

УДК 004.272

М.Г. Курносов, А.А. Пазников

ИНСТРУМЕНТАРИЙ ДЕЦЕНТРАЛИЗОВАННОГО ОБСЛУЖИВАНИЯ ПОТОКОВ ПАРАЛЛЕЛЬНЫХ МРІ-ЗАДАЧ В ПРОСТРАНСТВЕННО-РАСПРЕДЕЛЕННЫХ МУЛЬТИКЛАСТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ¹

Рассматриваются децентрализованные алгоритмы и программы обслуживания потоков параллельных задач в пространственно-распределенных вычислительных системах (ВС). Описывается функциональная структура разработанного программного пакета GBroker децентрализованной диспетчеризации параллельных МРІ-программ в пространственно-распределенных мультикластерных ВС. Приводятся результаты моделирования алгоритмов и оценка их эффективности при реализации на мультикластерных ВС.

Ключевые слова: *диспетчеризация параллельных программ, пространственно-распределенные вычислительные системы, GRID-системы.*

При решении сложных задач науки и техники широкое применение получили пространственно-распределенные вычислительные системы (ВС) – макроколлективы рассредоточенных вычислительных средств (подсистем), взаимодействующих между собой через локальные и глобальные сети связи (включая сеть Internet) [1, 2]. К таким системам относятся GRID-системы и мультикластерные ВС.

К значимым проблемам организации функционирования пространственно-распределенных ВС относится диспетчеризация параллельных программ. Для каждой программы, из поступивших в ВС, требуется определить ресурсы (подсистемы) для ее выполнения. В средствах диспетчеризации необходимо учитывать возможность изменения состава и загрузки распределенной ВС с течением времени.

Централизованные средства диспетчеризации программ имеют существенный недостаток: отказ управляющего узла ВС может привести к неработоспособности всей системы. Кроме того, в случае применения таких средств в больших масштабах ВС возрастают временные затраты на поиск требуемых ресурсов. Поэтому актуальной является задача разработки децентрализованных моделей, алгоритмов и программного обеспечения для диспетчеризации параллельных задач в распределенных ВС.

При децентрализованной диспетчеризации в вычислительной системе должен функционировать коллектив диспетчеров, осуществляющий выбор необходимых ресурсов для реализации программ. Это позволяет достичь и живучести больших масштабов ВС, то есть способности систем продолжать работу при отказах отдельных компонентов и подсистем.

Существует несколько пакетов централизованной диспетчеризации параллельных программ в пространственно-распределенных системах: GridWay, Common Scheduler Framework (CSF), Nimrod/G, Condor-G, GrADS, AppLeS,

¹ Работа выполнена в рамках интеграционного проекта № 113 СО РАН, при поддержке РФФИ (гранты № 09-07-13534, 09-07-90403, 08-07-00022, 10-07-05005, 08-07-00018), Совета по грантам Президента РФ для поддержки ведущих научных школ (грант НШ-5176.2010.9) и в рамках государственного контракта № 02.740.11.0006 с Минобрнауки РФ.

DIRAC, WMS и др. В пакете GridWay [3, 4] преследуется цель минимизации времени обслуживания задач, при этом учитывается время решения задачи и её пребывания в очереди; реализован механизм миграции задач между подсистемами. Среда CSF [5] предоставляет средства резервирования ресурсов на основе алгоритма Round-Robin и алгоритмов, созданных пользователем. В основе Nimrod/G [6] лежит экономическая модель, целью которой является поиск равновесного состояния между поставщиками и потребителями ресурсов посредством механизма аукциона. Диспетчер Condor-G [7], являющийся развитием системы Condor, ориентирован на использование в системах высокой пропускной способности (High-throughput Computing) и основан на алгоритмах Matchmaking и ClassAd. Предполагается, что пользователь самостоятельно выбирает подсистему из списка доступных; при этом поддерживается возможность создания пользовательских диспетчеров. Функционирование диспетчера WMS [8], используемого в GRID-проекте Enable Grid for e-Science, основывается на применении двух алгоритмов диспетчеризации: либо задача отправляется на подсистему для решения как можно быстрее, либо ожидает в очереди до тех пор, пока ресурс не станет доступным. В DIRAC [9] на подсистемах установлены агенты, которые при освобождении ресурсов запрашивают задачи из глобальной очереди.

В данной работе предлагаются децентрализованные алгоритмы и программное обеспечение диспетчеризации параллельных программ в пространственно-распределенных вычислительных и GRID-системах.

1. Постановка задачи

Пусть имеется пространственно-распределенная ВС, состоящая из H подсистем; N – суммарное количество элементарных машин (ЭМ) в подсистемах. Обозначим через n_i количество ЭМ, входящих в состав подсистемы $i \in S = \{1, 2, \dots, H\}$. Пусть также $b_{ij} = b(i, j, m)$ – пропускная способность канала связи между подсистемами $i, j \in S$ при передаче сообщений размером m байт ($[b(i, j, m)]$ = байт/с).

В каждой подсистеме присутствует диспетчер, который поддерживает очередь параллельных задач и осуществляет поиск вычислительных ресурсов для их выполнения (рис. 1).

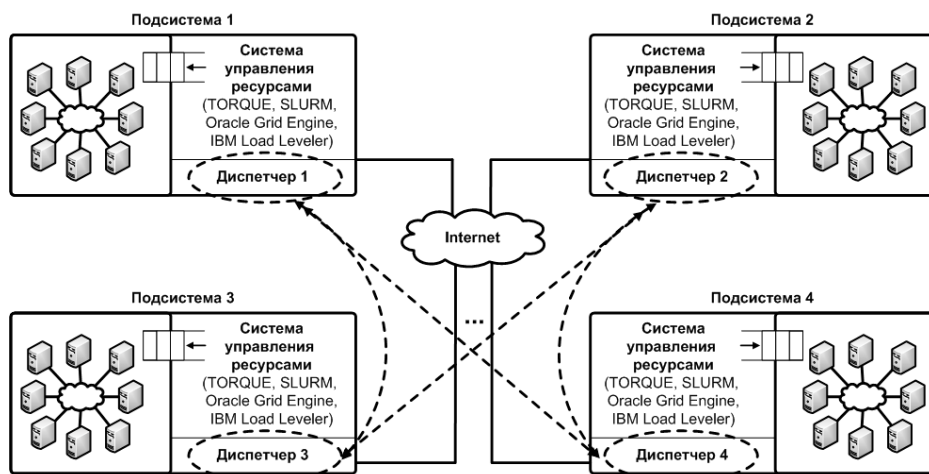


Рис. 1. Пример локальных окрестностей диспетчеров:
 $H = 4$, $L(1) = \{3, 4\}$, $L(2) = \{3, 4\}$, $L(3) = \{1, 2\}$, $L(4) = \{1, 2\}$

Коллектив диспетчеров представлен в виде ориентированного графа $G = (S, E)$, в котором вершинам соответствуют диспетчеры, а ребрам – логические связи между ними. Наличие дуги $(i, j) \in E$ в графе означает, что диспетчер i может отправлять ресурсные запросы (задачи из своей очереди) диспетчеру j . Множество всех вершин j , смежных вершине i , образуют её локальную окрестность $L(i) = \{j \in S: (i, j) \in E\}$.

Пользователь направляет задачу и запрос на выделение ресурсов одному из диспетчеров. Диспетчер (в соответствии с реализованным в нем алгоритмом) осуществляет поиск (суб)оптимальной подсистемы $j^* \in S$ для выполнения задачи пользователя.

Считаем, что задача характеризуется рангом r – количеством параллельных ветвей, ожидаемым временем t выполнения программы (walltime) и суммарным размером z исполняемых файлов и данных ($[z]$ = байт). Рассмотрим децентрализованный алгоритм обслуживания потоков параллельных задач в пространственно-распределенных ВС.

2. Децентрализованный алгоритм диспетчеризации параллельных задач

При поступлении программы пользователя в очередь подсистемы $i \in S$ её диспетчер выполняет следующий алгоритм:

1. У каждого диспетчера $j \in L(i) \cup \{i\}$ запрашивается оценка времени t_j , через которое программа может быть запущена на ресурсах подсистемы j в случае передачи программы в её очередь.

2. Через систему мониторинга определяется текущее значение пропускной способности $b_{ij} = b(i, j, z)$ канала связи между подсистемами i и j .

3. Отыскивается оценка времени $t(i, j)$ обслуживания программы в случае её перенаправления в очередь подсистемы j . Время обслуживания включает в себя время доставки исполняемых файлов и данных до подсистемы j , время ожидания t_j в очереди подсистемы и время t выполнения программы:

$$t(i, j) = z / b_{ij} + t_j + t.$$

4. Программа перенаправляется в очередь подсистемы j^* , в которой достигается минимальное значение времени $t(i, j)$ обслуживания параллельной программы

$$j^* = \arg \min_{j \in L(i) \cup \{i\}} \{t(i, j)\}.$$

Диспетчер выполняет описанный алгоритм поиска подсистем при наступлении одного из нижеследующих событий.

1. Поступление новой задачи в очередь от пользователя.

2. Удаление задачи из очереди, начало выполнения задачи, изменение параметров (количества ветвей, времени решение) одной или нескольких задач в очереди. Перечисленные события влекут за собой поиск новых подсистем для задач, стоящих в очереди после задач с модифицированными параметрами и состояниями.

3. Окончание временного интервала Δ с момента последнего запуска алгоритма поиска подсистем.

Периодический поиск подсистем для задач в очереди позволяет учесть динамически изменяющуюся загрузку ресурсов пространственно-распределенных вычислительных и GRID-систем. Описанный подход реализован в программном пакете GBroker децентрализованной диспетчеризации параллельных MPI-программ в пространственно-распределенных мультикластерных ВС.

3. Программный пакет GBroker

В Центре параллельных вычислительных технологий ГОУ ВПО «Сибирский государственный университет телекоммуникаций и информатики» (ЦПВТ ГОУ ВПО «СибГУТИ») и Лаборатории вычислительных систем Института физики полупроводников им. А.В. Ржанова СО РАН (ИФП СО РАН) создан и развивается программный пакет GBroker [10] децентрализованной диспетчеризации параллельных MPI-программ в пространственно-распределенных ВС. Пакет разработан на языке ANSI C для операционной системы GNU/Linux.

В пакет (рис. 2) входят диспетчер gbroker, клиентское приложение gclient и средство мониторинга производительности каналов связи netmon между подсистемами на уровне стека протоколов TCP/IP.

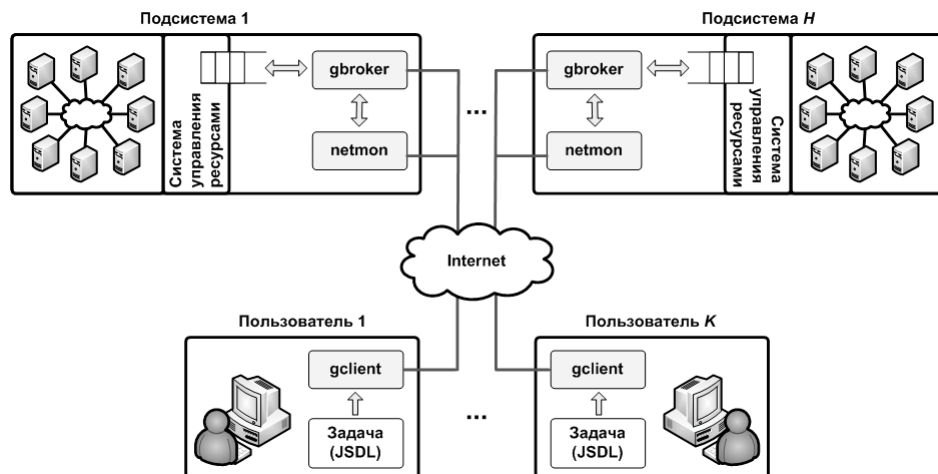


Рис. 2. Функциональная структура пакета GBroker

Модуль gbroker устанавливается на каждой из подсистем и на основе расширяемой архитектуры обеспечивает интерфейс с локальной системой пакетной обработки заданий (на данный момент – TORQUE). Модуль netmon устанавливается вместе с gbroker на подсистемах. Взаимодействуя друг с другом, сервисы netmon собирают информацию о производительности каналов связи между подсистемами. Модуль gclient реализует интерфейс между пользователем и системой.

Администратор настраивает локальные окрестности диспетчеров gbroker, указывая, какие диспетчеры с какими могут обмениваться задачами из своих очередей, и настраивает сервис netmon.

Пользователь формирует задание, состоящее из параллельной MPI-программы и паспорта на языке ресурсных запросов JSDL (Job Submission Description Language) и отправляет его средствами gclient любому из диспетчеров gbroker. Диспетчер в соответствии с описанным алгоритмом выбирает подсистему, на которой должна выполняться задача.

4. Моделирование алгоритмов и программных средств

Созданный инструментарий децентрализованной диспетчеризации параллельных задач исследован на действующей мультикластерной ВС, созданной ЦПВТ

ГОУ ВПО “СибГУТИ” совместно с Лабораторией ВС ИФП СО РАН. В экспериментах было использовано 3 сегмента мультикластерной ВС (рис. 3):

- кластер Xeon16: 4 узла (2 x Intel Xeon 5150, 16 процессорных ядер);
- кластер Xeon32: 4 узла (2 x Intel Xeon 5345, 32 процессорных ядра);
- кластер Xeon80: 10 узлов (2 x Intel Xeon 5420, 80 процессорных ядер).

Сеть связи между сегментами ВС Fast Ethernet.

На сегментах системы был установлен пакет GBroker и настроен компонент netmon (рис. 3). В локальную окрестность каждого диспетчера были включены диспетчеры всех кластеров.

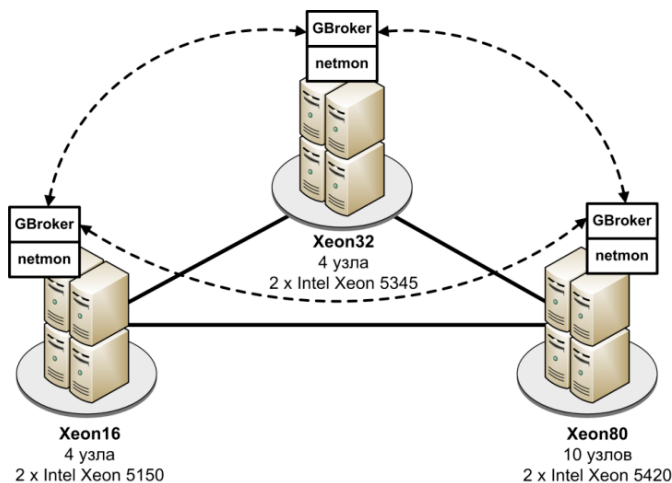


Рис. 3. Тестовая конфигурация пространственно-распределенной мультикластерной ВС ($H = 3$)

В качестве тестовых задач использовались MPI-программы из пакета NAS Parallel Benchmarks (NPB). Моделирование проводилось следующим образом. На выбранную подсистему поступал стационарный поток из $M = 500$ параллельных задач. Тестовые задачи выбирались псевдослучайным образом. Ранг r_i параллельной программы генерировался в соответствии с распределением Пуассона со значениями математического ожидания $r = \{4, 8, 12, 16\}$. Задачи поступали в очередь диспетчера кластера Xeon80 через интервалы времени t , экспоненциально распределенные со значениями интенсивности $\lambda = \{0.1, 0.2, 0.3, 0.4\}$, $[\lambda] = \text{с}^{-1}$.

Обозначим через t_i – момент времени поступления задачи $i \in \{1, 2, \dots, M\}$ на вход диспетчера, t'_i – момент времени начала решения задачи i , t''_i – момент завершения решения задачи i . Тогда суммарное время τ обслуживания (ожидания в очереди и решения) M задач составит

$$\tau = \max_{i=1,2,\dots,M} t''_i - \min_{i=1,2,\dots,M} t_i.$$

Для оценки эффективности алгоритма диспетчеризации использовались следующие показатели:

- среднее время T обслуживания задачи

$$T = 1/M \sum_{i=1}^M (t''_i - t_i);$$

- среднее время W пребывания задачи в очереди

$$W = 1/M \sum_{i=1}^M (t'_i - t_i);$$

- пропускная способность B системы

$$B = M / \tau.$$

На рис. 4 показана зависимость среднего времени обслуживания параллельных задач от интенсивности их поступления в систему.

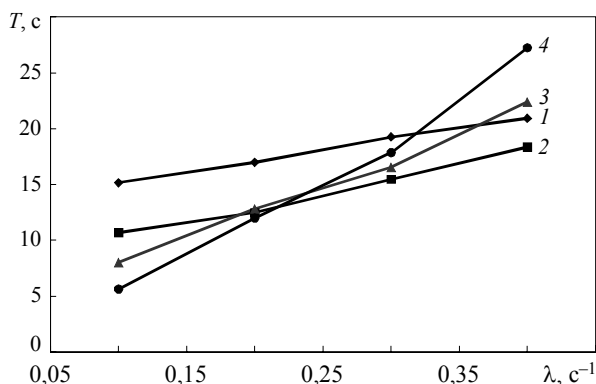


Рис. 4. Зависимости среднего времени T обслуживания задачи от интенсивности λ потока задач: 1 – $r = 4$; 2 – $r = 8$; 3 – $r = 12$; 4 – $r = 16$

Из графиков видно, что среднее время обслуживания задачи увеличивается с ростом интенсивности потока поступления задач. Это связано с образованием очередей задач на подсистемах ВС. Такое влияние интенсивности потока особенно значительно при рангах параллельных программ, близких к количеству процессорных ядер в подсистемах.

Среднее время W пребывания задачи в очереди незначительно (рис. 5) по сравнению с суммарным временем её обслуживания и в среднем не превосходит одной секунды (для рассматриваемой конфигурации ВС). Среднее время пребывания задачи в очереди не изменяется существенно как с ростом интенсивности потока поступления задач, так и с увеличением среднего ранга параллельных задач.

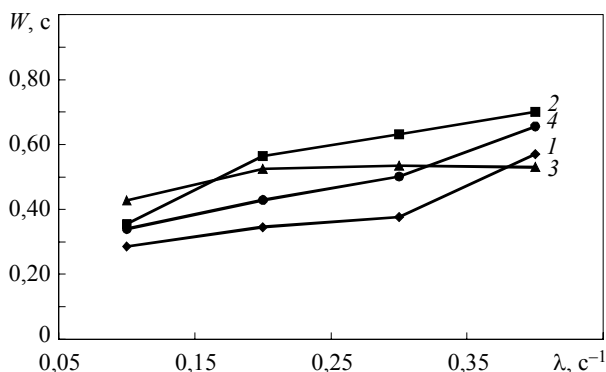


Рис. 5. Зависимости среднего времени W пребывания задачи в очереди от интенсивности λ потока задач: 1 – $r = 4$; 2 – $r = 8$; 3 – $r = 12$; 4 – $r = 16$

Пропускная способность системы (рис. 6) растет при увеличении интенсивности потока поступления задач. При больших значениях интенсивности происходит снижение пропускной способности системы, что связано с образованием очередей на подсистемах. С увеличением среднего ранга задач это влияние интенсивности на пропускную способность системы усиливается.

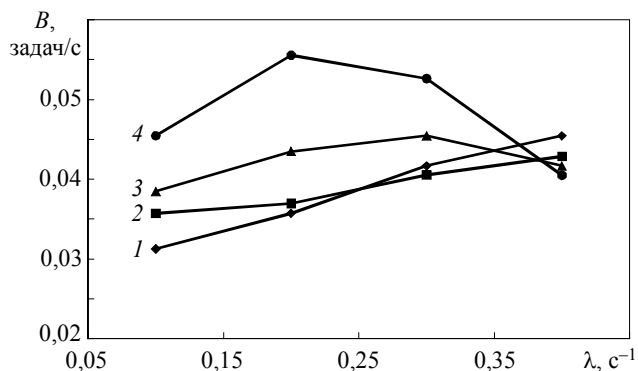


Рис. 6. Зависимости пропускной способности B системы от интенсивности λ потока задач: 1 – $r = 4$; 2 – $r = 8$; 3 – $r = 12$; 4 – $r = 16$

Заключение

Централизованные системы диспетчеризации большемасштабных распределенных ВС характеризуются вычислительной сложностью поиска требуемых ресурсов. Децентрализованная диспетчеризация параллельных программ в пространственно-распределенных вычислительных и GRID-системах позволяет повысить их живучесть. Кроме того, применение централизованных систем диспетчеризации в большемасштабных ВС ограничено вычислительной сложностью поиска требуемых ресурсов.

Результаты исследования созданного инструментария децентрализованной диспетчеризации параллельных MPI-программ на мультикластерной ВС показали, что среднее время обслуживания задач и при децентрализованной, и при централизованной диспетчеризации сопоставимы. Время диспетчеризации достаточно мало по сравнению со временем выполнения задач.

Инструментарий децентрализованной диспетчеризации – один из необходимых компонентов обеспечения живучести пространственно-распределенной мультикластерной ВС ЦПВТ ГОУ ВПО «СибГУТИ» и лаборатории ВС ИФП СО РАН.

ЛИТЕРАТУРА

1. Евреинов Э.В., Хорошевский В.Г. Однородные вычислительные систем. Новосибирск: Наука, 1978. 320 с.
2. Хорошевский В.Г. Архитектура вычислительных систем. М.: МГТУ им. Н.Э. Баумана. 2008. 520 с.
3. Montero R., Huedo E., Llorente I. Grid resource selection for opportunistic job migration // 9th International Euro-Par Conference. 2003. V. 2790. P. 366–373.
4. Huedo E., Montero R., Llorente I. A framework for adaptive execution on grids // Software – Practice and Experience (SPE). 2004. V. 34. P. 631–651.

5. Xiaohui W., Zhaohui D., Shutao Y. CSF4: A WSRF compliant meta-scheduler // Proc. of World Congress in Computer Science Computer Engineering, and Applied Computing. 2006. P. 61–67.
6. Buyya R., Abramson D., Giddy J. Nimrod/G: an architecture for a resource management and scheduling system in a global computational Grid // Proc. of the 4th International Conference on High Performance Computing in Asia-Pacific Region. 2000. P. 283–289.
7. Frey J. et. al. Condor-G: A computation management agent for multi-institutional grids // Cluster Computing. 2001. V. 5. P. 237–246.
8. Andreetto P., Borgia S., Dorigo A. Practical approaches to grid workload and resource management in the EGEE project // CHEP'04: Proc. of the Conference on Computing in High Energy and Nuclear Physics. 2004. V. 2. P. 899–902.
9. Caron E., Garonne V., Tsaregorodtsev A. Evaluation of meta-scheduler architectures and task assignment policies for high throughput computing // Technical report. Institut National de Recherche en Informatique et en Automatique. 2005.
10. Курносов М.Г., Пазников А.А. Децентрализованное обслуживание потоков параллельных задач в пространственно-распределенных вычислительных системах // Вестник СибГУТИ. 2010. № 2 (10). С. 79–86.

Курносов Михаил Георгиевич

Пазников Алексей Александрович

ГОУ ВПО «Сибирский государственный университет

телекоммуникаций и информатики»

E-mail: mkurnosov@gmail.com; apaznikov@gmail.com

Поступила в редакцию 3 декабря 2010 г.