

ЭФФЕКТИВНОСТЬ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА ПОСТРОЕНИЯ ДИСКРЕТНОЙ МАРКОВСКОЙ МОДЕЛИ ОДНОРОДНОГО КЛАСТЕРА

Фельдман Л.П., Михайлова Т.В.

Кафедра ПМИИ ДонНТУ

feldman@r5.dgtu.donetsk.ua, tanya@r5.dgtu.donetsk.ua

Abstract

Feldman L., Michailova T. Efficiency of parallel algorithm of construction of discrete Markov's models. In the work there is presented a discrete Markov's model algorithm, which allows fastening a Markov's model's computation.

Введение

Одной из основных проблем, возникающих при проектировании, эксплуатации и оптимизации параллельных и распределенных ВС является выработка рекомендаций по рациональному использованию ресурсов вычислительной среды. Эффективным способом разрешения этой проблемы может быть использование непрерывных [1, 2, 3] или дискретных аналитических моделей [4].

Дискретная модель кластера с совместным разделением дискового пространства [5] при использовании методов построения дискретных Марковских моделей [6] описана в [7]. Однако анализ кластерных систем с помощью этих моделей при большом количестве решаемых задач на ЭВМ сопряжен с большими временными затратами, так как количество состояний дискретной Марковской модели комбинаторно возрастает при увеличении количества задач. Для уменьшения временных затрат надо рапараллелить алгоритм построения дискретной модели. Оценим трудоемкость этого алгоритма и характеристики результатов распараллеливания. Данная статья является продолжением и развитием работы [7].

Оценка трудоемкости алгоритма расчета характеристик функционирования однородной кластерной системы

Алгоритм построения дискретной Марковской модели [7] состоит из двух частей: вычислений матрицы переходных вероятностей и вектора стационарных вероятностей. Оценим трудоемкость алгоритма вычисления матрицы переходных вероятностей.

Для этого алгоритма [7] необходимо выполнить $2(N-1)$ операций сложения, чтобы определить количество задач при переходе из состояния

\overline{m} в состояние $\overline{m'}$ ([7], формула (3)) и одну операцию вычитания, чтобы вычислить величину $\Delta = \sum_{s=1}^N m_s - \sum_{s=1}^N m'_s$. Далее, для отсечения нулевых переходных вероятностей, выполним три операции сравнения ([7], условие (4)). Временная сложность этой части алгоритма для вычисления одного элемента матрицы переходных вероятностей P $2(N+1) t_{cl}$, а для вычисления всех элементов $- L \times L \times 2(N+1) t_{cl}$.

Посчитаем оставшееся количество элементов в матрице P после проверки условия $\Delta \in \{1, 0, -1\}$. Для первой строки, соответствующей размещению 0 задач по N узлам, останется $C_{0+N-1}^{N-1} + C_{1+N-1}^{N-1} = N$ элементов.

Для строк, соответствующих размещению M задач по N узлам, количество которых C_{M+N-1}^{N-1} , останется $C_{M+N-1}^{N-1} + C_{M+N-2}^{N-1}$ элементов, т.е. $C_{M+N-1}^{N-1} (C_{M+N-1}^{N-1} + C_{M+N-2}^{N-1})$. Определим количество элементов, которое останется в строке для состояния с произвольным количеством задач j (соответствующих размещению j задач по N узлам). В каждой такой строке матрицы P останется следующее количество состояний $C_{j+N-2}^{N-1} + C_{j+N-1}^{N-1} + C_{j+N}^{N-1}$. Всего строк C_{j+N-1}^{N-1} , тогда во всех строках, соответствующих размещению j задач по N узлам, останется $C_{j+N-1}^{N-1} (C_{j+N-2}^{N-1} + C_{j+N-1}^{N-1} + C_{j+N}^{N-1})$ элементов. Учитывая, что $j = \overline{1, M-1}$, общее количество элементов равно $\sum_{j=1}^{M-1} C_{j+N-1}^{N-1} (C_{j+N-2}^{N-1} + C_{j+N-1}^{N-1} + C_{j+N}^{N-1})$. Таким образом, общее количество

ненулевых элементов для всей матрицы переходных вероятностей после проверки условия $\Delta \in \{1, 0, -1\}$ вычисляется следующим образом

$$K_1 = N + \sum_{j=1}^{M-1} C_{j+N-1}^{N-1} (C_{j+N-2}^{N-1} + C_{j+N-1}^{N-1} + C_{j+N}^{N-1}) + C_{M+N-1}^{N-1} (C_{M+N-1}^{N-1} + C_{M+N-2}^{N-1})$$

На следующем этапе алгоритма определения матрицы переходных вероятностей для каждого из оставшихся K_1 элементов матрицы P выполняются такие операции: N операций сравнения для вычисления вектора $\overline{\alpha}$ ([7], формула(2)), N операций вычитания для нахождения вектора \overline{i} ([7], формула(2)) N операций вычитания для определения условия ([7], условие(4)). Временная сложность этого этапа алгоритма для вычисления одного элемента матрицы P составляет $3N t_{cl}$, а для всех элементов $- K_1 3N t_{cl}$.

В следующей части алгоритма для определения множества J ([7], формула(5)) необходимо выполнить $N-1$ операций сравнения, а для

вычисления вектора $\bar{\beta}$ ([7], формула (12)) – N операций сложения. Временная сложность этой части алгоритма вычисления матрицы переходных вероятностей определяется как $(2N-1) t_{cl}$, соответственно, для вычисления всех элементов – $K_l (2N-1) t_{cl}$.

Определим трудоемкость следующего этапа алгоритма. Максимальное количество элементов в индикаторах распределения \bar{Z}, \bar{Y} , учитывая ограничение ([7], формула (11)), равно $C_{\alpha_2+N-2}^{N-2}$. Наибольшее значение элементов вектора $\bar{\alpha}$ равно элементам вектора \bar{k} (количеству устройств в узле), соответственно. Поэтому количество элементов в индикаторах распределения \bar{Z}, \bar{Y} равно $C_{k_2+N-2}^{N-2}$. Количество операций сложения для вычисления индикаторов \bar{Z}, \bar{Y} , определяемых ([7], формулы (11),(13)), вычисляется как $C_{k_2+N-2}^{N-2} * 2N$. Временные сложности этого этапа алгоритма для вычисления \bar{z}, \bar{y} для одного элемента матрицы переходных вероятностей и для всех ненулевых элементов равны, соответственно, $C_{k_2+N-2}^{N-2} * 2N * t_{cl}$, $K_l * C_{k_2+N-2}^{N-2} * 2N * t_{cl}$.

В следующей части алгоритма построения матрицы переходных вероятностей для вычисления ([7], формулы (19),(20)) учитывая, что $z_s \leq \alpha_s \leq k_s$, $2 \sum_{s=1}^N k_s$, для вычисления вероятности $P_r(\bar{z})$ для одного элемента \bar{z} ([7], формула (21)), необходимо выполнить следующее

количество операций умножения: $(2k_2 - 6) + 2 \sum_{s=1}^N k_s$, временная сложность

определяется как $[(2k_2 - 6) + 2 \sum_{s=1}^N k_s] t_{умн}$. Вычисление вероятности $P_r(\bar{y})$

производится по формуле ([7], формула (22)), в зависимости от величины

Δ . Количество операций умножения равно $(2k_2 - 6) + 2 \sum_{s=1}^N k_s + 7$.

Временная сложность вычисляется следующим образом

$[(2k_2 - 6) + 2 \sum_{s=1}^N k_s + 7] t_{умн}$. Общее количество операций умножения для

вычисления одного слагаемого переходной вероятности, т.е. $P_r(\bar{z}, \bar{y})$ (определяемой формулой [7],(23)), равно сумме операций умножений для

вычисления $P_r(\bar{z})$ и $P_r(\bar{y})$, т.е. $(4k_2 - 5) + 4 \sum_{s=1}^N k_s$, а для всех ситуаций,

определяемых индикаторами \bar{Z}, \bar{Y} , количество операций умножения и сложения равно, соответственно, $C_{k_2+N-2}^{N-2} * [(4k_2 - 5) + 4 \sum_{s=1}^N k_s]$,

$C_{k_2+N-2}^{N-2} - 1$. Временная сложность для этой части алгоритма вычисляется следующим образом

$$C_{k_2+N-2}^{N-2} * [(4k_2 - 5) + 4 \sum_{s=1}^N k_s] t_{умн} + (C_{k_2+N-2}^{N-2} - 1) t_{сл}.$$

Всего ненулевых элементов в матрице P - K_I , следовательно, временная сложность последнего этапа алгоритма вычисления переходной вероятности $P_r(\bar{m} \rightarrow \bar{m}')$ равна

$$\{K_I * \{C_{k_2+N-2}^{N-2} * [(4k_2 - 5) + 4 \sum_{s=1}^N k_s] t_{умн} + (C_{k_2+N-2}^{N-2} - 1) t_{сл}\}.$$

Итак, общая временная суммарная трудоемкость для вычисления матрицы переходных вероятностей определяется

$$L \times L \times 2(N+1) t_{сл} + K_I * 3N t_{сл} + K_I \{((2N+1)C_{k_2+N-2}^{N-2} - 1) t_{сл} + C_{k_2+N-2}^{N-2} [(4k_2 - 5) + \sum_{s=1}^N k_s] t_{умн}\}.$$

(1)

Учитывая, что $t_{сл} \approx t_{умн}$, формула (1) примет вид

$$Tl_{пол} = L \times L \times 2(N+1) t_{он} + K_I * 3N t_{он} + K_I \{(2N + 4k_2 + \sum_{s=1}^N k_s - 4) C_{k_2+N-2}^{N-2}\} t_{он}$$

(2)

Для определения трудоемкости вычисления вектора стационарных вероятностей. Представим СЛАУ в виде

$$\bar{\pi}(k) = \bar{\pi}(0) P^Z, \quad (3)$$

где $\bar{\pi}(0)$ – начальное распределение вероятностей состояний. Пусть $Z=K$ – степень, при которой матрица P не изменяется.

Возведение матрицы P в степень Z потребует $(1+\log_2 Z)$ операций умножения матрицы. При последовательной реализации расчета вектора стационарных вероятностей общее количество вычислений определяется как

$$T2_{пол} = (L^3 * t_{умн} + L^2 (L-1) * t_{сл}) * (1+\log_2 Z) + L * (L * t_{умн} + (L-1) * t_{сл})$$

или

$$T2_{пол} = (L^3 + L^2 (L-1)) * t_{он} (1+\log_2 Z) + L * 2 * (L-1) * t_{он}. \quad (4)$$

Вычислительная сложность последовательного алгоритма построения дискретной Марковской модели кластера с общей памятью определяется суммой (2) и (4)

$$T_{\text{посл}} = T1_{\text{посл}} + T2_{\text{посл}} = L \times L \times 2(N+1)t_{\text{он}} + K_1 * 3N t_{\text{он}} + K_1 \left\{ (2N + 4k_2 + \sum_{s=1}^N k_s - 4) C_{k_2+N-2}^{N-2} \right\} t_{\text{он}} + (L^3 + L^2(L-1))(1 + \log_2 Z) t_{\text{он}} + L * 2(L-1) * t_{\text{он}} \quad (5)$$

Отображение параллельного алгоритма Марковской модели на решетку процессоров

Пусть процессор SIMD-структуры имеет топологию- решетку процессорных элементов (ПЭ) $p=n^2$. Каждый ПЭ может выполнить любую операцию за один такт; время обращения к запоминающему устройству ПЭ пренебрежительно мало по сравнению со временем выполнения операции.

На каждом ПЭ с индексами i, j вычисляется один элемент матрицы переходных вероятностей p_{ij} . Алгоритм вычисления переходной вероятности между двумя произвольными состояниями приведен в [6].

Временная сложность вычисления одного элемента матрицы переходных вероятностей размерности L определяется как

$$T1 = 5N t_{\text{он}} + (2N + 4k_2 + \sum_{s=1}^N k_s - 4) C_{k_2+N-2}^{N-2} t_{\text{он}} . \quad (6)$$

Число ненулевых элементов матрицы переходных вероятностей определяется величиной K_1 . В этом случае временная сложность параллельного алгоритма вычисления матрицы переходных вероятностей вычисляется следующим образом

$$T1_{\text{парал}} = \left(\left\lfloor \frac{K_1}{p} \right\rfloor + 1 \right) \{ 5N t_{\text{он}} + (2N + 4k_2 + \sum_{s=1}^N k_s - 4) C_{k_2+N-2}^{N-2} t_{\text{он}} \} . \quad (7)$$

Временная сложность последовательного алгоритма определяется формулой (2). Ускорение параллельного алгоритма вычисления матрицы переходных вероятностей на решетке процессоров, определяемое как отношение выражений (2) и (7) для различных значений количества обрабатываемых на кластере задач представлено на рис. 1 для решетки процессоров $p=64$; при этом изменялась конфигурация кластера, т.е. увеличивалось количество устройств (серверов и/или дисков) в узлах. Эффективность параллельного алгоритма вычисления матрицы переходных вероятностей на решетке процессоров с увеличением количества решаемых задач M незначительно растет (рис. 2) и мало изменяется (в сотых долях) с ростом количества процессоров.

Эффективность параллельного алгоритма вычисления матрицы переходных вероятностей на решетке процессоров зависит от значений M и не зависит от значений k_2, k_3 .

Распараллелим вычисление вектора стационарных вероятностей на решетку процессоров размерности $p=n^2$ с использованием блочного подхода методом Фокса [8, 9]. Размер блоков матрицы переходных вероятностей P равен $(L/\sqrt{p}) \times (L/\sqrt{p})$.

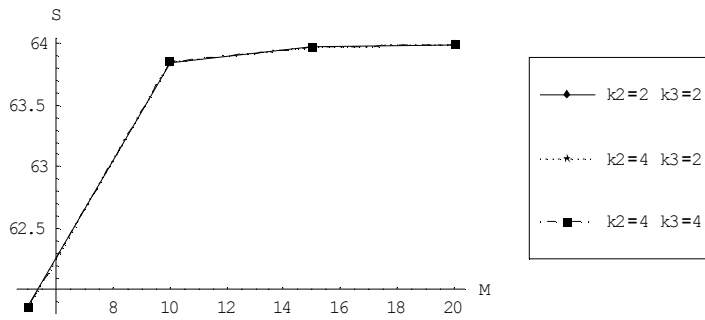


Рисунок 1. Ускорение параллельного алгоритма вычисления матрицы переходных вероятностей на решетке процессоров в зависимости от количества решаемых задач M

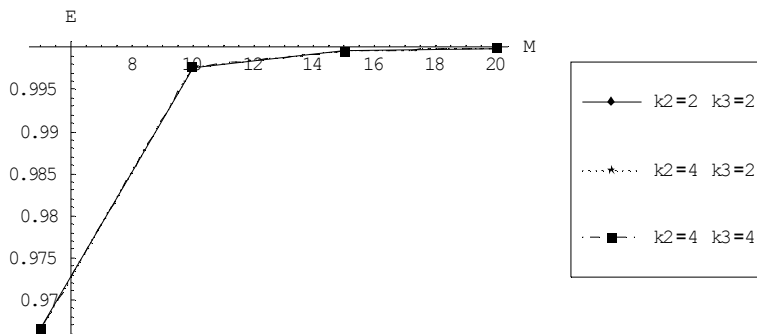


Рисунок 2. Эффективность параллельного алгоритма вычисления матрицы переходных вероятностей на решетке процессоров в зависимости от количества решаемых задач M

Временная сложность вычисления вектора стационарных вероятностей блочным методом Фокса определяется как

$$T_{2_{\text{парал}}} = \{2L^3 / p * t_{\text{сл}} + L^2 / p \sqrt{p} t_{\text{сдв}}\} (\log_2 Z + 1) + L^2 / p * t_{\text{сл}} + L / \sqrt{p} (\sqrt{p} - 1) t_{\text{сдв}}, \quad (8)$$

где $2L^3 / p$ – количество операций сложения для каждого процессора;

$t_{\text{сдв}}$ – максимальная длительность рассылки по строкам решетки;

$L^2 / p t_{\text{сдв}}$ – длительность рассылки блоков на одной итерации;

\sqrt{p} – количество итераций.

Порядок вычислений последовательного алгоритма – $L^3(1 + \log_2 Z)$.

Ускорение и эффективность параллельного алгоритма вычисления вектора стационарных вероятностей на решетке процессоров представлены на рис. 3, 4.

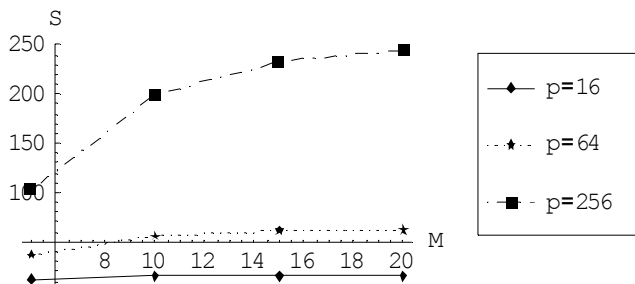


Рисунок 3. Ускорение параллельного алгоритма вычисления вектора стационарных вероятностей на решетке процессоров в зависимости от количества решаемых задач и процессоров

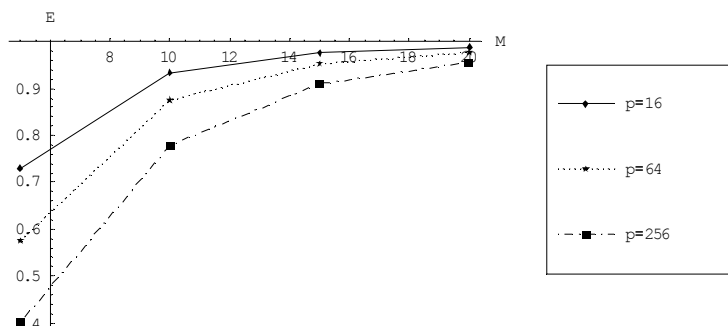


Рисунок 4. Эффективность параллельного алгоритма вычисления вектора стационарных вероятностей на решетке процессоров в зависимости от количества решаемых задач и процессоров

Временная сложность алгоритма построения модели Маркова определяется суммой (7) и (8).

Ускорение параллельного алгоритма асимптотически приближается к количеству процессоров (рис. 5) и не зависит от количества устройств (серверов и/или дисков) в узлах кластера. Поэтому в формулах оценки трудоемкости алгоритма вычисления матрицы переходных вероятностей можно отбросить слагаемые, соответствующие количеству устройств в узлах.

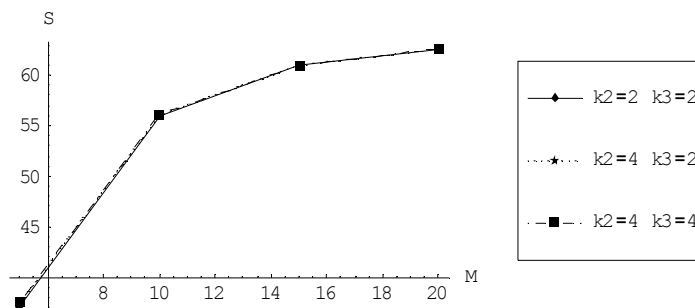


Рисунок 5. Ускорение параллельного алгоритма расчета модели Маркова на решетке процессоров в зависимости от количества решаемых задач M

Эффективность параллельного алгоритма расчета модели Маркова на решетке процессоров в зависимости от количества решаемых задач M , устройств и процессоров не изменяется и равна 0.99 при $M > 20$ (6).

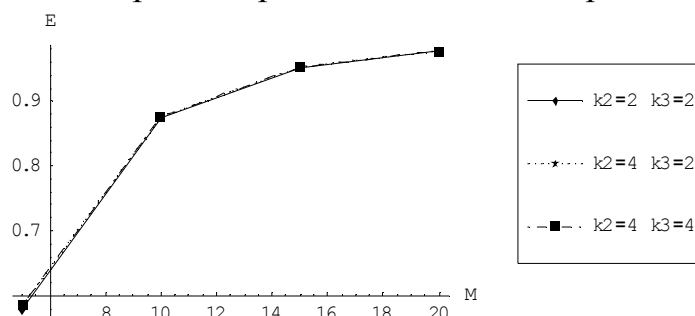


Рисунок 6. Эффективность параллельного алгоритма расчета модели Маркова на решетке процессоров в зависимости от количества решаемых задач M

Выводы

Полученные оценки трудоемкости алгоритма дискретной Марковской модели кластера с совместным использованием дискового пространства, которые зависят от структуры вычислительной среды, от количества обрабатываемых задач и от особенностей матрицы переходных вероятностей, и оценки распараллеливания этого алгоритма на SIMD-структуры позволяют эффективно использовать параллельный алгоритм дискретной модели на высокопроизводительных ВС, который обеспечивает высокую точность расчетов.

В дальнейшем можно получить оценки распараллеливания алгоритма дискретной Марковской модели кластера с совместным использованием дискового пространства на параллельные структуры другой топологии и выбрать оптимальную вычислительную среду по критерию цена/производительность.

Литература

1. Клейнрок Л. Вычислительные системы с очередями. — М.: Мир, 1979, 600с. Последовательно - параллельные вычисления: Пер. с англ. - М.: Мир, 1985. - 456 с.
2. Varki E. Response Time Analysis of Parallel Computer and Storage Systems //IEEE Transactions on Parallel and Distributed Systems, vol. 12, №11, nov.2001, pp.1146-1161
3. Cremonesi P., Gennaro C. Integrated Performance Models for SPMD Applications and MIMD Architectures //IEEE Transactions on Parallel and Distributed Systems, vol. 13, №7, jul.2002, pp.745-757
4. Авен О. И. и др. Оценка качества и оптимизация вычислительных систем. — М.: Наука, 1982, 464с.

5. Шнитман В. Современные высокопроизводительные компьютеры. Информационно-аналитические материалы центра информационных технологий, 1996: http://hardware/app_kis.
6. Фельдман Л.П., Дедищев В.А. Математическое обеспечение САПР. Моделирование вычислительных и управляющих систем. –Киев: УМК ВО, 1992, 256с
7. Михайлова Т.В. Параллельная реализация алгоритма построения дискретных моделей Маркова //Научные труды Донецкого государственного технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2003).-Донецк:ДонГТУ.- 2003 (в печати)
8. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. –Н.Новгород, ННГУ, 2001
9. Корнеев В.В. Параллельные вычислительные системы. –М.,1999, 312с.

Дата надходження до редакції 13.10.2006 р.