2Ф

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 рапараллелить 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 $q_i=\dfrac{\tau}{T_i}$ – 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, ($i=\overline{1,3}$), $r_i=1-q_i$.

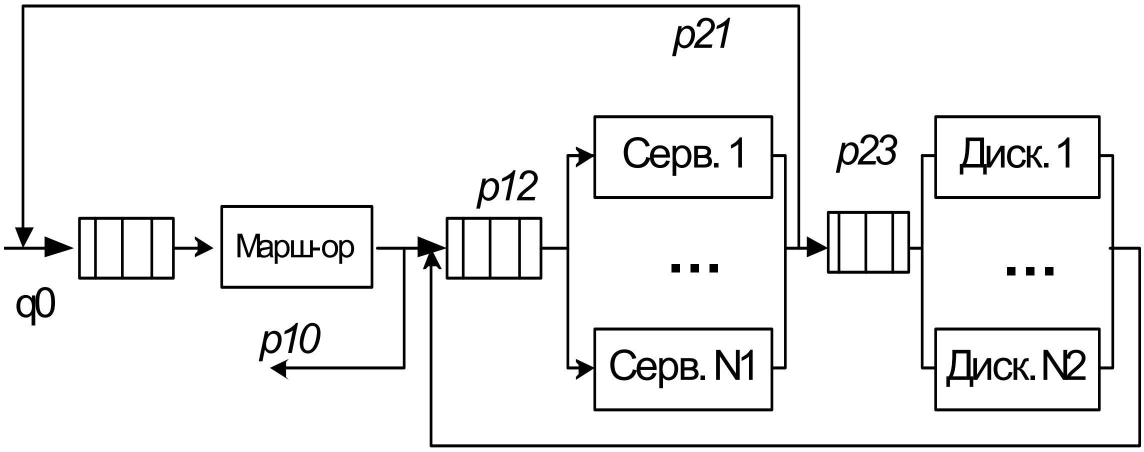

Рисунок 1 – Структурная схема Марковской модели кластера с совместным использованием дискового пространства

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

За состояние системы примем размещение M заявок по трем узлам $\overline{m}=(m_1,m_2,m_3)$, где m_i – количество задач в i-м узле. Определим все возможные состояния. Обозначим множество состояний через $S=\{(m_1,m_2,m_3)\,|\,\sum_{i=1}^{N} m_i = M\}$. 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:

$$L=\sum_{j=0}^{M} L_j\,. \tag{1}$$

Вектор $\overline{k}=(k_1,k_2,k_3)$ определяет количество устройств в каждом узле ВС. Определим переходные вероятности для каждой пары состояний $P_{ij}^{(k)}=P\{S_j^{(k)}\,|\,S_j^{(k-1)}\}$, 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 $\overline{m}$ в состояние $\overline{m'}$. В состоянии $\overline{m}$ обрабатывается $j = \sum_{s=1}^{N} m_s$ задач, а в состоянии $\overline{m'}$ – $j' = \sum_{s=1}^{N} m'_s$. Так как за один такт может поступить на обработку в ВС или покинуть ее одна задача, то разница $\Delta = j - j'$ должна удовлетворять условию

$$\Delta \in \{1,0,-1\}. \tag{2}$$

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

$$\alpha_S = min(m_S, k_S), \quad s = \overline{1,N}, \tag{3}$$

загруженных устройств в s-м узле.

Вычислим вектор $\bar{i}$

$$\bar{i} = \overline{m} - \overline{m'} = (i_1, i_2, ..., i_N), \tag{4}$$

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 $\overline{m}$ число задач, обслуженных s-м узлом обработки, не может быть больше α_s, то есть

$$i_S \le \alpha_S, \quad s = \overline{1,N}. \tag{5}$$

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

$$J = \{ S|_{i_S < 0} \}, \quad s = \overline{1,N}, s \ne 2. \tag{6}$$

Если $J \ne \varnothing$, 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)

$$|i_1| + \delta \le \alpha_2. \tag{7}$$

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

$$\sum_{s=3}^{N} |i_s| \le \alpha_2. \tag{8}$$

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

$$|i_1| + \delta + \sum_{s=3}^{N} |i_s| \le \alpha_2. \tag{9}$$

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

$$|i_2| \le \alpha_1 + \sum_{s=3}^{N} \alpha_s \ . \tag{10}$$

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

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

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

$$z_2 = \sum_{s=1, s \ne 2}^{N} z_S \ . \tag{11}$$

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

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

$$\gamma_2 \le z_2 \le \alpha_2 \ . \tag{12}$$

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

$$\beta_2 = \alpha_2 - \gamma_2; \tag{13}$$

$$\beta_S = \begin{cases} \alpha_S, & \text{если } i_S \le 0; \\ \alpha_S - i_S, & \text{если } i_S > 0, \quad s = \overline{1, N} \quad, s \ne 2. \end{cases}$$

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 $\overline{m} \to \overline{m'}$ узел s должен обслужить более α_s задач, что невозможно.

$$\gamma_S = \begin{cases} |\, i_S \,|, & если\ i_S \le 0; \\ 0, & если\ i_S > 0, \quad s = \overline{1,N}. \end{cases} \tag{14}$$

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

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

$$\gamma_{S \le} z_S \le \gamma_S + \beta_S\,. \tag{15}$$

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

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

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

$$0 \le l \le \sum_{s=1}^{n} \beta_{s+1}\,. \tag{16}$$

Элементы множества B могут быть получены по рекуррентной формуле:

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

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

Начальные условия вычисляются следующим образом:

$$B(0,n) = \{(0,0,...,0)\},$$
$$B(l,1) = \begin{cases} \{l\}, & если\ l \le \beta_1, \\ 0, & если\ l > \beta_1. \end{cases} \tag{18}$$

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

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

Элемент $\overline{\partial}\in D(\beta,N)$ 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 $0\le d_s\le\beta_s,\quad s=\overline{1,N}$. Все допустимые для перехода индикаторы распределения программ $\overline{z}$ равны:

$$\overline{z}=\overline{\partial}+\overline{\gamma}. \qquad (20)$$

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

$$\overline{y}=\overline{\partial}+\overline{e},$$

где

$$\overline{e_s}=\begin{cases}i_s, & если\ i_s>0,\\ 0, & если\ i_s\le 0,\ s=\overline{2,N},\end{cases}$$
$$e_1=\sum_{s=2}^{N}e_s. \qquad (21)$$

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

Рассчитаем матрицу переходных вероятностей.

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

Найдем вероятность для события, определяемого произвольным допустимым для перехода $\overline{m}\to\overline{m'}$ индикатором распределения программ $\overline{z}=(z_1,z_2,...,z_N)$. Значение компоненты z_2 равно числу задач, поступивших в узлы от α_2 серверов, работающих в состоянии $\overline{m}$. 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

$$P(z_1,...,z_N)=\frac{P(z_2)}{P(z_1)P(z_3)...P(z_N)}, \qquad (22)$$

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

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

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

$$P_r(\overline{z}) = C_{\alpha_2}^{z_2} P(\overline{z}) P(z_1, ..., z_N). \tag{24}$$

Аналогично находится вероятность события, определяемого индикатором $\overline{y}$ 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 $\overline{y}$, равна

$$P_r(\overline{y}) = \begin{cases} ((q_1 p_{10} q_0 + r_1 r_0)(1 - y_1) + (q_1 p_{12} r_0) y_1) \prod_{s=3}^{N} C_{\alpha_S}^{y_S} (q_S p_{S2})^{y_S} r_S^{\alpha_S - y_S}, j < M, \\ ((q_1 p_{10} + r_1)(1 - y_1) + (q_1 p_{12}) y_1) \prod_{s=3}^{N} C_{\alpha_S}^{y_S} (q_S p_{S2})^{y_S} r_S^{\alpha_S - y_S}, j = M. \end{cases} \tag{25}$$

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

$$P_r(\overline{z}, \overline{y}) = P_r(\overline{z}) P_r(\overline{y}), \tag{26}$$

вычисляемых по (24) и (25).

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

$$P_r(\overline{m} \to \overline{m'}) = \sum P_r(\overline{z}, \overline{y}) \tag{27}$$

для всех $\overline{z} \in \Im(\overline{m} \to \overline{m'})$ и соответствующих $\overline{y} \in \Re(\overline{m} \to \overline{m'})$.

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

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

$$P_r(\overline{y}) = \begin{cases} q_0(r_1(1-y_1)+(q_1p_{12})y_1)\prod_{s=3}^{N} C_{\alpha_S}^{y_S}(q_S p_{S2})^{y_S} r_S^{\alpha_S - y_S}, & j<M, \\ (r_1(1-y_1)+(q_1p_{12})y_1)\prod_{s=3}^{N} C_{\alpha_S}^{y_S}(q_S p_{S2})^{y_S} r_S^{\alpha_S - y_S}, & j=M. \end{cases} \quad (28)$$

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

Рассмотрим случай $\Delta = 1$. Количество задач, обрабатываемых в ВС при переходе $\overline{m} \to \overline{m'}$, уменьшается на одну. Если m_1 = 0, то вероятность перехода $\overline{m} \to \overline{m'}$ равна нулю, если m_1 >0, то вероятность – ненулевая, вычислим ее. Вероятности $P(\overline{z})$ и $P_r(\overline{z})$ вычисляются по (23) и (24). А вероятность $P_r(\overline{y})$ вычисляется по следующей формуле

$$P_r(\overline{y}) = \begin{cases} r_0 q_1 p_{10}\prod_{s=3}^{N} C_{\alpha_S}^{y_S}(q_S p_{S2})^{y_S} r_S^{\alpha_S - y_S}, & j<M, \\ q_1 p_{10}\prod_{s=3}^{N} C_{\alpha_S}^{y_S}(q_S p_{S2})^{y_S} r_S^{\alpha_S - y_S}, & j=M. \end{cases} \quad (29)$$

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

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

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

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

$$2(N+1)\ t_{сл}, \tag{35}$$

а для вычисления всех элементов матрицы переходных вероятностей –

$$L\times L\times\ 2(N+1)\ t_{сл}\,. \tag{36}$$

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 $\sum\limits_{j=1}^{M-1}C_{j+N-1}^{N-1}(C_{j+N-2}^{N-1}+C_{j+N-1}^{N-1}+C_{j+N}^{N-1})$.

Таким образом, общее количество ненулевых элементов для всей матрицы переходных вероятностей после проверки условия (2) вычисляется так:

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

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 $\overline{\alpha}$ (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 –

$$3N\ t_{сл}\,, \tag{38}$$

а для всех элементов матрицы P –

$$K_1\ 3N\ t_{сл}. \tag{39}$$

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

$$(2N\text{–}1)\ t_{сл}\,, \tag{40}$$

соответственно, для вычисления всех элементов –

$$K_1\ (2N\text{–}1)\ t_{сл}. \tag{41}$$

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

$$C_{k_2+N-2}^{N-2}. \tag{42}$$

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

$$C_{k_2+N-2}^{N-2} * 2N. \tag{43}$$

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

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

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

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

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

временная сложность которых определяется как

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

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

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

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

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

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

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

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

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

$$C_{k_2+N-2}^{N-2} - 1\,. \tag{53}$$

Временная сложность для этой части алгоритма вычисляется следующим образом:

$$C_{k_2+N-2}^{N-2} * [(4k_2 - 5) + 4\sum_{s=1}^{N} k_s]\ t_{умн} + (C_{k_2+N-2}^{N-2} - 1)\ t_{сл}\,. \tag{54}$$

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

$$K_1 * \{C_{k_2+N-2}^{N-2} * [(4k_2 - 5) + 4\sum_{s=1}^{N} k_s]\ t_{умн} + (C_{k_2+N-2}^{N-2} - 1)\ t_{сл}\,, \tag{55}$$

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

$$\begin{aligned} &L * L * 2(N+1)\ t_{сл} + K_1 * 3N * t_{сл} + \\ &+ K_1 * \{((2N+1)C_{k_2+N-2}^{N-2} - 1)\ t_{сл} + C_{k_2+N-2}^{N-2}[(4k_2 - 5) + \sum_{s=1}^{N} k_s]\ t_{умн}\} \end{aligned} \tag{56}$$

Учитывая, что $t_{сл} \approx t_{умн}$, (56) примет вид:

$$\begin{aligned} &T1_{посл} = L \times L \times 2(N+1)\ t_{оп} + K_1 * 3N\ t_{оп} + \\ &+ K_1\ \{(2N + 4k_2 + \sum_{s=1}^{N} k_s - 4)C_{k_2+N-2}^{N-2}\ \}\ t_{оп}\,. \end{aligned} \tag{57}$$

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

$$\overline{\pi}(k) = \overline{\pi}(0)P^R\,, \tag{58}$$

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

R – степень, при которой матрица P не изменяется.

Возведение матрицы P в степень R потребует $(1+log_2R)$ 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 – $(1+log_2R)$.

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

$$T2_{посл} = (L^3 * t_{умн} + L^2\ (L-1) * t_{сл}) * (1 + log_2K) + L * (L * t_{умн} + (L-1) * t_{сл})$$

или (при $t_{сл} \approx t_{умн}$)

$$T2_{посл} = (L^3 + L^2\ (L-1)\ ) * t_{оп}\ (1 + \log_2 K) + L * 2 * (L-1) * t_{оп}\,. \tag{59}$$

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_1 (37). Поэтому заменим сомножитель $L \times L$ в первом слагаемом (57) на K_1, приведем подобные. После преобразований временная сложность дискретной марковской модели имеет вид:

$$\begin{aligned} T_{посл} &= K_1 \times (5N + 2)\, t_{on} + \\ &+ K_1 \, \{(2N + 4k_2 + \sum_{s=1}^{N} k_s - 4) C_{k_2+N-2}^{N-2} \} \, t_{on} + \\ &+ (L^3 + L^2 (L-1))(1 + loq_2 K)\, t_{on} + L * 2(L-1) * t_{on}. \end{aligned} \tag{60}$$

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, № 1 1 . – 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 трудоміссткості

Статья поступила в редакцию 29.06.2006.