



сборки изделий из ткани в системах автоматизации проектирования легкой промышленности, а также при создании компьютерных игр и анимации.

Литература

1. Корнеев В.Д. Параллельное программирование кластеров : учеб. пособие / В.Д. Корнеев. – Новосибирск: Издательство НГТУ, 2008. – 312 с.
2. Хьюз К., Хьюз Т. Параллельное и распределенное программирование на C++.: Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 672 с.: ил.

М.С. Мезенцева

ОПТИМАЛЬНОСТЬ РАСПАРАЛЛЕЛИВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ НА ПРИМЕРЕ РЕШЕНИЯ СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ МЕТОДОМ ГАУССА

(Самарский государственный архитектурно-строительный университет)

Мы живем в эпоху прогресса, во время, когда наука не стоит на месте, появляются новые изобретения, которые облегчают человеку жизнь. Наблюдается бум построения мощных вычислительных систем. Та или иная страна старается быть первой в научном развитии. Становится все больше машин под названием «суперкомпьютер».[1, с.232] Что же под ним скрывается? Суперкомпьютер - вычислительная машина, значительно превосходящая по своим техническим параметрам большинство существующих компьютеров. Например, суперкомпьютер JUQUEEN имеет 458752 ядер, Stampede-462462 ядер. Поэтому тема распараллеливания, при котором в работе участвуют все ядра компьютера, носит актуальный характер.

В качестве математической модели, на которой были проведены исследования по распараллеливанию, был взят, пожалуй, известный всем метод Гаусса, предназначенный для решения системы линейных уравнений. Он применяется во многих сферах: экономике, математике, информатике. Метод Гаусса был реализован в QT на языке программирования C++.

Программа наряду с расчетом уравнений, осуществляет подсчет времени, в течение которого, происходили вычисления. Так при линейном алгоритме программы и при небольшом количестве уравнений, расчет системы происходит довольно быстро. Например, матрицу из 4 уравнений, программа приводит к ступенчатому виду буквально за 0,1 секунды. С увеличением числа уравнений, увеличивается и время. И уже 1000 уравнений преобразовываются в течение 17 секунд. Эти задержки во времени ощутимы и не совсем приятны. Чтобы не возникло этого явления, к программе применяется распараллеливание.

Распараллеливание проводилось на уровне данных. Одна и та же задача: поочередное вычитание одного уравнения из другого, применялась к множеству элементов данных. Это множество разбивалось на определенное количество порций.



Как только формируется матрица, на экран выводится соответствующее сообщение и в работу вступает параллелизм, при котором каждый поток производит вычисления в рамках своей порции. В результате чего в работе принимают участие столько ядер компьютера, сколько мы указали потоков. По завершению работы программы на экран выводится сообщение об окончании вычислений и время.

Как было сказано выше, вычисление 1000 уравнений заняло 17 секунд. Сформировав 2 порции по 500 уравнений, и задав 2 потока, время работы на 2-х ядерном компьютере уменьшилось до 10 секунд, на 4-х ядерном – до 9. Далее при увеличении потоков и уменьшении порций время вычислений на 2-х ядерном компьютере не уменьшалось, а при 6 потоках стало наоборот увеличиваться. Что же касается испытуемого 4-х ядерного компьютера, то при увеличении до 4-х потоков, время уменьшилось еще на 6 секунд. А далее произошла та же ситуация, что и в случае 2-х ядерного компьютера. Сначала время оставалось неизменным, а на 7 потоке стало увеличиваться. Это связано с тем, что возникает перенасыщенность потоками, они только мешают друг другу в вычислениях. Таким образом, оптимальное количество потоков, при котором вычисления будут осуществляться с максимальной скоростью, равно числу ядер в компьютере.

Мир не стоит на месте. Каким он будет через 10, 20 лет неизвестно. Ясно одно, что в сфере компьютеров будут еще перемены, и, может, возникнет другое решение, при котором в работе будут задействованы все ядра, ну а пока распараллеливание является оптимальным решением при работе с большим количеством данных.

Литература

1. Карпов В.Е. Введение в распараллеливание алгоритмов и программ / В.Е. Карпов // Компьютерные исследования и моделирование. – 2010. - №3. – С.232-240.
2. Сбитнев Ю. Распараллеливание программ [Электронный ресурс]. – [URL] <http://cluster.linux-ekb.info/pp.php>
3. Левин М. Средства автоматизированного распараллеливания программ [Электронный ресурс]. – [URL] <http://www.intuit.ru/studies/courses/1112/232/lecture/6033>