УДК 004.725.5

Реализация аппаратной поддержки коллективных операций в маршрутизаторе высокоскоростной коммуникационной сети с топологией «многомерный тор»

The implementation of hardware support for collective operations in high speed interconnect with multi-dimensional torus topology.

Сыромятников Е. Л., Макагон Д. В., Парута С. И., Румянцев А. А.
Syromyatnikov E. L., Makagon D. V., Paruta S. I., Rumyantsev A.A.

Аннотация

Коллективные операции используются в широком спектре задач для обмена данными между узлами системы, аппаратная их поддержка способствует повышению эффективности выполнения и масштабируемости параллельных программ. В статье описываются детали аппаратной реализации операций broadcast и reduce в сети с топологией многомерный тор, разработанной в ОАО «НИЦЭВТ».

Abstract

Collective operations are used in a wide range of applications to exchange data between the nodes of the system. Their hardware implementation allows for higher efficiency and scalability. This paper describes the hardware implementation of broadcast and reduce operations for the multi-dimensional interconnect developed in JSC "NICEVT".

Коллективные операции используются в широком спектре задач для обмена данными между узлами системы [1]. Примерами коллективных операций являются broadcast (рассылка данных от одного узла множеству узлов), reduce (сбор данных со множества узлов с применением коммутативной ассоциативной бинарной операции к ним, результат отсылается заданному узлу), scatter (распределение массива с одного узла по множеству узлов), gather (сбор массива со множества узлов), all reduce (аналогично reduce, но результат рассылается всем участникам операции), all gather (сбор массива со множества узлов на всех узлах множества), alltoall (распределение массива с каждого узла по остальным узлам). Коллективные операции относятся к основным примитивам взаимодействия вычислительных элементов в большинстве стандартов параллельного программирования, ориентированных на выполнение на системах с распределённой памятью (MPI [2], Shmem [3], PGAS-языки — UPC [4], X10 [5]); они могут составлять значительную часть коммуникационных обменов в процессе работы [1]. Реализация коллективных операций посредством операций «точка — точка» имеет ряд недостатков, таких как большая доля дублирующего трафика, плохая масштабируемость [6], поэтому аппаратная их поддержка способствует повышению эффективности выполнения и масштабируемости параллельных программ (см., например, [7]).

В ОАО «НИЦЭВТ» разрабатывается высокоскоростная коммуникационная сеть с топологией многомерный тор [8][9]; в рамках данной сети была реализована аппаратная поддержка двух коллективных операций — broadcast и reduce. Для этого была добавлена виртуальная подсеть, состоящая из двух виртуальных каналов с отдельными буферами и специальными правилами маршрутизации.

Виртуальная подсеть имеет топологию дерева, наложенного на многомерный тор (иллюстрация 1). В дереве задаётся корень, относительно которого вводятся два возможных направления движения по дереву: от корня и к корню. Каждому из направлений соответствует свой виртуальный канал. Узлы, из которых

движение от корня больше невозможно, называются листьями. Дерево строится с учётом порядка измерений: X, Y, Z, W (это позволяет предотвратить возможные дедлоки ввиду отсутствия циклов). Для построения дерева могут использоваться вспомогательные транзитные узлы — они логически не принадлежат дереву, но нужны для его связности (в данных узлах процессоры не посылают и не получают данных).

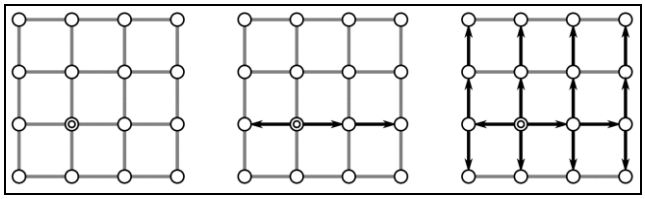

Рис. 1. Построение дерева.

При выполнении операции broadcast каждый узел при получении пакета от узла выше по дереву рассылает его всем узлам непосредственно ниже по дереву (иллюстрация 2). При инжектировании пакета в сеть не в корневом узле сначала генерируется запрос на broadcast, который посылается корню.

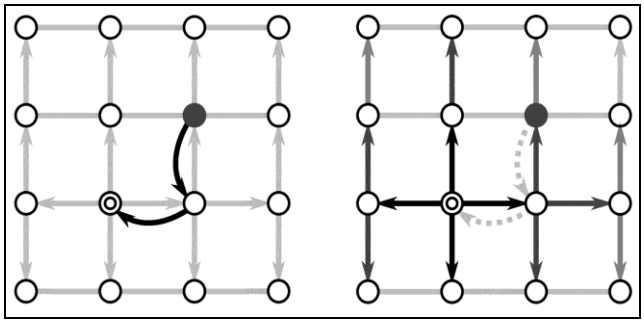

Рис. 1. Посылка запроса на broadcast и рассылка broadcast.

При выполнении операции reduce узел ждёт пакеты от хоста (в случае, если узел не является транзитным) и всех узлов непосредственно ниже по дереву, выполняет над ними указанную в пакете редукционную операцию (одну из заранее заданных коммутативных ассоциативных бинарных операций; в текущей реализации поддерживаются сумма, минимум, максимум) и отправляет результат выше по дереву. Операция reduce может завершаться (в корневом узле) посылкой результата одному узлу (операцией «точка — точка») или всем (операцией broadcast).

Задание дерева происходит на каждом узле посредством настройки таблицы маршрутизации коллективной подсети, при этом указываются:

- направления, в которых находятся узлы непосредственно ниже по дереву;
- направление, в котором находится узел непосредственно выше по дереву;
- является ли узел транзитным;
- является ли узел корнем.

Поскольку дерево задаётся распределённо (т. е. каждый узел обладает информацией только о своих соседях), необходимо гарантировать, что совокупность таблиц задаёт корректное дерево. Необходимыми условиями для этого являются:

- Условия, проверяемые при передаче пакетов:
  1) пакеты, приходящие от корня (пакеты broadcast) должны приходить по направлению, противоположному направлению к узлу выше по дереву;
  2) пакеты, идущие к корню (reduce, запросы на broadcast) должны приходить по направлениям от узлов ниже по дереву;
- Условия, проверяемые при задании таблицы маршрутизации:
  1) направления на узлы ниже по дереву могут быть только по измерениям, следующим за направлением вверх по дереву и по направлению, противоположному направлению вверх по дереву;
- Глобальные условия для таблиц маршрутизации:
  1) корень ровно один;
  2) если в каком-то узле выставлено направление вниз по дереву, то в этом направлении должен находиться принадлежащий дереву узел, в котором направление вверх по дереву выставлено противоположным данному.

Соблюдение условий, используемых при передаче пакетов и задании таблицы маршрутизации, позволяет избежать неограниченной рассылки пакетов при произвольном задании таблиц маршрутизации на узлах. Ограничения же на зависимость между таблицами маршрутизации позволяет гарантировать корректность задания дерева.

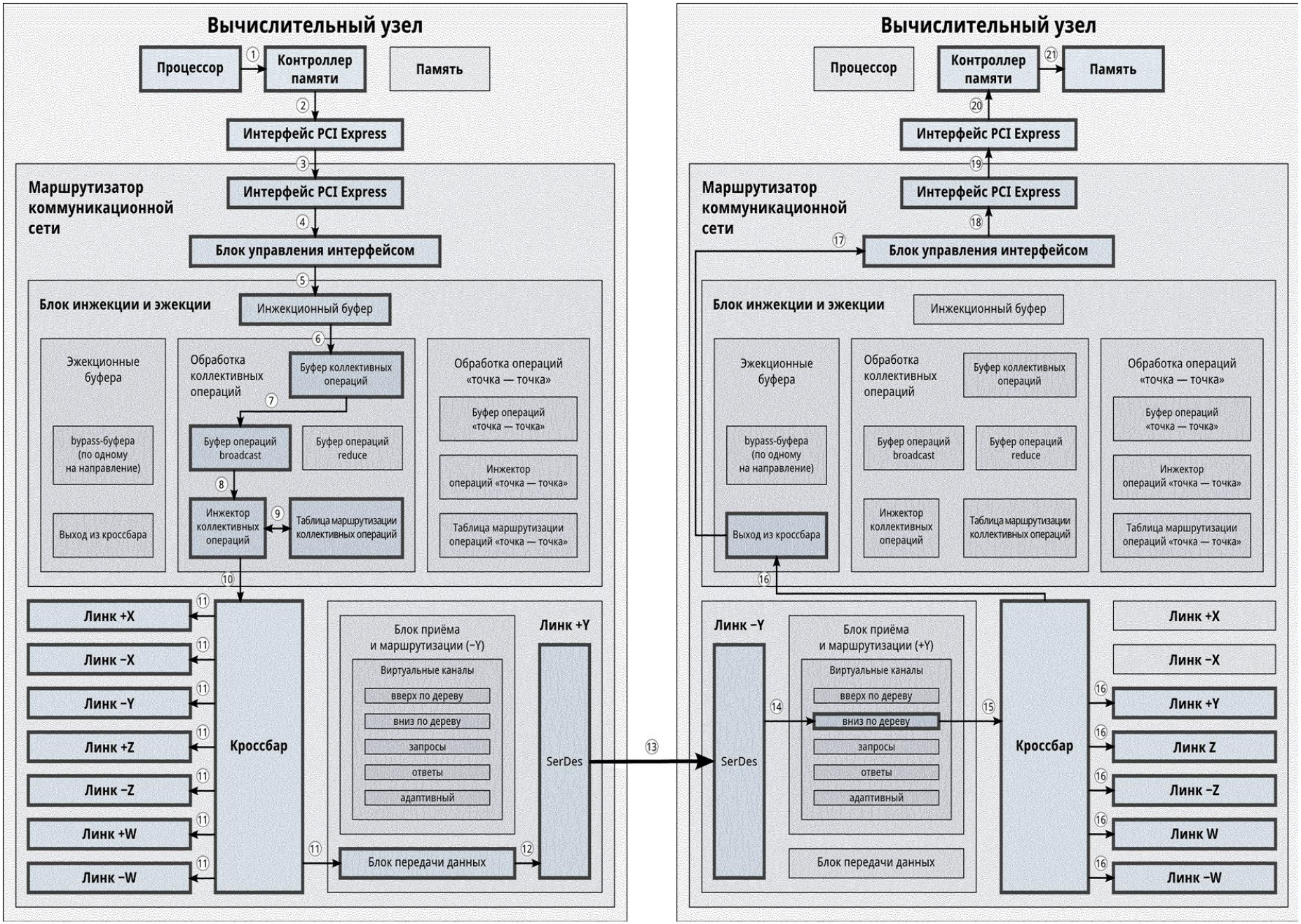

Рис. 3. Процесс отправки и приема broadcast внутри маршрутизатора

Процесс отправки и приёма broadcast внутри маршрутизатора состоит из следующих этапов (иллюстрация 3):

1 – процессор выполняет запись пакета специального вида по адресу, соответствующему кольцевому буферу маршрутизатора;

2 – контроллер памяти согласно внутренней таблице маршрутизации (PAT) отправляет запись в маршрутизатор посредством интерфейса PCI Express;

3 – запись передаётся по шине PCI Express;

4, 5 – блок управления интерфейсом обрабатывает адрес записи и помещает её в инжекционный кольцевой буфер;

6, 7 – из инжекционного буфера пакет попадает в буфер коллективных операций, откуда попадает в буфер операций broadcast;

8, 9 – инжектор коллективных операций извлекает пакет из буфера, на основе таблицы маршрутизации коллективных операций добавляет адресную информацию и принимает решение о маршрутизации пакета по определённым направлениям;

10, 11, 12, 13 – пакеты инжектируются через кроссбар и попадают в блок передачи данных, откуда через SerDes отправляются по среде передачи данных;

14 – в соответствии с информацией в переданном по сети пакете, он помещается во входной буфер соответствующего виртуального канала (в случае broadcast пересылка идёт по виртуальному каналу вниз по дереву);

15, 16 – блок приёма и маршрутизации обрабатывает пакет и через кроссбар отправляет его на остальные линки и на эжекцию;

17, 18, 19, 20, 21 – из эжекционного буфера пакет через блок управления интерфейсом отправляется как DMA-запись по PCI Express, что обрабатывает и выполняет контроллер памяти.

В рамках сети можно задавать различные пересекающиеся деревья; каждому дереву сопоставляется идентификатор, который указывается при задании таблицы маршрутизации и по которому производится выборка из неё при принятии решении о маршрутизации для пакета. Также для каждого дерева возможно задавать различные группы подгруппы узлов (идентификатор подгруппы, также, как и идентификатор дерева, указывается в пакете). Эти два механизма позволяют обеспечить эффективную поддержку механизма коммуникаторов в MPI и задание подмножеств узлов в Shmem (в которых участвует только часть узлов с определённым шагом).

Одновременно маршрутизатор поддерживает ограниченное число reduce (в текущей реализации — 32); каждый reduce, выполняющийся по данному дереву, имеет свой идентификатор (от 0 до 31). Наличие счётчиков отправленных и завершённых reduce в рамках узла позволяет организовывать управление потоком данных по подсети коллективных операций.

Базовые версии коллективных операций — односторонние асинхронные, т. е. управление процессору возвращается, как только операции отправлены в сеть, а результат записывается в память без активного участия принимающей стороны (т. е. аналогично удалённой записи). Это позволяет не блокировать процессор, эффективно совмещая ожидание окончания коллективных операций со счётом.

Для того, чтобы узнать, что коллективная операция завершилась и результат доступен всем участвовавшим в операции узлам, необходима синхронизация. Синхронизация выполняется посредством посылки специального reduce с результатом, рассылаемым всем узлам — детерминированный порядок передачи гарантирует, что все предыдущие broadcast для данного узла, а также reduce и запросы на broadcast для всех узлов завершатся до прихода результата этого reduce данному узлу. Для гарантирования получения broadcast всеми узлами дерева необходимо выполнить два синхронизирующих reduce подряд (это гарантирует получение другими узлами всех broadcast так как для рассылки результата второго синхронизирующего reduce необходимо, чтобы все его выполнили, а это возможно не ранее, чем все узлы выйдут из первого синхронизирующего reduce, то есть, получат broadcast первого синхронизирующего reduce, а, следовательно, получат все предыдущие broadcast'ы вследствие детерминированного порядка передачи пакетов в подсети). Для операции reduce также существует специальный программно-аппаратный механизм,

позволяющий узнать, что результат операции reduce уже доступен на данном узле.

Так как сеть ориентирована на эффективную поддержку Shmem и MPI, то входящие в них примитивы для реализации коллективных операций поддерживаются следующим образом:

Shmem:

1) реализация broadcast/reduce односторонняя асинхронная, что соответствует стандарту Cray Shmem;

2) присутствует аппаратная поддержка strides [3] посредством предопределённых деревьев с заданными group mask (16 деревьев позволяют покрыть все варианты stride с logPE_stride в диапазоне 0..7; при больших же значениях возможна рассылка посредством операций «точка — точка» без существенного снижения эффективности при размере сети менее 4—16 тысяч узлов);

MPI:

3) аппаратно поддерживаются операции MPI_Broadcast и MPI_Reduce, MPI_Allreduce;

4) аппаратно поддерживаются коммуникаторы созданием соответствующих деревьев и заданием group mask.

На данный момент завершается отладка коллективных операций на макетном образце коммуникационной сети третьего поколения (М3), состоящем из 9 узлов, соединённых в топологию двухмерный тор 3×3 [9]. Время выполнения broadcast для 9-ти узлов составляет 3,1 мкс, пропускная способность достигает 5,6 Гбит/с. По результатам имитационного моделирования для 512-узловой системы (тор 8×8×8), с сетью на базе прототипа на кристалле, увеличение пропускной способности на аппаратно поддерживаемых коллективных операциях относительно реализации их с помощью передач «точка — точка» составит 10,52 раз.

Литература

1. *Fox, Geoffrey C. and Johnson, Mark A. and Lyzenga, Gregory A. and Otto, Steve W. and Salmon, John K. and Walker, David W.,* Solving problems on concurrent processors. Vol. 1: General techniques and regular problems, 1988, Prentice-Hall, Inc., ISBN 0-13-823022-6

2. Message Passing Interface Forum, MPI: A Message-Passing Interface Standard, 1995, http://www.mpi-forum.org/docs/mpi-11-html/node64.html

3. *Karl Feind, Cray Research,* Shared Memory Access (SHMEM) Routines, 1995, http://www.cug.org/5-publications/proceedings_attendee_lists/1997CD/S95PROC/303_308.PDF

4. *Elizabeth Wiebel, David Greenberg, Steve Seidel, UPC* Collective Operations Specifications, 2003, http://upc.gwu.edu/docs/UPC_Coll_Spec_V1.0.pdf

5. *Vijay Saraswat, Bard Bloom, Igor Peshansky, Olivier Tardieu, David Grove,* X10 Language Specification, 2011, http://dist.codehaus.org/x10/documentation/languagespec/x10-latest.pdf

6. *Vasanth Bala, Jehoshua Bruck, Robert Cypher, Pablo Elustondo, Alex Ho, Ching-Tien Ho, Shlomo Kipnis, Marc Snir,* CCL: A Portable and Tunable Collective Communication Library for Scalable Parallel Computers, Parallel Processing Symposium: с. 835—844 , Proceedings, ISBN 0-8186-5602-6 , 1994, http://citeseer.ist.psu.edu/viewdoc/download;jsessionid=3D465188A4C42E5F58002758BE8B57C3?doi=10.1.1.155.3612&rep=rep1&type=pdf

7. *George Almási, Gábor Dózsa, C. Erway, Burkhardt Steinmacher-Burow.* Efficient Implementation of Allreduce on BlueGene/L Collective Network, Recent Advances in Parallel Virtual Machine and Message Passing Interface: с. 57—66, Springer Berlin / Heidelberg, 2005, http://dx.doi.org/10.1007/11557265_12

8. *А. А. Корж, Д. В. Макагон, А. А. Бородин, И. А. Жабин, Е. Р. Куштанов, Е. Л. Сыромятников, Е. В. Черемушкина.* Отечественная коммуникационная сеть 3D-тор с поддержкой глобально адресуемой памяти для суперкомпьютеров транспетафлопсного уровня производительности, Параллельные вычислительные технологии (ПаВТ'2010): Труды международной научной конференции (Уфа, 29 марта — 2 апреля 2010 г.): с. 227—237, Челябинск: Издательский центр ЮУрГУ, ISBN 978-5-696-03987-9, 2010, http://omega.sp.susu.ac.ru/books/conference/PaVT2010/full/134.pdf

9. *А. С. Симонов, И. А. Жабин, Д. В. Макагон.* Разработка межузловой коммуникационной сети с топологией «многомерный тор» и поддержкой глобально адресуемой памяти для перспективных отечественных суперкомпьютеров. — Научно-техническая конференция «Перспективные направления развития вычислительной техники», ОАО «НИЦЭВТ», 2011.