Математические вопросы кибернетики и вычислительной техники 23, 2004, 166-174.

Моделирование очереди многомашинного вычислительного комплекса*

Микаел К. Гюрджян и Владимир Г. Саакян

Институт проблем информатики и автоматизации НАН РА

Аннотация

При организации параллельных вычислений на многомашинном вычислительном комплексе (кластере) большую роль играет способ организации очереди и порядок обслуживания заданий. Одним из основных параметров такой системы является время ожидания задания в очереди до момента его поступления на выполнение. В данной работе рассматривается некоторый набор параметров, позволяющий составить рекуррентную систему уравнений и на их основе вычислить время ожидания.

При организации параллельных вычислений на многомашинном вычислительном комплексе (кластере) большую роль играет способ организации очереди и порядок обслуживания заданий [1,2,3].

Одним из основных параметров такой системы является время ожидания задания в очереди до момента его поступления на выполнение. Эта величина зависит от множества других параметров системы.

В данной работе рассматривается некоторый набор параметров, позволяющий составить рекуррентную систему уравнений для их вычисления и на их основе вычислить время ожидания.

Рассмотрим кластер, состоящий из M (M=1,2, ...) узлов. Для обслуживания на кластере поступает поток заданий. Каждое задание характеризуется длительностью выполнения β и числом требуемых узлов μ, где β положительная случайная величина, а μ целочисленная случайная величина, принимающая значения из множества 1, 2, ..., M.

Если кластер имеет не менее μ свободных узлов, то поступившее задание с параметрами (β, μ) начинает обслуживаться, а если нет, то становится в конец очереди.

Задания из очереди могут поступать на обслуживание в моменты завершения обслуживаний. Здесь наиболее типичными являются следующие схемы:

1. поступают на обслуживание задания в порядке поступления их в систему, при этом, учитываются все задания из очереди до первого, требующего больше узлов, чем имеется;
2. поступают на обслуживание задания в порядке поступления их в систему, при этом, задания в очереди перебираются до конца;
3. поступают на обслуживание задания в порядке числа требуемых узлов, при этом, задания в очереди перебираются до конца.

В данной работе рассматривается первая из описанных схем.

---

* Работа выполнена в рамках работ по проекту МНТЦ А-823

Для исследования системы введем следующие обозначения (i=1, 2, ...):

- $0=t_0<t_1<t_2<\ldots<t_i<\ldots$ – последовательные моменты времени, когда в систему поступают или покидают задания ($t_0=0$ вводится как точка отсчета времени для всех рассматриваемых в дальнейшем величин, если об этом не упоминается специально);
- $x_i=t_i-t_{i-1}$;
- ξ_i – промежуток времени между поступлениями i-1 и i-го задания;
- $\nu(t)$ – число заданий в очереди в момент времени t ;
- $\nu_i=\nu(t_i+0)$ – число заданий в очереди в момент времени t_i+0 (здесь и в дальнейшем момент t_i+0 означает, что в системе все изменения, связанные с поступлением или уходом заданий в момент t_i, уже учтены);
- $\delta(t)$ – число занятых узлов в момент времени t ;
- $\delta_i=\delta(t_i+0)$ – число занятых узлов в момент времени t_i+0 ;
- $\gamma(t)$ – число выполняемых заданий в момент времени t;
- $\gamma_i=\gamma(t_i+0)$ – число выполняемых заданий в момент времени t_i+0 ;
- $\Theta_i=\Theta(t_i+0)$ – число выполненных заданий в момент времени t_i+0 ;
- $\alpha(t)$ – число заданий, поступивших в систему до момента времени t;
- $\alpha_i=\alpha(t_i+0)$ – число заданий поступивших до момент времени t_i+0 (заметим, что $\alpha_i=\nu_i+\gamma_i+\Theta_i$);
- $Q(t)$ – вектор очереди в момент времени t, $Q(t)=\left(\left(b_1(t),n_1(t)\right),\left(b_2(t),n_2(t)\right),\ldots,\left(b_{\nu(t)}(t),n_{\nu(t)}(t)\right)\right)$;
- $Q_i=Q(t_i+0)$ – вектор очереди в момент времени t_i+0 , $Q_i=\left(\left(b_1^{(i)},n_1^{(i)}\right),\left(b_2^{(i)},n_2^{(i)}\right),\ldots,\left(b_{\nu_i}^{(i)},n_{\nu_i}^{(i)}\right)\right)$;
- $S(t)$ – вектор обслуживаемых заданий в момент времени t, $S(t)=\left(\left(c_1(t),m_1(t)\right),\left(c_2(t),m_2(t)\right),\ldots,\left(c_{\delta(t)_i}(t),m_{\delta(t)_i}(t)\right)\right)$;
- $S_i=S(t_i+0)$ – вектор обслуживаемых заданий в момент времени t_i+0 ; $S_i=\left(\left(c_1^{(i)},m_1^{(i)}\right),\left(c_2^{(i)},m_2^{(i)}\right),\ldots,\left(c_{\gamma_i}^{(i)},m_{\gamma_i}^{(i)}\right)\right)$, причем $0<c_1^{(i)}<c_2^{(i)}<\cdots<c_{\gamma_i}^{(i)}$.

Опишем состояние системы в момент времени t_1+0. В момент времени t_1 в систему поступает одно задание и сразу начинается его обслуживание. При этом $t_1=t_0+x_1$, $x_1=\xi_1$, $\nu_1=0$, $\delta_1=\mu_1$, $\gamma_1=1$, $\Theta_1=0$, $S_1=\left(\left(c_1^{(1)},m_1^{(1)}\right)\right)$, причем $c_1=\beta_1$, $m_1=\mu_1$.

Рассмотрим параметры состояния системы в момент времени t_{i+1} и попробуем выразить их рекуррентно через параметры системы в момент t_i.

Для этого введем индикаторную функцию χ(A), которая принимает значение 1, если событие A истинно и 0 в противном случае.

Так как $t_{i+1}=t_i+x_{i+1}$ есть момент времени, когда в систему или поступает (α_i+1)-ое задание, или же систему покидает (Θ_i+1)-ое обслуженное задание, то $x_{i+1}=\min\left(\xi_{\alpha_i+1},c_1^{(i)}\right)\chi(\gamma_i>0)+\xi_{\alpha_i+1}\chi(\gamma_i=0)$.

Вычислим число заданий в очереди в момент времени t_{i+1}. Для этого опишем возможные ситуации в системе в момент времени t_{i+1}, которые воздействуют на число заданий в очереди и попробуем выразить параметр ν_{i+1} через ν_i.

$$\nu_{i+1} = \begin{cases} \nu_i & \begin{cases} \text{a) если поступило новое задание, очередь пуста и число требуемых} \\ \quad \text{узлов для нового задания меньше числа свободных на кластере;} \\ \text{b) если систему покинуло обслуженное задание и очередь пуста;} \\ \text{c) если систему покинуло обслуженное задание, очередь не пуста} \\ \quad \text{и задание из головы очереди требует большего числа узлов,} \\ \quad \text{чем имеется свободных на кластере;} \end{cases} \\ \nu_i + 1 & \begin{cases} \text{a) если поступило новое задание, а очередь не пуста;} \\ \text{b) если поступило новое задание, очередь пуста, но число требуемых} \\ \quad \text{узлов для этого задания больше числа свободных на кластере;} \end{cases} \\ \nu_i - k & \begin{cases} \text{a) если систему покинуло обслуженное задание и из очереди} \\ \quad \text{на обслуживание пошли k заданий;} \end{cases} \end{cases}$$

$$\begin{aligned} \nu_{i+1} = \nu_i \Big[& \chi\left(\xi_{\alpha_i+1} < c_1^{(i)}, \nu_i = 0, \mu_{\alpha_i+1} \le \mathrm{M} - \delta_i\right) + \chi\left(\xi_{\alpha_i+1} > c_1^{(i)}, \nu_i = 0\right) + \\ & + \chi\left(\xi_{\alpha_i+1} > c_1^{(i)}, \nu_i > 0, n_1^{(i)} > \mathrm{M} - \delta_i + m_1^{(i)}, \gamma_i > 1\right)\Big] + \\ & + (\nu_i + 1)\left[\chi\left(\xi_{\alpha_i+1} < c_1^{(i)}, \nu_i > 0\right) + \chi\left(\xi_{\alpha_i+1} < c_1^{(i)}, \nu_i = 0, \mu_{\alpha_i+1} > \mathrm{M} - \delta_i, \gamma_i > 0\right)\right] + \\ & + (\nu_i - k)\left[\chi\left(\xi_{\alpha_i+1} > c_1^{(i)}, \nu_i = k, \sum_{j=1}^{k} n_j^{(i)} \le \mathrm{M} - \delta_i + m_1^{(i)}, k \ge 1\right) + \right. \\ & \left. + \chi\left(\xi_{\alpha_i+1} > c_1^{(i)}, \nu_i > k, \sum_{j=1}^{k} n_j^{(i)} \le \mathrm{M} - \delta_i + m_1^{(i)}, \sum_{j=1}^{k+1} n_j^{(i)} > \mathrm{M} - \delta_i + m_1^{(i)}, k \ge 1\right)\right]. \end{aligned}$$

Вычислим число выполняемых заданий в момент времени t_{i+1}. Для этого опишем возможные ситуации в системе в момент времени t_{i+1}, которые воздействуют на число выполняемых заданий в системе и попробуем выразить параметр γ_{i+1} через γ_i.

$$
\gamma_{i+1}=\begin{cases}
\gamma_i & \begin{cases}\text{a) если поступило новое задание, а очередь не пуста;}\\ \text{b) если поступило новое задание, очередь пуста и число требуемых}\\ \quad\text{узлов для нового задания больше числа свободных узлов на кластере;}\\ \text{c) если систему покинуло обслуженное задание, очередь не пуста, и из}\\ \quad\text{очереди на обслуживание поступило одно задание;}\end{cases}\\
\gamma_i-1 & \begin{cases}\text{a) если систему покинуло обслуженное задание и очередь пуста;}\\ \text{b) если систему покинуло обслуженное задание, очередь не пуста, а}\\ \quad\text{первое задание в очереди требует большего числа узлов, чем имеется}\\ \quad\text{свободных на кластере;}\end{cases}\\
\gamma_i+1 & \begin{cases}\text{a) если поступило новое задание, очередь пуста и число требуемых}\\ \quad\text{узлов для нового задания меньше числа свободных узлов на кластере;}\\ \text{b) если систему покинуло обслуженное задание и из очереди на}\\ \quad\text{обслуживание поступило два задания;}\end{cases}\\
\vdots & \\
\gamma_i+k & \begin{cases}\text{a) если систему покинуло обслуженное задание и из очереди на}\\ \quad\text{обслуживание поступило k+1 задание;}\end{cases}
\end{cases}
$$

$$
\begin{aligned}
\gamma_{i+1} = {} & \gamma_i\left[\chi\left(\xi_{\alpha_i+1}<c_1^{(i)},\nu_i>0\right)+\chi\left(\xi_{\alpha_i+1}<c_1^{(i)},\nu_i=0,\mu_{\alpha_i+1}>\mathrm{M}-\delta_i,\gamma_i>0\right)\right]+\\
&+(\gamma_i-1)\left[\chi\left(\xi_{\alpha_i+1}>c_1^{(i)},\nu_i=0\right)+\chi\left(\xi_{\alpha_i+1}>c_1^{(i)},\nu_i>0,n_1^{(i)}>\mathrm{M}-\delta_i+m_1^{(i)},\gamma_i>1\right)\right]+\\
&+(\gamma_i+1)\left[\chi\left(\xi_{\alpha_i+1}<c_1^{(i)},\nu_i=0,\mu_{\alpha_i+1}\le\mathrm{M}-\delta_i\right)+\right.\\
&+(\gamma_i+k)\left[\chi\left(\xi_{\alpha_i+1}>c_1^{(i)},\nu_i=k+1,k\ge0,\sum_{j=1}^{k+1}n_j^{(i)}\le\mathrm{M}-\delta_i+m_1^{(i)}\right)+\right.\\
&+\left.\chi\left(\xi_{\alpha_i+1}>c_1^{(i)},\nu_i>k+1,k\ge0,\sum_{j=1}^{k+1}n_j^{(i)}\le\mathrm{M}-\delta_i+m_1^{(i)},\sum_{j=1}^{k+2}n_j^{(i)}>\mathrm{M}-\delta_i+m_1^{(i)}\right)\right].
\end{aligned}
$$

Вычислим также число выполненных заданий в момент времени t_{i+1} и число заданий поступивших до момент времени t_{i+1}.

$$
\Theta_{i+1}=\Theta_i+\chi\left(\xi_{\alpha_i+1}<c_1^{(i)}\right),\ \alpha_{i+1}=\nu_{i+1}+\gamma_{i+1}+\Theta_{i+1}.
$$

Опишем вектор очереди. В момент времени t_{i+1} возможны следующие ситуации:

1. поступило новое задание и при этом:
   - очередь была пуста, число требуемых узлов для нового задания меньше числа свободных узлов на кластере, следовательно, задание поступает на обслуживание;
   - очередь пуста, но число требуемых узлов для нового задания больше числа свободных на кластере, следовательно, задание ставится в очередь;
   - очередь не пуста, следовательно, задание ставится в конец очереди.
2. обслуженное задание покинуло систему и при этом:
   - очередь пуста;
   - очередь не пуста, но число требуемых узлов для задания из головы очереди требует большего числа узлов, чем имеется свободных на кластере, следовательно, очередь не меняется;
   - очередь не пуста и из очереди на обслуживание поступает k (k=1,2,…,M) заданий, сумма требуемых узлов которых меньше числа свободных на кластере, следовательно, из головы очереди уходят k первых заданий.

Выразим вышеописанное в виде формул:

$$Q_{i+1}=\begin{cases} a)\ \left(\left(b_1^{(i+1)},n_1^{(i+1)}\right),\left(b_2^{(i+1)},n_2^{(i+1)}\right),\ldots,\left(b_{\nu_{i+1}}^{(i+1)},n_{\nu_{i+1}}^{(i+1)}\right)\right), \\ \qquad \text{где} \quad b_j^{(i+1)}=b_j^{(i)},\ n_j^{(i+1)}=n_j^{(i)},\ j=\overline{1..\nu_{i+1}}, \\ \qquad \text{если} \quad \nu_{i+1}=\nu_i; \\ b)\ \left(\left(b_1^{(i+1)},n_1^{(i+1)}\right),\left(b_2^{(i+1)},n_2^{(i+1)}\right),\ldots,\left(b_{\nu_{i+1}}^{(i+1)},n_{\nu_{i+1}}^{(i+1)}\right)\right), \\ \qquad \text{где} \quad b_j^{(i+1)}=b_j^{(i)},\ n_j^{(i+1)}=n_j^{(i)},\ j=\overline{1..\nu_i},\ b_{\nu_{i+1}}^{(i+1)}=\beta_{\alpha_i+1},\ n_{\nu_i+1}^{(i+1)}=\mu_{\alpha_i+1}, \\ \qquad \text{если} \quad \nu_{i+1}=\nu_i+1; \\ c)\ \left(\left(b_1^{(i+1)},n_1^{(i+1)}\right),\left(b_2^{(i+1)},n_2^{(i+1)}\right),\ldots,\left(b_{\nu_{i+1}}^{(i+1)},n_{\nu_{i+1}}^{(i+1)}\right)\right), \\ \qquad \text{где} \quad b_j^{(i+1)}=b_{j+k}^{(i)},\ n_j^{(i+1)}=n_{j+k}^{(i)},\ j=\overline{1..\nu_{i+1}},\ k\geq 1, \\ \qquad \text{если} \quad \nu_{i+1}=\nu_i-k. \end{cases}$$

Опишем вектор обслуживаемых заданий в момент времени t_{i+1}:

$$
S_{i+1} = \begin{cases}
a)\ \left(\left(c_1^{(i+1)}, m_1^{(i+1)}\right)\dots\left(c_{\gamma_{i+1}}^{(i+1)}, m_{\gamma_{i+1}}^{(i+1)}\right)\right), \text{ где } c_j^{(i+1)} = c_j^{(i)} - x_{i+1},\ m_j^{(i+1)} = m_j^{(i)},\ j = \overline{1..\gamma_{i+1}}, \\
\qquad \text{если} \quad 1)\ \xi_{\alpha_i+1} < c_1^{(i)},\ \nu_i > 0, \\
\qquad\qquad\quad 2)\ \xi_{\alpha_i+1} < c_1^{(i)},\ \nu_i = 0,\ \mu_{\alpha_i+1} > \mathrm{M} - \delta_i,\ \gamma_i > 0, \\
\qquad \text{или} \quad \gamma_{i+1} = \gamma_i; \\
b)\ \left(\left(c_1^{(i+1)}, m_1^{(i+1)}\right)\dots\left(c_{\gamma_{i+1}}^{(i+1)}, m_{\gamma_{i+1}}^{(i+1)}\right)\right), \text{ где } c_j^{(i+1)} = c_{j+1}^{(i)} - x_{i+1},\ m_j^{(i+1)} = m_{j+1}^{(i)},\ j = \overline{1..\gamma_{i+1}}, \\
\qquad \text{если} \quad 1)\ \xi_{\alpha_i+1} < c_1^{(i)},\ \nu_i = 0, \\
\qquad\qquad\quad 2)\ \xi_{\alpha_i+1} > c_1^{(i)},\ \nu_i > 0,\ n_1^{(i)} > \mathrm{M} - \delta_i + m_1^{(i)},\ \gamma_i > 1, \\
\qquad \text{или} \quad \gamma_{i+1} = \gamma_i - 1; \\
c)\ \left(\left(c_1^{(i+1)}, m_1^{(i+1)}\right)\dots\left(c_{\gamma_{i+1}}^{(i+1)}, m_{\gamma_{i+1}}^{(i+1)}\right)\right) = \mathbf{Sort}\left(\left(\widetilde{S}_i,\ \left(\mu_{\alpha_i+1}, \beta_{\alpha_i+1}\right)\right)\right), \\
\qquad \text{где} \quad \widetilde{S}_i = \left(\left(\widetilde{c}_1^{(i)}, \widetilde{m}_1^{(i)}\right)\dots\left(\widetilde{c}_{\gamma_i}^{(i)}, \widetilde{m}_{\gamma_i}^{(i)}\right)\right), \\
\qquad\qquad\quad \widetilde{c}_j^{(i)} = c_j^{(i)} - x_{i+1},\ \widetilde{m}_j^{(i)} = m_j^{(i)}\ j = \overline{1..\gamma_i}, \\
\qquad \text{если} \quad 1)\ \xi_{\alpha_i+1} < c_1^{(i)},\ \nu_i = 0,\ \mu_{\alpha_i+1} \le \mathrm{M} - \delta_i, \text{ или } \gamma_{i+1} = \gamma_i + 1; \\
d)\ \left(\left(c_1^{(i+1)}, m_1^{(i+1)}\right)\dots\left(c_{\gamma_{i+1}}^{(i+1)}, m_{\gamma_{i+1}}^{(i+1)}\right)\right) = \\
\qquad = \underbrace{\mathbf{Sort}\left(\dots\mathbf{Sort}\left(\mathbf{Sort}\left(\widetilde{S}_i, \left(b_1^{(i)}, n_1^{(i)}\right)\right), \left(b_2^{(i)}, n_2^{(i)}\right)\right)\dots\left(b_{k+1}^{(i)}, n_{k+1}^{(i)}\right)\right)}_{k+1}, \\
\qquad \text{где} \quad \widetilde{S}_i = \left(\left(\widetilde{c}_1^{(i)}, \widetilde{m}_1^{(i)}\right)\dots\left(\widetilde{c}_{\gamma_i}^{(i)}, \widetilde{m}_{\gamma_i}^{(i)}\right)\right), \\
\qquad\qquad\quad \widetilde{c}_j^{(i)} = c_{j+1}^{(i)} - x_{i+1}\ \ \widetilde{m}_j^{(i)} = m_{j+1}^{(i)},\ j = \overline{1..\gamma_i - 1}, \\
\qquad \text{если} \quad 1)\ \xi_{\alpha_i+1} > c_1^{(i)},\ \nu_i = k+1,\ k \ge 0,\ \sum_{j=1}^{k+1} n_j^{(i)} \le \mathrm{M} - \delta_i + m_1^{(i)}, \\
\qquad\qquad\quad 2)\ \xi_{\alpha_i+1} > c_1^{(i)},\ \nu_i > k+1,\ k \ge 0, \\
\qquad\qquad\quad \sum_{j=1}^{k+1} n_j^{(i)} \le \mathrm{M} - \delta_i + m_1^{(i)},\ \sum_{j=1}^{k+2} n_j^{(i)} > \mathrm{M} - \delta_i + m_1^{(i)}, \\
\qquad \text{или} \quad \gamma_{i+1} = \gamma_i + k,\ k \ge 0;
\end{cases}
$$

где $\mathbf{Sort}\left(\widetilde{S}_i, \left(b_j^{(i)}, n_j^{(i)}\right)\right) = \left(\left(\widetilde{c}_1^{(i)}, \widetilde{m}_1^{(i)}\right)\dots\left(\widetilde{c}_{k-1}^{(i)}, \widetilde{m}_{k-1}^{(i)}\right), \left(b_j^{(i)}, n_j^{(i)}\right), \left(\widetilde{c}_k^{(i)}, \widetilde{m}_k^{(i)}\right)\dots\left(\widetilde{c}_{\gamma_i+1}^{(i)}, \widetilde{m}_{\gamma_i+1}^{(i)}\right)\right)$,

$\widetilde{c}_1^{(i)} < \widetilde{c}_2^{(i)} < \dots < \widetilde{c}_{k-1}^{(i)} < b_j^{(i)} < \widetilde{c}_k^{(i)} < \dots < \widetilde{c}_{\gamma_i+1}^{\,i}$.

Пусть теперь $w(t,n)$ есть виртуальное время ожидания до начала обслуживания задания, поступившего в момент времени t и требующего n узлов.

Обозначим через $\overline{w}(t,n)$ виртуальное время ожидания до начала обслуживания задания, поступившего в момент времени t и требующего n узлов, когда очередь пуста.

Лемма: Имеет место следующее:

$$\overline{w}(t,n)=\begin{cases}0 & \text{, если } \nu(t)=0,\ n\le M-\delta(t),\\ c_1(t) & \text{, если } \nu(t)=0,\ n\le M-\delta(t)+m_1(t),\ \ n>M-\delta(t),\\ c_2(t) & \text{, если } \nu(t)=0,\ n\le M-\delta(t)+m_1(t)+m_2(t),\ \ n>M-\delta(t)+m_1(t),\\ \vdots & \\ c_{\gamma(t)}(t) & \text{, если } \nu(t)=0,\ n\le M-\delta(t)+\sum\limits_{j=1}^{\gamma(t)}m_j(t),\ \ n>M-\delta(t)+\sum\limits_{j=1}^{\gamma(t)-1}m_j(t).\end{cases} \quad (1)$$

Доказательство: При $\nu(t)=0,\ n\le M-\delta(t)$, задание поступившее в момент времени t и требующее n узлов сразу поступит на обслуживание (т.к. число требуемых узлов не превышает числа свободных на кластере).

При $\nu(t)=0,\ n>M-\delta(t)$, задание поступившее в момент времени t и требующее n узлов ставится в очередь. Задание будет ждать в очереди до тех пор, пока не освободится требуемое количество узлов, т.е.

$\overline{w}(t,n)=c_1(t)$, если $\nu(t)=0,\ n\le M-\delta(t)+m_1(t),\ \ n>M-\delta(t)$,

$\overline{w}(t,n)=c_2(t)$, если $\nu(t)=0,\ n\le M-\delta(t)+m_1(t)+m_2(t),\ \ n>M-\delta(t)+m_1(t)$,

$\overline{w}(t,n)=c_{\gamma(t)}(t)$, если $\nu(t)=0,\ n\le M-\delta(t)+\sum\limits_{j=1}^{\gamma(t)}m_j(t),\ \ n>M-\delta(t)+\sum\limits_{j=1}^{\gamma(t)-1}m_j(t)$.

Лемма доказана.

Теорем: При $t_i<t<t_{i+1},\ i=1,2,\ \dots$,

$$w(t,n)=\begin{cases}0 & \text{, если } \nu_i=0,\ n\le M-\delta_i,\\ \overline{w}(t_i,n)-t+t_i, & \text{если } \nu_i=0,\ \ n>M-\delta_i,\\ \overline{w}(\widetilde{t},n)+\widetilde{t}-t, & \text{где } \widetilde{t}=t_{i-1}+w_{i-1}(n^{(i)}_{\nu_i}),\ \text{ если } \nu_i>0.\end{cases}$$

Доказательство: При $\nu_i=0,\ n\le M-\delta_i$, из леммы следует, что $w(t,n)=w(t_i,n)=\overline{w}(t_i,n)=0$.

При $\nu_i=0,\ n>M-\delta_i$ из леммы следует, что $w(t_i,n)=\overline{w}(t_i,n)$. А задание поступившее в момент времени t и требующее n узлов будет ждать на время $t-t_i$ меньше, чем задание поступившее в момент времени t_i, т.е. $w(t,n)=\overline{w}(t_i,n)-t+t_i$.

Если $\nu_i>0$, то при поступлении задания в момент времени t и требующего n узлов, задание ставится в конец очереди. Опишем вектор очереди системы в момент времени t_i:

$Q_i=\left(\left(b_1^{(i)},n_1^{(i)}\right),\left(b_2^{(i)},n_2^{(i)}\right),\dots,\left(b_{\nu_i}^{(i)},n_{\nu_i}^{(i)}\right)\right)$, причем $\nu_i>0$.

Рассмотрим состояние системы в момент времени t_i.

a) Допустим, что момент времени t_i был моментом ухода уже обслуженного задания. Если момент времени t_i есть момент ухода обслуженного задания, то в момент времени t_{i-1}, $\nu_{i-1} > 0$ и $b_{\nu_i}^{(i)} = b_{\nu_{i-1}}^{(i-1)}, n_{\nu_i}^{(i)} = n_{\nu_{i-1}}^{(i-1)}$. Допустим, что нам известно время ожидания задания, если бы оно поступило в момент времени t_{i-1}. Так как $b_{\nu_i}^{(i)} = b_{\nu_{i-1}}^{(i-1)}, n_{\nu_i}^{(i)} = n_{\nu_{i-1}}^{(i-1)}$, то это время ожидания будет равно $w_{i-1}(n_{\nu_i}^{(i)})$. Следовательно, задание $(b_{\nu_i}^{(i)}, n_{\nu_i}^{(i)})$ пойдет на обслуживание в момент времени $\widetilde{t} = t_{i-1} + w_{i-1}(n_{\nu_i}^{(i)})$. Это означает, что в момент времени $\widetilde{t}$ очередь будет пуста (мы не рассматриваем те задания, которые поступают в систему после момента времени t, т.к. они не влияют на время ожидания задания поступившего в момент времени t). Следовательно, время ожидания задания, поступившего в момент времени t, до попадания его во главу очереди будет равно $\widetilde{t} - t$. После чего можно применить формулу (1) для момента времени $\widetilde{t}$. Следовательно, получим ,что $w(t,n) = \overline{w}(\widetilde{t}, n) + \widetilde{t} - t$.

b) Допустим, что момент времени t_i был моментом прихода нового задания. Так как $\nu_i > 0$, то задание , пришедшее в момент времени t_i и требующее $n_{\nu_i}^{(i)}$ узлов стало в очередь, если

- в момент времени t_{i-1} очередь была не пуста;
- в момент времени t_{i-1} очередь была пуста, но пришедшее задание требует большего числа узлов чем имеется свободных.

Следовательно, если бы задание пришло в момент времени t_{i-1} и требовало бы $n_{\nu_i}^{(i)}$ узлов для обслуживания, то оно простояло бы в очереди на время x_i больше, чем если бы оно пришло в момент времени t_i. Тогда $w_i(n_{\nu_i}^{(i)}) = w_{i-1}(n_{\nu_i}^{(i)}) - x_i$ и, следовательно, $\widetilde{t} = t_i + w_i(n_{\nu_i}^{(i)}) = t_i + w_{i-1}(n_{\nu_i}^{(i)}) - x_i = t_{i-1} + w_{i-1}(n_{\nu_i}^{(i)})$. Это означает , что в момент времени $\widetilde{t}$ очередь будет пуста. Следовательно время ожидания задания, поступившего в момент времени t, до попадания его в голову очереди будет равно $\widetilde{t} - t$. После чего можно применить формулу (1) для момента времени $\widetilde{t}$. Таким образом, $w(t,n) = \overline{w}(\widetilde{t}, n) + \widetilde{t} - t$. Теорема доказана.

Для иллюстрации вышесказанного рассмотрим следующий пример. Рассмотрим кластер, состоящий из M = 10 узлов. Для обслуживания на кластере поступает поток заданий. Для простоты будем считать, что моменты поступления и длительность выполнения заданий целочисленны. Время поступления и характеристики заданий представлены в таблице.

Таблица

Задание	Момент поступления (t)	Число требуемых узлов (μ)	Длительность выполнения (β)
I	1	8	4
II	3	4	7
III	8	3	2
IV	9	4	8
V	11	10	1

С помощью вышеперечисленных рекуррентных уравнений вычислим параметры состояния системы в последовательные моменты времени, когда в систему поступают или покидают задания:

$x_1 = 1,\ \nu_1 = 0,\ \delta_1 = 8,\ \gamma_1 = 1,\ \Theta_1 = 0,\ \alpha_1 = 1,\ |Q_1| = 0,\ S_1 = ((4,8))$;
$x_2 = 2,\ \nu_2 = 1,\ \delta_2 = 8,\ \gamma_2 = 1,\ \Theta_2 = 0,\ \alpha_2 = 2,\ Q_2 = ((7,4)),\ S_2 = ((2,8))$;
$x_3 = 2,\ \nu_3 = 0,\ \delta_3 = 4,\ \gamma_3 = 1,\ \Theta_3 = 1,\ \alpha_3 = 2,\ |Q_3| = 0,\ S_3 = ((7,4))$;
$x_4 = 3,\ \nu_4 = 0,\ \delta_4 = 7,\ \gamma_4 = 2,\ \Theta_4 = 1,\ \alpha_4 = 3,\ |Q_4| = 0,\ S_4 = ((2,3),(4,4))$;
$x_5 = 1,\ \nu_5 = 1,\ \delta_5 = 7,\ \gamma_5 = 2,\ \Theta_5 = 1,\ \alpha_5 = 4,\ Q_5 = ((8,4)),\ S_5 = ((1,3),(3,4))$;
$x_6 = 1,\ \nu_6 = 0,\ \delta_6 = 8,\ \gamma_6 = 2,\ \Theta_6 = 2,\ \alpha_6 = 4,\ |Q_6| = 0,\ S_6 = ((2,4),(8,4))$;
$x_7 = 1,\ \nu_7 = 1,\ \delta_7 = 8,\ \gamma_7 = 2,\ \Theta_7 = 2,\ \alpha_7 = 5,\ Q_7 = ((1,10)),\ S_7 = ((1,4),(7,4))$;
$x_8 = 1,\ \nu_8 = 1,\ \delta_8 = 4,\ \gamma_8 = 1,\ \Theta_8 = 3,\ \alpha_8 = 5,\ Q_8 = ((1,10)),\ S_8 = ((6,4))$;
$x_9 = 6,\ \nu_9 = 0,\ \delta_9 = 10,\ \gamma_9 = 1,\ \Theta_9 = 4,\ \alpha_9 = 5,\ |Q_9| = 0,\ S_9 = ((1,10))$;
$\nu_{10} = 0,\ \delta_{10} = 0,\ \gamma_{10} = 0,\ \Theta_{10} = 5,\ \alpha_{10} = 5,\ |Q_{10}| = 0,\ |S_{10}| = 0.$

После чего применяя доказанную теорему вычислим время ожидания до начала обслуживания заданий: $w(1,8) = 0$; $w(3,4) = 2$; $w(8,3) = 0$; $w(9,4) = 2$; $w(11,10) = 7$.

Литература

1. Л. Клейнрок Вычислительные системы с очередями, Москва, Мир, 1979, 600 с.
2. А.Бабичев и др. Параллельная обработка информации, Киев, Наукова Думка, 1985, С 278.
3. R. W. Wolff. Stochastic Modeling and the Theory of Queues. Englewood Cliffs, NJ: Prentice-Hall, 1989, P 560.

Բազմամեքենայական հաշվողական համալիրի հերթի մոդելավորումը

Մ. Ղ. Գյուրջյան, Վլադիմիր Գ. Սահակյան

Ամփոփում

Բազմամեքենայական հաշվողական համալիրի (կլաստերի) վրա հաշվարկներ կազմակերպելու ժամանակ մեծ դեր է կատարում հերթերի կազմակերպումը և խնդիրների սպասարկման հերթականությունը: Այսպիսի համակարգերի հիմնական պարամետրերից մեկն է խնդրի մինչ իր կատարումը հերթում սպասման ժամանակը: Տվյալ աշխատանքում դիտարկվում է պարամետրերի որոշ հավաքածու, որը թույլ է տալիս կազմել հավասարումների ռեկուրենտ համակարգ և այդ պարամետրերի հիման վրա հաշվարկել սպասման ժամանակը: