

Эффективность использования вычислительных ресурсов в процессах распределенной визуализации трехмерных моделей

Александр Панасюк, Дмитрий Астриков, Дмитрий Кузьмин

Институт космических и информационных технологий, Сибирский Федеральный Университет

panasmeister@gmail.com, astrikov.d@gmail.com, dkuzmin@sfu-kras.ru

Аннотация. *Распределенная визуализация трехмерных моделей является ресурсоемким процессом, возлагающим высокие требования к производительности вычислительной системы. Поэтому вопрос о решении проблемы управления этим процессом на высокопроизводительных ресурсах стоит очень остро. Особенно это касается так называемых рендерферм – сервисов, предоставляющих вычислительные ресурсы для визуализации трехмерных моделей, и облачных рендерферм, предлагающих те же возможности по запросу. В работе представлено исследование возможностей распределенной обработки данных в популярных программных пакетах САПР и визуализации (3DsMAX, VRay, MentalRay), а так же описание экспериментальной системы управления распределенным рендерингом. Результаты работы легли в основу создания облачной рендерфермы работающей на суперкомпьютерных ресурсах..*

Ключевые слова

Распределенная визуализация, облачные сервисы, SaaS, рендеринг, рендерферма, компьютерная графика, трехмерная графика, высокопроизводительные комплексы, суперкомпьютер.

1 Введение

На сегодняшний день компьютерная графика (КГ) занимает важное место во многих областях деятельности человека. Средствами КГ пользуются дизайнеры, архитекторы, ученые, СМИ (телевидение, печатные и Интернет-издания). Стоит так же отметить, что современная кино-индустрия чрезвычайно редко отказывается от использования трехмерного (3D) моделирования и визуальных эффектов в своих продуктах.

Дизайнеры, архитекторы, аниматоры, создавая свои проекты, тратят значительное количество времени и вычислительных ресурсов на осуществление визуализации созданного проекта, так называемый рендеринг. Так например обработка (визуализация) 10-минутного видеоролика в высоком разрешении (1920x1080 пикселей), в зависимости от сложности сцены, может выполняться от нескольких дней до нескольких недель (на среднем по мощности домашнем ПК). При этом если дизайнер вносит изменения в свой проект, то операция визуализации должна повториться. Соответственно задача 3D-рендеринга (3D-визуализации) является одной из самых ресурсоемких задач, которая требует больших вычислительных затрат и, что не мало важно, соответствующего специализированного программного обеспечения [1]. Это приводит к тому, что компании, занимающиеся созданием таких 3D проектов должны обладать мощными центрами обработки данных (ЦОД) и лицензиями на специализированное ПО для рендеринга. Зачастую они просто не могут себе позволить закупку и содержание таких ресурсов.

Одним из путей, позволяющих преодолеть ресурсные проблемы, является создание специализированных рендер-ферм аккумулирующих в себе мощные вычислительные ресурсы и специализированное программное обеспечение, предоставляемые в виде облачного сервиса (SaaS — от англ. Software as a Service — ПО как услуга). Такой подход позволяет существенно сократить время рендеринга и элиминировать затраты, связанные с приобретением и обслуживанием вычислительного комплекса.

Создание облачных сервисов рендеринга на базе высокопроизводительных комплексов учебных заведений позволит задействовать свободные ресурсы, для решения прикладных задач 3D-визуализации [2].

Для проведения исследований на суперкомпьютерном комплексе ИКИТ СФУ [3] была построена экспериментальная вычислительная система, позволяющая задавать любые конфигурации процессов рендеринга, а так же оперативно получать сведения об их работе. В основу работы легли данные, полученные в результате эксплуатации системы.[1]

Стоит так же отметить представленная система стала прототипом облачного сервиса трехмерной визуализации „Рендермама“ (<http://rendermama.com/>)

В работе представлен результат анализа конфигураций процессов распределенной визуализации трехмерных моделей.

2 Экспериментальная вычислительная система распределенной визуализации

Облачный сервис трехмерной визуализации создается на базе суперкомпьютерного центра Сибирского федерального университета. В составе центра 3 высокопроизводительные системы, объединяющие 280 счетных вычислительных серверов, SAN-сеть 8 Гбит/с, InfiniBand 20 Гбит/с. Гибридная составляющая комплекса — платформа NextIO Vcore Express с 4 GPU Tesla M2090.[4]

В настоящий момент, в лаборатории высокопроизводительных вычислений СФУ создана системы управления распределенной визуализацией (СУРВ), которая используется для создания сервиса распределенной визуализации.

2.1 Архитектура системы управления

Решение, лежащее в основе СУРВ, позволяет построить сервис визуализации на суперкомпьютерных ресурсах с минимальными временными, трудовыми и финансовыми затратами. Разработанная система предоставляет интерфейс для отправки 3D сцен на вычислительный кластер, постановки задач в очередь, мониторинга прогресса их выполнения и выдачи результатов обработки сцен пользователю. Для управления ресурсами кластера используется свободно распространяемое ПО: Torque PBS (Torque, Maui).

С точки зрения системы управления комплексом, задачи рендеринга — это процессы выполняющиеся от имени специального пользователя. Для решения этих задач выделяются специальные очереди, используемые только СУРВ.

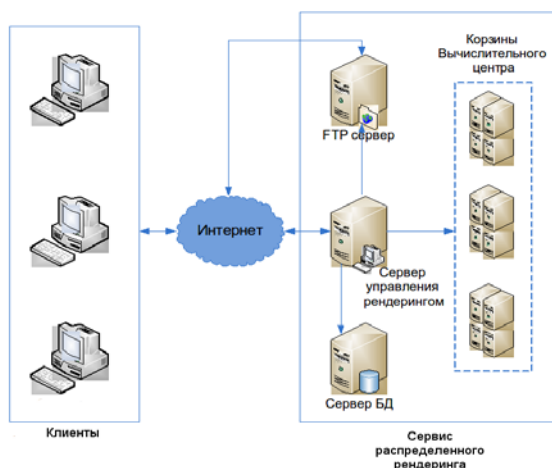


Диаграмма 1. Структура системы управления рендерингом.

Структура СУРВ включает в себя (Диаграмма 1):

- Сервер управления рендерингом;
- Сервер баз данных (MySQL);

- Сервер управления очередью заданий (Torque);
- Файловый сервер (FTP);
- Сетевая файловая система (NFS);
- Вычислительные узлы.

Сервер управления рендерингом отвечает за прием (через FTP) и проверку входных данных (архив с настройками рендеринга, трехмерными сценами и текстурами), создание и постановку задач в очередь обработки Torque, слежение за задачами и ресурсами кластера и выдачу результатов рендеринга (через FTP). Для выполнения этих функций разработан web-интерфейс и REST API.

Сервер управления рендерингом тесно связан с сервером управления очередью заданий. Первый обладает необходимым функционалом для постановки задач в очередь и отслеживания их состояния, а также, с помощью последнего, производит управление и мониторинг состояния вычислительных узлов и очередей заданий. Таким образом сервис может предоставлять конечным пользователям виртуализированные ресурсы, для клиента создается иллюзия неограниченного количества ресурсов и он может получить доступ к ним по запросу. Кроме того, Torque может управлять нагрузкой на вычислительные узлы (в той мере, в которой это необходимо при выполнении задач распределенной визуализации) без вмешательства администратора. Это подход к предоставлению ресурсов является обязательным атрибутом облачных сервисов [1].

Сервер БД используется для хранения учетных записей пользователей сервисов, некоторых входных данных для заданий и информации об их выполнении и затраченных для этого ресурсов.

FTP сервер принимает архивы с входными данными (файлы сцен, текстуры, настройки) для заданий и помещает их в сетевую ФС. Отвечает за выдачу результатов визуализации.

Вычислительные узлы, наравне с сервером управления распределенным рендерингом и FTP-сервером имеют доступ к NFS.

2.2 Апробация системы

Апробация СУРВ показала, что использование этого решения сокращает время рендеринга 3D сцен в 7-50 раз, в зависимости от сложности сцены.

Тестирование системы производилось с помощью сцен различной сложности (количество объектов, освещение, шейдеры, большие и маленькие размеры файлов сцен). Была произведена визуализация единичных изображений и анимаций с помощью программных пакетов MentalRay и Vray.

3 Анализ узких мест в системе распределенной визуализации

Одной из проблем, требующих серьезной проработки, является обеспечение эффективности использования вычислительных ресурсов. Для выявления узких мест в системе распределения ресурсов проводились испытания в различных режимах работы программ рендеринга. Сначала были определены основные конфигурации визуализации и их входные параметры.

В экспериментах для конфигурации процесса рендеринга задавался набор входных параметров включающих в себя:

1. Разрешение выходного изображения;
2. Количество частей изображения;
3. Формат выходного изображения.

Эти параметры оказывают основное влияние на время обработки и требования к вычислительной системе.

Разрешение выходного изображения влияет на время обработки трехмерной модели и на количество оперативной памяти, требуемой для процесса визуализации.

Количество частей изображения влияет на распределение нагрузки.

Формат выходного изображения влияет на время постобработки. В зависимости от него находится размер выходных файлов.

3.1 Способы распределения нагрузки

Распределенная обработка статической трехмерной сцены подразумевает ее разбиение на кусочки. То есть вся сцена делится на части, каждая из которых обрабатывается отдельным узлом вычислительного кластера. Таким образом реализуется распределение нагрузки на вычислительные узлы. Этот вариант подходит и для обработки динамических сцен – анимаций, но в этом случае вычислительные узлы обрабатывают кадры анимации целиком.

Возможен комбинированный вариант распределения нагрузки, когда кадры анимации делятся на части и каждая из них обрабатывается отдельно. Он был бы удобен для большого количества низких по производительности вычислительных узлов, например небольших компьютеров с процессорами ARM-архитектуры и весьма ограниченным объемом оперативной памяти.

Перед обработкой исходная сцена (статическая, либо динамическая) может быть разбита на части несколькими способами:

- Полосами
- Квадратами
- С использованием карты сложности (importance map)
- Кадрами (для анимаций)
- Комбинированно
- С использованием коэффициентов производительности вычислительных узлов (для гетерогенных систем)

Для систематизации информации о способах распределения нагрузки предложена следующая классификация:

- Индифферентные
 - Статическими полосами
 - Статическими квадратами
 - Кадрами
 - Комбинированно
- Адаптивные
 - Адаптивными полосами
 - Адаптивными квадратами
 - С использованием карты сложности
 - использованием коэффициентов производительности

Данная классификация строится на основе зависимости способа от входных данных, таких как разрешение выходного изображения, индекс производительности узла и т.д.

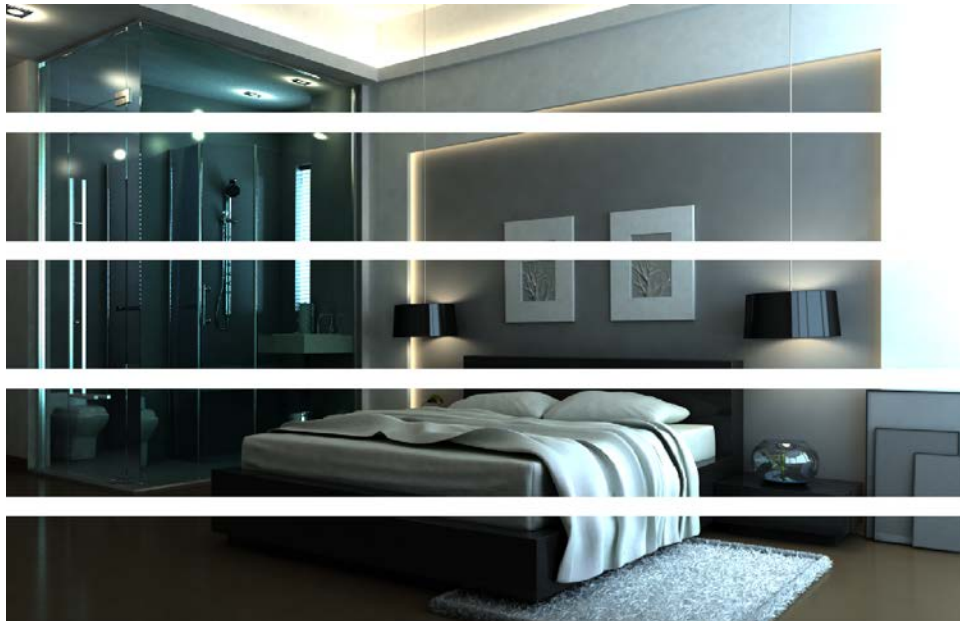
3.1.1 Индифферентные

Не зависят от входных данных, а следовательно не позволяют влиять на нагрузку. Индифферентное разделение сцены на полосы осуществляется заданием количества или размера вертикальных или горизонтальных полос. При этом не учитывается ни разрешение выходного изображения, ни сложность сцены.

Если задается количество полос на сцену, то вычислить их длину и ширину не составит большого труда. Однако судить о сложности каждой полосы, о времени и ресурсах, требуемых для ее обработки невозможно.

Следовательно, при количестве узлов, равном количеству полос, время обработки всей сцены будет равно времени рендеринга самой сложной полосы. Это неэффективно, так как когда завершится обработка самых легких полос, узлы, работавшие над ними, освободятся, а ресурсы, обрабатывающие остальную часть сцены останутся занятыми этой задачей. Причем разница во времени обработки полос может быть очень большой.

Например, сцена на Изображении 1. Здесь использовалось разбиение полосами. Время рендеринга отдельных полос отображено на Диаграмме 2.



Изображение 1. Сцена, разбитая на 5 полос.

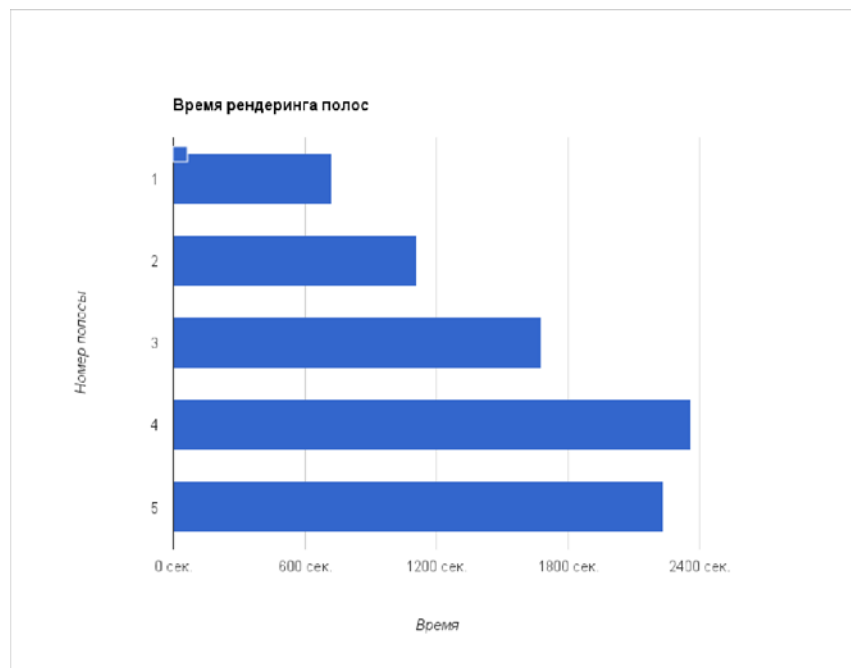


Диаграмма 2. Время рендеринга отдельных полос.

Из Диаграммы 2 очевидно, что полоса №4 обрабатывалась дольше остальных и примерно в 3 раза дольше полосы №1. Отсюда можно сделать эмпирическое предположение, что увеличение количества полос положительно повлияет на время рендеринга, то есть уменьшит его. Чтобы проверить это предположение был поставлен эксперимент. Результат эксперимента в виде графика изображен на Диаграмме 3.

Тестируемая сцена (Изображение 1) сначала обрабатывалась на 5 узлах (Диаграмма3, синий график). Таким образом была смоделирована ситуация, когда в вычислительной системе дефицит ресурсов. А затем та же сцена была визуализирована на количестве узлов, сопоставимом с количеством частей.

Синий график показывает, на участке от 1 до 25 полос их связь со временем рендеринга обратно пропорциональна, следовательно, наблюдается уменьшение времени рендеринга. Но после 25, число полос становится избыточным и ресурсы начинают использоваться неэффективно, что приводит к увеличению времени обработки. Это происходит из-за того, что время, затрачиваемое узлом на подготовку частей сцены к рендерингу начинает превосходить время самого рендеринга этих частей.

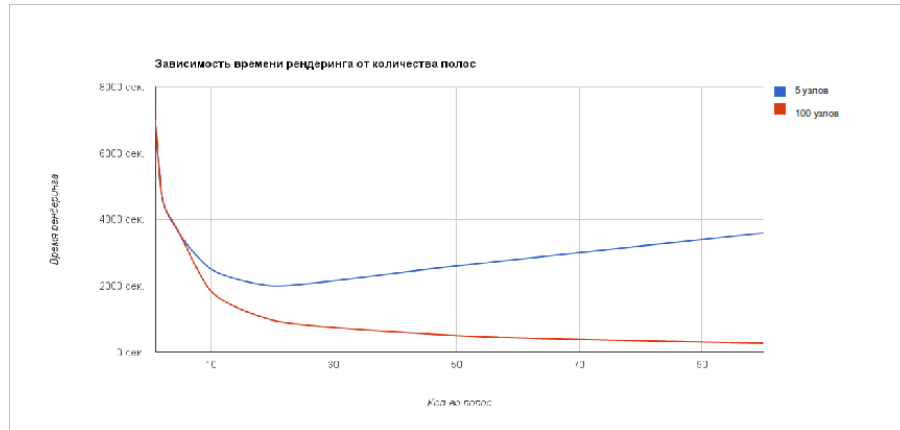


Диаграмма 3. График зависимости времени рендеринга от количества полос. Синим цветом – для 5 узлов, красным – для 100.

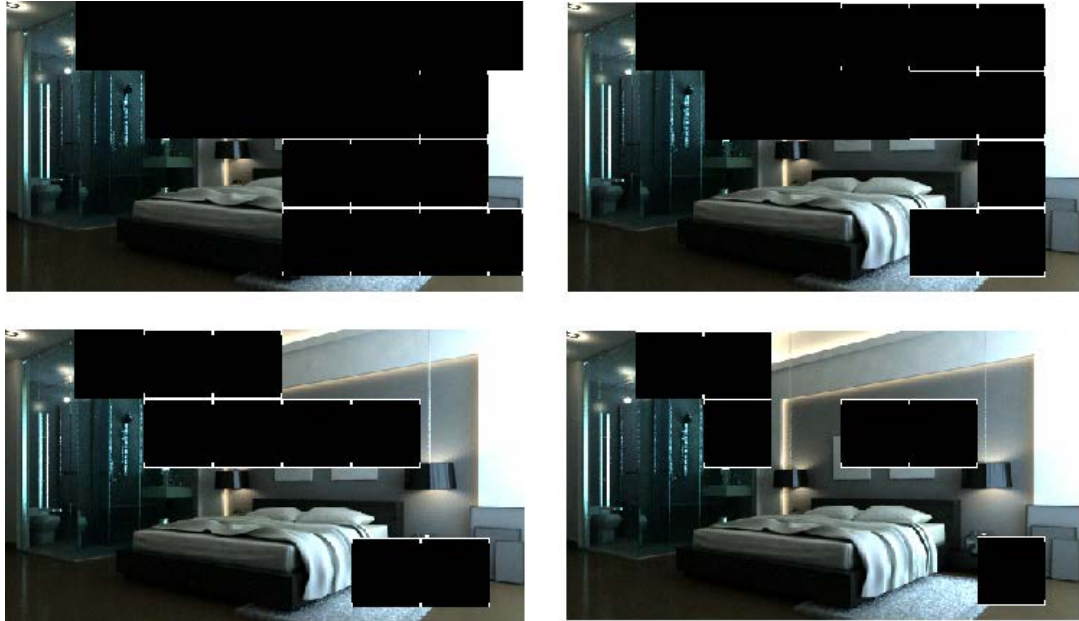
Красный график иллюстрирует плавное уменьшение времени рендеринга с увеличением числа полос. В отличие от случая с ограниченными вычислительными ресурсами, подготовка сцены происходит на каждом узле только один раз. Стоит отметить, что уменьшение времени визуализации наблюдается только до определенного порога, после которого увеличение числа полос не имеет смысла, и время рендеринга снова начинает расти. Значение этого порога прямо пропорционально зависит от таких факторов как сложность сцены и разрешение выходного изображения, но на данный момент его нельзя определить однозначно.

Данный способ используется в ПО для управления распределенной визуализацией Autodesk Backburner, который поставляется в комплекте с 3DsMAX.

Несомненно, минусом такого способа распределения нагрузки является независимость количества полос от разрешения сцены и ее сложности. А так же, ввиду зависимости одной из сторон полосы от разрешения, невозможность ограничить количество ресурсов, необходимых для обработки одной полосы. Так как другая сторона полосы этого изображения может перекрывать доступные для ее обработки ресурсы узла. И результат рендеринга может быть утрачен по причине дефицита ресурсов.

Разбиение сцены на равные квадраты является более эффективным, так как не зависит от ее геометрии (Изображение 2). Но так же как и в случае с полосами существует разница во времени рендеринга отдельных квадратов, которая может быть нивелирована уменьшением их размера. Такой способ распределения нагрузки реализован в продуктах MentalRay и VRay.

Покадровый рендеринг трехмерных анимаций во многом схож с разбиением на квадраты. В этом случае так же можно наблюдать разницу в рендеринге отдельных кадров анимации. Отметим, что такой способ используется везде: Autodesk Backburner, MentalRay, VRay.



Изображение 2. Случайные этапы рендеринга с применением распределения квадратами.

Чтобы избавиться от разницы во времени рендеринга отдельных частей сцены, индифферентные способы распределения нагрузки могут использоваться в комбинации между собой. Например:

- Полосы в сочетании с квадратами. Полоса обрабатывается на отдельном вычислительном узле, а узел разбивает ее на квадраты и каждый из них визуализируется одним потоком, на одном ядре ЦП.
- Кадры в сочетании с полосами (или квадратами). Каждый кадр разбивается на полосы (или квадраты), обрабатываемые одним узлом.

3.1.2 Адаптивные

Алгоритмы адаптивных способов используют информацию о геометрии или сложности сцены для создания более равномерного распределения нагрузки на вычислительные узлы.

На данный момент ведется исследование возможностей определения сложности сцен форматов 3DsMAX, VRay и MentalRay, но уже сейчас можно предполагать некоторые их свойства.

В отличие от распределения статическими полосами, использование адаптивных полос предусматривает анализ сложности сцены. Его результаты используются для задания ширины каждой полосы в зависимости от сложности обрабатываемого фрагмента, так, чтобы значение сложности фрагментов сцены были равны.

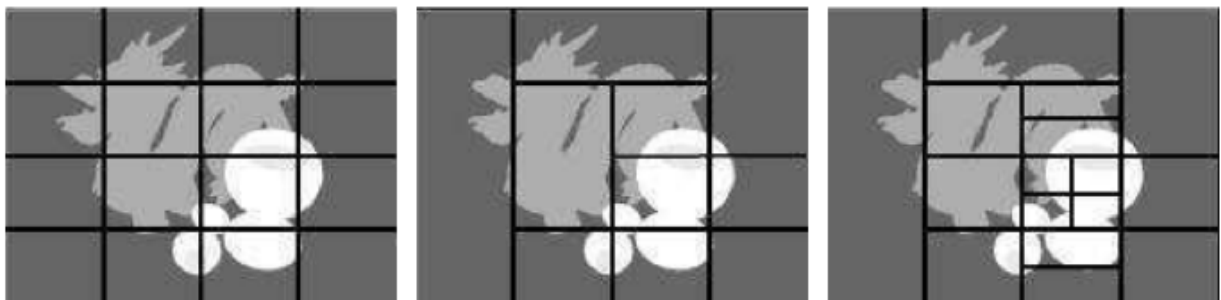
Так же дело обстоит и с адаптивными квадратами, но в отличие от адаптивных полос, они имеют фиксированные ширину и высоту, что позволяет этому способу более эффективно использовать ресурсы.

Карта сложности (importance map) [5] представляет собой матрицу, каждый элемент которой соответствует пикселю изображения и имеет значение коэффициента сложности. На Изображении 3 можно увидеть результат рендеринга сцены, а на Изображении 4 процесс построения карты сложности. На первом этапе картинка разбита на 16 равных частей (blind partitioning), на втором этапе части с одинаковыми значениями коэффициента сложности объединяются, на третьем производится балансировка отношения сложности к размеру. Метод построения карт сложности используется в мультиагентной системе распределенного рендеринга MAgArRO. [6]

Использование коэффициента производительности вычислительных узлов имеет место в гетерогенных вычислительных системах, поэтому в данной работе на рассматривается.



Изображение 3. Пример сцены для построения карты сложности.



Изображение 4. Карта сложности. Слева направо: разбиение на равные части, объединение одинаковых по сложности частей, балансировка отношения сложности к размеру.

3.2 Критерий эффективности использования ресурсов

Эффективность использования вычислительных ресурсов, наряду с оптимальностью, имеет большое значение при выборе способа производства вычислений. Поэтому встает вопрос о том как оценивать эти два параметра в процессах распределенной визуализации.

В данной работе предлагается определить коэффициент эффективности как разность единицы и отношения минимального и максимального времени рендеринга части сцены (Формула 1).

$$K_{odrr} = 1 - \frac{\min(t_p)}{\max(t_p)}$$

Формула 1. Коэффициент эффективности распределенного рендеринга. t_p – время рендеринга части сцены.

Иными словами, чем меньше разница во времени рендеринга между самой сложной частью и самой простой, тем эффективнее используются вычислительные ресурсы. На Диаграмме 4 изображена зависимость времени рендеринга частей сцены от их количества при условии равенства числа частей и вычислительных узлов для разбиения статичными полосами. Нетрудно заметить, что использование 100 полос позволит решить задачу визуализации гораздо эффективнее, чем 2.

В Таблице 1 приведены значения коэффициента эффективности.

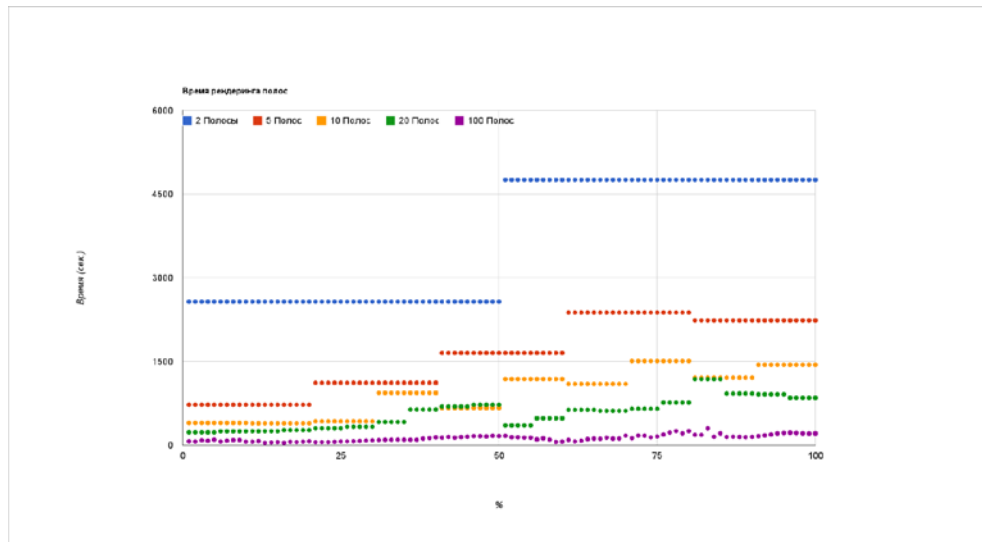


Диаграмма 4. Зависимость времени рендеринга частей сцены от их количества.

Таблица 1. Зависимость коэффициента эффективности от количества полос.

<i>Кол-во полос</i>	<i>Kodrr</i>
2	0.46
5	0.69
2	0.74
5	0.80
10	0.86

4 Заключение

В процессе исследования программных продуктов 3DsMAX, MentalRay, Vray были определены их возможности по распределенной обработке трехмерных сцен. Анализ результатов исследования позволил сделать выводы, что исследуемые программные продукты не эффективно используют вычислительные ресурсы. Был определен критерий эффективности использования ресурсов в процессах распределенной визуализации. Этот параметр позволил оценить узкие места в процессах распределенного рендеринга с использованием вышеупомянутых продуктов.

Дальнейшее исследование алгоритмов оценки сложности трехмерных сцен поможет определить конфигурации процессов рендеринга, которые будут более эффективно использовать вычислительные ресурсы. К ним относится алгоритм построения карты сложности, используемый в мультиагентной системе рендеринга, именуемой MAgArRo.

Источники

- [1] А.И. Панасюк, С.В. Маколов, Д.Ю. Астриков, Д.А Кузьмин: Визуализация трехмерных моделей на суперкомпьютерных ресурсах. *Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции*:69-74, 2012
- [2] Д. Астриков, Д. Кузьмин, А. Панасюк: Высокопроизводительные вычисления как сервис. *The 2nd International Conference on High Performance Computing Kyiv, Ukraine*:79-85, 2012
- [3] О комплексе. <http://cluster.sfu-kras.ru/>
- [4] Д. Кузьмин: Сибирский Федеральный Университет (СФУ). *журнал Суперкомпьютеры № 10* - М.: ООО "Издательство СКР Медиа":35, 2012
- [5] С. Gonzales-Morcillo, G. Weiss, D. Vallejo, L. Jimenez-Linares and J.J. Castro-Schez: A Multi-Agent Architecture for 3D Rendering Optimization. *Journal of Applied Artificial Intelligence*: 24:4, 313-349, 2010.
- [6] MultiAgent Approach to Rendering Optimization: MAgArRo, <http://www.esi.uclm.es/www/cglez/magarro/>