УДК 519.6

АДАПТИВНАЯ СИСТЕМА МАРШРУТИЗАЦИИ ДЛЯ ОТЕЧЕСТВЕННОЙ СИСТЕМЫ МЕЖПРОЦЕССОРНЫХ ОБМЕНОВ СМПО-10G

В. Г. Басалов, В. М. Вялухин
(РФЯЦ-ВНИИЭФ)

Представлен результат работ по созданию алгоритмов адаптивной системы маршрутизации для отечественной системы межпроцессорного обмена СМПО-10G. Приводится обоснование выбора локальной адаптивной маршрутизации в качестве оптимальной для СМПО-10G. Описываются разработанные механизм инициализации коммуникационной среды и алгоритм маршрутизации сообщений. Детализирован алгоритм выбора оптимального выходного порта коммутатора при передаче сообщения. Описывается механизм организации широковещательных обменов для СМПО-10G.

Ключевые слова: мультипроцессорная вычислительная система, система межпроцессорного обмена, СМПО-10G, топология коммуникационной среды, локальная адаптивная маршрутизация, механизм широковещания.

Введение

Система межпроцессорного обмена СМПО-10G является коммуникационной сетью и предназначена для построения мультипроцессорных систем. Посредством сети реализуется объединение вычислительных узлов в единую вычислительную систему (ВС) и обмен информацией между ними. Областью применения СМПО-10G являются ВС разного уровня производительности: от компактных суперЭВМ до больших систем, состоящих из тысяч вычислительных узлов.

Одной из ключевых частей, входящих в коммуникационное программное обеспечение СМПО-10G, является система маршрутизации, которая выполняет функции по инициализации коммуникационной сети, ее активации и вычислению оптимальных маршрутов передачи информации между вычислительными модулями (ВМ).

Требования, предъявляемые к системе маршрутизации современного уровня, очень жесткие. Она должна обрабатывать любые изменения топологии коммуникационной сети без внешнего вмешательства, гарантируя бесперебойное функционирование ВС. Дополнительная нагрузка, вносимая системой маршрутизации в коммуникационную сеть ВС при ее обнаружении, инициализации, активации и поддержании, должна быть минимизирована. Алгоритмы маршрутизации, реализованные в ней, должны обеспечивать передачу сообщений между узлами коммуникационной сети ВС по кратчайшему пути, избегая ситуаций *взаимных блокировок*[*] при обмене сообщениями.

Одним из сложных и сильно нагружающих сеть видов обменов являются коллективные операции. К ним, в частности, относится операция широковещательного обмена. Операции широковещания на уровне MPI-приложений [1] соответствует операция Broadcast. Эта операция является одним из элементов в реализации алгоритмов ряда других коллективных операций MPI (All_gether, All_reduce, barrier), активно используемых в приложениях, разрабатываемых в РФЯЦ-ВНИИЭФ.

Масштабирование *параллельного* вычислительного комплекса практически не приводит к

[*] *Взаимной блокировкой*, или *дедлоком* (англ. deadlock), называют ситуацию, при которой группа пакетов в сети не может продолжить движение из-за ожидания освобождения ресурсов, занятых самими этими пакетами. Если такая ситуация однажды возникла, то вся сеть или ее часть теряет способность передавать данные.

ухудшению времени выполнения двухточечных обменов, но существенно сказывается на времени выполнения коллективных обменов. Алгоритмы коллективных операций можно реализовывать либо с использованием последовательности двухточечных обменов, возлагая основную нагрузку на коммуникационное программное обеспечение вычислительных узлов, либо передать выполнение этих операций аппаратуре коммуникационной сети, что значительно повысит их эффективность. Поэтому система маршрутизации СМПО-10G должна обеспечивать аппаратную поддержку коллективных обменов без привлечения ресурсов ВМ.

Выполнение этих требований в разрабатываемой системе маршрутизации позволит снизить задержку при передаче сообщений и увеличить производительность ВС.

Выбор типа системы маршрутизации для СМПО-10G

Выбор типа системы маршрутизации и маршрутных алгоритмов напрямую зависит от архитектуры системы межпроцессорного обмена. Основными топологиями коммуникационных сетей ВС, создаваемых на базе архитектуры СМПО-10G, являются решетчатые структуры — двумерные и трехмерные решетки и торы. В связи с тем, что эти топологии имеют замкнутые циклические маршруты, важной задачей становится недопущение взаимных блокировок при обмене сообщениями. Алгоритмы, позволяющие гарантированно строить маршруты, свободные от взаимных блокировок, делятся на два класса: детерминированной маршрутизации и адаптивной маршрутизации [2].

Детерминированные маршрутные алгоритмы всегда используют постоянный путь между парой узлов коммуникационной сети и сильно зависят от ее топологии. Детерминированная маршрутизация может быть статической или динамической.

При статической маршрутизации запись в таблицу маршрутов вводится и изменяется вручную администратором каждый раз, когда происходит изменение топологии коммуникационной среды. К достоинствам статической маршрутизации можно отнести легкость отладки и конфигурирования в коммуникационных сетях небольшой размерности. Однако статическая маршрутизация не обладает устойчивостью к изменению топологии и поэтому не может быть использована в СМПО-10G. Статическая маршрутизация применялась в системах межпроцессорного обмена первого поколения в 2000—2005 гг. [3, 4].

При динамической маршрутизации запись в таблице маршрутов обновляется автоматически, что позволяет постоянно держать таблицу маршрутов в актуальном состоянии и вычислять оптимальные маршруты на основе текущей топологии сети.

Динамическая маршрутизация требует разработки *менеджера* коммуникационной сети, который отвечает за обнаружение, инициализацию, активацию и поддержание сети. При работе менеджер сети оказывает дополнительную нагрузку на коммуникационную сеть ВС, причем наибольшая нагрузка приходится на область узла, на котором запущен менеджер сети. При увеличении размера ВС повышается нестабильность коммуникационной сети, что может приводить к ситуациям, когда коммутаторы не успевают синхронизировать свои маршрутные таблицы. Централизованная динамическая маршрутизация реализована в менеджере сети OpenSM для коммуникационных сетей на основе архитектуры InfiniBand. Работы над этим программным пакетом продолжаются в течение 15 лет [5—7].

Одной из особенностей архитектуры СМПО-10G является размещение коммутаторов в ВМ. Отключение по какой-либо причине любого ВМ приводит к отключению его коммутатора СМПО-10G и, в свою очередь, к изменению топологии коммуникационной сети. Регулярные решетчатые топологии становятся иррегулярными с большим количеством циклов, что значительно осложняет построение свободных от взаимных блокировок маршрутов.

В связи с трудно преодолимыми недостатками детерминированной маршрутизации, как статической, так и динамической, принято решение положить в основу системы маршрутизации СМПО-10G локальную адаптивную маршрутизацию.

Адаптивная маршрутизация — это механизм, позволяющий изменять маршрут следования сообщения в сети, если часть сети вышла из строя, с выбором наиболее оптимального маршрута. Локальная адаптивная маршрутизация — распределенная маршрутизация с учетом локальной информации о состоянии сети. Она основана на использовании маршрутных таблиц, в которых для каждого узла получателя указано

несколько вариантов выходных трактов, упорядоченных по степени их предпочтительности.

К основным недостаткам адаптивной маршрутизации относятся:

- увеличение сложности коммутатора. В коммутатор требуется добавить блок выбора выходного канала: в зависимости от текущего состояния выходных портов коммутатор должен выбрать из группы возможных выходных портов оптимальный для передачи сообщения;
- не гарантированный порядок доставки пакетов от источника до приемника.

Наряду с перечисленными недостатками адаптивная маршрутизация имеет неоспоримые достоинства:

- возможность динамически изменять маршрут следования сообщения в коммуникационной сети, если часть сети по каким-либо причинам вышла из строя. Это особенно актуально для СМПО-10G, коммутаторы которой размещаются в вычислительных узлах, выключение любого из которых приводит к изменению топологии коммуникационной сети;
- нет необходимости постоянно отслеживать состояние коммуникационной сети и пересчитывать маршрутные таблицы. Вследствие этого отпадает необходимость в разработке менеджера коммуникационной сети. Отсутствие постоянно исследующего коммуникационную сеть менеджера снижает нагрузку на нее;
- более равномерная нагрузка на коммуникационную сеть по сравнению с детерминированной маршрутизацией за счет прокладки маршрута по наименее загруженным связям;
- возможность избежать состояния взаимной блокировки для иррегулярных коммуникационных сетей, содержащих циклы.

Механизм инициализации распределенной коммуникационной среды СМПО-10G

Процесс инициализации коммуникационной среды СМПО-10G начинается после того, как завершена ее физическая инициализация. Физическая инициализация означает, что порты смежных коммутаторов соединены линией связи и приведены в состояние готовности передачи служебных сообщений. Линия связи считается готовой к передаче служебных сообщений, когда порты на ее концах находятся в состоянии инициализации.

Механизм инициализации состоит из двух частей — первоначальной инициализации вычислительной системы и инициализации отдельных групп узлов в процессе работы.

Механизм первоначальной инициализации вычислительной системы начинает действовать после того, как на вычислительный комплекс будет подано электропитание и пройдет процесс его физической инициализации. Старт первоначальной инициализации осуществляется запуском специальной утилиты на любом ВМ мультипроцессорной системы. Эта утилита осуществляет запись в коммутатор СМПО-10G данного ВМ его координат (x, y, z), размерности (m, n, k) всей решетчатой топологии, признака топологии (решетка или тор) и идентификатора процесса инициализации. Получив эти данные, коммутатор заносит их в свою память и выставляет флаг, сообщающий о том, что его инициализация выполнена.

После того, как первый коммутатор был проинициализирован, он должен инициализировать смежные с ним коммутаторы. Для этого коммутатор отправляет через все свои инициализированные порты служебные сообщения. Сообщение содержит вычисленные координаты (x, y, z) соответствующие данному порту коммутатора, размерность (m, n, k) всей решетчатой топологии, признак топологии (решетка или тор) и идентификатор процесса инициализации.

Алгоритмы вычисления номеров смежных коммутаторов различаются для топологий *решетка* и *тор*. В основном эти различия касаются коммутаторов, имеющих среди своих координат равные 0, а также $m-1$, $n-1$, $k-1$.

В любом коммутаторе имеется информация о том, на изменение какой координаты (x, y или z) и какое именно изменение (увеличение ($+$) или уменьшение ($-$)) "направлен" каждый его порт. На основании этой информации и своих координат коммутатор вычисляет координаты смежных с ним коммутаторов коммуникационной сети.

Рис. 1 (см. также цветную вкладку) иллюстрирует процесс первоначальной инициализации ВС.

Получив служебное инициализационное сообщение, коммутатор коммуникационной сети про-

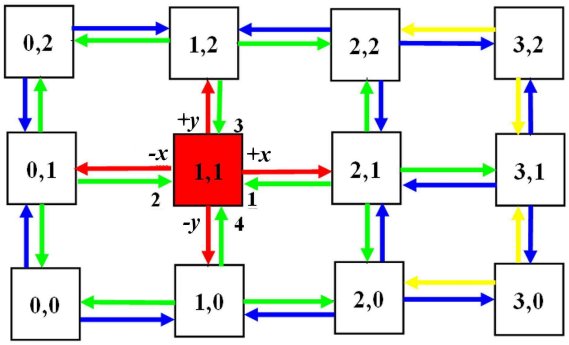

Рис. 1. Механизм первоначальной инициализации ВС

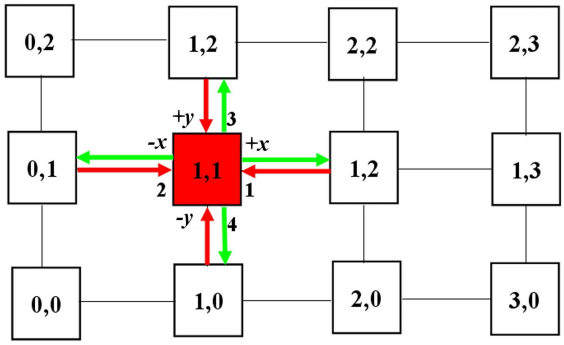

Рис. 2. Механизм инициализации повторно включаемых ВМ

веряет состояние своего флага инициализации. Если он "опущен", коммутатор сохраняет присланные данные, "поднимает" флаг инициализации и запускает процесс рассылки служебных инициализационных сообщений, содержащих координаты смежных ему узлов. Иначе, если флаг инициализации поднят, проверяется идентификатор процесса инициализации. В случае, если присланное значение не совпадает с хранящимся в коммутаторе, считается, что запущен новый процесс инициализации коммуникационной сети — присланные данные сохраняются и запускается процесс рассылки служебных инициализационных сообщений. Иначе, когда флаг инициализации поднят и присланный идентификатор процесса инициализации совпал с идентификатором, хранящимся в коммутаторе, сообщение отбрасывается как повторное и узел больше не распространяет служебных инициализационных сообщений.

Таким образом, начавшись в одном произвольном узле, процесс инициализации распространится через всю коммуникационную сеть и завершится, когда все коммутаторы будут проинициализированы. Контролировать инициализацию коммутаторов можно при помощи системы мониторинга СМПО-10G.

В процессе работы ВС возможны случаи выключения отдельных ВМ или даже их групп вследствие возникновения неисправностей либо с целью выполнения профилактических работ. Коммутаторы СМПО-10G, размещаемые внутри ВМ, тоже отключаются. Впоследствии при включении ВМ необходимо провести инициализацию коммутаторов включаемых ВМ. Этот механизм проиллюстрирован на рис. 2 (см. также цветную вкладку).

При включении коммутатора происходит физическая инициализация портов и переход порта на смежном коммутаторе из состояния *выключен* в состояние *инициализирован*. Обнаружив это изменение, смежный коммутатор вычисляет координату вновь включенного коммутатора, создает служебное инициализационное сообщение и посылает его в подключившийся порт. Затем он переводит порт в активное состояние. Приняв это сообщение, коммутатор выполняет алгоритм инициализации, описанный выше.

Таким образом, изменения в топологии коммуникационной сети, вызываемые включением ВМ, обрабатываются автоматически и без возрастания нагрузки на сеть.

Механизм адаптивной маршрутизации сообщений в СМПО-10G

В процессе создания механизмов системы маршрутизации СМПО-10G было рассмотрено несколько методов выбора оптимального выходного канала для передаваемого информационного сообщения [8]. Наиболее предпочтительным был признан метод, названный авторами методом *алгоритмически коммутируемой маршрутизации*. Его идея заключается в том, что вычисление оптимального выходного порта для передачи каждого транзитного информационного сообщения осуществляется коммутатором непосредственно в момент передачи.

Заголовок передаваемого пакета содержит адрес узла получателя, который представляет собой его координаты (Dx, Dy, Dz). Выделив эти координаты из заголовка пришедшего пакета, коммутатор сравнивает их со своими координатами (x, y, z), полученными при инициализации. Для этого из значений координат узла

получателя (Dx, Dy, Dz) вычитаются соответствующие им координаты текущего узла (x, y, z) и получается разность (Rx, Ry, Rz). Если координаты совпали, т. е. значения Rx, Ry и Rz равны 0, делается вывод, что пакет достиг узла назначения. Иначе, если существует хотя бы одно ненулевое значение в полученной разности, пакет считается транзитным и начинается выполнение алгоритма выбора оптимального выходного порта.

Основным достоинством метода алгоритмически коммутируемой маршрутизации является то, что нет необходимости заранее вычислять маршруты (при высокой вероятности изменения топологии вычислительной системы это становится нетривиальной задачей) и затрачивать на их хранение память коммутатора. Использование данного метода маршрутизации позволяет создавать ВС с неограниченным числом компонентов.

Недостатком этого метода является возможное увеличение задержки при пересылке сообщений по сравнению с табличным методом, поскольку коммутатор вычисляет для каждого сообщения оптимальный выходной порт непосредственно во время пересылки.

Алгоритм выбора оптимального выходного порта

Для выбора оптимального выходного порта проанализируем полученную разность (Rx, Ry, Rz). Анализ всегда начинается с Rx.

Если $Rx = 0$, переходим к анализу следующей координаты разности (Rx, Ry, Rz). Иначе, если $Rx > 0$, выходным портом считаем порт $+x$, если $Rx < 0$, то выходной порт $-x$.

Полученный выходной порт необходимо проверить на возможность передачи сообщения (порт должен быть в активном состоянии и очередь выходных пакетов должна иметь свободные места). Если порт подходит, он считается оптимальным, и анализ завершается. Иначе, если передача сообщения через выбранный порт невозможна (порт не в активном состоянии либо очередь выходных пакетов заполнена), а $Ry \vee Rz = 0$, в качестве выходного порта выбирается первый из портов $+y$, $-y$, $+z$, $-z$, способный к передаче сообщений. Он считается оптимальным, и анализ завершается. В ситуации, когда все порты коммутатора не способны к передаче сообщений через определенный промежуток времени, пакет уничтожается, освобождая место в приемном буфере. Выполнение этих условий предотвращает возникновение состояний взаимной блокировки в коммуникационной сети.

Если $Ry \vee Rz \neq 0$, переходим к анализу следующей координаты разности (Rx, Ry, Rz).

Повторяем анализ для каждой координаты Rx, Ry, Rz, пока не найдем свободного выходного порта.

Авторы разработали алгоритмы выбора оптимального выходного порта для топологий коммуникационной сети в виде двумерных и трехмерных решеток и торов. Эти алгоритмы реализованы в блоке управления коммутацией пакетов аппаратной части коммутатора СМПО-10G.

Адаптивный механизм широковещания СМПО-10G

При отсутствии менеджера коммуникационной сети, который отслеживает ее состояние, механизм широковещания СМПО-10G должен автоматически обеспечивать передачу широковещательных сообщений при высокой вероятности изменения топологии коммуникационной сети. В основу механизма широковещания СМПО-10G, названного *логическим деревом*, был положен принцип адаптивности. Идея этого механизма состоит в том, что широковещательный маршрут представляет собой дерево обменов, построенное на массиве узлов, занятых конкретной задачей. На рис. 3 (см. также цветную вкладку) приведена схема работы механизма широковещания *логическое дерево* второй степени.

ВМ — источник широковещательного сообщения — формирует пакет, в заголовке которого в качестве адреса получателя указываются координаты корневого узла логического дерева, а в качестве источника — координаты текущего уз-

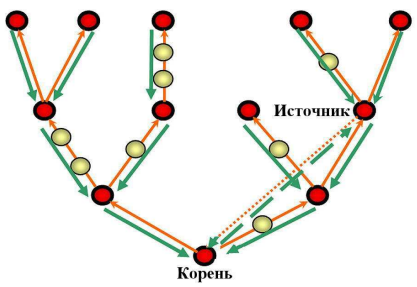

Рис. 3. Схема работы механизма широковещания *логическое дерево* второй степени

ла. После формирования пакета он передается в аппаратный модуль СМПО-10G источника.

Определив, что из ВМ получен широковещательный пакет, аппаратный модуль СМПО-10G сравнивает координаты источника и приемника сообщения. Если они не равны, значит, сообщение еще не достигло узла корня логического дерева — пакет передается получателю с помощью адаптивного механизма *точка-точка*, как указывает на рис. 3 точечно-пунктирная стрелка (красного цвета). Если же координаты совпадают, аппаратный модуль СМПО-10G сразу начинает процесс рассылки широковещательных пакетов с использованием широковещательной маршрутной информации.

Получив широковещательный пакет из канала связи, корневой аппаратный модуль СМПО-10G сбрасывает пакет в ВМ и транслирует этот пакет дальше согласно имеющейся у него для данной задачи маршрутной информации. Формат маршрутной информации приведен на рис. 4, где TaskID — идентификатор задачи; DST1 — координаты первого узла получателя пакета — первые три байта, а четвертый байт отведен под CF (complete flag) — флаг завершения цепочки; DST2 — то же самое для второго узла получателя.

Аппаратный модуль СМПО-10G рассылает широковещательный пакет по адресам, указанным в маршрутной информации учитывая состояния полей CF в DST1 и DST2. Если значение CF в DST1 равно 1, то происходит замена координат получателя в заголовке пакета и его отправка. На рис. 3 эти операции показаны стрелками (на цветном рисунке — стрелки красного цвета), ведущими к *листьям* логического дерева. Иначе, если значение CF в DST1 равно 0, координаты получателя не рассматриваются и цепочка считается законченной. Далее происходит такой же анализ DST2. Ситуация, когда оба поля CF в DST1 и DST2 равны 0, означает, что достигнута конечная вершина логического дерева. Определив такую ситуацию, аппаратный модуль СМПО-10G формирует и отсылает подтверждающее сообщение об удачном завершении широковещательной операции согласно маршрутной информации обратного хода. Формат маршрутной информации обратного хода представлен на рис. 5. На рис. 3 указанные операции обозначены стрелками, ведущими к корню логического дерева и пунктирной стрелкой — из узла корня в узел источника (на цветном рисунке — стрелки зеленого цвета). После того, как ВМ — источник широковещательного сообщения получит пакет-подтверждение, широковещательная операция будет считаться завершенной.

Время выполнения операции широковещания с подтверждением для механизма, представленного на рис. 3, при плотном распределении задачи по ВМ можно оценить по формуле

$$T = (d + 2\log_2 N)t,$$

где d — *диаметр* области ВМ, занятой задачей, т. е. число транзитных участков между наиболее удаленными ВМ; N — количество ВМ, занятых задачей; t — время выполнения одной пересылки.

В процессе запуска MPI-приложения загрузчик задач распределяет процессы запускаемого приложения по ВМ вычислительного комплекса. После этого запускается механизм инициализации MPI-процессов. В ходе инициализации каждый MPI-процесс формирует локальную коммуникационную таблицу. Для этого на стадии инициализации (функция MPI_Init) производится вызов функции GetIDComm низкоуровневой библиотеки доступа. Получив все локальные коммуникационные таблицы, процесс *мастер* формирует глобальную коммуникационную таблицу. При этом каждая глобальная коммуникационная таблица MPI-процесса содержит необходимую информацию обо всех процессах, участвующих в межпроцессорных обменах в рамках одной задачи. Таблицы атрибутов располагаются в соответствии с рангами (rank) процессов, образуя хэш-таблицу с прямой адресацией.

Для организации механизма широковещания процесс *мастер* запускает функцию IPCS_10G_Multicast, которая вычисляет маршрутную информацию для операций ши-

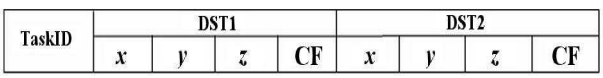

Рис. 4. Формат прямой маршрутной информации

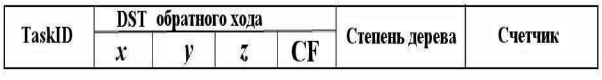

Рис. 5. Формат маршрутной информации обратного хода

роковещания и размещает ее в глобальной коммуникационной таблице.

После того, как глобальная информационная таблица создана, она рассылается на все узлы, занятые задачей. Информация, относящаяся к механизму маршрутизации, на каждом узле своя, она заносится в память аппаратного модуля СМПО-10G с помощью вызова функции PutMulticast низкоуровневой библиотеки доступа.

Освобождение ресурсов аппаратного модуля СМПО-10G происходит в функции MPI_Finalize посредством вызова функции EndMulticast низкоуровневой библиотеки доступа.

К достоинствам предложенного механизма широковещания можно отнести следующее:

- заголовок широковещательного пакета имеет постоянный размер и не зависит от количества узлов, на которых размещена задача;
- маршрутная информация пересылается только на те узлы, на которых размещается задача в процессе ее инициализации;
- пересылка пакета подчиняется принципам адаптивности и становится не чувствительной к изменениям топологии вычислительного комплекса;
- предложенная организация маршрутизации широковещания может быть с пользой применена для выполнения обратной глобальной вычислительной операции (sum, min, max и др.) над данными, расположенными в различных ВМ, с выдачей результата одному ВМ и рассылкой всем остальным ВМ.

Авторы благодарят Ю. Г. Бартенева за ряд полезных замечаний и уточнений алгоритмов, описанных в данном разделе.

Заключение

В работе получены следующие результаты:

- обоснован тип системы маршрутизации для отечественной системы межпроцессорных обменов СМПО-10G, отвечающий всем требованиям, предъявляемым к современным системам маршрутизации;
- разработаны механизмы и алгоритмы, обеспечивающие инициализацию коммуникационной сети и адаптацию при отказах;
- разработаны и реализованы в аппаратуре СМПО-10G маршрутные алгоритмы для коммуникационных сред с топологиями двумерных и трехмерных решеток и торов;
- разработан адаптивный механизм широковещания СМПО-10G.

Направление дальнейших работ связано с адаптацией механизмов и алгоритмов системы маршрутизации СМПО-10G к новым топологиям коммуникационной среды, применение которых позволит увеличить производительность системы межпроцессорных обменов.

Список литературы

1. MPI: A Message Passing Interface standard // Message Passing Interface Forum. March 22, 1994. http://www.mpi-forum.org.
2. *Flich J. J., Robles A., L'opez P., Duato J.* Supporting fully adaptive routing in InfiniBand networks // IEEE Computer Society Press. April 2003. P. 165—172.
3. *Попов В. С., Степаненко С. А., Холостов А. А.* Архитектура аппаратных средств системы межпроцессорного обмена мультипроцессорной системы МП-Х // Вопросы атомной науки и техники. Сер. Математическое моделирование физических процессов. 2002. Вып. 4. С. 61—64.
4. *Байков Э. Г., Басалов В. Г., Варгин А. М. и др.* Система межпроцессорных обменов для мультипроцессорных сред // Межд. семинар «Супервычисления и мат. моделирование». Саров, 6—10 октября 2003 г.
5. InfiniBandTM Architecture Release 1.0.a. Volume 1 — General Specifications. Jule 19, 2001.
6. *Khapyorsky S.* OpenSM and InfiniBand Management Update. http://www.openfabric.org/archives/spring2008sonoma/wednesdey/opensm.pdf.
7. *Bermudez A., Casado R., Quiles F. J. et al.* Evaluation of a subnet management mechanism for InfiniBand networks // Proc. of Int. Conf. on Parallel Proc. USA, Kaohsiung. October 9, 2003. P. 117—124.
8. *Dally W., Towles B.* Principles and Practices of Interconnection Networks. San Francisco: Morgan Kaufmann Publishers, 2004.

Статья поступила в редакцию 10.01.12.