

УДК 519.6

## Реализация параллельного алгоритма построения марковских моделей

Фельдман Л.П., Михайлова Т.В., Ролдугин А.В.  
Донецкий национальный технический университет  
feldman@ r5.dgtu.donetsk.ua, tanya@ r5.dgtu.donetsk.ua

### Abstract

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

### Введение

Создание и использование современных суперкомпьютерных систем стало в наше время одним из основных направлений в ведущих странах мира. Одной из основных проблем, возникающих при проектировании и эксплуатации параллельных и распределенных вычислительных систем (ВС) является выработка рекомендаций рационального использования ресурсов вычислительной среды. Одним из эффективных способов решения этой проблемы может быть использование результатов 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]. Анализ кластерных систем с помощью этих моделей при большом количестве решаемых задач на ЭВМ требует больших временных затрат, так как количество состояний дискретной Марковской модели комбинаторно возрастает при увеличении количества задач. Для решения этой проблемы предложен параллельный алгоритм построения дискретной модели [7]. Сравним теоретическую трудоемкость этого алгоритма и характеристики распараллеливания с реальными. Данная статья является продолжением и развитием работ [7, 8].

### Теоретическая оценка трудоемкости параллельного алгоритма расчета характеристик функционирования кластерной системы

Алгоритм построения дискретной Марковской модели [7,8] состоит из двух частей: вычисления матрицы переходных вероятностей и вектора стационарных вероятностей.

Теоретическая оценка трудоемкости алгоритма вычисления матрицы переходных вероятностей, учитывая, что  $t_{ст} \approx t_{ум}$ , определяется

$$T I_{носл} = L \times L \times 2(N+1)t_{он} + K_I * 3N t_{он} + K_I \left\{ (2N + 4k_2 + \sum_{s=1}^N k_s - 4) C_{k_2+N-2}^{N-2} \right\} t_{он} \quad (1)$$

где

$L$  – количество состояний марковской модели;  
 $N$  – количество узлов вычислительного кластера;  
 $k_s$  – количество устройств в  $s$ -ом узле,  $s=1..N$ ;  
 $K_I$  – общее количество ненулевых элементов для всей матрицы переходных вероятностей после проверки условия  $\Delta \in \{1,0,-1\}$  вычисляется следующим образом

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

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

$$\bar{\pi}(k) = \bar{\pi}(0)P^Z, \quad (2)$$

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

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

Вычислительная сложность последовательного алгоритма построения дискретной Марковской модели кластера с общей памятью вычисляется как сумма (2) и (4)

$$T_{носл} = T I_{носл} + T2_{носл} \quad (4)$$

### Отображение параллельного алгоритма Марковской модели на решетку процессоров

Пусть процессор 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

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

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

$$T1_{парал} = (\lfloor \frac{K_1}{p} \rfloor + 1) \{ 5N t_{on} + (2N + 4k_2 + \sum_{s=1}^N k_s - 4) C_{k_2+N-2}^{N-2} t_{on} \} \quad (6)$$

Временная сложность последовательного алгоритма определяется (1). Ускорение параллельного алгоритма вычисления матрицы переходных вероятностей на решетке процессоров, определяемое как отношение выражений (1) и (6) 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.

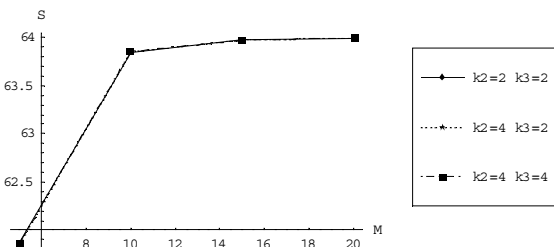


Рисунок 1- Ускорение параллельного алгоритма вычисления матрицы переходных вероятностей.

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  $(L/\sqrt{p}) \times (L/\sqrt{p})$ .

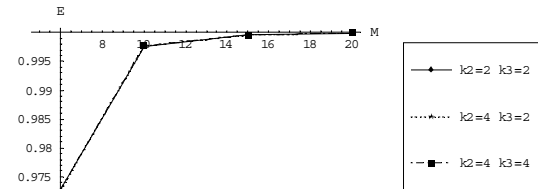


Рисунок 2- Эффективность параллельного алгоритма вычисления матрицы переходных вероятностей

Временная сложность вычисления вектора стационарных вероятностей блочным методом Фокса определяется как

$$T2_{парал} = \{ 2L^3 / p * t_{cl} + L^2 / p \sqrt{p} t_{cov} \} * (\log_2 Z + 1) + L^2 / p * t_{cl} + L / \sqrt{p} (\sqrt{p} - 1) t_{cov} \quad (7)$$

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

$t_{cov}$  - максимальная длительность рассылки по строкам решетки;

$L^2 / p t_{cov}$  - 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, рис.4.

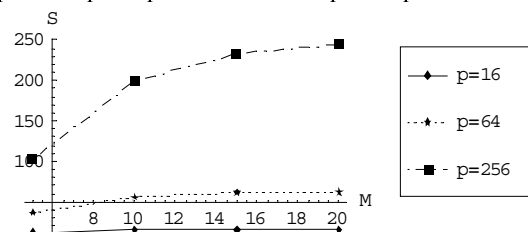


Рисунок 3 - Ускорение параллельного алгоритма вычисления вектора стационарных вероятностей.

Временная сложность алгоритма построения модели Маркова определяется суммой (6) и (7). Ускорение параллельного алгоритма асимптотически приближается к количеству процессоров (рис.5) и не зависит от количества

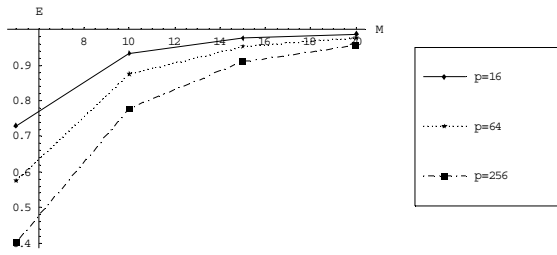


Рисунок 4- Эффективность параллельного алгоритма вычисления вектора стационарных вероятностей.

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.

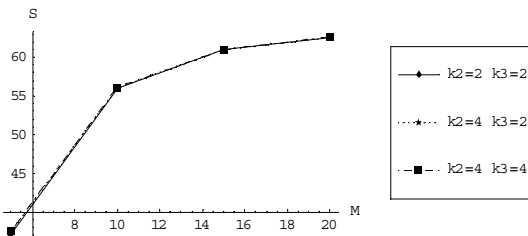


Рисунок 5- Ускорение параллельного алгоритма расчета модели Маркова.

Эффективность параллельного алгоритма расчета модели Маркова на решетке процессоров в зависимости от количества решаемых задач  $M$ , устройств и процессоров не изменяется и равно 0.99 при  $M > 20$  (5).

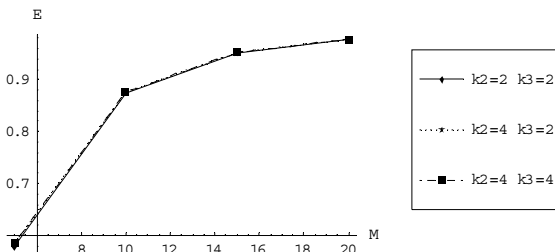


Рисунок 6- Эффективность параллельного алгоритма расчета модели Маркова

### Реализация параллельного алгоритма

Вычисление элемента матрицы переходных вероятностей не зависит от соседних элементов, следовательно, в основе параллельной реализации этой части параллельного алгоритма можно использовать принцип распараллеливания по данным.

В качестве базовой подзадачи взят расчет строки матрицы переходных вероятностей, которая характеризуется одинаковой вычислительной трудоемкостью, что позволяет сбалансировать загрузку вычислительных модулей кластера.

Каждый процесс определяет какие строки матрицы переходных вероятностей должен вычислить и отправить управляющему процессу. Схема параллельного алгоритма приведена на рис.7.

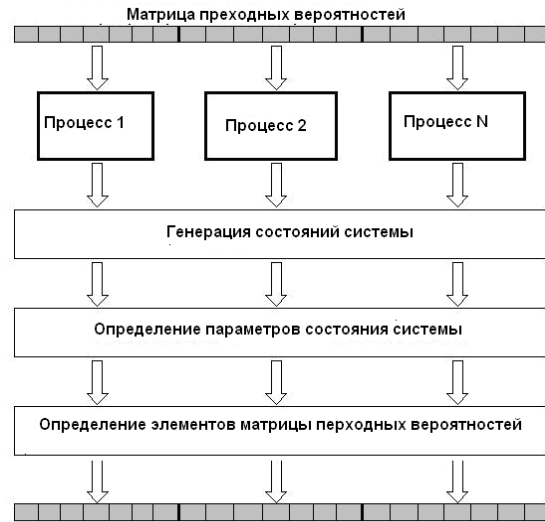


Рисунок 7 - Параллельный алгоритм построения матрицы переходных вероятностей

Расчет стационарных вероятностей реализован с использованием итерационного алгоритма, в котором в качестве базового используется алгоритм умножения матрицы на вектор. Схема алгоритма расчета вектора стационарных вероятностей представлена на рис.8.

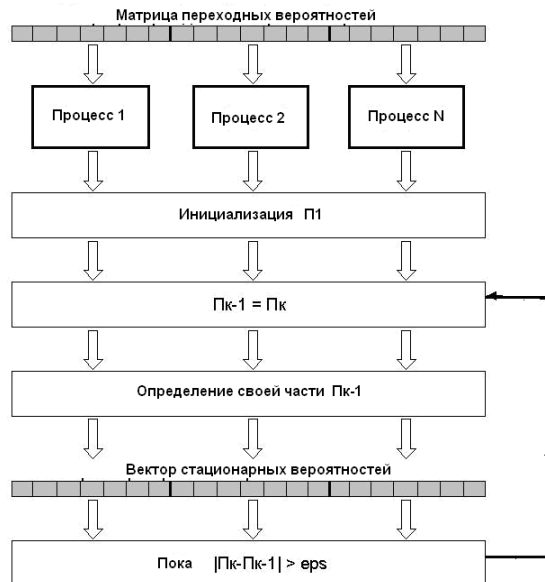


Рисунок 8 - Параллельный алгоритм расчета вектора стационарных вероятностей

Расчет задачи проводился на кластере, состоящего из двух, трех и четырех вычислительных модулей, реализованных на базе

процессоров AMD Athlon 3000+ (2ГГц), и сети Fast Ethernet.

Общее время (в секундах) для расчета характеристик моделируемого кластера приведено в табл.1.

Доля расчета матрицы переходных вероятностей от общего времени работы задачи составляет 0.65 (табл.2), что подтверждает теоретические расчеты.

Таблица 1. Общее время (в секундах) работы задачи

Размер матрицы	Количество процессоров		
	2	3	4
1771	57	40	31
3276	192	132	105
7770	1021	731	596

Таблица 2. Доля расчета матрицы переходных вероятностей от общего времени работы

Размер матрицы	Количество процессоров		
	2	3	4
1771	0.74	0.69	0.66
3276	0.73	0.69	0.66
7770	0.71	0.68	0.65

Ускорение параллельного алгоритма несколько(рис.9) хуже, чем теоретическое. Это объясняется тем, что топология кластера – «звезда», следовательно, время передачи между ВМ возрастает. Но несмотря на это, ускорение увеличивается с ростом количества процессоров для решаемых задач с количеством состояний 1771 и 3276.

Для задачи с количеством состояний 56 ускорение меньше 0.5. Следовательно, для этой задачи количество обменов превышает количество вычислительных операций на ВМ.

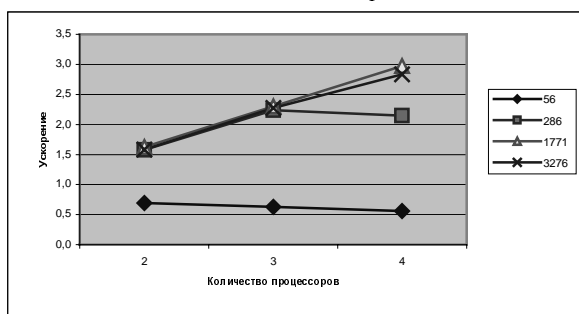


Рисунок 9 – Ускорение параллельного алгоритма

## 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.

При реализации на кластере параллельного алгоритма дискретной марковской модели кластера доказана его достаточно высокая эффективность и ускорение при большой размерности решаемых задач.

В дальнейшем можно определить оптимальное количество процессоров для распараллеливания подобных алгоритмов 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](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. Параллельный алгоритм построения дискретной модели Маркова // Искусственный интеллект. Интеллектуальные и многопроцессорные системы. Материалы Международной научно-технической конференции, 25-30 сентября 2006г., Таганрог-Донецк-Минск, 2006.
8. Фельдман Л.П., Михайлова Т.В. Использование аналитических методов для оценки эффективности многопроцессорных вычислительных систем. //Электронное моделирование, Т.29, №2, 2007.- С.17-27
9. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. –Н.Новгород, ННГУ, 2001
10. Корнеев В.В. Параллельные вычислительные системы. –М.,1999, 312с.

Поступила в редколлегию 10.03.2009