



В.А. Засов, К.К. Смаилов

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ ИССЛЕДОВАНИЯ ЭФФЕКТИВНОСТИ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ РЕШЕНИЯ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

(Самарский государственный университет путей сообщения)

Многие прикладные задачи управления решаются на основе анализа математических моделей процессов и систем представленных в виде дифференциальных уравнений в частных производных. Наиболее распространенным подходом к численному решению дифференциальных уравнений является метод конечных разностей или метод сеток [1]. Сеточные методы характеризуются большими временными затратами на решение, определяемыми величиной $T = kmN^2$, где N есть количество узлов по каждой из координат области определения D уравнения, m - число операций, выполняемых для одного узла сетки, k - количество итераций метода до выполнения условий погрешности вычислений [1]. Одним из перспективных направлений повышения производительности при решении таких сложных задач является использование параллельных вычислительных систем (ВС) и параллельных алгоритмов решения дифференциальных уравнений.

Оценка эффективности применяемых параллельных алгоритмов производится на основе вычисления ряда показателей эффективности: ускорения $S_p(n)$; эффективности $E_p(n)$; стоимости $C_p(n)$ вычислений и др.[1,2], определяемых при использовании параллельного алгоритма для p процессоров (ядер) и числа узлов сетки N .

Показатели эффективности параллельных алгоритмов являются противоречивыми и действия по повышению значений одного из показателей могут привести к ухудшению значения другого показателя. Например, повышение ускорения $S_p(n)$ обычно может быть обеспечено за счет увеличения числа p процессоров, что приводит, как правило, при фиксированной сложности задачи к уменьшению эффективности $E_p(n)$.

Поэтому важно определить минимальную сложность задачи (размер сетки), при которой для заданного числа процессоров (ядер) и ускорения обеспечиваются требуемые значения показателей эффективности.

Кроме того, эффективность параллелизма в значительной мере зависит от того, в какой мере параллельный алгоритм учитывает архитектуру конкретной параллельной ВС[1,2].

Таким образом, эффективное применение параллельных алгоритмов в конкретной вычислительной системе основано на определении некоторого компромиссного сочетания требуемых значений разных показателей эффективности: $S_p(n)$, $E_p(n)$, $C_p(n)$ для заданной размерности задачи или выбора размерности задачи, при которой будут обеспечиваться требуемые значения группы показателей.



Актуальность определения такого компромиссного сочетания значений показателей эффективности еще более возрастает при разработке высокопроизводительных вычислительных систем для бортового применения и мобильного приложений.

Из вышесказанного следует актуальность проведения этапа тестирования конкретной вычислительной системы для получения значений показателей эффективности параллельных алгоритмов, реализуемых на ее платформе.

Для решения этой актуальной задачи разработан тестовый программный комплекс позволяющий выполнять вычислительные эксперименты для определения показателей эффективности различных параллельных алгоритмов решения дифференциальных уравнений в частных производных.

В качестве тестового примера используется численное решение задачи Дирихле для уравнения Пуассона, которая определяется как задача нахождения функции $u = u(x, y)$, удовлетворяющей в области определения D уравнению

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), (x, y) \in D,$$
$$u(x, y) = g(x, y) \in D^0,$$

и принимающей значения $g(x, y)$ на границе D^0 области D , функции задаются при постановке задачи. Эта модель широко применяется для описания стационарных тепловых полей, процессов теплопередачи с внутренними источниками тепла, деформации упругих пластин и др. физических процессов.

Программный комплекс позволяет определять показатели эффективности параллельных алгоритмов и программ, выполняемых на многоядерных ВС, использующих технологию многопоточной обработки OpenMP. Для бортовых высокопроизводительных систем многоядерная архитектура ВС наиболее эффективна, а ограничение числа ядер (обычно до 8) в универсальных процессорах CPU преодолевается использованием графических процессоров (GPU) с числом ядер достигающим 10^3 .

В программном комплексе для увеличения масштабируемости параллельных алгоритмов применяется технология CUDA (Computer Unified Device Architecture) на основе применения графических процессоров GPU.

Разработанная программный комплекс позволяет:

- произвести выбор для моделирования из базы алгоритмов того или иного последовательного или параллельного алгоритма решения дифференциальных уравнений в частных производных: алгоритмов Гаусса-Зейделя с классической и волновой схемой расчета, алгоритмов Гаусса-Якоби, алгоритмы с чередованием обработки четных и нечетных строк, алгоритмы с шахматным разбиением области вычислений, с использованием технологии CUDA и др.;
- произвести выбор для моделирования размер и шаг сетки, количество потоков для проведения расчетов;



- произвести оценку ускорения и времени решения;
- вычислить зависимости времени решение от числа узлов, а также от числа используемых ядер;
- получить сравнительные оценки времени проведения решения дифференциальных уравнений для разных видов алгоритмов;
- проводить вычислительные эксперименты с целью определения затрат на синхронизацию потоков, исключения неоднозначности вычислений и взаимной блокировки в параллельных алгоритмах;
- сформировать обзорные отчеты в табличной и графической форме по результатам вычислительных экспериментов.

Пользовательский интерфейс программного комплекса представлен на рисунке 1, а примеры результатов вычислительных экспериментов в графической и табличной формах – на рисунке 2 и таблице.

Программный комплекс эффективно использовался при разработке систем мониторинга по динамическим параметрам крупномасштабных объектов транспортной инфраструктуры – рельсовых цепей, мостов, сортировочных горок [3]. Проведенные на комплексе вычислительные эксперименты позволили разработать программное обеспечение и выбрать оптимальную по соотношению «производительность-цена» архитектуру вычислительной системы для обработки в реальном времени сигналов системы автоматической локомотивной сигнализации [3,4], обеспечивающей безопасность движения поездов.

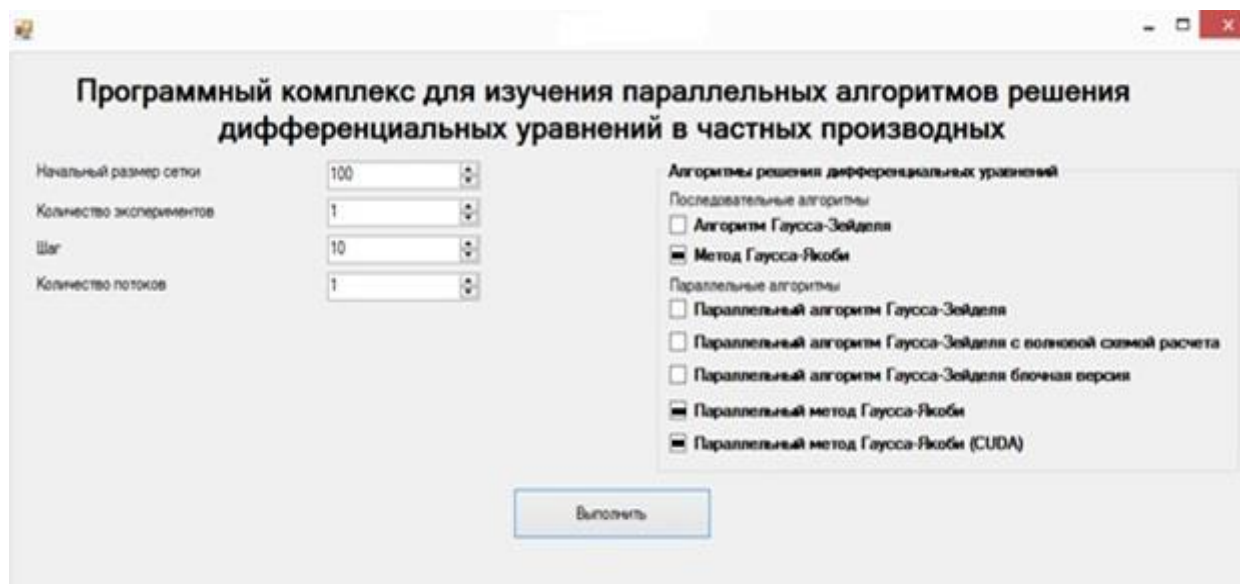


Рисунок 1 – Интерфейс программного комплекса

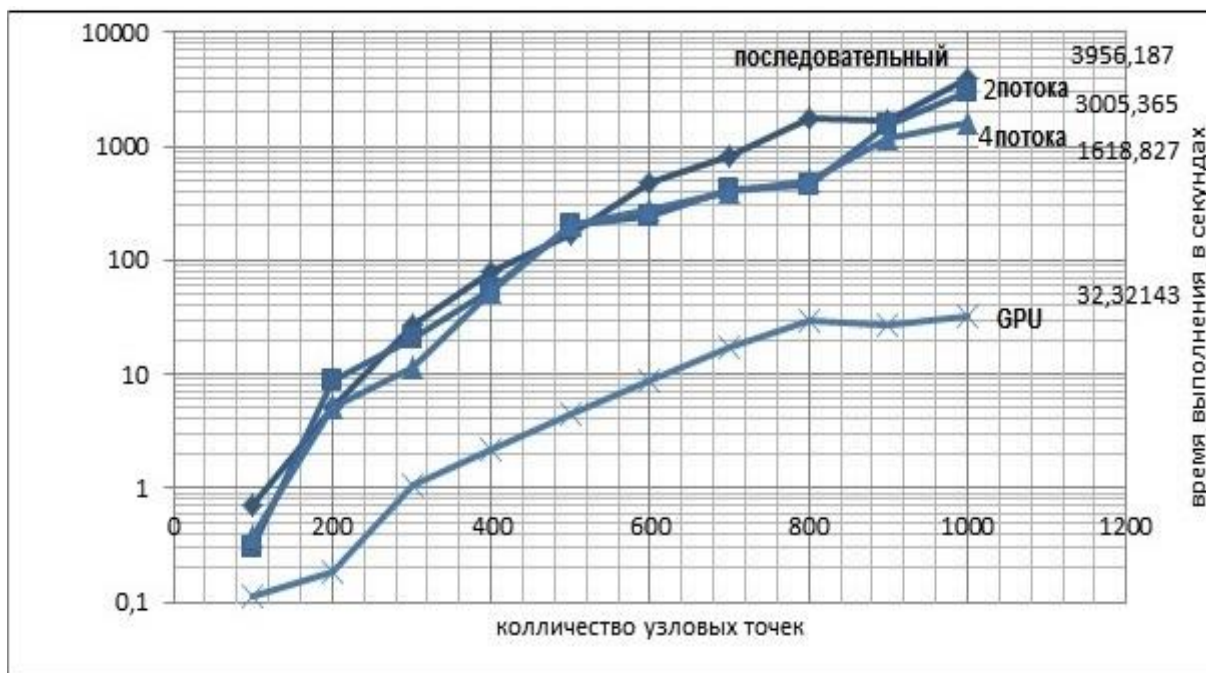


Рисунок 2 – Зависимость времени решения задачи Дирихле методом Гаусса-Якоби от числа ядер и количества узловых точек

Таблица - Время и ускорение решения дифференциального уравнения Пуассона параллельными алгоритмами в зависимости от числа ядер

Количество узловых точек дифференциальных уравнений	Последовательный метод Гаусса-Якоби	Параллельный метод Гаусса-Якоби					
		2 потока		4 потока		GPU	
		Время (сек)	Ускорение	Время	Ускорение	Время	Ускорение
100	0,703525	0,295874	1,376208	0,383992	1,351991	0,11204	9,778235
200	5,14226	8,824108	1,427296	5,031053	1,88388	0,186467	41,17471
300	27,45753	20,65497	1,389197	11,39528	1,72433	1,046881	55,76511
400	79,22281	54,22827	1,47745	53,88307	1,942143	2,22455	60,26964
500	169,7012	202,3873	1,331636	206,0666	1,965262	4,453022	66,55243
600	470,1044	243,7881	1,372349	276,015	1,945489	8,707889	71,63771
700	824,5963	426,2822	1,300534	395,2749	1,95244	17,53718	72,75664
800	1735,924	457,0978	1,394176	490,8026	1,941489	30,05934	73,17973
900	1683,662	1563,369	1,420016	1168,412	1,487688	27,16877	74,75528
1000	3956,187	3005,365	1,405529	1618,827	1,979912	32,32143	74,60107

Литература

1. Гергель В.П. Теория и практика параллельных вычислений: учебное пособие. – М.: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2007. – 423 с.: ил.
2. Параллельные алгоритмы. Разработка и реализация /Ю.К. Демьянович и др. – М.: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2012. – 344 с.: ил.
3. Засов В.А., Ромкин М.В. Параллельные вычисления в задаче разделения сигналов в многомерных динамических системах //Труды VI международной конференции «Параллельные вычисления и задачи управления, РАСО-



2012». – М.: Институт проблем управления РАН им. В.А. Трапезникова, 2012. – С. 96-102.

4. Засов В.А., Никоноров Е.Н. Параллельные вычисления в задачах мониторинга объектов по параметрам динамических процессов //Труды V международной конференции «Параллельные вычисления и задачи управления, РАСО-2010». – М.: Институт проблем управления РАН им. В.А. Трапезникова, 2010. – С. 335-341.

А.А. Иглин, С.В. Востокин

РАЗРАБОТКА ВЫЧИСЛИТЕЛЬНЫХ ШАБЛОНОВ ДЛЯ АНАЛИЗА ДИНАМИЧЕСКИХ СИСТЕМ И ПРОЦЕССОВ НА СУПЕРКОМПЬЮТЕРЕ

(Самарский университет)

Динамические системы изучаются при решении большого количества задач в разных областях физики, химии, механики, биологии, экологии и других. Весьма распространённым случаем задания математической модели динамической системы является конечное число дифференциальных уравнений, при решении которых находятся зависимости переменных от времени и анализируется фазовый портрет системы, наглядно характеризующий изменение состояния системы во времени [1].

Для нахождения решений таких систем и построения фазовых портретов удобно использовать средства автоматизации вычислений. В настоящее время широко распространены различные вычислительные шаблоны и предметно-ориентированные языки программирования, имеющие синтаксис и семантику, напоминающие обычный математический язык (решение описывается набором формул, таблиц и так далее). За счёт использования подобных средств процесс разработки программы значительно упрощается, а также от разработчика не требуется навыков программирования на языках общего назначения [2]. Концепция предметно-ориентированного программирования подразумевает, что от разработчика требуется только представить решение в виде формальной модели (математических формул) и описать его на предметном языке. Затем на основе такой программы автоматически генерируется исполняемый код на языке общего назначения [3].

Чтобы эффективно производить вычисления на большом наборе исходных данных, необходимы большие вычислительные мощности: грид-системы, кластеры, суперкомпьютеры. Использование ресурсов суперкомпьютера позволяет обрабатывать большие объёмы данных за меньшее время по сравнению с производением вычислений на локальной машине, что позволяет значительно увеличить размерность решаемых задач, а также снизить затраты времени на производство вычислений. На данный момент для исполнения параллельных программ на суперкомпьютере «Сергей Королёв» используется система Тем-