УДК 004.272.43

Т. С. Дьячук, Р. К. Кудерметов

РАСПРЕДЕЛЕНИЕ ЗАДАНИЙ В МНОГОПРОЦЕССОРНОЙ СИСТЕМЕ

Разработана математическая модель оптимального распределения заданий для параллельной вычислительной системы с учетом загруженности ее вычислительных узлов. Представлены результаты экспериментальных исследований модели, которые демонстрируют значительный прирост производительности по сравнению с равномерным распределением параллельного задания по узлам кластера.

ВВЕДЕНИЕ

Производительность распределенных и параллельных компьютерных систем, предназначенных для выполнения сложных вычислительных задач, существенно зависит от правильного распределения вычислительных заданий между узлами этих систем. Распределение заданий между параллельно работающими вычислительными узлами называют балансировкой нагрузки. Существует множество стратегий, методов и программных средств балансировки нагрузки в кластерных системах (PBS, MOSIX и др.) [1, 2]. Однако многообразие задач и структур вычислительных систем вызывают необходимость разработки и исследования все новых подходов к балансировке, которые более тонко учитывают особенности решаемых вычислительных задач и средств для их решения.

В данной работе предложена модель балансировки нагрузки для Beowulf-кластера при решении на нем одновременно нескольких задач (заданий), реализованных в виде параллельных программ. Кластер представляет собой множество компьютеров (узлов), объединенных сетью, работающих под правлением операционной системы Red Hat Linux 7.2. Для организации параллельных вычислений используется стандарт MPI 1, реализованный в библиотеке mpich 1.2.4 Аргоннской национальной лаборатории США.

ПОСТАНОВКА ЗАДАЧИ УПРАВЛЕНИЯ ЗАДАНИЯМИ

Под заданием будем понимать параллельную программу, части которой могут выполняться параллельно на различном количестве вычислительных узлов кластера, причем на одном узле может выполняться несколько частей параллельной программы, в зависимости от их распределения между узлами кластера. Задания могут возникать в произвольные моменты времени, в зависимости от потребностей пользователей, получающих доступ к ресурсам кластера.

Эффективное распределение множества заданий зависит от различных факторов, в частности загруженности и объема свободной памяти узлов кластера, интенсивности обмена узлов через сеть, а так же и от характера самих заданий, например, сложности, их приоритета, количества параллельных частей. Распределение будет эффективным, если каждое из заданий выполняется за минимально возможное время.

Следует заметить, что время, затрачиваемое на распределение заданий, должно быть существенно меньшим по сравнению со временем выполнения заданий.

В данной работе авторами предложен подход, при котором учитывается только степень загруженности узлов кластера, и задача эффективного распределения сформулирована как задача оптимизации с нелинейной целевой функцией:

$$\Theta(\Omega(\beta_1, \beta_2, \ldots, \beta_N), \Psi) \to \min, \qquad (1)$$

где Θ – время выполнения задания; Ψ – функция, характеризующая задачу (например: степень ее параллелизма, требуемая память и т. д.); Ω – функция, характеризующая кластерную систему; β_i – загруженность i-го узла кластера; N – число узлов кластера.

МОДЕЛЬ РАСПРЕДЕЛЕНИЯ ЗАДАНИЙ

Назовем размерностью задания $S_{\max}$ количество подзадач, на которое может быть разбито задание, и, соответственно, такое количество параллельных ветвей может иметь программа, реализующая задание. Таким образом, максимальное число узлов, которое может быть задействовано для выполнения задания, равно его размерности.

Введем обозначения:

N – число узлов кластера;

T – время выполнения задания на одном узле, при условии, что узел не загружен другими заданиями;

s – оптимальное количество ветвей задания для текущей загрузки узлов кластера ($s = 1 \ldots S_{\max}$);

$t = T/s$ – время выполнения одной подзадачи на узле, при условии, что он не загружен другими заданиями;

x_i – число подзадач, выполняемых на i-м узле ($x_i \geq 0$, $i = 1, \ldots, N$);

k_i – коэффициент замедления выполнения заданий на i-м узле, за счет уже выполняющихся заданий на данном узле. Коэффициент рассчитывается, исходя из процессорного времени, которое может быть выделено пользовательской подзадаче и зависит от приоритетов процессов, выполняющихся на данном узле.

Напомним, что любая программа, не являющаяся частью ядра, может выполняться в ОС Linux лишь в виде процесса. У каждого процесса есть приоритет планирования, который по умолчанию равен 20. Приоритет может быть изменен при помощи системного вызова nice, вычитающего свое значение из 20. Поскольку значение nice находится в диапазоне от −20 до +19, приоритеты всегда попадают в промежуток от 1 до 40. Чем выше приоритет, тем большая доля процессорного времени выделяется процессу [3].

Обозначим значение nice для уже выполняющихся процессов на i-м узле кластера как ni_{ig}, где g – порядковый номер процесса. Таким образом, каждый процесс на узле кластера имеет значение ni_{ig} в диапазоне от −20 до +19. Для пользовательских подзадач введем значение nice как ni, по умолчанию оно равно 0 (т. е. $ni = 0$).

Будем считать, что доля процессорного времени d, выделяемая процессу g, пропорциональна его приоритету:

$$d = (20 - ni_{ig}) \cdot \frac{1}{\sum_{j=0}^{J} (20 - ni_{ij})}, \qquad (2)$$

где J – число процессов на i-м узле кластера.

В общем случае коэффициент замедления для процесса g находится как

$$k_i = \frac{1}{d} = \frac{\sum_{j=0}^{J} (20 - ni_{ij})}{(20 - ni_{ig})}. \qquad (3)$$

Поскольку пользовательские подзадачи по умолчанию будут иметь значение nice, равное 0, то

$$k_i = \frac{\sum_{j=0}^{J} (20 - ni_{ij}) + x_i \cdot 20}{20} = \sum_{j=0}^{J} \left(1 - \frac{ni_{ij}}{20}\right) + x_i. \qquad (4)$$

Введем переменную z_i, характеризующую степень загруженности i-го узла кластера:

$$z_i = \sum_{j=0}^{J} \left(1 - \frac{ni_{ij}}{20}\right). \qquad (5)$$

Данная переменная является суммой приоритетов всех процессов, выполняющихся на узле, разделенная на 20. Например, если значение nice для всех процессов, выполняющихся на узле, равно 0, то степень загруженности равна числу процессов на данном узле $z_i = J$. Максимальное значение $z_i = 2J$ достигается при значении nice равному −20 для всех процессов.

Таким образом, коэффициент замедления равен

$$k_i = z_i + x_i. \qquad (6)$$

Время выполнения подзадач на i-м узле L_i

$$L_i = t \cdot k_i = k_i \cdot \frac{T}{s}. \qquad (7)$$

Так как время выполнения параллельной задачи будет равно максимальному значению L_i и T является постоянным, то получаем функцию, которую необходимо минимизировать:

$$L_i = \max_i \left(\frac{k_i}{s}\right). \qquad (8)$$

Таким образом, необходимо найти

$$L_{\text{опт}} = \min_s \max_i \left(\frac{z_i + x_i}{s}\right). \qquad (9)$$

Решая задачу минимизации, находим оптимальное число подзадач s и число подзадач на каждой машине x_i, причем $s = \sum_i x_i$. После определения всех x_i задание распределяется между соответствующими i-ми узлами кластера.

РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТАЛЬНОЙ ПРОВЕРКИ МОДЕЛИ

Экспериментальная проверка предложенной модели распределения проводилась на пяти узлах кластера. Поскольку в данной модели из характеристик кластера учитывалась только степень загруженности узлов, задания были выбраны такими, которые не требуют интенсивного обмена данными между узлами кластера. Значение размерности $S_{\max}$ принято равным 22, то есть каждое задание могло состоять из 22 подзадач.

Для проверки эффективности системы было рассмотрено три случая:

– равномерная загруженность всех узлов кластера:

$$z_1 = z_2 = z_3 = z_4 = z_5 = z;$$

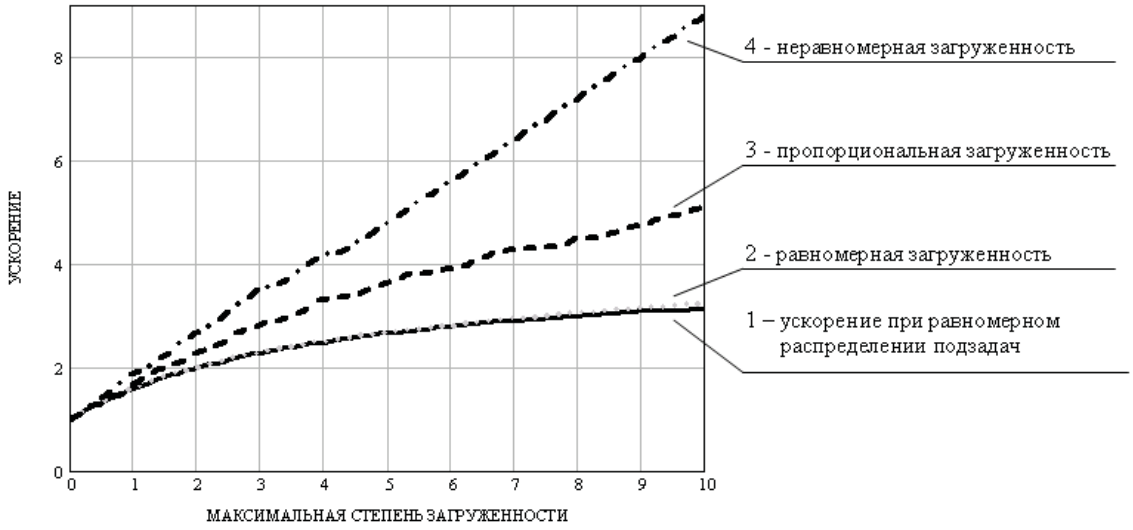

Рисунок 1 – Результаты исследований

– пропорциональная загруженность всех узлов кластера:

$$z1 = 0,\ z2 = z/4,\ z3 = 2z/4,\ z4 = 3z/4,\ z5 = z;$$

– неравномерная загруженность всех узлов кластера:

$$z1 = z2 = z3 = z4 = 0,\ z5 = z.$$

Значение z назовем максимальной степенью загруженности узлов кластера;

T1 – время выполнения параллельной программы при простом распределении по одной подзадаче на каждый из узлов кластера, то есть $x_1 = x_2 = x_3 = x_4 = x_5 = 1$;

T2 – время выполнения параллельной программы при равномерном распределении по четыре подзадачи на каждый из узлов кластера, то есть $x_1 = x_2 = x_3 = x_4 = x_5 = 4$;

T3, T4, T5 – время выполнения параллельной программы по разработанному алгоритму для равномерной, пропорциональной и неравномерной загруженности узлов кластера соответственно.

Для каждого случая загруженности были рассмотрены следующие показатели:

– ускорение работы параллельного задания при равномерном распределении подзадач между пятью узлами кластера. При этом на каждом узле выполнялось по 4 подзадачи, то есть число подзадач ≈*S*max, время выполнения T2. Ускорение равно T1/T2. Так как T1 и T2 зависят от узла с максимальной загруженностью, то для всех трех случаев загруженности значения ускорения будут одинаковы, график 1 рис. 1;

– ускорение работы параллельного задания при распределения подзадач по разработанному алгоритму (время выполнения T3, T4 и T5 соответственно для каждого случая). Ускорение равно T1/T3, T1/T4 и T1/T5, графики 2, 3 и 4 рис. 1.

Сравнения этих показателей и показывают эффективность разработанной системы.

На рис. 1 построена зависимость ускорения от максимальной степени загруженности для каждого случая.

Графики 1 и 2 совпали, так как при равномерной загруженности алгоритм равномерно распределяет подзадачи по всем узлам кластера. Наибольшее ускорение было получено при неравномерной загруженности кластера график 4.

ВЫВОДЫ

Разработанный алгоритм ориентирован на использование сведений о загруженности узлов, собранных непосредственно перед его работой. Он рассчитан на то, что нет значительной динамики в изменении загруженности за время его работы. Наибольший эффект достигается при неравномерной загруженности узлов кластера. В ходе дальнейших исследований планируется расширить данную модель, учитывая большее число факторов, разработать систему организации очередей заданий, предусмотреть гарантированное освобождение всех ресурсов, используемых задачей, от ее «обломков» при завершении, а также провести апробацию алгоритма при более сложных вариантах загруженности.

ПЕРЕЧЕНЬ ССЫЛОК

1. *Лацис А.* Как построить и использовать суперкомпьютер. – М.: Бестселлер, 2003. – 240 с.
2. *Букатов А. А., Дацюк В. Н., Жегуло А. И.* Программирование многопроцессорных вычислительных систем. – Ростов-на-Дону. Издательство ООО «ЦВВР», 2003. – 208 с.
3. *Таненбаум Э.* Современные операционные системы. 2-е изд. – СПб.: Питер, 2002. – 1040 с.

Надійшла 25.09.07

Розроблена математична модель оптимального розподілу завдань для паралельної обчислювальної системи з урахуванням завантаженості її обчислювальних вузлів. Представлені результати експериментальних досліджень моделі, які демонструють значний приріст продуктивності в порівнянні з рівномірним розподілом паралельного завдання по вузлах кластера.

The mathematical model of tasks optimum distribution for the parallel computing system is developed in view its computing nodes load. Results the model experimental researches which show a significant increase of productivity in comparison with uniform distribution of the parallel task on nodes of a cluster are submitted.

УДК 004.087

А. А. Егошина

ФОРМАЛЬНАЯ МОДЕЛЬ СЛОВООБРАЗОВАТЕЛЬНОЙ СЕМАНТИКИ

В статье рассмотрен вопрос формализации семантики естественного языка. Предложена формальная модель словообразовательной семантики, учитывающая специфику словообразовательного значения в сравнении с заданными грамматическими и лексическими значениями производного слова.

ВВЕДЕНИЕ

Семантический компонент уже достаточно давно признается необходимой частью полного описания языка. Семантика как раздел лингвистики отвечает на вопрос, каким образом человек, зная слова и грамматические правила какого-либо естественного языка, оказывается способным передать с их помощью самую разнообразную информацию о мире (в том числе и о собственном внутреннем мире), даже если он впервые сталкивается с такой задачей, и понимать, какую информацию о мире заключает в себе любое обращенное к нему высказывание, даже если он впервые слышит его.

Семантика как информация, передаваемая языком или какой-либо его единицей (словом, грамматической формой слова), представляет собой не жестко детерминированную систему. Ячейка семантики (полнозначное слово) организована по принципу «семантического треугольника» [1] и может быть схематично представлена следующим образом.

Схема, представленная на рис. 1, резюмирует семантические отношения, т. е. аналогичным образом семантика организована во всех единицах языка.

Свой вклад в формирование общих принципов семантического описания вносят разные теории языка. Семантика как наука начала развиваться еще во второй половине 19 века. Фундаментальные лингвистико-гносеологические концепции В. Фон Гумбольта, А. А. Потебни, В. Вудта и др. определили первый этап развития

Рисунок 1 – Семантический треугольник