ОДНОРОДНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ ИЗ МИКРО-ЭВМ
(Вычислительные системы)
1983 год Выпуск 97

УДК 681.51:519.8

ВЫБОР СПОСОБА ПОИСКА ДАННЫХ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ С ПРОГРАММИРУЕМОЙ СТРУКТУРОЙ

В.Г. Кербель

I. Общие положения. Вычислительные системы с программируемой структурой находятся под пристальным вниманием исследователей различных областей знаний. Впервые указанный класс систем рассмотрен в работе [I], где обоснована возможность получения высокой производительности при решении ряда задач линейной алгебры и дифференциального исчисления. В дальнейшем начались работы по созданию параллельных алгоритмов, соответствующих языков программирования, операционных систем, средств автоматической подготовки параллельных программ, а также различных алгоритмов планирования (разбиения системы на подсистемы) вычислений на ВС с программируемой структурой [2-7]. Стало ясно, что планировать использование ресурсов таких систем необходимо не только между различными задачами, реализуемыми на системе одновременно, но и при организации параллельных вычислений для одной задачи [7,8] в зависимости от характеристик системы и параметров решаемой задачи. В этой связи в работе исследуется поведение ВС с программируемой структурой при поиске данных в распределенном по машинам банке данных. Предполагается, что система состоит из однотипных элементарных машин равной производительности и время передачи запроса на данные и получения ответа пренебрежимо мало по сравнению со временем поиска этого данного. Перестраиваемость структуры вычислительной системы связывается с эффективной реализацией выбранного способа поиска данных.

Рассматриваются три способа (правила) поиска.

I. "Ищу сам-прошу всех", т.е. запрос, поступивший в машину, обрабатывается ею, и если данные не найдены, то передается всем остальным машинам системы.

2. "Ищем все", т.е. поступивший запрос дублируется во всех машинах системы.

3. "Ищу сам-прошу соседа", т.е. запрос, поступивший в машину, обрабатывается ею, и если данные не найдены, то передается соседней, например, правой машине (предполагается, что машины системы по связям замкнуты в кольцо).

Введем обозначения: N - число машин в системе или подсистеме, занятых поиском данных, t - среднее время обслуживания запроса одной машиной, p - вероятность найти данные "у себя", τ - среднее время между поступлениями запросов на одну машину.

По критерию "минимум среднего времени обслуживания запроса" требуется найти границы применимости указанных способов поиска.

2. Детерминированное обслуживание

Предположим, что система обслуживает набор из n запросов. Задача состоит в том, чтобы указать правило, позволяющее уменьшить общее время обслуживания этого набора.

Минимальное время, необходимое для обслуживания одного запроса по правилу I, есть:

$$T_1^{min} = t + (1-p)t; \qquad (1)$$

по правилу 2 -

$$T_2^{min} = t; \qquad (2)$$

по правилу 3 -

$$T_3^{min} = t+(1-p)t + \frac{(1-p)(N-2)}{N-1}\,t + \ldots + \frac{1-p}{N-1}\,t = t\cdot\left(1 + \frac{N(1-p)}{2}\right). \qquad (3)$$

Здесь предполагается, что если данные на запрос с вероятностью p не найдены в машине, они с равной вероятностью $(1-p)/(N-1)$ могут быть найдены в любой другой машине.

Максимальное время обслуживания одного запроса слагается из ожидания в очереди и собственно обслуживания. Следует иметь в виду, что обслуживание запроса считается завершенным, если его завершили (при необходимости) все N машин. Итак, для правила I:

$$T_1^{max} = t + (1-p)nt; \qquad (4)$$

для правила 2 -

$$T_2^{max} = nt; \qquad (5)$$

для правила 3 -

$$T_3^{max} = \left(\left[\frac{n-1}{N}\right] + 1\right)\left(1 + \frac{N(1-p)}{2}\right)t \ . \qquad (6)$$

Однако предполагается, что $n \le N$, т.е. из каждой ЭМ берется не более одного запроса в набор, а для правила 3 $(n=N)$ удается "закрутить" все запросы в одно кольцо (рис.I). В этом предположе-

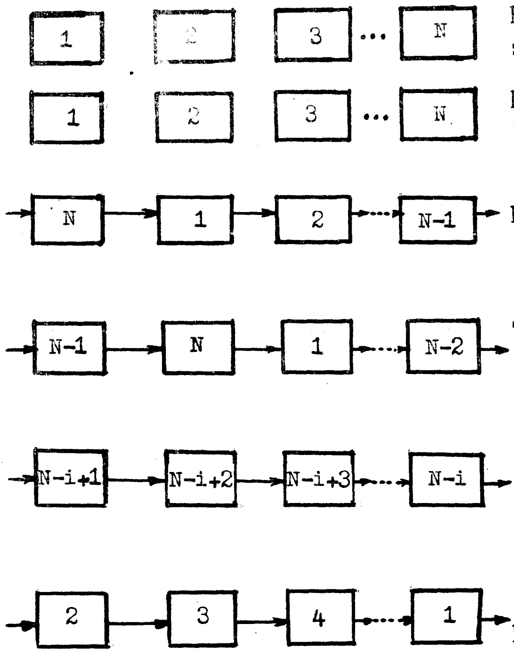

Рис. I

нии значения (3) и (6) равны и определяют среднее время поиска по правилу 3.

Определение среднего времени поиска по правилам I и 2 требует уточнения способа распределения запросов по машинам системы. Структурное распределение запросов (рис.2) позволяет организовать сбалансированную работу системы; при этом среднее время поиска для правила I:

$$T_1 = (T_1^{min} + T_1^{max})/2 = \left(1 + \frac{(n+1)(1-p)}{2}\right)t; \tag{7}$$

для правила 2 –

$$T_2 = (T_2^{min} + T_2^{max})/2 = \frac{n+1}{2}t. \tag{8}$$

Сравнивая значения (8) и (7), находим, что

$$T_2 \le T_1 \Leftrightarrow n \le (2-p)/p. \tag{9}$$

При равновероятном распределении банка по машинам, т.е. при $p = 1/N$, имеем $n \le 2N-1$, но так как $n \le N \le 2N - 1$, то можно сделать вывод, что правило 2 в с е г д а л у ч ш е правила I (в предположениях $nt < \tau$, $n \le N$ и структурном распределении запросов).

Сравнивая значения (8) и (3), находим, что

$$T_2 \le T_3 \Leftrightarrow n \le 1 + N(1-p) . \tag{10}$$

При $p = 1/N$ имеем $T_2 \le T_3$ при $n \le N$, т.е. правило 2 л у ч ш е правила 3.

Сравнивая значения (7) и (3), находим, что правило I л у ч ш е правила 3, если

$$T_1 \le T_3 \Leftrightarrow n \le N-1 . \tag{11}$$

Оценки (9)-(II) справедливы при структурном распределении запросов в машинах системы. Достигаться это может средствами операционной системы, осуществляющими планирование ресурсов в машине. В общем же случае необходимое время обслуживания отдельного запроса (и всего набора в целом) следует брать как максимальное из равенств (4)-(6). Имеем:

$$T_2 \le T_1 \Leftrightarrow n \le 1/p \Leftrightarrow p \le 1/n ;$$

$$T_2 \le T_3 \Leftrightarrow n \le 1+N(1-p)/2 \Leftrightarrow p \le 1-2(n-1)/N;$$

$$T_1 \le T_3 \Leftrightarrow (n \le N/2).$$

Следовательно, при равновероятном распределении данных по машинам и $n \le N$ запросы необходимо обслуживать по правилу 2, если $n \le (N + 1)/2$, в противном случае – по правилу 3.

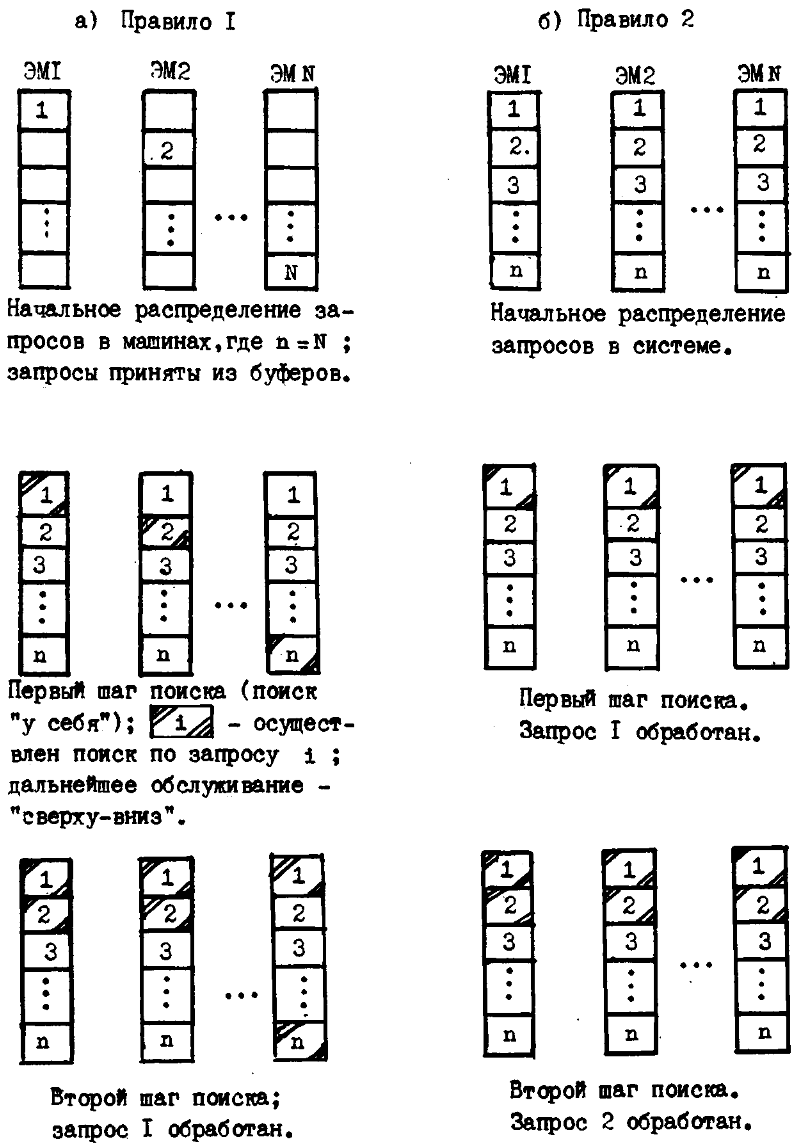

Рис.2. Структурное распределение запросов: а) поиск по правилу I; б) поиск по правилу 2.

ПРИМЕРЫ.

I. В системе I0 машин, поиск равновероятный, время поиска в одной машине t, $n \le N$ - число запросов в наборе. Имеем:

$$T_2 \le T_1 \Leftrightarrow n \le 10;$$

$$T_2 \le T_3 \Leftrightarrow n \le 1 + 10\,(1-0.1)/2 = 5.5;$$

$$T_1 \le T_3 \Leftrightarrow n \le 5.$$

Отсюда, выбирая допустимые значения, имеем:

$$T_{\text{обсл.}} = \begin{cases} nt \quad, \text{ если } n \le 5 \text{ и правило 2;} \\ 5{,}5t\,, \text{ если } n > 5 \text{ и правило 3.} \end{cases}$$

2. Число машин I0, банки соседей (по кольцу) дублированы и поиск по ним разрешен, т.е. $p = 3/10$ и время поиска увеличивается в 3 раза, $n \le N$ - число запросов. Тогда:

$$T_2 \le T_1 \Leftrightarrow n \le 10/3 \approx 3{,}3;$$

$$T_2 \le T_3 \Leftrightarrow n \le 4{,}5;$$

$$T_1 \le T_3 \Leftrightarrow n \le 5\,.$$

Отсюда:

$$T_{\text{обсл}} = \begin{cases} 3nt \quad, \text{ если } n \le 3 \text{ и правило 2;} \\ 11{,}4t, \text{ если } n = 4 \text{ и правило I;} \\ 13{,}5t, \text{ если } n > 4 \text{ и правило 3.} \end{cases}$$

В ы в о д. Поиск по дублированному от соседей банку разрешать не следует, так как увеличивается время обслуживания набора запросов.

3. Число машин I0, в каждой ЭМ задублированы данные соседей, т.е. $p = 3/10$, поиск в дублях разрешен только для "своего" запроса. Тогда $T_1 = (3+0{,}7n)t$; $T_2 = nt$; $T_3 = (3+7/2)t$. Окончательно:

$$T_{\text{обсл}} = \begin{cases} nt \quad, \text{ если } n \le 6 \text{ и правило 2;} \\ 6{,}5t, \text{ если } n > 6 \text{ и правило 3.} \end{cases}$$

Ясно, что время поиска при $n \ge 6$ увеличивается.

4. Число машин N , дубль m - кратный, т.е. в каждой ЭМ задублированы данные от $(m-1)$ ЭМ таким образом, чтобы данные в любых соседних ЭМ не пересекались; поиск по всем данным разрешен только для своего запроса.

Организация указанного дублирования осуществляется по правилу:

$$M_i = \{L_\nu : \nu = (i+kq) \bmod N,\ k=\overline{0,m-1}\}, \quad i = \overline{1,N},$$

где $q = \left[\frac{N}{m}\right]+1$, а M_i - множество банков i-й ЭМ, L_ν - начальный банк ν-й ЭМ. Вероятность p определяется как $p = m/N$. Тогда:

$$T_1 = \left(m + \frac{N-m}{N}n\right)t; \quad T_2 = nt; \quad T_3 = \frac{N^2+m^2+N-m}{2N}t.$$

Отсюда:

$$T_2 \le T_1 \Leftrightarrow n \le N;$$

$$T_1 \le T_3 \Leftrightarrow n \le (N-m+1)/2;$$

$$T_2 \le T_3 \Leftrightarrow n \le (N^2+m^2+N-m)/2N.$$

Легко показать, что время поиска увеличивается. Следовательно, поиск по дублю нерационален, а задачу дублирования данных в сосредоточенных системах можно решать независимо от задачи поиска. Поиск следует вести только по собственным данным.

5. $N = 10$, $p = 0{,}4$ - только в силу специфики запросов, n - объем набора. Имеем:

$$T_{обсл} = \begin{cases} nt, & \text{если } n \le 2, \text{ правило 2;} \\ (1+0{,}6n)t, & \text{если } 2 < n \le 5, \text{ правило 1;} \\ 4t, & \text{если } 5 \le n \le 10, \text{ правило 3.} \end{cases}$$

3. Обслуживание потока запросов

Функционирование системы с программируемой структурой при обслуживании потока запросов представлено схемой на рис.3, где $\boxed{|||}$ - буфер (1 - входных запросов, 2 - запросов в систему, 3 - запросов на поиск, 4 - результата поиска, 5 - ответа); F - функция определения ответа (например, конъюнкция признаков ответов всех машин по одному запросу); ▬ - собственный запрос; — - системный запрос. При детерминированном обслуживании системой набора запросов ее функционирование в целом такое же. Работа операционной системы ВС с программируемой структурой по организации рассылки запросов и передачи ответа в статье не рассматривается.

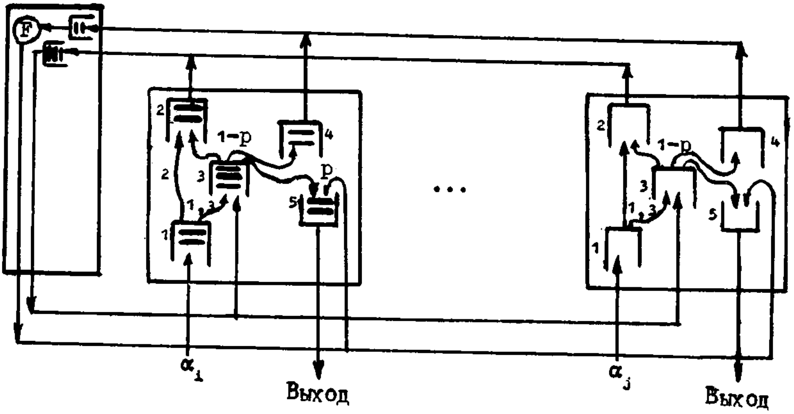

Рис. 3

Пусть $\alpha = 1/\tau$ - интенсивность входящего потока требований, $\beta = 1/t$ - интенсивность обслуживания запросов. Предполагается, что входной поток пуассоновский, а обслуживание подчинено экспоненциальному закону. Такое предположение позволяет получить "худшие" оценки в поведении системы и воспользоваться известными результатами теории массового обслуживания.

Итак, предполагается простейший поток запросов с параметром α на каждую машину системы. С вероятностью р этот поток будет обслужен в машине и с вероятностью 1-р передан в систему. Суммарное значение входящего на поиск в машину потока определяется для каждого правила обслуживания по своему.

Для правила I имеем:

$$\alpha_\Sigma^1 = \alpha + (1-p)(N-1)\alpha \, . \tag{12}$$

Для правила 2 -

$$\alpha_\Sigma^1 = N\alpha \, . \tag{13}$$

Для правила 3 простые подсчеты дают

$$\alpha_\Sigma^3 = \alpha + \frac{N(1-p)}{2}\,\alpha . \tag{14}$$

Отметим, что легко получить значения, аналогичные (I2)-(I4), для потоков α_i на машину i, однако для простоты изложения все значения α_i предполагаются равными α. Чтобы очередь запросов не росла бесконечно, необходимо выполнение для каждого правила неравенства

$$\rho_i = \alpha_\Sigma^i/\beta < 1, \quad i = 1,2,3. \qquad (15)$$

Известно [9,I0], что при заданных выше потоке и законе обслуживания среднее время ожидания запроса в очереди до начала обслуживания

$$W = \frac{\rho}{\beta(1-\rho)}, \qquad (16)$$

а среднее время собственно обслуживания

$$\bar{x} = 1/\beta. \qquad (17)$$

Следовательно, среднее время обслуживания запроса, определяемое как сумма (I6) и (I7), есть $\bar{\bar{W}} = W+\bar{x} = 1/(\beta-\alpha_\Sigma^i)$, $i=1,2,3$. По определению правила I, время обслуживания запроса T_1 = время "у себя" + $(1-p)\cdot$(максимум времени "у других"). В силу предполагаемой однородности имеем:

$$T_1 = \frac{1}{\beta-\alpha_\Sigma^1} + \frac{1-p}{\beta-\alpha_\Sigma^1} = \frac{2-p}{\beta-(1+(N-1)(1-p))\alpha}.$$

Аналогично для правила 2 -

$$T_2 = \frac{1}{\beta-\alpha_\Sigma^2} = \frac{1}{\beta-N\alpha}.$$

Несложные подсчеты позволяют определить

$$T_3 = \frac{2+N(1-p)}{2(\beta-\alpha_\Sigma^3)} = \frac{2+N(1-p)}{2\beta-(2+N(1-p))\alpha}.$$

В общем случае, когда на i-ю ЭМ поступает простейший поток заявок с интенсивностью α_i, p_i - вероятность найти данные "у себя", β_i - интенсивность обслуживания, выражения для определения T_i (i = I,2,3) выводятся по приведенной выше схеме.

Например, для правила I имеем суммарный входной поток α^j в машину с номером j :

$$\alpha^j = \alpha_j + \sum_{\substack{i=1 \\ i \neq j}}^{N} (1-p_i)\alpha_i \ .$$

Время от момента поступления требования в j-ю ЭМ до его полного обслуживания есть:

$$T_1^j = \frac{1}{\beta_j - \alpha_j - \sum\limits_{\substack{i=1 \\ i \neq j}}^{N} (1-p_i)\alpha_i} + (1-p_j) \max_{\substack{k=\overline{1,N} \\ k \neq j}} \frac{1}{\beta_k - \alpha^k} =$$

$$= \frac{1}{\beta_j - \alpha_j - \sum\limits_{\substack{i=1 \\ i \neq j}}^{N} (1-p_i)\alpha_i} + \frac{1-p_j}{\min\limits_{\substack{k=\overline{1,N} \\ k \neq j}} (\beta_k - \alpha_k - \sum\limits_{\substack{i=1 \\ i \neq k}}^{N} (1-p_i)\alpha_i)} \ .$$

Аналогичные выражения можно получить для определения времени обслуживания запроса по правилам 2 и 3. Практические расчеты проведены на ЭВМ по составленной ФОРТРАН-програме.

В дальнейшем изложении полагается, что $\forall i = \overline{1,N}$ $(\alpha_i = \alpha;\ \beta_i = \beta;\ p_i = p)$.

График зависимости среднего времени обслуживания заявки при различном числе машин и различных правилах приведен на рис.4.Расчет проведен для следующих значений параметров: $\beta = 8$, $\alpha_\Sigma = N$, $p = 0.1$ и $p = 0.3$. На рис.4 видно, что при $p = \max(1/N, 0{,}3)$ необходимо использовать правило 2 при $N < 5$, правило 3 при $N \geq 9$, в других случаях - правило I. Вероятность p характеризует распределение банка по машинам, степень его дублирования и частично характер поступающих запросов.

Например, реляционный банк данных, центральный каталог которого представлен матрицей $M \times M$, при равномерном распределении по N машинам определит значение $p = 1/N$, если M делится на N нацело, а в каждой машине будет по равному количеству строк (столбцов) матрицы. Однако стремление к обеспечению надежности и сохранности банка приведет к его частичному дублированию, возможно, m-кратному. Тогда значение p есть m/N. С другой стороны, на j-ю ЭМ могут поступать запросы в основном по j-у атрибуту,и тогда даже при равновероятном распределении банка по машинам значение p_j может не зависеть от N, т.е. $p_j = \max(1/N, p_0^j)$, где p_0^j - вероятность найти данные "у себя" для j-й машины.

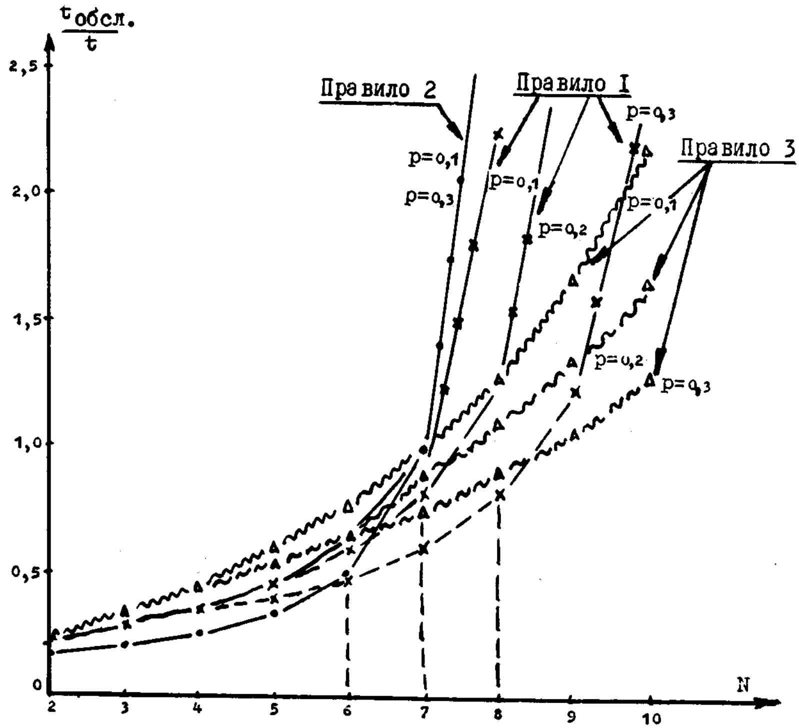

Рис. 4

Аналитическое нахождение точек пересечения кривых, характеризующих время обслуживания заявки по различным правилам, требует решения неравенств: $T_2 \le T_1$, $T_2 \le T_3$, $T_1 \le T_3$.

Из неравенств приходим к условиям:

$$T_2 \le T_1 \Leftrightarrow \rho \le \frac{1-p}{N-p} \Leftrightarrow N \le p + \frac{1-p}{\rho} . \qquad (18)$$

Здесь и в дальнейшем $\rho = \alpha/\beta$.

При равновероятном распределении банка имеем:

$$T_2 \le T_1 \Leftrightarrow \rho \le \frac{1}{N+1} \Leftrightarrow N \le \frac{1-\rho}{\rho} .$$

Из неравенства $T_2 \le T_3$ приходим к условию:

$$T_2 \le T_3 \Leftrightarrow \rho \le \frac{N(1-p)}{(N-1)(2+N(1-p))} .$$

При равновероятном поиске

$$T_2 \le T_3 \Leftrightarrow \rho \le \frac{1}{N+1} \Leftrightarrow N \le \frac{1-\rho}{\rho} . \qquad (19)$$

Наконец, из неравенства $T_1 \le T_3$ определяем:

$$T_1 \le T_3 \Leftrightarrow \rho \le \frac{1}{N(1-p)+2} \Leftrightarrow N \le \frac{1-2\rho}{\rho(1-p)} , \quad N > 2 , \ p < 1. \qquad (20)$$

Если $(N=2) \vee (p=1)$ значения T_1 и T_3 равны.

При $p = 1/N$ условие (20) равносильно условию (19).

4. Выводы

Результаты машинного расчета о выборе способа поиска данных в распределенном банке позволяют подтвердить гипотезу о том, что одновременный поиск (правило 2) эффективен лишь при малой загрузке системы (граница определяется условием (18)). Правило - I "ищу сам-прошу всех" - бессмысленно при равновероятном распределении банка, однако с увеличением p (вероятности "найти у себя") оно эффективней правила 3. Граница эта находится неравенством (20). "Спокойный" поиск по правилу 3 - "ищу сам-прошу соседа" - наиболее эффективен при насыщении системы и обеспечивает, что очень важно, наибольшее выживание системы (очереди на поиск в машинах ограничены).

В работе решена задача о выборе способа поиска при фиксированной интенсивности обслуживания и росте интенсивности потока с увеличением числа машин. Исследование задачи при фиксированной интенсивности потока и увеличении интенсивности обслуживания, с увеличением числа машин, будет представлено отдельной работой автора.

Литература

1. ЕВРЕИНОВ Э.В., КОСАРЕВ Ю.Г. Однородные вычислительные системы высокой производительности. - Новосибирск: Наука, 1966,- 308 с.

2. МИРЕНКОВ Н.Н. Параллельные алгоритмы для решения задач на однородных вычислительных системах.-В кн.: Вычислительные системы. Вып. 57. Новосибирск, 1973, с. 3-32.

3. ЕВРЕИНОВ Э.В., ХОРОШЕВСКИЙ В.Г. Однородные вычислительные системы. - Новосибирск: Наука, 1978. - 317 с.

4. Высокопроизводительные вычислительные системы. - Всесоюз. совещание. Тезисы докл. Ч. I-3. Тбилиси, 1981. - 600 с.

5. КЕРБЕЛЬ В.Г. Реализация экспериментального распараллеливания алгоритмов. -В кн.: Математическое обеспечение вычислительных систем (Вычислительные системы, вып. 78). Новосибирск, 1979, с. 54-69.

6. КЕРБЕЛЬ Н.К. О диалоговой системе параллельного программирования. -В кн.: Математическое обеспечение вычислительных систем (Вычислительные системы, вып. 89). Новосибирск, 1981, с.89-97.

7. КЕРБЕЛЬ В.Г. Вопросы мультиобработки и параллельного программирования на однородных вычислительных системах. - Дис. на соискание ученой степени канд.техн.наук. - Новосибирск, 1979.

8. ГРОПШЕН В.О., ТРАХТЕНГЕРЦ Э.А. Модели распараллеливания вычислений при решении экстремальных комбинаторных задач. -Автоматика и телемеханика, 1982, № 3, с. 151-164.

9. КЛЕЙНРОК Л. Теория массового обслуживания. -М.: Машиностроение, 1979. - 432 с.

10. КЛЕЙНРОК Л. Вычислительные системы с очередями. -М.: Мир, 1979. - 600 с.

Поступила в ред.-изд.отд.
15 октября 1982 года