Реализация и тестирование параллельной версии пакета визуализации Povray

Аннотация

Для реалистичной визуализации научных данных используется пакет трассировки лучей Povray 3.6.1. Для ускорения процесса визуализации была разработана параллельная версия пакета на основе технологии MPI. Изучена эффективность предложенного метода параллелизации.

1 Введение

Важным этапом в задаче численного моделирования физических процессов является интерпретация и визуализация полученных в результате численного эксперимента массивов данных. Одним из подходов к визуализации данных является получение изображения высокой четкости, как максимально приближенного к реальности, то есть фотореалистичного изображения, так и искуственное изображение с определенным наложением оттенков, отражающим геометрию объекта. В данной работе для получения визуализации высокой четкости используется метод трассировки лучей. За основу взят пакет трассировки лучей Povray 3.6.1. Метод трассировки лучей является вычислительно трудоемким, поэтому процесс визуализации может занимать большую часть времени расчета. Для ускорения пакета предложена и реализована эффективная параллелизация данного пакета на основе технологии MPI, что позволило получить ускорение, порпорциональное числу предоставленных процессоров.

2 Параллельная реализация

Параллельная реализации пакета Povray 3.6.1 основана на модели "начальник"-"подчиненный". Процессор с нулевым рангом помечается как "начальник" и выполняет задачу по распределению и приему обработанных частей изображения.

На исходную область изображения размеров $N \times M$ пикселей накладываются квадратные подобласти размером $K \times K$ пикселей. Если N или M не кратно K, то крайние подобласти обрезаются. Таким образом исходная задача делится на $\lceil N/K \rceil \times \lceil M/K \rceil$ подзадач. В данной работе параметр K равен 16.

Процессор-"начальник" распределяет среди "подчиненных" по L подзадач и ожидает выполненную работу при помощи синхронной операции приема MPI_Recv.

После выполнения приема результат записывается во временный массив и, если процессор "подчиненный" закончил свои L задач, "начальник" отправляет новые L задач при помощи асинхронной команды MPI_Isend. В данной работе параметр L равен 15. Задачи можно передавать как по порядку, так и в произвольном порядке. В данной работе задачи передавались по порядку.

Затем данные из временного массива записываются в конечное изображение.

Для выполнения асинхронных пересылок сделана многократная буферизация данных, то есть перед очередной пересылкой процессор проверяет свободен ли очередной буфер отправки при помощи команды MPI_Wait, затем записывает данные в этот буфер и вызывает команду на асинхронную отправку MPI_Isend. Для отправки новых задач процессор "начальник" хранит массив из $B_M = 256$ буферов. Аналогичный метод многократной буфферизации в сочетании с асинхронными пересылками используется для отправки выполненных данных процессором- "подчиненным", массив состоит из $B_S = 16$ буферов.

В итоге процессор-"начальник" выполняет следующие шаги:

1. синхронная рассылка первоначальных номеров задач всем узлам;

2. при получении выполненной задачи:
   (a) прием задачи во временный массив;
   (b) асинхронная отправка номеров новых задач (или -1, если новых задач нет) с текущего буфера, если "подчиненный" выполнил все задачи;
   (c) ожидание следующего буфера отправки номеров задач;
   (d) переключение на следующий буфер отправки номеров задач;
   (e) запись пикселей из временного массива в конечную картинку;
   (f) при наличии графической оболочки - вывод полученного массива пикселей на экран;
3. если все задачи выполнены, то запись картинки в файл и выход, иначе переход к пункту 1.

Процессоры-"подчиненные" одновременно выполняют следующие шаги:

1. прием номеров задач;
2. выполнение текущей задачи, если задача имеет номер -1, то ожидание отправки всех буферов и выход;
3. асинхронная отправка выполненной задачи с текущего буфера;
4. ожидание следующего буфера хранения выполненной задачи;
5. переключение на следующий буфер хранения выполненной задачи;
6. если все задачи выполнены, то ожидание новых номеров задач, переход к пункту 2.

Такой комплексный подход с использованием

1. модели "начальник"-"подчиненный",
2. асинхронных пересылок,
3. многократных буфферизаций,
4. выполнения подчиненным по несколько задач,

направлен на то, чтобы получить как адаптивную балансировку загрузки узлов так и на то чтобы сгладить задержку, вызванную латентностью сети. Первая цель выполняется благодаря тому, что процессор, загруженный областью, требующей большего времени расчета, не будет получать новых задач, пока не завершит выполнение своей. Таким образом задачи будут распределятся между менее загруженными процессорами. Латентность сети снижается благодаря множественной буферизации и тому, что фактически пересылки, где возможно, производятся одновременно с расчетом.

3 Тестирование

Параллельная версия тестировалась на кластере состоящие из 23 узлов, с двумя четырехядерными процессорами Intel Xeon 2.6 Ghz в каждом. Тест проводился на сцене, результат рассчета которой изображен на рисунке (1). Целью теста было проверить эффективность получения изображения размера 4096×3072 пикселей в зависимости от использования большого числа процессоров. Сцена имеет большое число геометрически сложных переотражающих поверхностей, преломляющих областей, а так же областей с рефракциией, на равне

с обычными поверхностями, тем самым является хорошим тестом на балансировку задачи между процессорами. Следует отметить, что часть времени уходит на загрузку сцены.

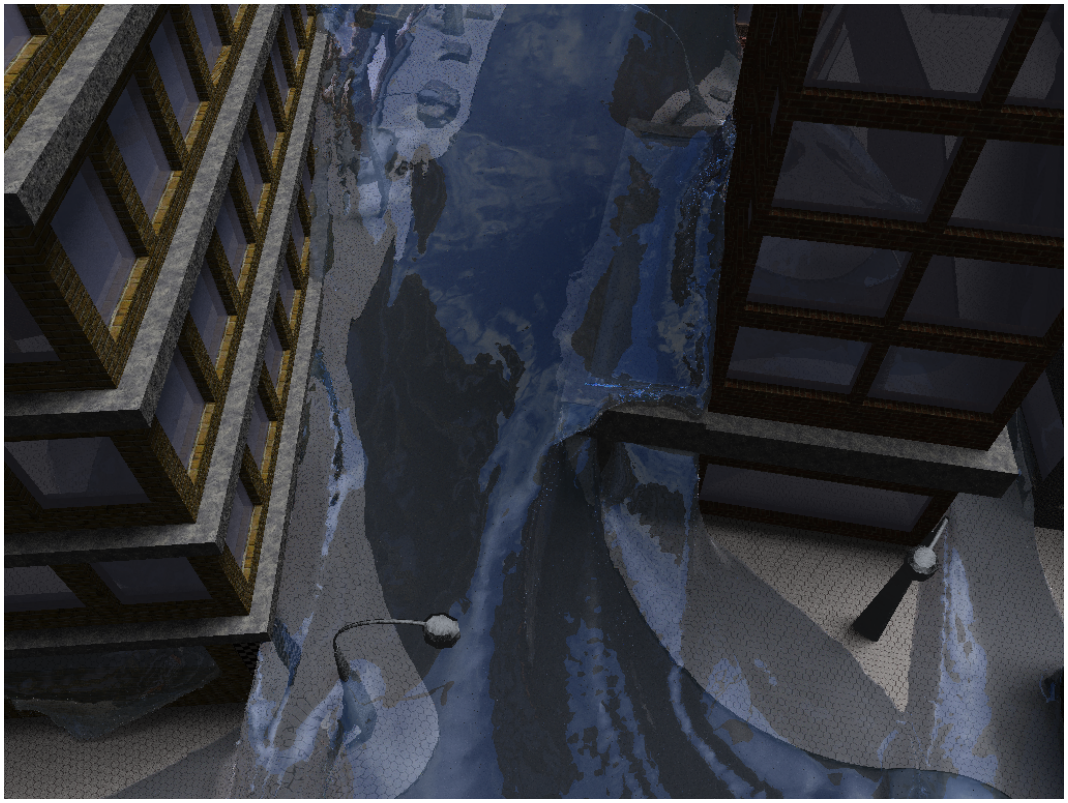

Рис. 1: Задача о затоплении города.

Ускорение, полученное при визуализации с разрешением 4096×3072 сцены, изображенной на рисунке 1, представлено в таблице 1:

Число вычислительных ядер	15	30	60	120
Время в секундах	367	195	107	62

Таблица 1: Время визуализации для одного кадра 4096×3072 (в секундах).

Таблица демонстрирует хорошую эффективность параллельной реализации пакета Povray 3.6.1 при визуализации сложных сцен.

4 Недостатки

Текущая реализация имеет следующие недостатки:

1. часть времени занимает чтение и обработка файлов сцены, в связи с характером реализации пакета Povray, данный этап сложно ускорить;

2. не параллелизован алгоритм переизлучения света.

В рамках проделанной работы не изучалась возможность ускорения данных этапов и алгоритмов.

5 Применение

Пакет широко применялся для получения визуализации для задачи моделирования вязкой и вязкопластичной жидкости со свободной поверхностью. С помощью пакета была получина коллекция видеоматериалов:
www.inm.ras.ru/research/freesurface, dodo.inm.ras.ru/research/freesurface

Так же результаты визуализаций представлены в следующих работах:

1. Nikitin K., Olshanskii M., Terekhov K., Vassilevski Yu. A numerical method for the simulation of free surface flows of viscoplastic fluid in 3D. *J. Comp.Math.* 29(6), 605-622, 2011.

2. Yu.V.VASSILEVSKI, K.D.NIKITIN, M.A.OLSHANSKII, K. M. TEREKHOV CFD technology for 3D simulation of large-scale hydrodynamic events and disasters *Russ. J. Numer. Anal. Math. Modelling*, Vol. 27, No. 4, pp. 399–412 (2012)

Патч к исходному коду Povray 3.6.1, выполняющий параллелизацию, доступен по адресу http://dodo.inm.ras.ru/ terekhov/mpipovray.html

При выполнении конфигурации измененной версии povray используется следующие флаги: ./configure CXX="mpicxx -DHAVEMPI"C="mpicc -DHAVEMPI"COMPILED_BY="your name <email@address>"

Вместо mpicxx и mpicc необходимо использовать доступные mpi-компиляторы.