

Модель децентрализованного управления передачей данных в многопроцессорных компьютерных системах

06, июнь 2015

Можаров Г. П.^{1,*}

УДК: 681.322.013.2

¹Россия, МГТУ им. Н.Э. Баумана

^{*}mojarov_g@mail.ru

Введение

Уменьшение габаритов компьютерных систем (КС) делает вполне реальным создание кластеров, состоящих из нескольких соединённых между собой КС. Количество входящих в один кластер КС определяется как возможностью распараллеливания задачи и требованиями к вычислительной мощности, так и наличием эффективных алгоритмов управления взаимосвязью входящих в кластер КС [1-3]. Отличительной особенностью КС является возможность изменения архитектуры кластера и числа входящих в кластер КС. Для управления обменом информацией между входящими в кластер КС используется полностью децентрализованный протокол управления; это позволяет одновременно передавать информацию между произвольным числом узлов. Входящие в один кластер узлы могут синхронизироваться как от общего генератора тактовой частоты, так и от независимых генераторов. Передача информации между каждой парой узлов строго синхронизирована, что обеспечивает получение вовремя требуемых операндов, параметров [4,5]. По сравнению с централизованным управлением применение полностью децентрализованного управления обеспечивает более высокую живучесть и потенциально неограниченные возможности расширения.

Обоснование и построение модели децентрализованного управления в КС

В качестве узлов в системе используем двухпроцессорные КС. В каждом узле имеется свой автономный арбитр и от двух до шести шин, используемых для соединения данного узла с его ближайшими соседями. Шесть шин, соединяющих узел с его левым, правым, верхним, нижним, находящимся перед ним и находящимся за ним соседними узлами, обозначаются l , r , u , d , f и b , соответственно. В зависимости от числа узлов, подсоединённых к узлу, используются различные подсистемы шин.

нённых к каждому узлу кластера, возможна реализация архитектуры двухмерного массива, куба, кольца и т.д. [1,5,6]

Реализованный в каждом узле автономный арбитр обеспечивает накопление и постановку в очередь запросов на доступ к памяти, поступающих как от процессоров данного узла, так и от соседних узлов. Запросы данного узла на доступ к памяти соседнего узла обозначаются O_x , где x – идентификатор узла, к памяти которого производится обращение, $x = l, r, u, d, f$ или b . Запросы на доступ к памяти данного узла от внешних узлов обозначаются I_γ , где γ – идентификатор узла, от которого поступил запрос, $\gamma = r, l, d, u, b$ или f .

Для любой пары непосредственно соединённых друг с другом узлов запросы O_x и I_γ называются взаимно соответствующими друг другу или совместимыми, если они относятся к одной и той же линии связи. Совместимыми являются, например, пары запросов O_l и I_r , O_u и I_d и т.д.

Возникновение в каком-либо узле запроса O_x приводит возникновению в одном из соседних узлов соответствующего данному запросу I_γ . Обработка запроса реализуется двумя арбитрами – запрашивающего узла и запрашиваемого узла. В ответ на поступивший запрос арбитр запрашиваемого узла выставляет сигнал разрешения доступа к памяти; этот сигнал обозначается G_γ . Для запрашивающего узла этот сигнал является сигналом подтверждения и обозначается A_x . После завершения доступа к памяти сигналы разрешения и подтверждения сбрасываются, а запросы O_x и I_γ удаляются из соответствующих очередей.

Арбитр каждого узла в течение одного цикла памяти занимается обслуживанием только одного запроса. В такой ситуации характеристики системы существенно зависят от используемой дисциплины обслуживания, поскольку любой запрос может быть удовлетворён только в том случае, когда в данном цикле памяти свободны арбитры запрашивающего и запрашиваемого узлов [6-8].

Пусть i – число запросов, находящихся в очереди какого-либо узла, а $\lambda_{i,i-1}$ – вероятность уменьшения этой очереди в течение одного цикла памяти до $(i-1)$ элементов, то есть вероятность того, что в течение данного цикла памяти запрос на доступ будет удовлетворён.

«Лучшей» дисциплиной является первоочередное обслуживание всех поступивших извне запросов до обслуживания сгенерированных в данном узле запросов, поскольку при этом увеличивается вероятность того, что арбитры соседних узлов смогут «договориться» между собой [5,9].

В принципе при такой дисциплине обслуживания запросов возможно возникновение взаимных блокировок в системе, когда в замкнутой цепи из нескольких узлов будет цир-

кулировать соответствующим образом упорядоченная последовательность запросов, так что в любом цикле ни один из запросов не сможет быть выполнен [7,9,10].

Представляет интерес оценка сложности реализации децентрализованных арбитров и сопоставление со сложностью и стоимостью реализации общесистемного централизованного арбитра. Приблизённо сложность можно считать пропорциональной числу состояний, в которых может пребывать цифровая система. В кластере с трёхмерной архитектурой, состоящем из R строк, C столбцов и L слоёв, в котором каждый узел соединён с I соседними узлами, число состояний, в которых может пребывать каждый узел, равно $1 + 2I$. Таким образом, сложность централизованного арбитра, управляющего всеми RCL узлами, пропорциональна величине

$$(1 + 2I)^{RCL}.$$

В системе с децентрализованными арбитрами сложность каждого арбитра пропорциональна величине

$$(1 + I) \sum_{i=0}^I C_i^I,$$

где C_i^I – число сочетаний из I элементов по i .

Известно, что стоимость реализации цифровой системы как функция сложности, определяемой числом состояний, содержит как линейный, так и логарифмический члены. Соотношение между коэффициентами при линейном и логарифмическом членах зависят от выполняемых функций, методов реализации системы и т.д. Таким образом, для отношения стоимости множества децентрализованных арбитров к стоимости одного централизованного арбитра имеем

$$C = \frac{RCL \left[\alpha (1 + I) \sum_{i=0}^I C_i^I + \beta \log_2 (1 + I) \sum_{i=0}^I C_i^I \right]}{\gamma (1 + 2I)^{RCL} + \delta (RCL) \log_2 (1 + 2I)},$$

где α , β , γ и δ – коэффициенты пропорциональности при линейном и логарифмическом членах в зависимостях стоимости от сложности.

В КС с полностью децентрализованным управлением все запросы на доступ к памяти обслуживаются одновременно двумя арбитрами – запрашивающего узла и запрашиваемого узла; поэтому эффективность системы определяется вероятностью того, что арбитры соседних узлов в каждом цикле смогут «договориться» между собой и обеспечить выполнение какого-либо из существующих запросов.

Пусть P – вероятность того, что в течение текущего цикла процессором данного узла будет сгенерирован запрос на доступ к памяти какого-либо из соседних узлов. Полагая, что вероятность обращения ко всем узлам, соединённым с данным, одна и та же, получаем, что для любого узла вероятность принять в течение одного цикла запрос на доступ к его памяти от какого-либо конкретного узла равна $P/(2I)$. Коэффициент «2» в знаменателе обусловлен тем, что при асинхронном функционировании входящих в один кластер уз-

лов синхронизация в запрашиваемом узле с вероятностью $1/2$ отстаёт от синхронизации в запрашивающем узле, и с вероятностью $1/2$ опережает её. При отставании запрос не может быть обработан в течение данного цикла.

Каждый узел одновременно выступает в роли, как запрашиваемого узла, так и запрашивающего узла. Поэтому для каждого узла вероятность получения в течение одного цикла ровно i запросов равна

$$r_i = (1-P)C_i^I \left(\frac{P}{2I}\right)^i \left(1 - \frac{P}{2I}\right)^{I-i} + PC_{i-1}^I \left(\frac{P}{2I}\right)^{i-1} \left(1 - \frac{P}{2I}\right)^{I-(i-1)}, \quad (1)$$

причём $\sum_{i=0}^{I+1} r_i = 1$.

Из анализа выражения (1) следует, что величины r_i , $i=1, 2, \dots, I+1$ с увеличением i уменьшаются экспоненциально и не очень существенно зависят от I .

В таблице приведены значения коэффициентов r_i для некоторых частных значений P и I . Предельный случай $P=1$, $I=6$ соответствует ситуации, когда в кластере с трёхмерной архитектурой все шесть узлов, окружающих данный, выполняют программу, находящуюся в памяти данного узла.

Таблица. Зависимость вероятностных коэффициентов r_i от значений величин P и I

P	I	r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7
0,2	2	0,722	0,2565	0,021	0,0005	—	—	—	—
0,2	6	0,7232	0,2543	0,0216	0,0008	0,000019	$2,3 \cdot 10^{-6}$	$1,53 \cdot 10^{-9}$	$0,43 \cdot 10^{-11}$
1	6	0	0,5933	0,3236	0,0735	0,0089	$6,1 \cdot 10^{-4}$	$0,22 \cdot 10^{-4}$	$0,33 \cdot 10^{-6}$

Среди имеющейся в любом конкретном узле очереди из i запросов как максимум один запрос сгенерирован процессором данного узла и адресован одному из соседних узлов (запрос типа O_x), а все остальные запросы поступили из соседних узлов (запросы типа I_y). В течение данного цикла памяти может быть выполнен любой из запросов типа I_y ; для того, чтобы мог быть выполнен запрос типа O_x , необходимо, чтобы он был воспринят запрашиваемым узлом и от последнего поступило подтверждение. Для среднего времени обслуживания запросов типа O_x имеем

$$m = 1 + \sum_{i=0}^{I+1} (i) r_i + \frac{1}{2} \sum_{i=1}^{I+1} (i-1) r_i.$$

Это время включает ожидание обработки в запрашиваемом узле. Численный анализ показывает, что в зависимости от значения величины P среднее время ожидания лежит в диапазоне от 1,2 до 2,8 циклов. С учётом потерь времени на передачу данных из одного узла в другой величина m должна быть несколько увеличена. Приблизённо среднее время m можно считать величиной постоянной и равной примерно 2,5 циклов.

Для вероятности уменьшения в течение одного цикла очереди запросов на единицу $\lambda_{i,i-1}$ или, другими словами, вероятности того, что в течение цикла будет обслужен запрос типа O_x или I_γ , имеем:

$$\lambda_{i,i-1} = \frac{IC_{i-1}^I \left(1 - \frac{1}{m}\right) + C_i^I}{IC_{i-1}^I + C_i^I}.$$

Вероятность обслуживания в течение одного цикла двух запросов (обязательно разных типов) равна

$$\lambda_{i,i-2} = \frac{IC_{i-1}^I (1/m)}{IC_{i-1}^I + C_i^I}.$$

Видно, что при $i \geq 2$, $\lambda_i = \lambda_{i,i-1} + \lambda_{i,i-2} = 1$; следовательно, в течение цикла новый арбитр обслуживает либо один запрос, либо два. Продолжительность обслуживания запросов типа I_γ составляет один цикл, запросов типа O_x – приблизительно 2,5 циклов.

Для определения эффективности использования процессоров системы используется представленная на рис. 1 диаграмма переходов из одних состояний в другие; под состоянием каждого узла понимается длина очереди запросов i ; используемые на рис. 1 обозначения: $P(i)$ – стационарная вероятность пребывания в состоянии i , $\lambda_i = \lambda_{i,i-1} + \lambda_{i,i-2}$ при $i \geq 2$, $\lambda_1 = \lambda_{10}$.

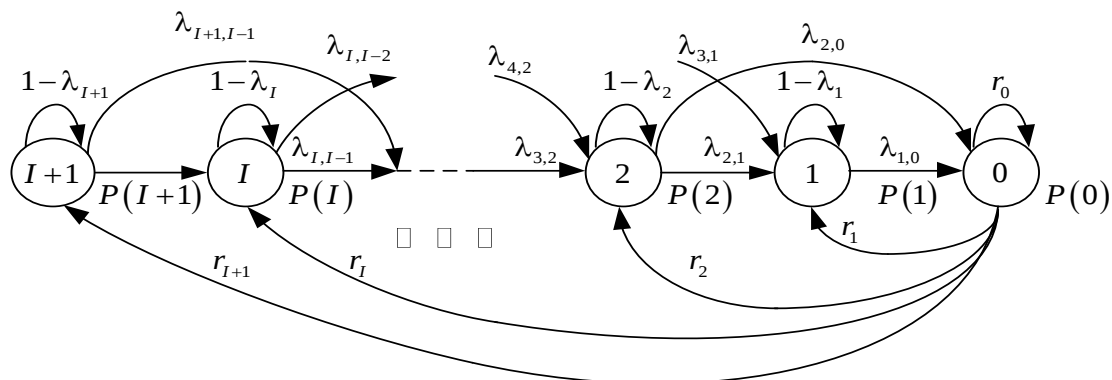


Рис. 1. Граф состояний системы

Коэффициент использования процессоров узла численно равен вероятности отсутствия очереди запросов $P(0)$ [8-10]:

$$U = P(0) = T^{-1} = \left[1 + \sum_{i=0}^{I+1} \sum_{j=i}^{I+1} \left(\frac{k_{ji} r_j}{\lambda_i} \right) \right]^{-1}. \quad (2)$$

Здесь k_{ji} – суммарная вероятность перехода из состояния j в состояние i ; сумма берётся по всем возможным путям перехода. Например, из состояния $j=3$ переход в состояние $i=1$ возможен как непосредственно, так и через состояние $i=2$. Поэтому

$$k_{3,1} = \frac{(\lambda_{3,2})(\lambda_{2,1})}{(\lambda_{3,2} + \lambda_{3,1})(\lambda_{2,1} + \lambda_{2,0})} + \frac{\lambda_{3,1}}{(\lambda_{3,2} + \lambda_{3,1})} = (\lambda_{3,2})(\lambda_{3,1}) + \lambda_{3,1}. \quad (3)$$

Аналогичным образом

$$\begin{aligned} k_{4,1} &= (\lambda_{4,3})(\lambda_{3,2})(\lambda_{2,1}) + (\lambda_{4,3})(\lambda_{3,1}) + (\lambda_{4,2})(\lambda_{2,1}), \\ k_{4,2} &= (\lambda_{4,3})(\lambda_{3,2}) + (\lambda_{4,2}) \end{aligned} \quad (4)$$

и т.д. Уравнения (1-4) дают зависимость коэффициента использования процессоров от вероятности генерации запроса на доступ к памяти соседнего узла. Графически эта зависимость представлена на рис. 2, где по оси абсцисс – вероятность генерации запроса на доступ к памяти соседнего узла P , по оси ординат – коэффициент использования процессоров U .

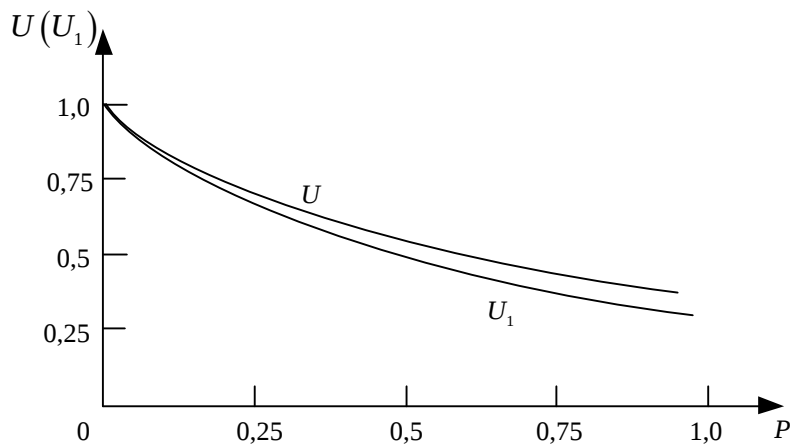


Рис. 2. Зависимость величин U и U_1 , от значения величины P

В первоначальной версии арбитра в каждом цикле мог обслуживаться только один запрос; следовательно, для этого арбитра $\lambda_{i,i-2} = 0$, и вместо (4) для коэффициента использования процессоров имеем

$$U_1 = \left[1 + \sum_{i=1}^{I+1} \sum_{j=i}^{I+1} \frac{r_j}{\lambda_i} \right]^{-1}.$$

Заключение

По сравнению с централизованным управлением применение полностью децентрализованного управления обеспечивает более высокую живучесть, выигрыш в быстродей-

ствии по сравнению с классической фон-неймановской архитектурой – от 2-3,5 раз и имеет потенциально неограниченные возможности расширения.

Недостаток предложенного подхода к созданию кластеров – уменьшение скорости передачи больших блоков данных из памяти одного узла в память другого узла в 2,5-3 раза по сравнению с предельной скоростью, определяемой быстродействием памяти.

Список литературы

1. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы: Общий подход / Пер. с англ. - М.: БИНОМ. Лаборатория знаний, 2009. - 406 с. [Miller R., Boxer L. Algorithms Sequential And Parallel: A Unified Approach. Prentice Hall, New Jersey. 2000].
2. Погребной А.В. Определение числа и топологии размещения станций многопроцессорной вычислительной системы // Известия Томского политехнического университета. - 2006. - Т. 309. - № 7. - С. 160-164.
3. Хританков А.С. Модели и алгоритмы распределения нагрузки. Алгоритмы на основе сетей СМО // Информационные технологии и вычислительные системы. - 2009. № 3. С. 33-48.
4. Мусина Л.В., Руденко Ю.М. Временная задержка на вычислительных модулях при реализации граф-схем // Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение. Спец. выпуск «Информационные технологии и компьютерные системы». - 2011. - С. 70-74.
5. Коваленко В.Н., Коваленко Е.И., Корягин Д.А., Семячкин Д.А. Управление параллельными заданиями в гриде с неотчуждаемыми ресурсами. - М.: ИПМ им. М.В. Келдыша РАН, 2007. - 28 с.
6. Рябченко А.А., Самосват Е.А. О числе подграфов в случайном графе Барабаши-Альберт // Изв. РАН. Сер. матем. - 2012. - Т. 6, № 3. - С. 183-202.
7. Grechnikov E.A., Gusev G.G., Ostroumova L.A., Pritykin Yu.L., Raigorodskii A.M., Serdyukov P., Vinogradov D.V., Zhukovskiy M.E. Empirical Validation of the Buckley-Osthus Model for the Web Host Graph // The proceedings of The 21st ACM Conference on Information and Knowledge Management, 2012. - Pp. 1577-1581.
8. Grechnikov E.A. The degree distribution and the number of edges between nodes of given degrees in the Buckley-Osthus model of a random web graph // Internet Math. - 2012. - V. 8, no 3. - Pp. 257-287.
9. Райгородский А.М. Линейно-алгебраический метод в комбинаторике. - М.: МЦНМО, 2007. - 136 с.
10. Алон Н., Спенсер Дж. Вероятностный метод. Москва, Бином. Лаборатория знаний, 2007, 320 с. [Alon N., Spencer J.H. The Probabilistic Method. Second edition, Tel Aviv and New York. 2000, 301 p.]