Алгоритмы децентрализованной диспетчеризации параллельных программ в пространственно-распределенных вычислительных системах*

М.Г. Курносов, А.А. Пазников
Институт физики полупроводников СО РАН, Новосибирск

Рассматриваются алгоритмы децентрализованной диспетчеризации в пространственно-распределенных вычислительных системах параллельных программ с целью минимизации времени их обслуживания. Приводятся результаты натурных экспериментов на действующей мультикластерной вычислительной системе.

Введение

Сегодня для решения сложных задач науки и техники широкое распространение получили пространственно-распределенные вычислительные системы (ВС). Они представляют собой макроколлективы пространственно рассредоточенных ВС (вычислительных подсистем), взаимодействующих между собой через локальные и глобальные сети связи [1]. К таким системам относятся GRID-системы и мультикластерные ВС.

При организации функционирования распределенных ВС возникает задача диспетчеризации параллельных программ, поступающих в систему. Для каждой программы необходимо определить, на каких вычислительных ресурсах каких подсистем она будет выполняться. В средствах диспетчеризации должны учитываться динамичность состава систем и переменная загрузка их ресурсов.

Существующие программные средства диспетчеризации параллельных программ (GridWay, Common Scheduler Framework, Nimrod/G, Condor-G, GrADS, AppLeS) являются централизованными [2]. В таких системах отказ управляющего узла может привести к неработоспособности всей ВС. Кроме того, применение централизованных систем диспетчеризации в большемасштабных ВС ограничено вычислительной сложностью поиска требуемых ресурсов.

При децентрализованной схеме диспетчеризации в пространственно-распределенной ВС функционирует коллектив диспетчеров, при-

* Работа выполнена при поддержке РФФИ (гранты № 08-07-00018, 09-07-00095, 09-07-13534) и Совета по грантам Президента РФ (грант № НШ-2121.2008.9).

нимающий решение о выделении ресурсов для программ пользователей. Это позволяет достичь живучести ВС, т.е. ее способности продолжать работу при отказах отдельных вычислительных подсистем.

Постановка задачи

Пусть распределенная ВС состоит из H вычислительных подсистем и включает N элементарных машин (ЭМ), n_i – количество ЭМ в составе подсистемы $i \in S = \{1, 2, \ldots, H\}$.

Известны значения $b_{ij} = b(i, j, m)$ – пропускной способности канала связи между подсистемами $i, j \in S$ при передаче сообщений размером m байт ($[b(i, j, m)]$ = байт/с).

На каждой подсистеме присутствует диспетчер, который поддерживает очередь параллельных программ и осуществляет поиск вычислительных ресурсов для их выполнения.

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

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

Считаем, что программа характеризуется рангом r – количеством параллельных ветвей, ожидаемым временем t выполнения программы и суммарным размером z исполняемых файлов и данных ($[z]$ = байт).

Децентрализованный алгоритм диспетчеризации

При поступлении программы в очередь диспетчера i он опрашивает диспетчеры $j \in L(i)$ из локальной окрестности и получает значения текущего количества задач в их очередях и оценку времени t_j, через которое программа может быть запущена на ресурсах подсистемы j в случае передачи задачи в её очередь. Через систему мониторинга определяются значения пропускных способностей b_{ij} каналов связи между подсистемами. Выбирается подсистема $i^* = \operatorname{argmin}\{T(j)\}$, $j \in L(i)$, $T(i) = z / b(i, j, z) + t_j + t$, обеспечивающая минимальное значение времени T обслуживания параллельной программы. Время обслуживания включает время доставки файлов и данных до выделенной подсисте-

мы, время ожидания в очереди подсистемы и время выполнения программы.

Затем программа пересылается в очередь подсистемы i^*. Важно отметить, что при передаче задачи в очередь подсистемы i^*, для нее аналогично периодически будет осуществляться поиск ресурсов. Это обеспечивает адаптацию под динамически изменяющуюся загрузку ресурсов пространственно-распределенных ВС.

Программный пакет GBroker

В Центре параллельных вычислительных технологий ГОУ ВПО «Сибирский государственный университет телекоммуникаций и информатики» (ЦПВТ ГОУ ВПО «СибГУТИ») ведется разработка программного пакета GBroker децентрализованной диспетчеризации программ в распределенных ВС.

В пакет (рис. 1) входят диспетчер gbroker, который устанавливается на каждой из подсистем и взаимодействует с локальным менеджером задач, клиентская часть gclient, реализующая интерфейс между пользователем и распределенной ВС, и модуль netmon мониторинга производительности каналов связи на уровне стека протоколов TCP/IP.

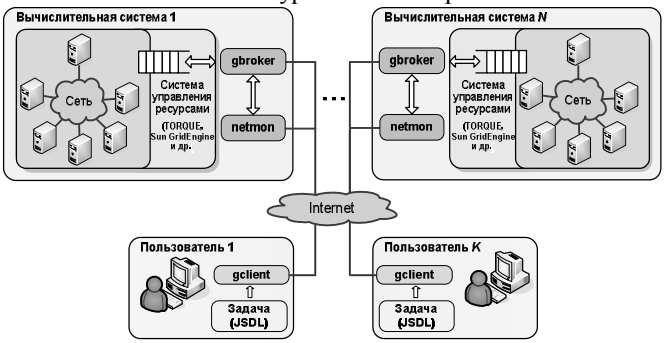


Рис. 1. Функциональная структура пакета GBroker

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

Пользователь формирует задание, состоящее из параллельной MPI-программы и паспорта на языке ресурсных запросов JSDL, и отправляет его средствами gclient любому из диспетчеров gbroker. Диспетчер в соответствии с описанным алгоритмом выбирает подсистему, на которой затем выполняется программа.

Результаты экспериментов

На мультикластерной системе ЦПВТ ГОУ ВПО «СибГУТИ» проведено исследование разработанного алгоритма диспетчеризации и программного средства. Использовались кластеры следующих конфигураций: Xeon16 (4 узла – 2 x Intel Xeon 5150), Xeon32 (4 узла – 2 x Intel Xeon E5345), Xeon80 (10 узлов – 2 x Intel Xeon E5420). Сеть связи между кластерами – Fast Ethernet.

В качестве тестовых задач использовались MPI-программы из пакета NAS Parallel Benchmarks.

На выбранную подсистему поступал стационарный поток из M параллельных задач. Ранг r_i параллельной программы распределен по закону Пуассона с параметрами $\lambda = \{0{,}25r_0, 0{,}5r_0, 0{,}75r_0, r_0\}$, где $r_0 = 80$ – максимальное количество процессорных ядер. Заявки на обслуживание поступали через интервалы времени t, экспоненциально распределенные с параметрами $\mu = \{0{,}1$ с, 0,2 с, 0,3 с, 0,4 с$\}$.

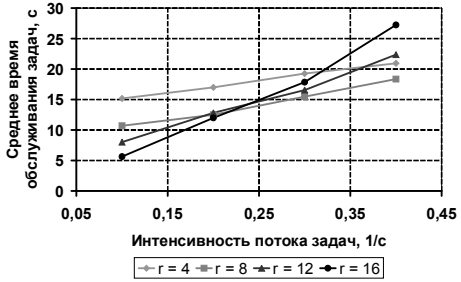


Рис. 2. Зависимости среднего времени обслуживания задач от интенсивности λ потока задач

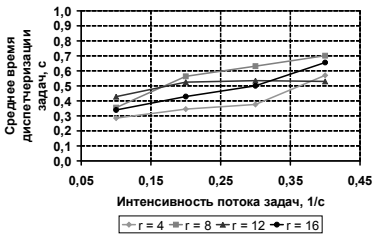


Рис. 3. Зависимости среднего времени диспетчеризации задач от интенсивности λ потока задач

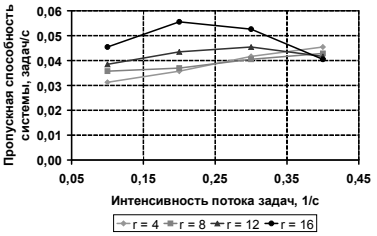


Рис. 4. Зависимости пропускной способности системы от интенсивности λ потока задач

Среднее время диспетчеризации (рис. 2, 3) задач незначительно по сравнению со временем их обслуживания. Пропускная способность системы (рис. 4) растет при увеличении интенсивности потока поступ-

ления задач. Снижение пропускной способности при больших значениях интенсивности обусловлено образованием очередей в системе пакетной обработки заданий.

Заключение

Децентрализованная диспетчеризация параллельных программ в распределенных ВС позволяет повысить живучесть последних. Среднее время обслуживания задач при децентрализованной диспетчеризации сопоставимо с централизованной. Время диспетчеризации достаточно мало по сравнению со временем выполнения программ.

В дальнейшем планируется выполнить интеграцию с программным пакетом GridWay. Планируется ввести учет производительности процессорных ядер на подсистемах и реализовать возможность выделения группы подсистем для решения параллельной программы.

Литература

1. **Хорошевский В. Г.** Архитектура вычислительных систем. М.: МГТУ им. Н. Э. Баумана, 2008.
2. **Gaweda I., Wilk C.** Grid Brokers and Metaschedulers. Market Overview // Technical Report. Kraków, 2006.