Г. И. Борзунов, Т. В. Петрова, Е. А. Сучкова

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ МАСШТАБИРОВАНИЯ НАБОРА ЗАДАЧ ПРИ ПОИСКЕ ЭКСТРЕМАЛЬНЫХ РАЗБИЕНИЙ

В реализации криптосистем [1] и в исследованиях вычислительной стойкости криптоалгоритмов [2] все большее значение приобретают распределенные вычисления. Для эффективной реализации распределенных вычислений требуется обеспечить равномерную вычислительную нагрузку для всех процессоров кластера и минимальные информационные потоки передачи данных между этими процессорами [3, 4]. Математической моделью решения этой задачи является поиск разбиения множества вершин взвешенного графа на непересекающиеся подмножества (блоки) с максимально близкими значениями суммарных весов вершин подмножеств (блоков) и при минимальном значении максимальной суммы весов ребер, соединяющих вершины из разных подмножеств. При этом вершины графа соответствуют подзадачам и взвешены временной сложностью этих подзадач, а ребра представляют собой обмен сообщениями между подзадачами и взвешены соответствующими коммуникационными затратами. Если задано число используемых процессоров n, то данная задача формулируется следующим образом. Пусть задан неориентированный взвешенный граф G(V,E) с числом вершин, равным m: $V = \{v_1,\ldots,v_m\}$, и числом ребер, равным k: $E = \{e_1,\ldots,e_k\}$. Функция F1 ставит в соответствие каждой вершине графа v_i ее вес q_i, аналогично функция F2 ставит в соответствие каждому ребру графа e_i его вес r_i. Рассмотрим разбиения множества вершин V графа G(V,E) на n непересекающихся подмножеств $(V_1,\ldots,V_n)$. Этими подмножествами вершин определяются подграфы $G_1(V_1,E_1),\ldots, G_n(V_n,E_n)$. Введем обозначение $C_i(V_i)$ для совокупности ребер, соединяющих вершины i-го подмножества с вершинами всех других подмножеств разбиения. Тогда $Q_i = \sum F1(V_i)$ будет представлять собой сумму весов вершин i-го подмножества, а $R_i = \sum F2(C_i(V_i))$ — сумму весов ребер данного сечения по i-му подмножеству вершин того же разбиения. Теперь задача оптимального разделения графа состоит в поиске разбиения множества его вершин $P = \{ V_1,\ldots,V_n\}$, для которого выполняется следующее условие:

$$Qmax = \max((Q_i + R_i), I \in \{1, 2, \ldots n\}) \rightarrow \min. \quad (1)$$

Предложенные ранее алгоритмы Eq4_0 и Eq4_1 [5, 6] обеспечивают равномерную вычислительную нагрузку процессоров, число которых не превышает 4. Для выявления причин, ограничивающих эффективность применения указанных алгоритмов, следует рассмотреть базовую операцию, используемую в этих алгоритмах при масштабировании. В качестве базовой операции (подзадачи) был принят поиск экстремального (в смысле условия 1) разбиения среди разбиений, число блоков которых задано как параметр. Этот параметр представлял собой целое число n, принадлежащее последовательности (1, 2, …, m), где m — число элементов в множестве, для которого осуществляется поиск экстремального разбиения. Таким образом, число проверяемых вариантов разбиений в каждой базовой операции (размерность подзадачи) было равно числу Стирлинга второго рода [7]: S(m, n), где m — число элементов в множестве, для которого перечисляются и анализируются разбиения, n — количество блоков в указанных разбиениях. При этом общая временная сложность поиска экстремального разбиения множества, мощность которого равна m, будет составлять число Белла (B(m)). При фиксированном значении m эти числа, в зависимости от числа блоков в разбиениях (n), имеют существенно разную величину. На рис. 1 приводится график зависимости значений S(m,n) от значений n при m = 10.

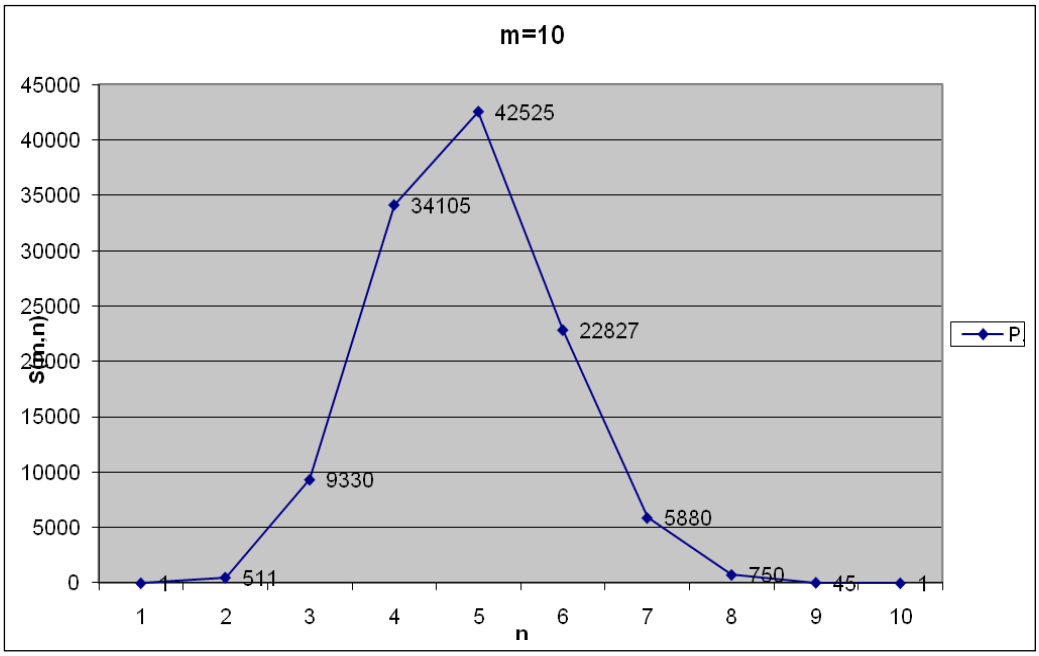

Рис. 1.

График на рис. 1 наглядно демонстрирует неравномерность временной сложности рассматриваемых базовых подзадач. Графики зависимостей S(m,n) от значений n при значениях m = 11, 12, …20 имели форму, аналогичную форме графика, приведенного на рис. 1. Максимум этих зависимостей определяет объем вычислений при выполнении максимальной по временной сложности базовой операции. Приведенные ниже данные (см. таблицу 1) позволяют количественно оценить неравномерность изменения чисел Стирлинга второго рода S(m, n) при фиксированном значении m в зависимости от числа блоков в разбиении (n = 1, 2, …, m): максимальное число Стирлинга второго рода превосходит среднее значение в 3,7 раза при m = 10 и в 5,9 раз при m = 20.

Таблица 1.

m	B(m)	(B(m)/ m)	max(S(m,n), n =1, 2, …, m)	max S(m, n)/ (B(m)/ m)	max S(m, n)/ B(m)
10	115975	11597,5	42525	3,666738521	0,366673852
11	678570	61688,18	246730	3,999631578	0,363602871
12	4213597	351133,1	1379400	3,928425049	0,327368754
13	27644437	2126495	9321312	4,383415586	0,337185814
14	190899322	13635666	63436373	4,652238744	0,332302767
15	1382958545	92197236	420693273	4,562970537	0,304198036
16	10480142147	655008884	3281882604	5,010439833	0,31315249
17	82864869804	4874404106	25708104786	5,27410207	0,310241298
18	6,82077E + 11	37893155898	1,97462E + 11	5,211032935	0,28950183
19	5,83274E + 12	3,06986E + 11	1,70975E + 12	5,569467658	0,293129877
20	5,17242E + 13	2,58621E + 12	1,51709E + 13	5,86609166	0,293304583

При этом доля объема вычислений, определяемых максимальным числом Стирлинга, от объема вычислений, выполняемых при полном переборе, составляет 37 % при m = 10 и 29 % при m = 20. В среднем при m = 10, 11, …, 20 максимальное S(m, n) превосходит в 4,7 раза среднее значение последовательности этих чисел S(m, n), определенное при фиксированном значении m и при изменении n от 1 до m, и доля объема вычислений, определяемых максимальным S(m, n),

от объема вычислений, выполняемых при полном переборе, составляет 32 %. Приведенные выше оценки показывают, что равномерное распределение вычислительной нагрузки в многопроцессорных системах при поиске экстремальных разбиений требует разделения базовой операции (подзадачи) перечисления и анализа разбиений, содержащих заданное число блоков, на более мелкие операции (подзадачи), требующие меньшего объема вычислений. В качестве новой базовой операции предлагается принять поиск экстремального разбиения среди разбиений, характеристические векторы которых соответствуют заданному вектору спецификации [5, 8]. Число векторов спецификаций, определяющих разбиение множества m на n блоков, определяется как число сочетаний из (m — 1) элемента по n элементов: C(m — 1, n). Пусть вектор спецификации разбиений множества на n блоков имеет вид: $(1^{\rho 1},2^{\rho 2},\ldots,n^{\rho n})$, где показатели степени определяют коэффициенты повторения оснований степени в векторе спецификации и удовлетворяют соотношениям: $\rho_1 + \rho_2 + \ldots + \rho_n = m$, $\rho_1 >= 1$, $\rho_2 >= 1$, … , $\rho_n >= 1$. Тогда число характеристических векторов, соответствующих данной спецификации, будет равно произведению:

$$t = 2^{(\rho 2 - 1)} * 3^{(\rho 3 - 1)} * \ldots * n^{(\rho n - 1)}. \qquad (2)$$

Согласно предложенной оценке, число характеристических векторов достигает максимума при максимальном значении $\rho_n = (m — n)$ и, как следствие, при обращении в 1 всех остальных параметров: $\rho_2 = 1$, $\rho_3 = 1$, …, $\rho_{n-1} = 1$. Вектор спецификации или просто спецификацию, определяющую максимальное количество характеристических векторов, будем называть максимальной спецификацией. В таблице 2 для каждого значения m (m = 10, 11, …, 20) приводятся: n — число блоков в разбиениях, при котором достигается максимум количества характеристических векторов; $n^{(m-n)}$ — максимальное число характеристических векторов, определяемых одной спецификацией при заданных значениях m и n; d — доля вычислительной нагрузки $d = (n^{(m-n)})/B(m)$, требуемой для генерации и анализа соответствующих максимальной спецификации характеристических векторов относительно полного объема вычислений при поиске минимального разбиения, удовлетворяющего заданным ограничениям. Анализ данных таблицы 2 показывает, что значение d меняется от 0,026945 до 0,001329 при изменении m от 10 до 20. В первом случае обеспечивается существование необходимых условий для равномерного распределения вычислительной нагрузки между 37 процессорами, а во втором случае (при m = 20) обеспечивается существование необходимых условий для равномерного распределения вычислительной нагрузки уже между 752 процессорами.

Таблица 2.

m	n	$n^{(m-n)}$	$d = (n^{(m-n)})/B(m)$
10	5	3125	0,026945
11	5	15625	0,023026
12	5	78125	0,018541
13	6	279936	0,010126
14	6	1679616	0,008798
15	6	10077696	0,007287
16	7	40353607	0,00385
17	7	282475249	0,003409
18	7	1977326743	0,002899
19	8	8589934592	0,001473
20	8	68719476736	0,001329

Приведенный на рис. 2 график представляет данные таблицы 2 в наглядной форме, которая позволяет выявить общую тенденцию: при увеличении мощности множества, для которого

осуществляется поиск разбиения, удовлетворяющего заданным условиям, уменьшается доля базовой операции от общего объема вычислений, что позволяет равномерно загрузить вычислениями большее число процессоров. Таким образом, повышается эффективность масштабирования параллельных вычислений с использованием предложенной базовой операции (подзадачи).

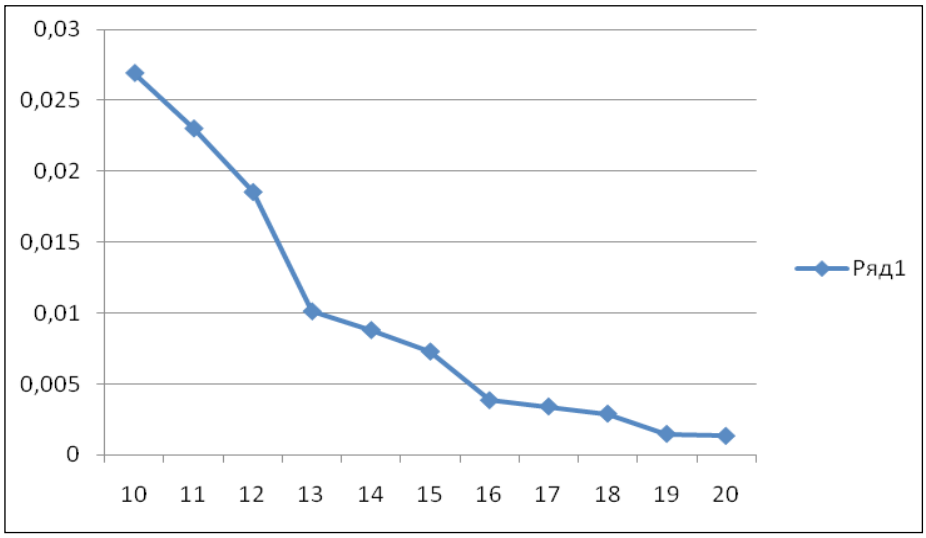

Рис. 2.

Выводы

Применение в качестве базовой подзадачи операции поиска экстремального разбиения среди разбиений, число блоков которых задано как параметр, обеспечивало балансировку вычислительной нагрузки при условии, что количество используемых процессоров не превышает 4.

Использование в качестве базовой подзадачи операции поиска экстремального разбиения среди разбиений, характеристические векторы которых соответствуют заданному вектору спецификации, позволяет значительно улучшить масштабирование набора подзадач при поиске экстремальных разбиений, удовлетворяющих заданным ограничениям: при увеличении размерности задачи m увеличивается число равномерно загружаемых процессоров. Так, при m = 10 обеспечивается существование необходимых условий для равномерного распределения вычислительной нагрузки между 37 процессорами, а при m = 20 обеспечивается существование необходимых условий для равномерного распределения вычислительной нагрузки уже между 752 процессорами. Это значительно повышает эффективность комбинаторного метода рационального распределения вычислительной нагрузки в многопроцессорных системах.

СПИСОК ЛИТЕРАТУРЫ:

1. *Нестеренко М. Ю., Полежаев П. Н.* Разработка параллельного алгоритма возведения длинных чисел в степень по модулю для криптосистемы RSA // Материалы шестого научно-практического семинара «Высокопроизводительные вычисления на кластерных системах» / Под ред. проф. Р. Г. Стронгина. СПб., 2007. С. 105–112.

2. *Бабенко Л. К., Курилкина А. М.* Распараллеливание криптоаналитического метода «разделяй и побеждай» для каскадных шифров // Материалы XII Всероссийской научно-практической конференции «Проблемы информационной безопасности в системе высшей школы». М.: МИФИ, 2005. С. 14–15.

3. *Гергель В. П.* Теория и практика параллельных вычислений. М.: БИНОМ. Лаборатория знаний, 2007. – 423 с.

4. *Борзунов Г. И., Войнов А. Е., Петрова Т. В.* Анализ методов повышения эффективности распределенных вычислений при решении задач безопасности информационных технологий // Безопасность информационных технологий. 2009. № 4. С. 57–60.

5. *Борзунов Г. И.* Двоичный поиск и параллельное программирование при минимизации количества необходимых проборок основ в ремиз // Известия вузов. Технология текстильной промышленности. 2009. № 2. С. 99–101.
6. *Борзунов Г. И., Войнов А. Е., Сучкова Е. А.* Выбор базового алгоритма для расчета минимального количества процессоров, обеспечивающего достижение заданного значения коэффициента ускорения // Безопасность информационных технологий. 2010. № 1. С. 45–46.
7. *Липский В.* Комбинаторика для программистов. М.: Мир, 1988. – 200 с.
8. *Романовский И. В.* Алгоритмы решения экстремальных задач. М.: Наука, 1977. – 352 с.