

УДК 681.3

Л.П. Фельдман Т.В. Михайлова

Донецкий национальный технический университет, г.Донецк, Украина f
eldman@ r5.dgtu.donetsk.ua, tanya@ r5.dgtu.donetsk.ua

Дискретная модель Маркова однородного кластера

В статье предлагается дискретная марковская модель однородного кластера и оценка ее трудоемкости.

Для исследования эффективности функционирования кластеров и выработки рекомендаций рационального использования ресурсов могут быть применены непрерывные [1-3] или дискретные аналитические модели [4].

Дискретная модель по сравнению с непрерывной – более точная. Границы использования этих двух моделей приведены в [5]. Однако анализ кластерных систем с помощью дискретных 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 распараллелить алгоритм построения дискретной модели кластера и оценить трудоемкость этого алгоритма и характеристики результатов распараллеливания. Построим дискретную модель однородного кластера и оценим ее трудоемкость.

1. Дискретная модель однородного кластера

Однородные кластеры с совместным использованием дискового пространства [6] используют СУБД Oracle Parallel Server и Informix [7].

Структурная схема марковской модели кластера с совместным использованием дискового пространства приведена на рис. 1. Количество серверов-приложений – $N1$. Каждый из них может обратиться к данным, расположенным на дисках, количество которых $N2$. Ввиду ограниченных вычислительных возможностей будем считать, что количество задач, обрабатываемое такой вычислительной системой, не более M .

Допустим, задачи, обрабатываемые на таком кластере, однородные и имеют следующие характеристики: p_{12} – вероятность запроса к одному из $N1$ серверов, p_{23} – вероятность запроса к одному из $N2$ дисков, p_{21} – вероятность завершения обслуживания одним из $N1$ серверов, p_{10} – вероятность завершения обслуживания задачи, q_0 – вероятность появления новой задачи.

Для вычислительных ресурсов примем, что длительности времен обслуживания заявок на входном узле, сервере приложений или дисковом массиве соответственно

имеют геометрическое распределение со средним, равным T_i , ($i = \overline{1,3}$). Тогда $q_i = \frac{\tau}{T_i}$ – вероятность завершения обслуживания задачи на входном узле, сервере приложений или дисковом массиве соответственно, ($i = \overline{1,3}$), r_i – вероятность продолжения обслуживания задачи на входном узле, сервере приложений или дисковом массиве соответственно, ($i = \overline{1,3}$), $r_i = 1 - q_i$.

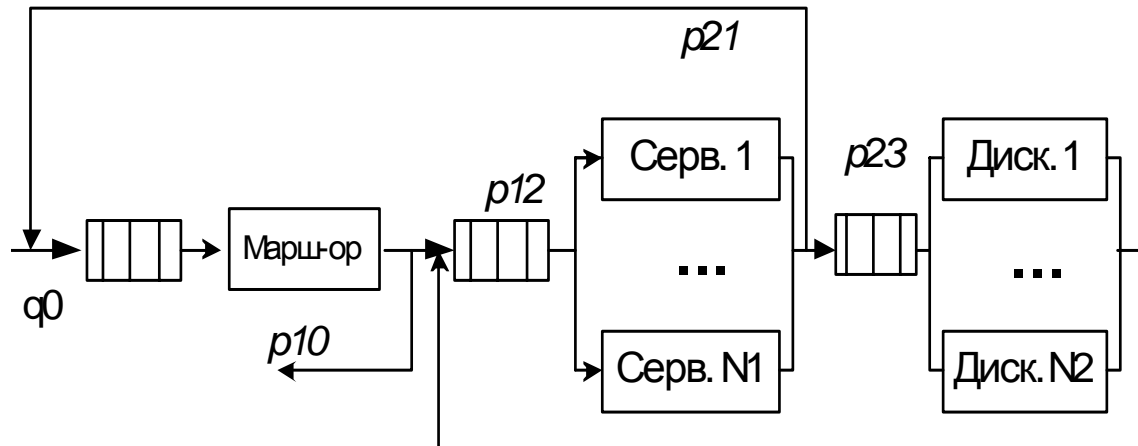


Рисунок 1 – Структурная схема Марковской модели кластера с совместным использованием дискового пространства

Для построения дискретной модели кластера, приведенного на рис.1 используем методику, описанную в [7].

За состояние системы примем размещение M заявок по трем узлам $\bar{m} = (m_1, m_2, m_3)$, где m_i – количество задач в i -м узле. Определим все возможные состояния. Обозначим множество состояний через $S = \{ (m_1, m_2, m_3) \mid \sum_{i=1}^N m_i = M \}$. Число состояний системы для одного и того же количества задач $j = \sum_{i=1}^N m_s$ равно числу размещений j задач по N узлам и определяется по формуле $L_j = C_{j+N-1}^{N-1}$, общее количество состояний вычисляется так:

$$L = \sum_{j=0}^M L_j. \quad (1)$$

Вектор $\bar{k} = (k_1, k_2, k_3)$ определяет количество устройств в каждом узле ВС. Определим переходные вероятности для каждой пары состояний $P_{ij}^{(k)} = P\{S_j^{(k)} \mid S_i^{(k-1)}\}$, т.е. вероятности перехода из состояния S_i , в котором она находилась на $(k-1)$ -м шаге, в состояние S_j на k -м шаге. Определим матрицу переходных вероятностей P .

Рассмотрим переход из произвольного состояния \bar{m} в состояние \bar{m}' . В состоянии \bar{m} обрабатывается $j = \sum_{s=1}^N m_s$ задач, а в состоянии \bar{m}' – $j' = \sum_{s=1}^N m'_s$. Так как за один такт может поступить на обработку в ВС или покинуть ее одна задача, то разница $\Delta = j - j'$ должна удовлетворять условию

$$\Delta \in \{1, 0, -1\}. \quad (2)$$

Введем вектор $\bar{\alpha}$, s -я компонента которого определяет число

$$\alpha_s = \min(m_s, k_s), \quad s = \overline{1, N}, \quad (3)$$

загруженных устройств в s -м узле.

Вычислим вектор \bar{i}

$$\bar{i} = \bar{m} - \bar{m}' = (i_1, i_2, \dots, i_N), \quad (4)$$

каждая компонента i_s которого представляет изменение числа задач в s -м узле. Если $i_s > 0$, то на рассматриваемом переходе из узла s должны уйти минимум i_s задач. Если $i_s < 0$, то в узел s должны прийти $|i_s|$ задач. При любом переходе из \bar{m} число задач, обслуженных s -м узлом обработки, не может быть больше α_s , то есть

$$i_s \leq \alpha_s, \quad s = \overline{1, N}. \quad (5)$$

Определим множество J номеров узлов обработки, в которых $i_s < 0$, т.е.

$$J = \{s \mid i_s < 0\}, \quad s = \overline{1, N}, s \neq 2. \quad (6)$$

Если $J \neq \emptyset$, то величины $\gamma_s = |i_s|$ ($s \in J$) определяют минимально возможное число программ, которые поступают к тем узлам обработки, номера s которых принадлежат множеству J . Число программ, поступивших в один из таких узлов s равно $|i_s|$. Количество поступивших задач к распределительному узлу от серверов не может превышать количества работающих серверов (за исключением случая, когда появляется новая задача или покидает ВС обслуженная задача, о чем свидетельствует символ δ)

$$|i_1| + \delta \leq \alpha_2. \quad (7)$$

Аналогично, количество поступивших задач к дискам от серверов не может превышать количества работающих серверов

$$\sum_{s=3}^N |i_s| \leq \alpha_2. \quad (8)$$

От серверного узла задачи поступают и к пользователям и к дискам, поэтому общее количество ушедших задач не превышает количества работающих серверов

$$|i_1| + \delta + \sum_{s=3}^N |i_s| \leq \alpha_2. \quad (9)$$

Количество поступивших задач к серверам не превышает количества работающих клиентов и дисков

$$|i_2| \leq \alpha_1 + \sum_{s=3}^N \alpha_s. \quad (10)$$

Чтобы в общем виде описать алгоритм построения матрицы переходных вероятностей, надо откорректировать вектор \bar{i} следующим образом: $i_1 = i_1 + \delta$, а остальные компоненты оставить без изменения. Эта операция позволит не учитывать пришедшую и покинувшую в рассматриваемом такте задачу при распределении задач по узлам ВС.

Если $J \neq \emptyset$, то величина $\gamma_2 = \sum_{S \in J, S \neq 2} |i_s|$ определяет количество задач от серверов приложений к дискам и маршрутизатору. Если $J = \emptyset$, то $\gamma_2 = 0$.

Переход $\bar{m} \rightarrow \bar{m}'$ возможен, если соблюдаются условия (2), (5), (9) и (10).

Распределение программ, обслуженных в серверном узле, по узлам системы определим N -мерным вектором \bar{Z} , компонента которого равна количеству задач, обслуженных серверами и распределенных по остальным узлам системы:

$$z_2 = \sum_{s=1, s \neq 2}^N z_s. \quad (11)$$

Назовем этот вектор индикатором распределения задач, обслуженных серверным узлом по остальным узлам системы. Распределение задач, обслуженных узлами системы, зададим N -мерным вектором \bar{Y} , компонента y_s ($s = \overline{1, N}, s \neq 2$) которого равна числу задач, обслуженных s -м узлом, а компонента $y_2 = \sum_{s=1, s \neq 2}^N y_s$ равна общему числу задач, поступивших в серверный узел от всех остальных. Вектор \bar{Z} назовем индикатором распределения задач, обслуженных узлами системы.

Найдем множества допустимых индикаторов распределения задач \bar{Z} и \bar{Y} для произвольного перехода $\bar{m} \rightarrow \bar{m}'$, которые соответственно обозначим через $\mathfrak{Z} = \{ \bar{Z} \}$ и $\mathfrak{Y} = \{ \bar{Y} \}$. Число программ, которые могут поступить от серверного узла, заключено в следующих границах:

$$\gamma_2 \leq z_2 \leq \alpha_2. \quad (12)$$

Для определения возможности распределения этих задач по каналам введем вектор $\bar{\beta}$, компоненты которого определяются формулами

$$\beta_2 = \alpha_2 - \gamma_2; \quad (13)$$

$$\beta_s = \begin{cases} \alpha_s, & \text{если } i_s \leq 0; \\ \alpha_s - i_s, & \text{если } i_s > 0, \quad s = \overline{1, N}, \quad s \neq 2. \end{cases}$$

Компонента β_2 равна максимально возможному числу программ, которые дополнительно, сверх γ_2 , могут быть обслужены серверным узлом. Аналогично, β_s равна максимально возможному числу программ, которые дополнительно, сверх

обязательно поступивших в s -й узел обработки, могут еще поступить в него, если допустить, что в s -й узел поступает больше задач, тогда при переходе $\bar{m} \rightarrow \bar{m}'$ узел s должен обслужить более α_s задач, что невозможно.

$$\gamma_s = \begin{cases} |i_s|, & \text{если } i_s \leq 0; \\ 0, & \text{если } i_s > 0, \quad s = \overline{1, N}. \end{cases} \quad (14)$$

Вектор, компоненты которого определяются формулами (11) и (12), является допустимым для рассматриваемого перехода индикатором распределения по узлам системы задач, обслуженных серверным узлом.

Вектор $\bar{\gamma}$ определяет для данного перехода распределение по каналам минимального числа γ_2 программ, обслуженных процессорным узлом. Отсюда следует, что индикатор распределения программ \bar{Z} будет допустим для перехода $\bar{m} \rightarrow \bar{m}'$, если выполняются условия

$$\gamma_s \leq z_s \leq \gamma_s + \beta_s. \quad (15)$$

Чтобы сделать дальнейшие выводы, надо в векторах $\bar{\gamma}, \bar{\beta}, \bar{i}, \bar{\alpha}, \bar{Z}, \bar{Y}$ переставить местами первую и вторую компоненты.

Определим множество \mathfrak{Z} допустимых для перехода $\bar{m} \rightarrow \bar{m}'$ индикаторов распределения программ для общего случая. Эта задача сведется к задаче размещения l программ, где $0 \leq l \leq \beta_s$, по $N-1$ каналу, если в любом s -м из них не может быть более β_s программ. Множество \mathfrak{Z} является объединением всех таких размещений, полученных для каждого целого $l = \overline{0, \beta_s}$.

Обозначим через $\mathfrak{R}(l, n)$ множество размещений l задач по n узлам, если в любом из s узлов может быть не более β_s программ и соблюдается условие

$$0 \leq l \leq \sum_{s=1}^n \beta_{s+1}. \quad (16)$$

Элементы множества B могут быть получены по рекуррентной формуле:

$$B(l, n) = \bigcup_{j \in \{j_0, j_M\}} B(l-j, n-1) \times \{j\}, \quad (17)$$

где $0 \leq l \leq \sum_{s=1}^n \beta_{s+1}$, $j_0 = \max(0, l - \sum_{s=1}^{n-1} \beta_{s+1})$, $j_M = \min(l, \beta_{n+1})$.

Начальные условия вычисляются следующим образом:

$$\begin{aligned} B(0, n) &= \{(0, 0, \dots, 0)\}, \\ B(l, 1) &= \begin{cases} \{l\}, & \text{если } l \leq \beta_1, \\ 0, & \text{если } l > \beta_1. \end{cases} \end{aligned} \quad (18)$$

Множество всевозможных размещений $l = \overline{0, \beta_1}$ задач по $N-1$ каналу системы с учетом ограничений β_s получим как объединение множеств, определяемых формулой (17), то есть

$$D(\beta, N) = \bigcup_{l=\{0, \beta_1\}} \{l\} \times B(l, N-1). \quad (19)$$

Элемент $\bar{\delta} \in D(\beta, N)$ является N -мерным вектором, первая компонента которого равна $d_1 = \sum_{s=2}^N d_s = l$ числу задач, распределенных по $N-1$ узлу системы с учетом ограничений, определяемых формулами (14), то есть s -я компонента $0 \leq d_s \leq \beta_s$, $s = \overline{1, N}$. Все допустимые для перехода индикаторы распределения программ \bar{z} равны:

$$\bar{z} = \bar{\delta} + \bar{\gamma}. \quad (20)$$

Для каждого допустимого на переходе $\bar{m} \rightarrow \bar{m}'$ размещения $\bar{\delta} \in D(\beta_1, N)$, определяющего распределение по каналам системы программ, поступивших от процессоров, найдем соответствующий допустимый на том переходе индикатор \bar{y} числа задач, обслуженных каналами по формуле

$$\bar{y} = \bar{\delta} + \bar{e},$$

где

$$e_s = \begin{cases} i_s, & \text{если } i_s > 0, \\ 0, & \text{если } i_s \leq 0, s = \overline{2, N}, \end{cases} \quad (21)$$

$$e_1 = \sum_{s=2}^N e_s.$$

Для того, чтобы отобразить реальную структуру моделируемой ВС, переставим опять первую и вторую компоненты векторов $\bar{\gamma}$, $\bar{\beta}$, \bar{i} , $\bar{\alpha}$, \bar{Z} , \bar{Y} .

Рассчитаем матрицу переходных вероятностей.

Рассмотрим случай $\Delta = 0$, т.е. количество задач, обрабатываемых в ВС при переходе $\bar{m} \rightarrow \bar{m}'$, остается неизменным.

Найдем вероятность для события, определяемого произвольным допустимым для перехода $\bar{m} \rightarrow \bar{m}'$ индикатором распределения программ $\bar{z} = (z_1, z_2, \dots, z_N)$. Значение компоненты z_2 равно числу задач, поступивших в узлы от α_2 серверов, работающих в состоянии \bar{m} . Число различных комбинаций, когда из α_2 работающих процессоров z_2 завершают обслуживание своих программ, равно $C_{\alpha_2}^{z_2}$. Для любой такой комбинации, содержащей номера z_2 -го процессора, число различных размещений z_s обслуженных программ, когда в s -й канал поступают программы от z_s процессоров, равно числу перестановок из z_2 с повторениями, то есть

$$P(z_1, \dots, z_N) = \frac{P(z_2)}{P(z_1)P(z_3) \dots P(z_N)}, \quad (22)$$

где $z_2 = \sum_{\substack{s=1 \\ s \neq 2}}^N z_s$. Любой из остальных серверов, обслуживающих программы, может

продолжать ее обслуживание. Вероятность такого исхода равна величине, определяемой формулой

$$P(\bar{z}) = \left[\prod_{s=1, s \neq 2}^N (q_2 p_{2s})^{z_s} \right] r_2^{\alpha_2 - z_2}. \quad (23)$$

где $r_i = 1 - q_i$ и одинакова для всех таких исходов, тогда вероятность наступления события, определяемого индикатором \bar{z} , будет равна

$$P_r(\bar{z}) = C_{\alpha_2}^{z_2} P(\bar{z}) P(z_1, \dots, z_N). \quad (24)$$

Аналогично находится вероятность события, определяемого индикатором \bar{y} распределения обслуженных задач первым и третьим узлами. Надо учесть, что первый узел может покинуть задача и вместо нее поступить новая в случае, когда $y_1 = 0$. Вероятность такого события $(q_1 p_{10} q_0 + r_1 r_0)(1 - y_1)$. Если $y_1 = 1$, задача завершает обслуживание в первом узле и переходит во второй. Вероятность такого события $(q_1 p_{12} r_0) y_2$. Общая вероятность события, определяемого индикатором \bar{y} , равна

$$P_r(\bar{y}) = \begin{cases} ((q_1 p_{10} q_0 + r_1 r_0)(1 - y_1) + (q_1 p_{12} r_0) y_2) \prod_{s=3}^N C_{\alpha_s}^{y_s} (q_s p_{s2})^{y_s} r_s^{\alpha_s - y_s}, & j < M, \\ ((q_1 p_{10} + r_1)(1 - y_1) + (q_1 p_{12}) y_2) \prod_{s=3}^N C_{\alpha_s}^{y_s} (q_s p_{s2})^{y_s} r_s^{\alpha_s - y_s}, & j = M. \end{cases} \quad (25)$$

Таким образом, вероятность события, определяемого индикатором распределения программ \bar{z} и соответствующим ему индикатором обслуженных каналами программ \bar{y} , равна произведению вероятностей

$$P_r(\bar{z}, \bar{y}) = P_r(\bar{z}) P_r(\bar{y}), \quad (26)$$

вычисляемых по (24) и (25).

Вероятность перехода $\bar{m} \rightarrow \bar{m}'$ определяется теперь как сумма вероятностей событий для всех допустимых на этом переходе индикаторов \bar{z} и \bar{y} , вычисляемых по (24), то есть

$$P_r(\bar{m} \rightarrow \bar{m}') = \sum P_r(\bar{z}, \bar{y}) \quad (27)$$

для всех $\bar{z} \in \mathfrak{Z}(\bar{m} \rightarrow \bar{m}')$ и соответствующих $\bar{y} \in \mathfrak{R}(\bar{m} \rightarrow \bar{m}')$.

Рассмотрим случай $\Delta = -1$, т.е. количество задач, обрабатываемых в ВС при переходе $\bar{m} \rightarrow \bar{m}'$, увеличивается на одну. Если $m_2 = 0$, то вероятность перехода $\bar{m} \rightarrow \bar{m}'$ равна нулю, если $m_2 > 0$, то вероятность не равна нулю. Рассчитаем ее.

Вероятности $P(\bar{z})$ и $P_r(\bar{z})$ вычисляются по формулам (23) и (24). А вероятность $P_r(\bar{y})$ вычисляется по следующей формуле

$$P_r(\bar{y}) = \begin{cases} q_0(r_1(1-y_1) + (q_1 p_{12})y_1) \prod_{s=3}^N C_{\alpha_s}^{y_s} (q_s p_{s2})^{y_s} r_s^{\alpha_s - y_s}, & j < M, \\ (r_1(1-y_1) + (q_1 p_{12})y_1) \prod_{s=3}^N C_{\alpha_s}^{y_s} (q_s p_{s2})^{y_s} r_s^{\alpha_s - y_s}, & j = M. \end{cases} \quad (28)$$

Вероятность события, определяемого индикатором распределения программ \bar{z} и соответствующим ему индикатора обслуженных каналами программ y равна произведению вероятностей, вычисляемого по формулам (26), а вероятность перехода $\bar{m} \rightarrow \bar{m}'$ – как сумма вероятностей событий для всех допустимых на этом переходе индикаторов \bar{z} и y , вычисляемых по (24), то есть по (26) для всех $\bar{z} \in \mathfrak{Z}(\bar{m} \rightarrow \bar{m}')$ и соответствующих $y \in \mathfrak{R}(\bar{m} \rightarrow \bar{m}')$.

Рассмотрим случай $\Delta=1$. Количество задач, обрабатываемых в ВС при переходе $\bar{m} \rightarrow \bar{m}'$, уменьшается на одну. Если $m_l = 0$, то вероятность перехода $\bar{m} \rightarrow \bar{m}'$ равна нулю, если $m_l > 0$, то вероятность – ненулевая, вычислим ее. Вероятности $P(\bar{z})$ и $P_r(\bar{z})$ вычисляются по (23) и (24). А вероятность $P_r(\bar{y})$ вычисляется по следующей формуле

$$P_r(\bar{y}) = \begin{cases} r_0 q_1 p_{10} \prod_{s=3}^N C_{\alpha_s}^{y_s} (q_s p_{s2})^{y_s} r_s^{\alpha_s - y_s}, & j < M, \\ q_1 p_{10} \prod_{s=3}^N C_{\alpha_s}^{y_s} (q_s p_{s2})^{y_s} r_s^{\alpha_s - y_s}, & j = M. \end{cases} \quad (29)$$

Таким образом, вероятность события, определяемого индикатором распределения программ \bar{z} и соответствующим ему индикатором обслуженных каналами программ y равна произведению вероятностей, вычисляемой по (26), а вероятность перехода $\bar{m} \rightarrow \bar{m}'$ – как сумма вероятностей событий для всех допустимых на этом переходе индикаторов \bar{z} и y , вычисляемых по (24), то есть по (27) для всех $\bar{z} \in \mathfrak{Z}(\bar{m} \rightarrow \bar{m}')$ и соответствующих $y \in \mathfrak{R}(\bar{m} \rightarrow \bar{m}')$.

Для определения вектора стационарных вероятностей состояний $\bar{\pi} = (\pi_0, \pi_1, \dots, \pi_M)$ необходимо решить систему линейных алгебраических уравнений (СЛАУ),

$$\bar{\pi} = \bar{\pi} P, \quad (30)$$

соответствующую рассмотренной модели ВС.

Основные характеристики вычисляются аналогично как в [4].

Оценим трудоемкость последовательного алгоритма построения дискретной Марковской модели.

2. Оценка трудоемкости алгоритма

Алгоритм построения дискретной Марковской модели состоит из двух частей: вычисления матрицы переходных вероятностей и вектора стационарных вероятностей. Оценим трудоемкости каждой из частей и просуммируем их.

Оценим трудоемкость алгоритма вычисления матрицы переходных вероятностей.

Для первого этапа алгоритма необходимо выполнить $2(N-1)$ операций сложения, чтобы определить количество задач при переходе из состояния \bar{m} в состояние \bar{m}' и одну операцию вычитания, чтобы вычислить величину Δ . Далее, для отсеечения нулевых переходных вероятностей, выполним три операции сравнения (условие 2). Временная сложность первого этапа алгоритма для вычисления одного элемента матрицы P –

$$2(N+1) t_{cl}, \quad (35)$$

а для вычисления всех элементов матрицы переходных вероятностей –

$$L \times L \times 2(N+1) t_{cl}. \quad (36)$$

Посчитаем оставшееся количество элементов в матрице переходных вероятностей P после проверки условия (2). Для первой строки, соответствующей размещению 0 задач по N узлам, останется $C_{0+N-1}^{N-1} + C_{1+N-1}^{N-1} = N+1$ 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, то есть $C_{M+N-1}^{N-1} (C_{M+N-1}^{N-1} + C_{M+N-2}^{N-1})$.

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 (2) вычисляется так:

$$K_1 = N + \sum_{j=1}^{M-1} C_{j+N-1}^{N-1} (C_{j+N-2}^{N-1} + C_{j+N-1}^{N-1} + C_{j+N}^{N-1}) + C_{M+N-1}^{N-1} (C_{M+N-1}^{N-1} + C_{M+N-2}^{N-1}). \quad (37)$$

На втором этапе алгоритма определения матрицы переходных вероятностей (вычисление (3, 4)) для каждого из оставшихся K_1 элементов матрицы P выполняются такие операции: N операций сравнения для вычисления вектора $\bar{\alpha}$ (3), N операций вычитания для нахождения вектора \bar{i} (4), N операций вычитания для определения условия (5). Временная сложность этого этапа алгоритма для вычисления одного элемента матрицы P –

$$3N t_{cl}, \quad (38)$$

а для всех элементов матрицы P –

$$K_1 3N t_{cl}. \quad (39)$$

На третьем этапе алгоритма (вычисление (6), (13)) для определения множества J (6) необходимо выполнить $N-1$ операций сравнения, а для вычисления вектора $\bar{\beta}$ (13) – N операций сложения. Временная сложность третьего этапа алгоритма вычисления матрицы переходных вероятностей для одного элемента определяется как

$$(2N-1) t_{cl}, \quad (40)$$

соответственно, для вычисления всех элементов –

$$K_1 (2N-1) t_{cl}. \quad (41)$$

Определим трудоемкость следующего этапа алгоритма. Максимальное количество элементов в индикаторах распределения \bar{Z}, \bar{Y} , учитывая ограничение (12), равно $C_{\alpha_2+N-2}^{N-2}$. Наибольшее значение элементов вектора $\bar{\alpha}$ равно элементам вектора \bar{k} (количеству устройств в узле). Поэтому количество элементов в индикаторах распределения \bar{Z}, \bar{Y} равно

$$C_{k_2+N-2}^{N-2}. \quad (42)$$

Количество операций сложения для вычисления индикаторов \bar{Z}, \bar{Y} , определяемых (20) и (21), вычисляется как

$$C_{k_2+N-2}^{N-2} * 2N. \quad (43)$$

Временные сложности следующего этапа алгоритма ((14), (17) – (21)) для вычисления \bar{z}, \bar{y} для одного элемента матрицы переходных вероятностей и для всех ненулевых элементов равны соответственно:

$$C_{k_2+N-2}^{N-2} * 2N * t_{сл}, \quad (45)$$

$$K_1 * C_{k_2+N-2}^{N-2} * 2N * t_{сл}. \quad (46)$$

На последнем этапе алгоритма построения матрицы переходных вероятностей (вычисление (22) – (26)) количество операций для вычисления (22) и (23), учитывая, что $z_s \leq \alpha_s \leq k_s$, будет равно $2 \sum_{s=1}^N k_s$. Таким образом, чтобы вычислить вероятность $P_r(\bar{z})$ для одного элемента \bar{z} (24), необходимо выполнить следующее количество операций умножения

$$(2k_2 - 6) + 2 \sum_{s=1}^N k_s, \quad (47)$$

временная сложность которых определяется как

$$[(2k_2 - 6) + 2 \sum_{s=1}^N k_s] t_{умн}. \quad (48)$$

Вычисление вероятности $P_r(\bar{y})$ производится по одной из формул (25), (28), (29), в зависимости от величины Δ . Наибольшее количество операций умножения приходится на формулу (25) и равно

$$(2k_2 - 6) + 2 \sum_{s=1}^N k_s + 7. \quad (49)$$

Временная сложность нахождения вероятности $P_r(\bar{y})$ (25) вычисляется следующим образом:

$$[(2k_2 - 6) + 2 \sum_{s=1}^N k_s + 7] t_{умн}. \quad (50)$$

Общее количество операций умножения для вычисления одного слагаемого переходной вероятности, то есть $P_r(\bar{z}, \bar{y})$ (определяемой (26)), равно сумме операций умножений для вычисления $P_r(\bar{z})$ и $P_r(\bar{y})$:

$$(4k_2 - 5) + 4 \sum_{s=1}^N k_s, \quad (51)$$

а для всех событий, определяемых индикаторами \bar{Z}, \bar{Y} , количество операций умножения и сложения равно соответственно:

$$C_{k_2+N-2}^{N-2} * [(4k_2 - 5) + 4 \sum_{s=1}^N k_s], \quad (52)$$

$$C_{k_2+N-2}^{N-2} - 1. \quad (53)$$

Временная сложность для этой части алгоритма вычисляется следующим образом:

$$C_{k_2+N-2}^{N-2} * [(4k_2 - 5) + 4 \sum_{s=1}^N k_s] t_{y_{mn}} + (C_{k_2+N-2}^{N-2} - 1) t_{cl}. \quad (54)$$

Всего ненулевых элементов в матрице P равно K_1 , следовательно, временная сложность пятого этапа алгоритма вычисления переходной вероятности $P_r(\bar{m} \rightarrow \bar{m}')$ равна

$$K_1 * \{ C_{k_2+N-2}^{N-2} * [(4k_2 - 5) + 4 \sum_{s=1}^N k_s] t_{y_{mn}} + (C_{k_2+N-2}^{N-2} - 1) t_{cl} \}, \quad (55)$$

Итак, общая временная суммарная трудоемкость для вычисления матрицы переходных вероятностей определяется суммой (36, 39, 41, 46, 55) и равна

$$L * L * 2(N + 1) t_{cl} + K_1 * 3N * t_{cl} + K_1 * \{ ((2N + 1) C_{k_2+N-2}^{N-2} - 1) t_{cl} + C_{k_2+N-2}^{N-2} [(4k_2 - 5) + \sum_{s=1}^N k_s] t_{y_{mn}} \} \quad (56)$$

Учитывая, что $t_{cl} \approx t_{y_{mn}}$, (56) примет вид:

$$T1_{носл} = L \times L \times 2(N + 1) t_{on} + K_1 * 3N t_{on} + K_1 \{ (2N + 4k_2 + \sum_{s=1}^N k_s - 4) C_{k_2+N-2}^{N-2} \} t_{on}. \quad (57)$$

Определим трудоемкость вычисления вектора стационарных вероятностей. Представим СЛАУ (30) в виде

$$\bar{\pi}(k) = \bar{\pi}(0) P^R, \quad (58)$$

где $\bar{\pi}(0)$ – начальное распределение вероятностей состояний,

R – степень, при которой матрица P не изменяется.

Возведение матрицы P в степень R потребует $(1 + \log_2 R)$ операций умножения матрицы на матрицу вместо $R-1$ аналогичных операций, так как перемножение осуществляется по правилу: сначала перемножается матрица P на себя – получена степень 2, затем аналогично перемножается матрица P^2 на себя – получена степень 4 и так далее, пока не будет получена степень R . Всего таких операций для получения нужной степени – $(1 + \log_2 R)$.

При последовательной реализации расчета вектора стационарных вероятностей (58) временная сложность определяется как

$$T2_{носл} = (L^3 * t_{y_{mn}} + L^2 (L-1) * t_{cl}) * (1 + \log_2 K) + L * (L * t_{y_{mn}} + (L-1) * t_{cl})$$

или (при $t_{cl} \approx t_{y_{mn}}$)

$$T2_{носл} = (L^3 + L^2 (L-1)) * t_{on} (1 + \log_2 K) + L * 2 * (L-1) * t_{on}. \quad (59)$$

Временная сложность последовательного алгоритма построения дискретной марковской модели однородного кластера определяется суммой (57) и (59). Упростим вычисления, так как известны нулевые элементы матрицы переходных вероятностей для данного типа моделей, то есть условие (2) не обязательно вычислять для каждого состояния. Из любого состояния можно перейти в состояние с таким же количеством задач или на единицу больше/меньше. Число всех ненулевых элементов равно K_l (37). Поэтому заменим сомножитель $L \times L$ в первом слагаемом (57) на K_l , приведем подобные. После преобразований временная сложность дискретной марковской модели имеет вид:

$$T_{\text{пол}} = K_1 \times (5N + 2) t_{\text{он}} + K_1 \left\{ (2N + 4k_2 + \sum_{s=1}^N k_s - 4) C_{k_2+N-2}^{N-2} \right\} t_{\text{он}} + (L^3 + L^2(L-1))(1 + \log_2 K) t_{\text{он}} + L * 2(L-1) * t_{\text{он}}. \quad (60)$$

Заключение

Получены 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, который позволяет получить более точные характеристики.

Однако с увеличением количества состояний эта и подобные модели не считаются на традиционных ЭВМ. Дискретная марковская модель кластера легко распараллеливается на различные параллельные структуры. В связи с большим объемом изложенного невозможно привести результаты оценки эффективности параллельного алгоритма этой модели.

Литература

1. Клейнрок Л. Вычислительные системы с очередями. – М.: Мир, 1979. – 600 с.
2. Varki E. Response Time Analysis of Parallel Computer and Storage Systems // IEEE Transactions on Parallel and Distributed Systems. – 2001. – Vol. 12, № 11. – P. 1146-1161.
3. Cremonesi P., Gennaro C. Integrated Performance Models for SPMD Applications and MIMD Architectures // IEEE Transactions on Parallel and Distributed Systems. – 2002. – Vol. 13, № 7. – P. 745-757.
4. Авен О.И. и др. Оценка качества и оптимизация вычислительных систем. – М.: Наука, 1982. – 464 с.
5. Михайлова Т.В. Оценка точности непрерывной и дискретной моделей Маркова // Научные труды Донецкого государственного технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2005). – 2005. – Вып. 93. – С. 114-122
6. Шнитман В. Современные высокопроизводительные компьютеры. Информационно-аналитические материалы центра информационных технологий. – 1996. – http://hardware/app_kis.
7. Цилькер Б.Я., Орлов С.А. Организация ЭВМ и систем. – М.: Питер, 2004. – 668 с.
8. Фельдман Л.П., Дедищев В.А. Математическое обеспечение САПР. Моделирование вычислительных и управляющих систем. – Киев: УМК ВО, 1992. – 256 с.

Л.П. Фельдман Т.В. Михайлова

Дискретна модель Маркова однорідного кластера

У статті пропонується дискретна марковская модель однорідного кластера й оцінка її трудомісткості

Статья поступила в редакцию 29.06.2006.