

УДК 681.3..06

К.С. Солнушкин (3 курс, каф. ИБКС); М.В. Хлудова, к.т.н., доц.

ПОДДЕРЖКА СИММЕТРИЧНОЙ МУЛЬТИПРОЦЕССОРНОЙ ОБРАБОТКИ В ОПЕРАЦИОННЫХ СИСТЕМАХ LINUX И SUNOS

В настоящее время возрастающие мощности современных вычислительных систем позволяют решать все более разнообразные задачи из различных областей науки и техники. Однако рано или поздно оказывается, что даже самых быстрых компьютеров недостаточно для решения поставленных задач. В таком случае на помощь приходят параллельные вычисления – применение нескольких процессоров для выполнения одной работы. Понятие параллельных вычислений (параллельной обработки) включает в себе широкий спектр методов, начиная от распределенных вычислений и заканчивая симметричной мультипроцессорной обработкой (SMP).

Симметричная мультипроцессорная обработка – предельный случай параллельных вычислений, при котором идентичные процессоры работают в составе единой вычислительной системы и взаимодействуют через аппаратные средства. В SMP-системе все аппаратные ресурсы (оперативная память, диск, графический адаптер) представлены в единственном экземпляре, однако у каждого процессора, как правило, свой собственный кэш. Особенности аппаратной конфигурации требуют поддержки со стороны операционной системы для совместного доступа нескольких процессоров к разделяемым ресурсам.

SMP-системы достаточно популярны, так как, во-первых, один компьютер с несколькими процессорами обходится дешевле, чем система для распределенных вычислений с той же производительностью, состоящая из нескольких компьютеров, во-вторых, используя SMP-системы, можно добиться необходимой производительности уже сегодня, не дожидаясь выпуска нового поколения процессоров. Тем не менее, такая архитектура обладает и недостатками, связанными с необходимостью обеспечивать гармоничную работу нескольких процессоров.

Цель данной работы заключается в том, чтобы рассмотреть на практике реализацию поддержки SMP в такой распространенной операционной системе как Linux. ОС Linux была выбрана, в первую очередь, потому, что она используется на широком спектре аппаратных платформ, таких, как x86, Alpha, UltraSparc, SparcServer, PowerPC и др. Кроме того, исходный код ядра этой ОС предоставлен для открытого доступа. Поддержка SMP появилась в ядре ОС Linux, начиная с версии 2.0, и впоследствии постоянно улучшалась.

В работе рассмотрены примитивы параллельного программирования, используемые ядром ОС Linux (атомарные операции, проверка и установка, семафоры, блокировка в цикле), а также пояснена их реализация в ядре для архитектуры x86. Практическая часть работы реализует комплекс программ, совершающих перемножение матриц, что является хорошо распараллеливаемой вычислительной задачей. Для организации параллельной обработки используется такое понятие мультипроцессорного программирования, как нити.

Программный комплекс состоит из трех модулей, написанных на языке “С”, каждый из которых выполняет одну и ту же работу, но по разным алгоритмам. Общей частью всех модулей является перемножение двух квадратных матриц, A и B , с размером стороны $Size$ ячеек каждая (значение $Size$ задается директивой `#define` в тексте программы). Также в программе используется разделяемый ресурс – матрица D размером $Size$ строк и $2*Size$ столбцов. Каждый из модулей производит поэлементное вычисление двух произведений: AxB и BxA , при этом произведение AxB записывается в левую половину матрицы D , а произведе-

ние ВхА – в правую. Первый модуль не использует нитей, а просто последовательно вычисляет оба произведения. Второй модуль порождает две нити, каждая из которых вычисляет свое произведение. Это дает возможность использовать параллельную обработку. Третий модуль также порождает две нити, но, кроме того, использует блокировку на основе MutEx для контроля доступа к разделяемому ресурсу – матрице D: при этом, когда одна из нитей собирается сделать запись в элемент матрицы D, вторая нить блокируется, а после завершения операции записи продолжает работу.

Для проведения тестов были выбраны две категории вычислительных систем: однопроцессорные (их представляли Intel 486 66 MHz и Intel Celeron 566 MHz под управлением Linux) и двухпроцессорные (2xIntel Celeron 500 Mhz под управлением SunOS 5.8). Программные комплексы для систем под управлением Linux и SunOS различались, так как для Linux была использована библиотека функций pthread, а для SunOS – библиотека thread разработки Sun Microsystems. Совместная работа модулей координировалась с помощью программы, написанной на языке “shell”, которая обеспечивала компиляцию, запуск и автоматический подсчет времени работы каждого модуля. В результате исследований были получены результаты, сведенные в табл. 1.

Таблица 1. Результаты прогона тестов на платформах Celeron 566 Mhz и 2xCeleron 500 MHz.

Размер матрицы, элементов	Тип приложения	Время, затраченное на перемножение двух матриц, с		
		Без использования нитей	2 нити	2 нити и MutEx
		Celeron 566 MHz / 2xCeleron 500 MHz		
100x100		0,090 / 0,142	0,096 / 0,095	0,100 / 0,106
200x200		0,946 / 1,190	0,963 / 0,699	1,010 / 0,749
300x300		3,886 / 4,614	3,856 / 2,590	3,983 / 2,741
400x400		9,306 / 10,736	9,263 / 5,862	9,456 / 6,110
600x600		34,073 / 38,402	34,006 / 20,575	34,496 / 21,158

Результаты по платформе Intel 486 66 MHz не были отображены, так как длительность прогона тестов на этом процессоре была около 12 раз выше, чем на другой однопроцессорной системе, поэтому не удалось бы увидеть разницу между результатами «быстрых» платформ.

Тесты для размера матрицы 100 ячеек были выполнены настолько быстро, что сказались погрешность, внесенная необходимостью считывания исполняемого модуля с диска и загрузки его в память. Тем не менее, для размеров в диапазоне 200..600 четко прослеживается тенденция, характеризующаяся следующими факторами:

1. Для единого модуля (без использования нитей) видно, что платформа с процессором Celeron 566 MHz работает быстрее, чем платформа с двумя процессорами Celeron 500 MHz. Это связано с тем, что в данном тесте не используется второй процессор двухпроцессорной платформы.
2. В случае использования нитей однопроцессорная платформа ведет себя примерно так же, как и в случае единого модуля. Измеренное время примерно одинаково с точностью до статистической погрешности.
3. Двухпроцессорная платформа при использовании нитей показывает почти в два раза более высокие результаты. Двукратного увеличения производительности не удается добиться из-за издержек, необходимых планировщику задач для обеспечения работы нитей.
4. При использовании не только нитей, но и MutEx, обе платформы показывают более вы-

сокое время выполнения, чем без использования `Mutex`. Это связано с тем, что каждая из нитей блокируется и не может выполняться во время того, как другая нить производит запись в матрицу `D`.