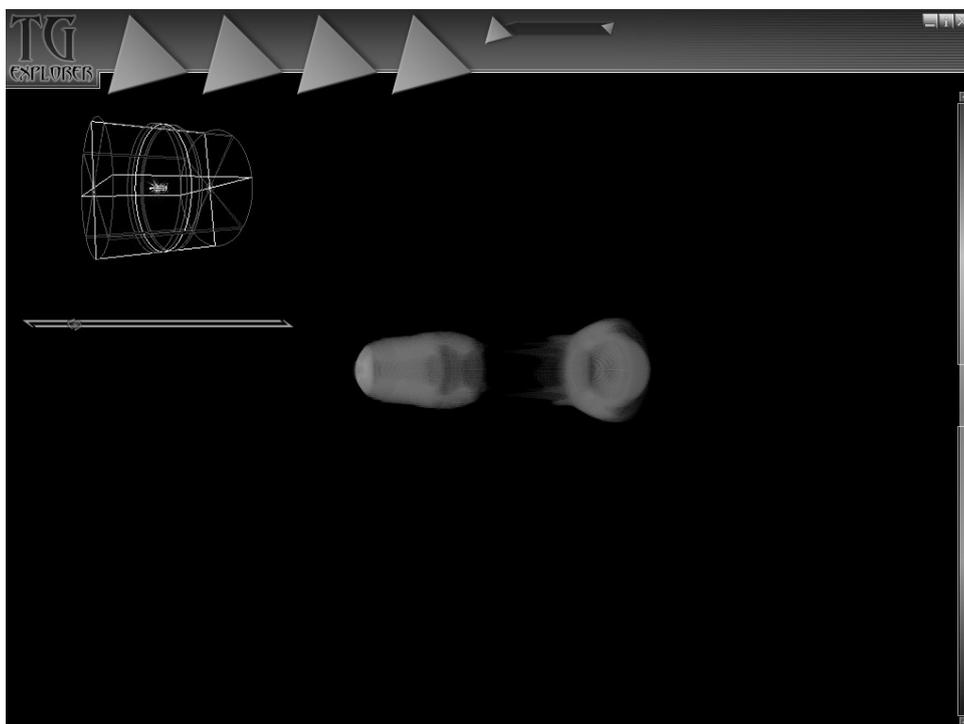


Рис 3. Визуализация сетки порядка  $10^5$  примитивов.

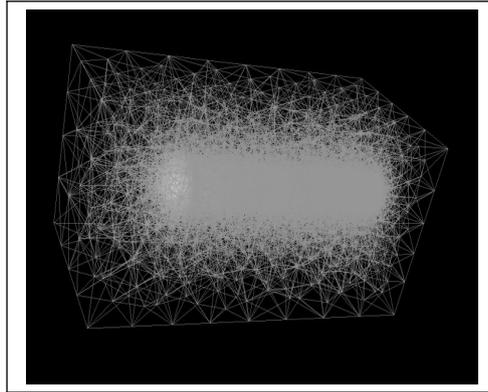


Рис. 4 Тетраэдральная структура сетки

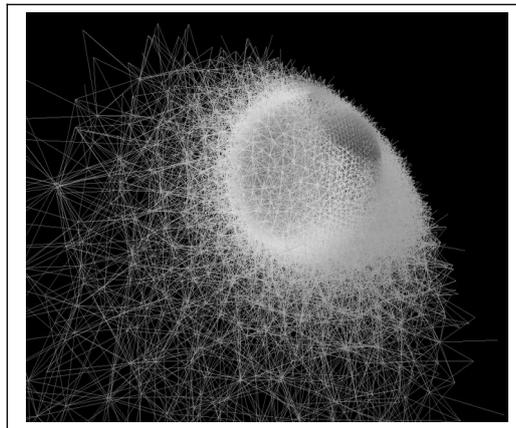


Рис 5. Визуализация сетки порядка  $10^5$  узлов