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

Моделирование нагрузки в неоднородных многопроцессорных компьютерных системах

07, июль 2015

Можаров Г. П.[1,*]

УДК: 681.142.1

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

Введение

Исследуется численный метод для анализа двухпроцессорных неоднородных компьютерных систем (КС) с детерминированными стратегиями загрузки и проводится сравнение нескольких таких систем, причём исследуются стратегии, минимизирующие время отклика, время обслуживания и максимизирующие пропускную способность. Показано, что системы, где максимизируется пропускная способность, являются наилучшими по рабочим характеристикам.

Рассматривается неоднородная КС с несколькими процессорами. Каждый процессор предоставляется в виде обслуживающего устройства с очередью. Это – типичная открытая модель с очередями для m независимых процессоров [1,2]. Процессор i полностью описывается средней интенсивностью обслуживания μ_i. Задания поступают в КС с интенсивностью λ. Промежутки между заданиями одинаково распределены и не зависят друг от друга. Распределение предлагается экспоненциальным с математическим ожиданием $1/\lambda$. Поступающее задание загружается в i-й процессор с вероятностью p_i. Так как один из m процессоров должен быть выбран, то $\sum_i^m p_i = 1$.

Обслуживание происходит в режиме поступления заданий. Время обработки задания предполагается независимой экспоненциально распределённой переменной с математическим ожиданием $1/\mu_i$.

Поведение КС может быть описано дискретным марковским процессом в непрерывном времени [2-4]. Мгновенное состояние s представляет собой вектор $(n_1, n_2, ..., n_i, ..., ..., n_m)$, где n_i есть число заданий в i-й очереди, включая обслуживаемое задание. Вероятности p_i выбираются либо произвольно, либо как функции параметров и текущего со-

стояния системы. Если вероятности не, зависит от состояния системы, то и недетерминированная стратегия загрузки не зависит от состояния системы.

Недетерминированные стратегии загрузки

Модели с недетерминированными стратегиями обслуживания рассматриваются как сети с очередями, что широко используется для анализа и прогноза обслуживания [1].

Недетерминированные стратегии, не зависящие от состояния системы, обычно используются в моделях с очередями, так как методика решения этих моделей хорошо разработана. Для открытых моделей с экспоненциально распределёнными очередями, вероятности равновесного состояния представляются в форме произведения. Для замкнутых моделей локальное равновесие существует в системах с экспоненциально распределённым временем обслуживания. Вероятности равновесного состояния определяются в форме произведения [4-6]. Если p_i фиксированы, то система процессоров может быть декомпозирована, и её можно анализировать как m независимых $M/M/1$ систем; i-ой системе соответствует интенсивность поступления заявок λp_i и интенсивность обслуживания μ_i. Если $\lambda p_i < \mu_i$, то i-я система не достигла насыщения, и среднее время цикла обслуживания задания равно $1/(\mu_i - \lambda p_i)$.

Далее, поскольку p_i есть доля заданий, обслуживаемых процессором i, то среднее время цикла обслуживания (T_{cp}) представляет собой взвешенную сумму времён циклов обслуживания для процессоров:

$$T_{cp} = \sum_{i=1}^{m} \frac{p_i}{\mu_i - \lambda p_i}. \qquad (1)$$

Естественно выбрать p_i пропорциональными скорости обслуживания процессора i, т.е. $p_i = \dfrac{\mu_i}{\sum_{j=1}^{m} \mu_j}$.

Стратегия, где p_i выбираются таким образом, что не минимизирует время цикла обслуживания. Однако она может гарантировать, что каждая очередь не будет насыщена до тех пор, пока $\lambda < \sum_{j=1}^{m} \mu_j$.

Из определения p_i и предыдущего неравенства следует, что

$$\lambda p_i = \lambda \frac{\mu_j}{\sum_{j=1}^{m} \mu_j} = \mu_i \frac{\lambda}{\sum_{j=1}^{m} \mu_j} < \mu_i.$$

Используя (1) и определение p_i, можно вычислить среднее время цикла обслуживания данной стратегии

$$T_{cp}^{PR} = \sum_{i=1}^{m} \frac{p_i}{\mu_i - \lambda p_i} = \frac{m}{\sum_{j=1}^{m} \mu_j - \lambda}. \qquad (2)$$

Детерминированные стратегии загрузки

В систему процессоров, описанную в предыдущем подразделе, включён диспетчер, распределяющий задания между процессорами. Если при поступлении задания КС находится состоянии $s = (n_1,\ n_2,\ ...,\ n_m)$, то задание будет направлено в очередь $q(s,\ C)$, определяемую величиной C.

Эта величина представляет собой критерии функционирования системы. Значение критерия при поступлении задания в i-ю очередь составляет $C(n_1,\ n_2,\ ...,\ n_{j+1},\ ...,\ n_m)$.

Детерминированность стратегии загрузки заданий определяется детерминированностью критерия функционирования [4].

Рассмотрим различные стратегии диспетчера.

Стратегия, минимизирующая время отклика. В данном случае поступающее, заранее помещается в процессор с наименьшим временем цикла. Данная стратегия может быть описана следующим образом. Пусть КС находится в состоянии $s = (n_1,\ n_2,\ ...,\ n_m)$. Тогда:

а) поступающее задание размещается в очереди i, если отношение её длины к интенсивности обслуживания минимально, то есть

$$\frac{n_{i+1}}{\mu_i} = \min\left\{ \frac{n_k + 1}{\mu_k} \middle| k = 1,\ 2,\ ...,\ m \right\};$$

б) если минимальное отношение не единственно, то диспетчер выбирает среди полученного множества процессор с максимальной интенсивностью обслуживания.

Формальная запись данной стратегии имеет следующий вид.

$$R(n_1,\ n_2,\ ...,\ n_k + 1,\ ...,\ n_m) \square \frac{n_k + 1}{\mu_k},$$

$$R_{\min}(s) \square \min\left\{ R(n_1,\ n_2,\ ...,\ n_k + 1,\ ...,\ n_m) \middle| k = 1,\ 2,\ ...,\ m \right\},$$

$$Q(s,\ R) \square \left\{ k \middle| R_{\min}(s) = R(n_1,\ n_2,\ ...,\ n_k + 1,\ ..., n_m) \right\}.$$

Тогда $q(s,\ R) = i$, где $i \in Q(s,\ R)$ и $\mu_i < \mu_j,\ \forall j \in Q(s,\ R),\ j \neq i.$

Стратегия, минимизирующая время обслуживания системы

Данная стратегия улучшает характеристики функционирования, хотя она и не является оптимальной стратегией загрузки. Пусть система находится в состоянии $s = (n_1,\ n_2,\ ...,\ n_m)$, тогда время системы $T(n_1,\ n_2,\ ...,\ n_m)$ определяется как ожидаемое

время завершения заданий, наводящихся в очередях. Это определение рекурсивно записывается в виде

$$T\left(n_1,\ n_2,\ ...,\ n_m\right)=E\ \text{[время обслуживания всех заданий]}\ =$$

$$=E\ \text{[время обслуживания следующих заданий]}\ +$$

$$+E\ \text{[время обслуживания остальных заданий]}\ =$$

$$=\frac{1}{\sum\limits_{\substack{j=1\\ n_j\neq 0}}^{m}\mu_j}+\sum\limits_{\substack{k=1\\ n_j\neq 0}}^{m}\left[\frac{\mu_k}{\sum\limits_{\substack{j=1\\ n_j\neq 0}}^{m}\mu_j}T\left(n_1,\ n_2,\ ...,\ n_k-1,\ ...,\ n_m\right)\right];$$

$$T\left(0,\ 0,\ ...,\ n_k,\ ...,\ 0,\ 0\right)=\frac{n_k}{\mu_k}.$$

Величина $1\Big/\sum\limits_{\substack{j=1\\ n_j\neq 0}}^{m}\mu_j$ представляет собой среднее время от момента, когда система находится в состоянии $\left(n_1,\ n_2,\ ...,\ n_m\right)$ до тех пор, пока очередное из $\sum\limits_{i=1}^{m}n_i$ заданий не будет обслужено. Это следует из предположения об экспоненциальном распределении времени обслуживания. Выражение $\mu_k\Big/\sum\limits_{\substack{j=1\\ n_j\neq 0}}^{m}\mu_j$ есть вероятность того, что следующее задание, обслуженное КС, находилось в очереди k. Если задание выходит из очереди k, то среднее время обслуживания оставшихся в очереди заданий может быть определено из рекурсивного соотношения, приведённого выше. Выбрав $T\left(n_1,\ n_2,\ ...,\ n_m\right)$ за критерий, можно сформулировать стратегию загрузки следующим образом [7,8]. Пусть

$$T_{\min}\left(s\right)\square\ \min\left\{T\left(n_1,\ n_2,\ ...,\ n_k+1,...,n_m\right)\middle|k=1,\ 2,\ ...,\ m\right\},$$

$$Q\left(s,\ R\right)\square\ \left\{k\middle|T_{\min}\left(s\right)=T\left(n_1,\ n_2,\ ...,\ n_k+1,\ ...,\ n_m\right)\right\}.$$

Тогда $q\left(s,\ R\right)=i$, где $i\in Q\left(s,\ R\right)$ и $\mu_i<\mu_j,\ \forall j\in Q\left(s,\ R\right),\ j\neq i.$

В случае двухпроцессорных КС

$$T\left(n_1,\ n_2\right)=\frac{1}{\mu_1+\mu_2}+\frac{\mu_1}{\mu_1+\mu_2}T\left(n_1-1,\ n_2\right)+\frac{\mu_2}{\mu_1+\mu_2}T\left(n_1,\ n_2-1\right),$$

$$T\left(n_1,\ 0\right)=\frac{n_1}{\mu_1}\ \text{и}\ T\left(0,\ n_2\right)=\frac{n_2}{\mu_2}.$$

Стратегия, максимизирующая пропускную способность системы

Стратегии, минимизирующие время отклика процессора и системное время зависят от длины очередей и интенсивностей обслуживания. Диспетчер может также использовать

информацию об интенсивности поступления заданий. Покажем, каким образом можно улучшить функционирование с помощью этой добавочной информации. Рассмотрим систему с одним процессором, эквивалентную по своим функциональным характеристикам многопроцессорной КС. Пусть μ_e есть интенсивность обслуживания данной КС. Кроме того, μ_e зависит от распределения нагрузки между очередями. После обслуживания очередного задания диспетчер перераспределяет нагрузку таким образом, чтобы минимизировать среднее время цикла обращения задания. С увеличением интенсивности обслуживания время цикла должно уменьшаться. Интенсивность обслуживания μ_e в течение времени между двумя поступлениями заявок представляет собой пропускную способность КС, распределение изменяется только в моменты поступления заданий. Стратегия размещает поступившее задание в той очереди, изменение длины которой максимизирует ожидаемое значение пропускной способности КС в интервале времени до поступления следующего задания.

Пропускная способность КС есть сумма пропускных способностей процессора. Средняя пропускная способность i-го процессора, обозначаемая $T_{\Pi_i}(n_i)$, определяется следующим образом. Пусть Q_{ik} есть вероятность того события, что k заданий покидают i-й процессор в течение периода времени τ при условии, что первоначально в очереди было n_i заданий. Поскольку время обслуживания i-м процессором распределено экспоненциально, то процесс обслуживания задании является пуассоновским, и вероятность Q_{ik} задаётся выражениями [6,9,10]

$$Q_{ik} = \frac{(\mu_i \tau)^k e^{-\mu_i \tau}}{k!}, \; 0 \le k < n_i,$$

$$Q_{in_i} = 1 - \sum_{k=1}^{n_i - 1} Q_{ik} = \sum_{k=n_i}^{\infty} \frac{(\mu_i \tau)^k e^{-\mu_i \tau}}{k!}.$$

Ожидаемая пропускная способность процессора i в интервале времени τ между двумя заявками есть среднее число заданий, обслуживаемых в единицу времени. Эта величина равна $\sum_{k=1}^{n_i} \frac{k \cdot Q_{ik}}{\tau}$, где n_i есть первоначальное число заданий в очереди i. Вероятность того, что интервал имеет длину τ, равна $f(\tau) = \lambda e^{-\lambda \tau}$.

Введём обозначение $X = \mu_i / (\lambda + \mu_i)$. Тогда средняя пропускная способность процессора (Π_i) может быть описана выражением

$$T_{\Pi_i}(n_i) = \lambda \left[\sum_{k=1}^{n_i - 1} X^k + n_i \sum_{k=n_i}^{\infty} \frac{X^k}{k} \right] = \lambda \left[\sum_{k=1}^{n_i - 1} \left(X^k - \frac{n_i}{k} X^k \right) + n_i \sum_{k=1}^{\infty} \frac{X^k}{k} \right] =$$

$$= \lambda \left[\sum_{k=1}^{n_i - 1} \left(1 - \frac{n_i}{k} \right) X^k - n_i \ln(1 - X) \right].$$

Если система находится в состоянии $\left(n_1,\ n_2,\ ...,\ n_m\right)$, то средняя пропускная способность $T_\Pi\left(n_1,\ n_2,\ ...,\ n_m\right)$ есть сумма пропускных способностей $T_{\Pi_i}\left(n_i\right)$, т.е.

$$T_\Pi\left(n_1,\ n_2,\ ...,\ n_m\right)=\sum_{i=1}^{m}\lambda\left[\sum_{k=1}^{n_i+1}\left(1-\frac{n_i}{k}\right)\left(\frac{\mu_i}{\lambda+\mu_i}\right)^k-n_i\ln\left(\frac{\lambda}{\lambda+\mu_i}\right)\right].$$

Стратегия, максимизирующая среднюю пропускную способность КС, формально записывается в следующем виде. Пусть

$$T_{\Pi_{\max}}\left(s\right)\square\ \max\left\{T_\Pi\left(n_1,\ n_2,\ ...,\ n_k+1,\ ...,\ n_m\right)\middle|k=1,\ 2,...,\ m\right\},$$

$$Q\left(s,\ R\right)\square\ \left\{k\middle|T_{\Pi_{\max}}\left(s\right)=T_\Pi\left(n_1,\ n_2,\ ...,\ n_k+1,\ ...,\ n_m\right)\right\}.$$

Тогда $q\left(s,\ R\right)=i$, где $i\in Q\left(s,\ R\right)$ и $\mu_i<\mu_j,\ \forall j\in Q(s,\ R),\ j\neq i.$

Основным достоинством данной стратегии является то, что она использует информацию о средней интенсивности поступления заданий.

Рекурсивная процедура анализа двухпроцессорных КС

Рассмотрим замкнутую двухпроцессорную неоднородную КС с интенсивностью поступления λ и интенсивностями обслуживания процессорами μ_1 и μ_2, где $\mu_1>\mu_2$. Диспетчер посылает поступающие задания в одну из двух очередей. КС описывается набором состояний $\left(i,\ j\right)$, где числа i и j есть количества заданий на первом и втором процессорах. Множество состояний разбивается на два подмножества – сбалансированных и несбалансированных состояний. Из любого несбалансированною состояния КС за конечное число переходов попадает в сбалансированное состояние. Если заданы равновесные вероятности состояний P_{ij}, то бесконечное множество состояний может быть сведено к множеству с конечным числом состоянии. Такое приближенное представление даёт незначительную ошибку, если КС не насыщена и число состояний достаточно велико, поскольку P_{ij} быстро сходятся к нулю при $i\square\ 0$ и $j\square\ 0$.

Рекурсивная процедура анализа двухпроцессорных КС начинается с определения значения вероятности состояния $\left(N_1,\ N_2\right)$. Затем вычисляются предварительные вероятности состояния, предшествующего $\left(N_1,\ N_2\right)$ на траектории сбалансированных состояний. Эта процедура выполняется рекурсивно до тех пор, пока не будет вычислена вероятность состояния, предшественником которого является $\left(0,\ 0\right)$. Тогда все ранее определённые вероятности нормализуются. Приведём описание данного алгоритма с помощью рекурсивной процедуры. X есть переменная, значение которой равно вероятности граничного состояния $\left(N_1,\ N_2\right)$; $preb\left(i,\ j\right)$ означает состояние, являющееся предшественником $\left(i,\ j\right)$ на сбалансированной траектории.

1. Установить $(i,\ j)=(N_1,\ N_2)$.

2. Установить $P_{ij}=\text{const}$ (некоторая произвольная величина).

3. От $K=(N_1+N_2)$, до 1 выполнить шаги 4 -15.

4. Установить $(i',\ j')=preb(i,\ j)$.

5. Если $i=i'$, то выполнить шаги 6-8.

6. Установить $P_{0j}=X$.

7. Для $I=1,\ ...,\ i-1$ найти зависимость $P_{I,j}$ от X , используя уравнение глобального баланса (УГБ) состояния $(I-1,\ j)$.

8. Решая УГБ состояния $(i-1,\ j)$, вычислить X .

9. Если $j=j'$, то выполнить шаги 10-12.

10. Установить $P_{i0}=X$.

11. Для $J=1,\ ...,\ j-1$ найти зависимость $P_{i,J}$ от X , используя УГБ состояния $(i,\ J-1)$.

12. Решая УГБ состояния $(i,\ j-1)$, вычислить X .

13. Решая УГБ состояния $(i,\ j)$, вычислить $P_{i',j'}$.

14. Установить $(i,\ j)=(i',\ j')$.

15. Нормализовать все P_{ij} .

Поскольку уравнение глобального баланса каждою состояния за исключением $(N_1,\ N_2)$ и $(0,\ 0)$ используется один раз, и время решения этого уравнения не зависит от N_1 и N_2, то объем используемою времени пропорционален общему количеству состояний $|S|=(N_1+1)(N_2+1)$.

При рассмотрении двухпроцессорных КС удобнее заменить функции максимума и минимума оператором отношения R . Пусть $C(i+1,\ j)$ и $C(i,\ j+1)$ есть значения критерия после поступления задания в первую или вторую очередь. Предполагается, что $\mu_1>\mu_2$ Тогда стратегию загрузки можно описать в виде

$$q(s,\ C)=\begin{cases}1, \text{ если } C(i+1,\ j)\,\mathrm{R}\,C(i,\ j+1)=true,\\ 2, \text{ иначе,}\end{cases}$$

где R эквивалентно ≥.

Сформулируем свойства этой стратегии, позволяющие использовать приведенную выше рекурсивную процедуру.

Свойство 1. Стратегия загрузки является детерминированной. Тогда траектория стратегии загрузки есть непрерывная цепочка переходов, соответствующих поступлению новых заданий. Эта траектория начинается в состоянии $(0,\ 0)$.

Свойство 2. Траектория стратегии разбивает состояние системы на два подмножества. Для каждого состояния $(i,\ j)$ в верхнем несбалансированном подмножестве

$$C(i+1,\ j+n)\,\mathrm{R}\,C(i,\ j+n+1) = \text{true} \quad \forall n \geq 0,$$

а для состояния $(i,\ j)$ из нижнего несбалансированного подмножества

$$C(i+n,\ j+1)\,\mathrm{R}\,C(i+n+1,\ j) = \text{true} \quad \forall n \geq 0.$$

Заключение

Для любой двухпроцессорной КС со стратегией, удовлетворяющей свойствам 1 и 2, может быть получено решение с помощью рекурсивной процедуры. Стратегии, минимизирующие время отклика, системное время, максимизирующие пропускную способность КС, удовлетворяют данным свойствам. Наилучшей является стратегия, максимизирующая пропускную способность.

Список литературы

1. Андреев А.М., Можаров Г.П., Сюзев В.В. Многопроцессорные вычислительные системы: теоретический анализ, математические модели и применение. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2011. - 334 с.
2. Кельберт М.Я., Сухов Ю.М. Вероятность и статистика в примерах и задачах. Т. 2: Марковские цепи как отправная точка теории случайных процессов и их приложения. - М.: МЦНМО, 2010. - 560 с
3. Foss S., Shneer S., Turlikov A. *Stability of a Markov-modulated Markov chain, with application to a wireless network governed by two protocols.* Stochastic Systems. 2012. Vol. 2, no. 1, pp. 208-231.
4. Топорков В.В. Модели распределенных вычислений. Москва, Физматлит, 2011. - 320 с.
5. Райгородский А.М. Комбинаторика и теория вероятностей. - Долгопрудный: Издательский Дом «Интеллект», 2013. - 117 с.
6. Райгородский А.М. *Теория вероятностей и комбинаторная геометрия* // Математика в задачах. М.: МЦНМО, 2009. С. 381-384.
7. Андреев А.М., Березкин Д.В., Можаров Г.П., Свирин Ил.С. *Математическое моделирование надежности компьютерных систем и сетей* // Вестник МГТУ им. Баумана. Сер. «Приборостроение». - 2012. Спец. выпуск «Моделирование и идентификация компьютерных систем и сетей». - с. 3-46.

8. Peskir G., Shiryaev A. Optimal stopping and free-boundary problems. Basel: Birkhauser, 2006. 500 p.
9. Стенли Р. Перечислительная комбинаторика: Пер. с англ. Т. 2. - М.: Мир, 2009. - 767 с. [Stanley R.P. Enumerative Combinatorics. Vol. 2. Camdridgt University Press, 2001 – 531 p.].
10. Хританков А.С. «*Модели и алгоритмы распределения нагрузки. Модель коллектива вычислителей. Модели с соперником*». Информационные технологии и вычислительные системы, № 2. - 2009. С. 65-80.