УДК 004.942:004.272.26

СРАВНЕНИЕ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ MPI И OPENCL ДЛЯ ГЕНЕРАЦИИ ВОЛНОВОЙ ПОВЕРХНОСТИ

Ганкевич И.Г., Дегтярев А.Б., **Соэ Моэ Лвин,**
Санкт-Петербургский государственный морской технический университет,
e-mail: gig.spb@gmail.com

Аннотация

В работе производится сравнение различных аппаратно-программных подходов к генерации волновой поверхности. Подробно описывается модель авторегрессии, общий ход решения задачи, приводятся отдельные алгоритмы для каждого из подходов и оценивается их эффективность. Эта модель наиболее актуальна при реализации виртуального полигона моделирования поведения динамических объектов и в инструментальных средствах тестирования бортовых корабельных интеллектуальных систем реального времени. В статье присутствует краткое описание технологий MPI и OpenCL, с выделением преимуществ и недостатков каждой из них.

Ключевые слова: *модель авторегрессии, MPI, OpenCL, генерация волновой поверхности, параллельные вычисления.*

Введение

Морское волнение – сложный физический процесс, и зачастую в задаче поиска ошибок, визуализация его в реальном времени оказывается эффективнее стандартных математических методов проверки. Также особо необходимо отметить важность этого процесса в задачах виртуального моделирования поведения морских объектов в условиях нерегулярного волнения и настройки базы знаний бортовых интеллектуальных систем. Такая визуализация возможна при достаточной скорости генерации волновой поверхности, поэтому поиск новых методов решения этой проблемы является актуальной задачей на сегодняшний день.

Модель авторегрессии

Волновая поверхность – это пространственно-временное поле, каждая точка которого является взвешенной суммой предыдущих по времени точек и белого шума (некоторой случайной переменной). Это называется авторегрессионной зависимостью [1].

$$z_{x,y,t} = \sum_{i=0}^{p_1}\sum_{j=0}^{p_2}\sum_{k=0}^{p_3}\varphi_{i,j,k}\cdot z_{x-i,y-j,t-k} + \varepsilon_{x,y,t}$$

Коэффициенты авторегрессии φ, необходимые для генерации ряда, можно получить из заданной автоковариационной функции, решив систему уравнений Юла-Уокера.

$$A\varphi = b$$
$$a_{i,j} = \gamma_{|x(i)-x(j)|,|y(i)-y(j)|,|t(i)-t(j)|}$$
$$b_i = \gamma_{x(i),y(i),t(i)},$$
$$x(i) = \mathrm{mod}((i-1)/(p_1p_2), p_1)$$
$$y(i) = \mathrm{mod}((i-1)/p_3, p_2)$$
$$t(i) = \mathrm{mod}(i-1, p_3)$$

Выше приведены формулы системы уравнений Юла-Уокера для трехмерной задачи, где $\gamma_{i,j,k}$ – дискретные значения автоковариационной функции.

Теоретически количество коэффициентов авторегрессии p стремится к бесконечности. На практике же выбирается автоковариационная функция, достаточно быстро стремящаяся к нулю, тогда значения φ_i при некотором i становятся пренебрежительно малыми. Одна из возможных аппроксимаций ковариационной поверхности приведена ниже.

$$\gamma(x,y,t) = e^{-\alpha(|x|+|y|+|t|)}\cdot\cos(\beta\cdot x)\cdot\cos(\beta\cdot y)\cdot\cos(\beta\cdot t)$$

При рассмотрении натурных данных в качестве такой поверхности рассматривается ее оценка, что приводит к необходимости применения метода наименьших квадратов при нахождении коэффициентов авторегрессии. Однако, исследования показали [1,3], что в этом случае модель авторегрессии выгоднее строить по уже сглаженной автоковариационной функции, а не по натурным данным.

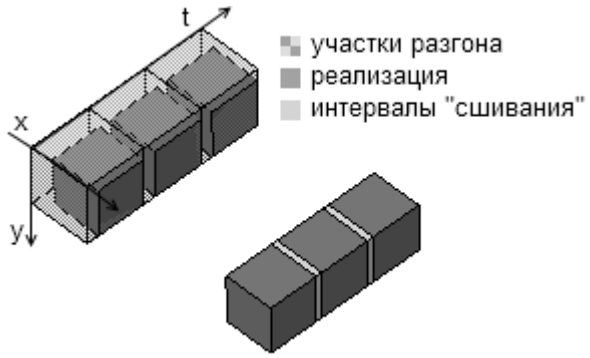

Рис 1. Волновая поверхность в памяти устройства

Поэтому целесообразнее подобрать оптимальный порядок системы вручную, минуя метод наименьших квадратов, часто используемый в решении одномерной задачи. Матрица системы Юла-Уокера положительна определена и симметрична, следовательно ее можно разложить на две треугольные методом Холецкого, и, решив системы, найти коэффициенты авторегрессии.

Помимо коэффициентов для создания волновой поверхности одной из важных составляющих алгоритма является генерация нормально распределенной случайной величины (белый шум). В качестве метода в работе использована параллельная реализация Вихря Мерсенна (Mersenne Twister). Алгоритм запускает в каждом потоке отдельный генератор с уникальными параметрами, задающимися специальной подпрограммой dcmt [6]. Это позволяет минимизировать коррелированность между последовательностями чисел этих генераторов [7].

Рис. 2. Соединение частей реализации по двухсторонней авторегрессионной зависимости

Получив коэффициенты авторегрессии и массив случайных величин, можно приступать к генерации волновой поверхности. Создание происходит в несколько этапов. На первом этапе генерируется реализация заведомо большего размера, разделенная на независимые части для каждого потока. В начале каждой части находится промежуток разгона алгоритма, на котором для генерации величин используются не все коэффициенты авторегрессии. Эти участки удаляются из реализации, оставляя несвязанные части. Алгоритм завершается соединением этих частей по двухсторонней авторегрессионной зависимости (интервал между ними изначально заполнен нулями) [8]. Такая процедура именуется сшиванием.

$$z_{x,y,t} = \sum_{i=0}^{p_1}\sum_{j=0}^{p_2}\sum_{k=0}^{p_3}\varphi_{i,j,k} \cdot z_{x-i,y-j,t-k}$$

$$+\sum_{i=0}^{p_1}\sum_{j=0}^{p_2}\sum_{k=0}^{p_3}\varphi_{i,j,k} \cdot z_{x+i,y+j,t+k} + \varepsilon_{x,y,t}$$

Проверка характеристик волновой поверхности

По окончанию генерации поверхности важно проверить ее достоверность, то есть сравнить реализацию и модель. Сравнение осуществляется по нескольким характеристикам: автоковариационная функция, дисперсия, спектр, распределения волновых аппликат, уклонов, высот и периодов волн. Выборочные результаты сравнения представлены на рисунках 3-6.

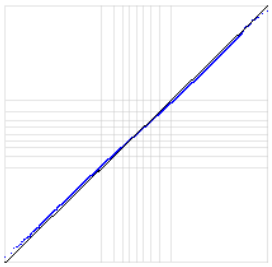

Рис. 3. Распределение волновых аппликат

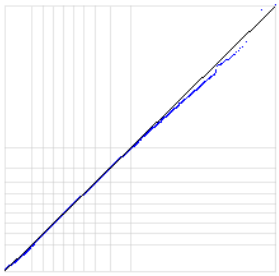

Рис 4. Распределение высот волн

Рис. 5. Сечение автоковариационной функции параллельно оси OX

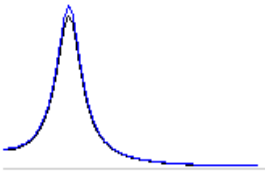

Рис. 6. Сечение спектра параллельно оси OX

Сравнение реализаций на MPI и OpenCL

Технология OpenCL, или Open Computing Language, предоставляет пользователю средства запуска программ на вычислительном устройстве и специальный язык

программирования. Особенностью и главным преимуществом данной технологии является возможность запуска программ на графических платах. Большинство транзисторов на таких платах предназначены для обработки данных, а не для кэширования или управления потоком выполнения. Число ядер на графическом устройстве превосходит число ядер у центрального процессора на порядок, хотя они и проигрывают по частоте в разы. Таким образом, особенности внутреннего устройства графических плат позволяют им одновременно выполнять большое число потоков, получая теоретический выигрыш в скорости по сравнению с процессорами.

Алгоритм генерации временного ряда, реализованный на OpenCL, обладает несколькими характерными чертами. Он работает тем быстрее, чем больше частей присутствует в реализации временного ряда (число частей должно быть меньше максимального числа потоков устройства). При малом их количестве не все параллельные потоки устройства запускаются, что значительно замедляет алгоритм (строки 1-4 таблицы 1). При большом количестве частей вынужденно уменьшается интервал между ними, что приводит к некачественному сшиванию. Накладные расходы на заполнение участка разгона минимальны, это подтверждают строки 7-8 таблицы 1, где область разгона отсутствует.

	N	Z_1			Z_2			P_1	P_2	I	t_{OCL}, с	t_{MPI}, с
		x	y	t	x	y	t					
1	6	32	32	768	64	64	1020	128	170	64	18,46	2,10
2	12	32	32	768	64	64	1020	64	85	32	15,29	1,52
3	24	32	32	768	64	64	1008	32	42	16	10,47	0,72
4	48	32	32	1536	64	64	1536	32	32	16	11,46	0,73
5	12	32	32	384	64	64	768	32	64	16	10,15	1,06
6	12	16	16	384	32	32	768	32	64	16	2,28	0,15
7	12	32	32	768	32	32	768	64	64	16	2,27	0,14
8	24	32	32	768	32	32	768	32	32	16	1,58	0,15

Таблица 1. Сравнение эффективности вариантов генерации волновой поверхности при помощи технологии OpenCL и MPI. N – количество частей, Z_1 – размер реализации, Z_2 – размер реализации вместе с участками разгона, P_1 – размер части, P_2 – размер части с разгоном, I – размер интервала, t_{OCL} и t_{MPI} – время работы программ.

Технология MPI, или Message Passing Interface, предоставляет пользователю средства для запуска программы сразу в нескольких процессах, которые могут выполняться на различных устройствах, соединенных между собой. В отличие от OpenCL, эта технология поддерживает только центральные процессоры. Она не позволяет всем процессам использовать общую память, как это происходит на графическом устройстве, тем не менее, распараллеливание программы на MPI упрощено и имеет меньше ограничений по сравнению с OpenCL.

Алгоритм генерации волновой поверхности, реализованный на MPI, заключается в следующем. Каждый процесс генерирует отдельную часть временного ряда. Затем каждому процессу передается небольшой участок предыдущей части, достаточный для их сшивания. Далее части соединяются и посылаются на одно устройство для проверки результата.

Алгоритм работает тем быстрее, чем больше процессов участвуют в работе (строки 1-4 таблицы 2). С увеличением размера реализации линейно возрастает время работы (строки 2, 5, 7 таблицы 2).

	N	Z_1			Z_2			P_1	P_2	I	t_{MPI}, с
		x	y	t	x	y	t				

1	2	64	64	768	128	128	1024	384	512	32	13,46
2	4	64	64	768	128	128	1024	192	256	32	16,6
3	8	64	64	768	128	128	1024	96	128	32	11,9
4	16	64	64	768	128	128	1024	48	64	32	11,31
5	4	64	64	1536	128	128	1792	384	448	32	22,62
6	4	64	64	384	128	128	640	96	160	32	11,05
7	4	64	64	3072	128	128	3328	768	832	32	41,07

Таблица 2. Эффективность распараллеливания программы под MPI. Обозначения те же, что в таблице 1.

Заключение

Программы тестировались на следующих устройствах: видеокарта с 112 потоковыми процессорами с частотой 1,5 Ггц, позволяющая запускать около 10 тыс. потоков одновременно и 2 двухъядерных процессора с частотой 3 Ггц, позволяющие запускать сразу 4 потока.

Несмотря на превосходство видеокарты по вычислительной мощности, алгоритм генерации временного ряда на MPI работает эффективнее алгоритма на видеокарте. OpenCL предоставляет весьма скудные средства синхронизации, даже с тонкой настройкой эффективно использовать все доступные потоки устройства проблематично. Преимуществом OpenCL можно было бы считать использование единой памяти для расчетов и визуализации, но при большой нагрузке на устройство графические средства видеокарты отключаются, и картинка на экране останавливается.

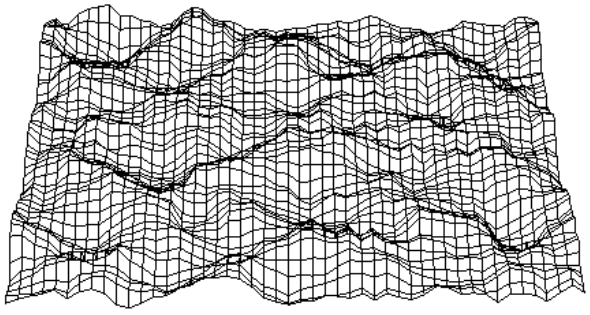

Рис. 6. Сгенерированная волновая поверхность.

Использование MPI дает выигрыш в производительности и максимальном размере реализации, но визуализация влечет за собой накладные расходы на пересылку данных устройству. OpenCL же при малом количестве данных позволяет отображать ход решения в реальном времени. Таким образом, использование того или иного подхода для моделирования волновой поверхности зависит от размера задачи и целей работы.

Литература

1. Бокс Дж., Дженкинс Г. Анализ временных рядов. М.: Мир, 1974
2. Давидан И. Н., Лопатухин Л. И., Рожков В. А. Ветровое волнение как вероятностный гидродинамический процесс. Л.: Гидрометеоиздат, 1978
3. Рожков В. А., Трапезников Ю. А. Вероятностные модели океанологических процессов. Л.: Гидрометеоиздат, 1990
4. Гирс И. В., Русецкий А. А., Нецветаев Ю. А. Испытания мореходных качеств судов. Л.: Судостроение, 1977
5. Голуб Дж., Ван Лоун Ч. Матричные вычисления. М.: Мир, 1999
6. Podlozhnyuk V. Parallel Mersenne Twister, 2007
7. Makoto Matsumoto, Takuji Nishimura Dynamic Creation of Pseudorandom Generators
8. Бухановский А.В., Иванов С.В. Параллельная обработка данных в информационных управляющих системах. //Сб. докладов ВНК «Управление и информационные технологии» УИТ-2003, С.-Петербург, 2003, т.2, с.64-68