электронный научно-технический журнал
ИНЖЕНЕРНЫЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51036. ISSN 2307-0595

# Модель распределения ресурсов в сети процессоров

# 08, август 2015

Можаров Г. П.[1,*]

УДК: 681.325.5

[1]Россия, МГТУ им. Н.Э. Баумана
[*]mojarov_g@mail.ru

## Введение

В традиционных расписаниях для компьютерных сетей задания распределяются между процессорами таким образом, чтобы минимизировать среднее время прохождения задания через компьютерную сеть, в том числе, за счёт постулирования распределения во времени заявок, выдаваемых отдельными процессами [1-3].

Качество планировщика может оцениваться по двум позициям: во-первых, насколько хорошо по производительности, во-вторых, насколько эффективно в смысле накладных расходов он реагирует на запросы ресурсов со стороны потребителей. Иными словами, пользователи заинтересованы в быстром и эффективном доступе к ресурсам и вовсе не желают разбираться с побочными эффектами, которые сопровождают управление ими. Планировщик в системе управления ресурсами опирается на информацию о запускаемой работе, которая предоставляется пользователем. В традиционных расписаниях для параллельных систем [2] процесс остаётся под управлением одной операционной системы. В сетях должен существовать не один, а множество вариантов распределения ресурсов [4] (в зависимости от наступления внешних по отношению к программе событий, динамично меняющегося окружения из-за пространственно-временной конкуренции различных программ и т.п.).

Проведём сравнительный анализ предлагаемой (заявки поступают в каждое устройство памяти равномерно распределёнными во времени и независимо друг от друга) и традиционной модели расписаний, чтобы нагляднее продемонстрировать требования к моделям планирования в распределённых масштабируемых средах.

## Математическое описание модели распределения ресурсов в сети

Рассмотрим сеть, состоящей из $n$ процессоров $P_1$, ..., $P_n$ и $m$ устройств $R_1$, ..., $R_m$, являющихся ресурсами общего пользования. Каждый процессор $P_i$ нуждается в доступе к устройствам в определённой последовательности и на определённые интервалы времени. Эта последовательность определена заранее и не зависит от других процессоров. Не может изменить эту последовательность и сам процессор $P_i$.

Пусть $R_j$ являются запоминающими устройствами. Тогда сеть содержит $n$ процессоров, работающих по $n$ независимым программам, и каждый процессор генерирует собственный поток адресов.

Предполагается, что время цикла памяти равно $\delta$ и что каждый процессор $P_i$ делает все запросы $c_{ij}$ на устройства $R_j$ за рассматриваемый интервал времени, равный $N\delta$, где $N$ – целое число. Предполагается далее, что сеть находится в устойчивом состоянии и что интервал $N\delta$ достаточно представителен.

В отличие от обычного подхода, когда постулируется распределение во времени заявок, выдаваемых отдельными процессами, в данной модели задаётся распределение времён прихода заявок на отдельные устройства $R_j$. Предполагается, что заявки будут поступать в $R_j$ равномерно распределёнными во времени и независимо друг от друга.

Нижняя оценка среднего времени ожидания, определяемого как время от момента поступления заявки в $R_j$ до удовлетворения заявки, может быть получена следующим образом. Заявка поступает либо тогда, когда $R_j$ свободно, и тогда заявка немедленно обслуживается, либо тогда, когда $R_j$ занято, и в этом случае заявка ожидает, пока $R_j$ закончит обслуживание другой заявки. Другими словами, задержки вычисляются при предположении, что ждёт обслуживания максимум одна заявка [5,6]. Нижняя граница среднего времени ожидания есть произведение вероятности того, что $R_j$ оказывается занятым в момент поступления заявки, на среднее время, которое требуется $R_j$ для завершения текущей задачи. Вероятность того, что $R_j$ занято в данный момент, равна $\frac{1}{N}\sum_k c_{kj}$.

Для завершения текущей задачи устройству $R_j$ требуется время $\delta$, если система действует синхронно и все заявки поступают в дискретные интервалы. Если же система действует асинхронно, то в среднем требуется время $\delta/2$. Итак, для завершения текущей задачи требуется время $\alpha\delta$, где $\alpha_{\text{син}} = 1$, $\alpha_{\text{асин}} = 1/2$. Время ожидания для устройства $R_j$ равно $\frac{\alpha\delta}{N}\sum_k c_{kj}$.

Процессор $P_i$ за время $N\delta$ $c_{ij}$ раз запрашивает устройство $R_j$, т. е. суммарное время ожидания процессором $P_i$ устройства $R_j$ равно $\frac{\alpha\delta}{N} c_{ij} \sum_k c_{kj}$.

Если просуммировать последнее выражение по всем устройствам $R_j$, то получится нижняя граница суммарного среднего времени ожидания процессора $P_i$. За тот же интервал $N\delta$ процессор $P_i$ должен успеть отработать со всеми необходимыми ему устройствами $R_j$, т.е. потратить время $\sum_j c_{ij}\delta$. Итак, получается неравенство

$$\sum_j c_{ij}\delta + \sum_j \frac{\alpha\delta}{N} c_{ij} \sum_k c_{kj} \le N\delta$$

или (1)

$$\sum_j \frac{c_{ij}}{N} + \alpha \sum_j \sum_k \frac{c_{ij}}{N}\frac{c_{kj}}{N} \le 1.$$

Чтобы найти верхнюю границу для использования процессора, достаточно зафиксировать $n$, $m$ и матрицу $c_{ij}$ и решить неравенство (l) для наименьшего $N$, удовлетворяющего всем $i$ [7-10].

Для специального случая, когда $c_{ij} = c$ для всех $i$ и $j$, неравенство (1) принимает вид

$$\left(\frac{mc}{N}\right) + \alpha \frac{n}{m}\left(\frac{mc}{N}\right)^2 \le 1. \quad (2)$$

Отсюда

$$\frac{mc}{N} \le \frac{m}{2\alpha n}\left(\sqrt{1+\frac{4\alpha n}{m}} - 1\right) \quad \text{или} \quad \frac{mc}{N} \le \frac{m}{2n}\left(\sqrt{1+\frac{4\alpha n}{m}} - 1\right) \quad (3)$$

для синхронно действующей системы.

Так как процессор имеет доступ к ресурсам в течение времени $mc\delta$ за полный период $N\delta$, то величина $mc/N$ может трактоваться как средний коэффициент использования процессора. Из неравенства (3) видно, что средний коэффициент использование процессора зависит только от отношения $r = n/m$ и не зависит от абсолютных значений $n$ и $m$. На рис. 1 приведена зависимость коэффициента использования процессора от $n/m$ для описываемой модели 1 (кривая 1), а также для модели 2, в которой заявки процессоров никогда не конфликтуют друг с другом (кривая 2)

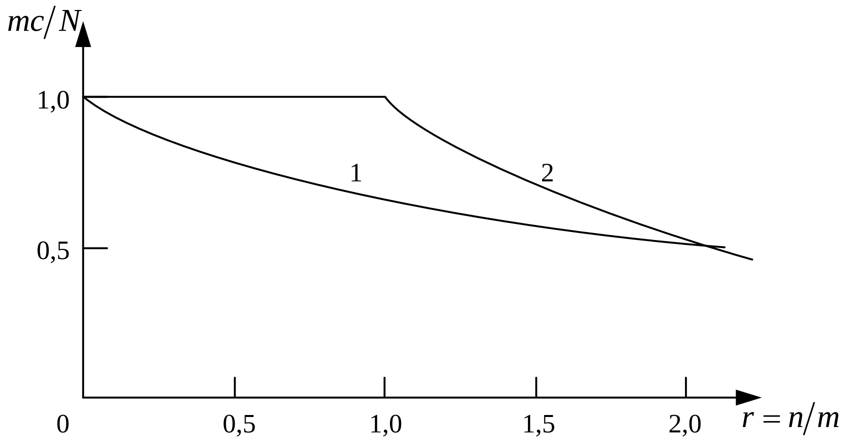


**Рис. 1.** Зависимость коэффициента использования процессора от $n/m$ для предлагаемой модели 1 (кривая 1) и для модели 2 (кривая 2), в которой заявки процессоров не конфликтуют друг с другом

Если обе части неравенства (3) умножить на $n/m$, то получится соотношение $\frac{nc}{N} \le \frac{1}{2}\left(\sqrt{1+\frac{4n}{m}}-1\right)$, определяющее верхнюю границу для коэффициента использования памяти $nc/N$. Зависимость $nc/N$ от $n/m$ для тех же двух моделей приведена на рис. 2.

## Сравнение с традиционными моделями и заключение

В модели 2, в которой заявки процессоров никогда не конфликтуют друг с другом, рассматривалась устойчивая система из $n$ процессоров и $m$ запоминающих устройств. Вычислялось среднее число непересекающихся заявок от процессоров в течение одного цикла. Каждый процессор получает доступ к любому устройству с равной вероятностью. Выражение для максимума коэффициента использования памяти имеет вид

$$U_{\text{mem}} = \sum_{i}^{\min} i \frac{i!S(n,\ i)\binom{m}{n}}{m^{n+1}}.$$

Это уравнение эквивалентно выражению

$$U_{\text{mem}} = 1-\left(1-\frac{1}{m}\right)^{n}.$$

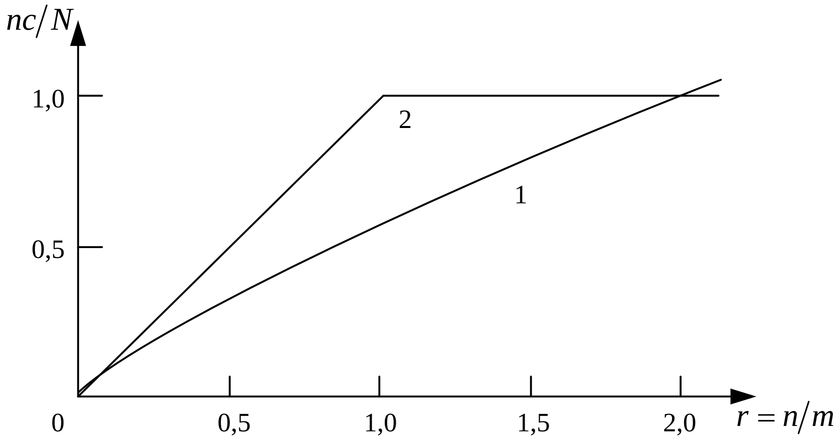


**Рис. 2.** Зависимость $nc/N$ от $n/m$ для моделей 1 и 2

Для фиксированного $r=n/m$ при увеличении $m$ это выражение стремится к $1-e^{-r}$, т.е. коэффициент использования зависит только от отношения $n/m$. Как в предложенной модели, так и в модели 2, общее увеличение системы (количество процессоров $n$ и запоминающих устройств $m$) не ведёт к уменьшению коэффициента использования $r$, а увеличение устройств $m$ при фиксированном числе процессоров $n$ (или наоборот) приводит к уменьшению коэффициента использования до $1/n$ и $1/m$ соответственно. На рис. 3 приведены зависимость коэффициента использования памяти от $r=n/m$ для описанной модели (кривая 1) и модели, в которой заявки процессоров никогда не конфликтуют друг с другом (кривая 2). Из рис. 3 следует, что предлагаемая модель на интервале значений $r$ от 0 до 1,15 даёт меньшую верхнюю границу для коэффициента использования памяти, чем модель 2, в которой заявки процессоров никогда не конфликтуют друг с другом.

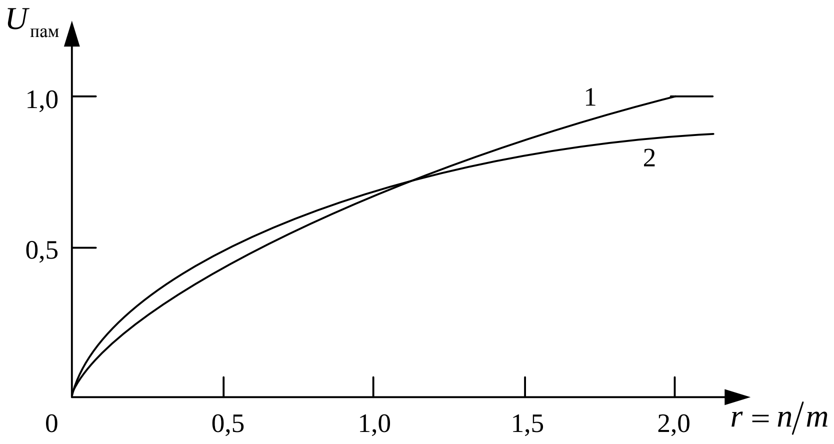


**Рис. 3.** Зависимость коэффициента использования памяти от $r = n/m$ для описанной модели 1 (кривая 1) и модели 2 (кривая 2)

Неравенство (2), по-видимому, имеет более общий характер и справедливо для широкого класса моделей. Например, оно выполняется для $n$ процессоров, взаимодействующих друг с другом, причём $c_{ij}$ есть суммарное время, которое процессор $i$ ждёт процессора $j$, а $m = n$, $\delta = 1$, $\alpha = 1/2$.

Рассмотренные процедуры могут быть использованы для динамического управления параллельными процессами на основе приоритетных расписаний выполнения параллельных программ.

## Список литературы


1. Муравьева-Витковская Л.А. Моделирование интеллектуальных систем. - СПб: НИУ ИТМО, 2012. - 145 с.
2. Топорков В.В. Модели распределенных вычислений. Москва, Физматлит, 2011. - 320 с.
3. Руденко Ю.М. *Учет зависимостей программных модулей по данным и последовательностям их выполнения при параллельных вычислениях* // Известия высших технических заведений. Технические науки. - Поволжский регион, 2009. - Вып. 3. - С. 67-75.
4. Пустовалов Е.В., Тюрликов А.М. *Случайный множественный доступ в векторном дизъюнктивном канале* // Проблемы передачи информации. 2013. Т.49, № 2. С. 17-33.

5. Kobliakov A., Turlikov A., Vinel A. *Distributed queue random multiple access algorithm for centralized data networks* // Proc. of the 10th IEEE International Symposium on Consumer Electronics (ISCE’06). - St.-Petersburg, Russia: 2006. - Pp. 290-295.
6. Flajolet P., Sedgewick R. Analytic combinatorics, Cambridge University Press, Cambridge, 2009. - 271 p.
7. Andreev S., Turlikov A., Vinel A. *Contention-based polling efficiency in broadband wireless networks* // Proc. of the 15th International Conference on Analytical and Stochastic Modeling Techniques and Applications. - 2008. - P. 295-309.
8. Ландо С.К. Введение в дискретную математику. - М.: МЦНМО, 2012. - 265 с.
9. Meshkov V.R., Omelchenko A.V., Petrov M.I., Tropp E.A. *Dyck and Motzkin triangles with multiplicities* // Mosc. Math. J. 2010. V. 10, no 3, pp. 611-628.
10. Райгородский А.М. Модели случайных графов. - М.: МЦНМО, 2011. - 136 с.