УДК 004.382.2:004.021.004.72

Варианты реализации барьерной синхронизации для высокоскоростных коммуникационных сетей с топологией «многомерный тор»

Д.В. Макагон
начальник отдела высокоскоростных коммуникационных сетей, ОАО «НИЦЭВТ»
E-mail: makagond@nicevt.ru
Е.Л. Сыромятников
науч. сотрудник, ОАО «НИЦЭВТ»
E-mail: syromiatnikov@nicevt.ru

Рассмотрено несколько вариантов реализации аппаратной поддержки барьерной синхронизации в высокоскоростной коммуникационной сети с топологией «многомерный тор». Проведено сравнение реализации барьерной синхронизации через двусторонние и односторонние коммуникации, а также с использованием агрегации пакетов в сетевом адаптере и минимизацией трафика в соответствии с топологией сети.

***Ключевые слова**: коллективные операции, синхронизационные операции, барьерная синхронизация, RDMA, топология «многомерный тор».*

This paper describes several approaches to barrier synchronization hardware support in high-speed interconnection networks with multi-dimensional torus topology. The comparison of implementations based on two-sided and one-sided communication models is presented, as well as implementations using packet aggregation in network adapter and topology-aware traffic minimization.

***Keywords:** collective operations, synchronization operations, barrier synchronization, RDMA, torus topology.*

Введение

Барьерная синхронизация используется как один из ключевых элементов существенного числа параллельных алгоритмов. Эффективная и масштабируемая реализация ее помогает избежать снижения масштабируемости задач при выполнении на большом числе вычислительных узлов.

Базовые определения

Прежде чем рассматривать различные способы выполнения барьерной синхронизации, следует дать определения различных понятий, используемых в дальнейшем, ввиду того, что в различных источниках можно встретить значительное число не вполне эквивалентных определений различных сущностей (см., например, [4–7]).

Задача – множество процессов, предназначенных для достижения единой цели, выполняемых на наборе вычислительных узлов, объединенных коммуникационной сетью и использующих сеть для взаимодействия между собой посредством обмена данными. Случай использования нескольких коммуникационных сетей сводится к случаю с использованием одной коммуникационной сети.

Процесс – последовательно выполняемый на вычислительном узле код (команды исполнителя). Код выполняется независимо от других узлов, т.е. отсутствует неявная внешняя синхронизация различных процессов. Случай использования нескольких потоков и/или процессов на одном узле сводится к последовательному путем введения дополнительного этапа внутриузловой синхронизации между процессами/потоками и выделения мастер-процесса, код которого опять же выполняется последовательно. Рассмотрение вопросов внутриузловой синхронизации выходит за рамки данной статьи (некоторые вопросы внутриузловой синхронизации и синхронизации процессов, работающих в рамках многопроцессорной системы с общей памяти, рассмотрены в [4, 5]). Процесс характеризуется точкой выполнения и данными, доступными ему локально. Данные могут изменяться этим процессом и другими процессами задачи (с участием данного процесса в случае двусторонних коммуникаций или без его участия в случае односторонних).

Коммуникационная операция – единица взаимодействия процессов посредством коммуникационной сети в рамках задачи. Коммуникационная операция имеет некий тип, который определяет ее операции, семантику (запись, чтение, атомарная операция). Коммуникационная операция может состоять из одного или нескольких пакетов – единиц сетевого взаимодействия в рамках коммуникационной сети. Коммуникационная операция может находиться в одном из следующих состояний:

– операция еще не была вызвана;
– операция была вызвана;
– данные из источника скопированы;
– данные инжектированы в сеть;
– данные получены из сети;
– данные записаны в целевое месторасположение;
– операция выполнена.

В зависимости от вида коммуникационных операций (блокирующие/неблокирующие, синхронные/асинхронные), управление процессам, вызывавшим операции, может возвращаться в различные моменты времени. Конкретные действия, выполняемые в рамках различных состояний хода коммуникационной операции, определяются для каждого из процессов-участников взаимодействия согласно ее коммуникационной операции, семантике. Коммуникационные операции в зависимости от их семантики и архитектуры сети могут накладывать ограничения на состояния и переходы между со-стояниями.

Синхронизационная гарантия – некое свойство взаимного расположения точек вы- полнения процессов во времени и состояний обменов данными, декларируемое для каждого отдельного процесса или для всех процессов задачи и выполняемое для одного или всех процессов задачи. Важно отметить, что декларирование и выполнение свойства есть независимые сущности: например, в случае синхронизационной гарантии «все коммуникационные операции, вызванные данным процессом, выполнены» гарантия декларируется для одного процесса, но выполняться должна для всех. Случай реализации синхронизационной гарантии для подмножества процессов задачи сводится к данному посредством определения пустого множества предъявляемых свойств к процессам,.не входящим в подмножество.

Синхронизационная операция – (служебная) операция обмена данными, выполнение которой (возврат управления после выхода) дает определенные синхронизационные гарантии процессам-участникам.

Виды синхронизационных гарантий

Изначально, барьерная синхронизация есть синхронизационная операция, позволяющая удостовериться в том, что ни один процесс-участник синхронизационного взаимодействия не продолжит работу до тех пор, пока все элементы не достигнут барьера (такое определение барьера встречается, например, в [8, 4, 5]). При этом (как указано там же) барьер обычно используется для получения гарантий того, что можно продолжать вычисления. Это в свою очередь означает, что все коммуникационные операции выполнились и все данные были получены (данным процессом или всеми).

В случае же использования неблокирующих, асинхронных и односторонних операций ситуация становится нетривиальной (а указанное выше определение — не вполне рабочим): процесс может дойти до барьера, но при этом может случиться так, что упомянутые коммуникационные операции еще не завершились (начиная от ситуации, когда не все операции возымели эффект, вплоть до ситуации, когда не все данные были считаны из узла- отправителя).

При синхронизации процессов, выполняющихся параллельно и обменивающихся данными в те или иные моменты работы программы посредством неблокирующих, асинхронных и односторонних операций, необходимы различные гарантии относительно состояния других узлов или доставки пакетов. Среди них можно выделить следующие, применяющиеся к операциям, вызванным до вызова синхронизирующей операции:

1) все узлы вошли в барьер (вызвали синхронизационную операцию);

2) все данные от этого узла (которые необходимо отправить в рамках выполнения коммуникационных операций, вызванных до вызова синхронизационной операции) были оформлены в виде пакетов и инжектированы в сеть;

3) данные (посланные в рамках коммуникационных операций, вызванных до вызова синхронизационной операции) от всех узлов получены;

4) все данные, посылаемые после синхронизационной операции, будут получены строго после данных, посланных до синхронизационной операции. Возможна ситуация, когда данные были получены, но эффекта не возымели, например, в случае атомарных операций без возврата значения. В этом случае имеется в виду, что данные получены и эффект от них применен в полной мере. Заметим, что данная гарантия не более сильная, чем предыдущая.

5) все операции, вызванные данным узлом до вызова синхронизационной операции, завершены.

Виды синхронизационных операций

В различных ситуациях может требоваться (является достаточным) различный набор гарантий. Далее можно определить барьерные операции, предоставляющие тот или иной набор гарантий.

1. *Барьер*. Дает гарантии, указанные в списке синхронизационных гарантий в разделе «Виды синхронизационных гарантий» под пунктами 1, 3 (и, как следствие, 4), 5 для всех процессов, участвующих в операции. Отметим, что здесь гарантию 5 можно заменить на гарантию получения всех пакетов всеми узлами как эквивалентную применительно ко всем узлам.

2. *Полубарьер*. Дает гарантии 1 и 3 для всех процессов, участвующих в операции.

3. *Fence*. Дает гарантию 4 для процесса, вызывающего данную операцию.

4. *Quiet*. Дает гарантию 5 в отношении некоего целевого узла для процесса, вызывающего данную операцию.

Как можно заметить, барьер и полубарьер различаются наличием гарантии завершения всех операций, вызванных данным узлом (последний ее не предоставляет). Однако факт выхода из полубарьера предоставляет гарантию 3, а совокупность данных гарантий, выполняемых для всех узлов, есть более сильная гарантия, нежели гарантия 5. Как следствие, для получения гарантий, предоставляемых барьером, достаточно выполнить два полубарьера.

Барьерная синхронизация в случае использования неблокирующих операций

В случае использования неблокирующих операций необходимо иметь возможность получать информацию о состоянии отправки данных из локальных буферов. Ввиду особенностей операций (отсюда, невозможности основываться на возврате управления) нижним уровнем (программным или аппаратным) должна быть предоставлена отдельная возможность получения состояния выполнения той или иной неблокирующей операции. С точки зрения интерфейса это обычно реализуется посредством проверки handle, возвращаемого неблокирующей операцией.

Барьерная синхронизация в случае использования односторонних операций

В случае использования односторонних операций необходимо принимать специальные меры для получения гарантий того, что все отправленные пакеты были получены и исполнены, так как приемник не располагает информацией о данных, которые ему посылались (ввиду посылки их без его ведома), а отправитель может не располагать информацией о состоянии доставки пакетов.

Как следствие, при использовании для сетевых обменов односторонних коммуникаций, необходимо иметь тот или иной механизм, позволяющий предоставить гарантии барьерной синхронизации.

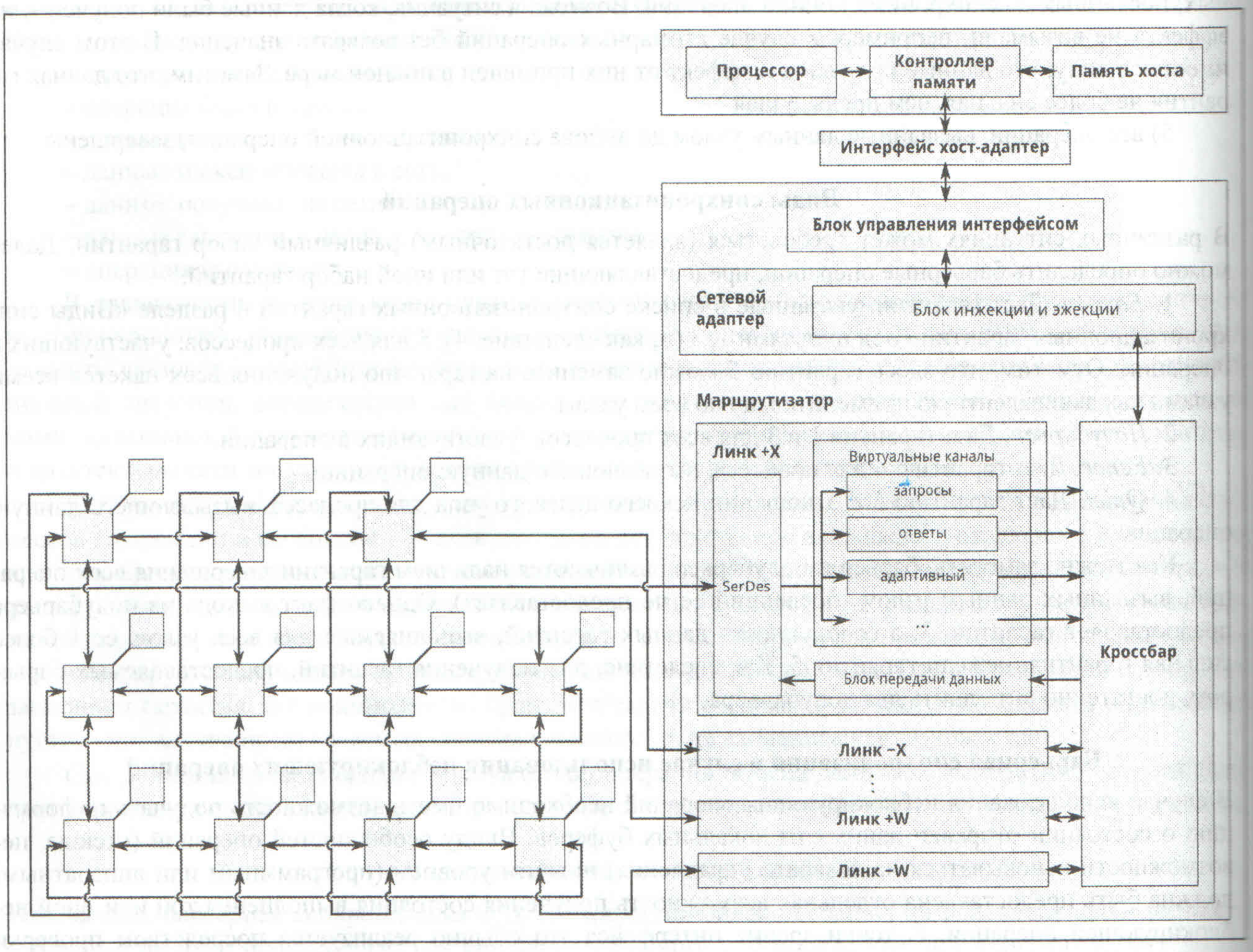
Подтверждения всех пакетов. Данный метод используется, например, в суперкомпьютерах фирмы Cray [1]. С одной стороны, из недостатков можно отметить значительную долю служебного трафика, уходящую на доставку подтверждений. С другой стороны, предоставление тех или иных синхронизационных гарантий реализуется относительно просто.

Детерминированный порядок передачи пакетов. С одной стороны, позволяет исключить необходимость хранения неподтвержденных пакетов. С другой стороны, детерминированная маршрутизация (являющаяся одним из основных способов достижения детерминированного порядка передачи) имеет ряд недостатков.

Подсчет пакетов требует больших буферов (для ряда синхронизационных гарантий – по числу узлов) для хранения информации о счетчиках.

Архитектура сети

В данном разделе приведено описание сети, в рамках которой проводится дальнейшее рассмотрение операций барьерной синхронизации (см. рисунок).



Архитектура коммуникационной сети

Сеть состоит из множества узлов, включающих в себя вычислительную часть (процессор) и коммуникационную, состоящую из адаптера коммуникационной сети и маршрутизатора. Сеть имеет топологию «многомерный тор». Каждый узел представляет собой вычислительный узел (обычно, на базе платформы с одним или несколькими процессорами с архитектурой x86) с подключенным (по интерфейсу PCI Express) адаптером коммуникационной сети.

Вычислительный узел может выполнять операции с памятью удаленных узлов сети посредством пакетов, записываемых в определенный регион адаптера и обрабатываемых адаптером по их получении. Для работы с памятью узла адаптер использует пакеты memory write и memory read request стандарта PCI Express.

Доступны следующие операции:

– запись в память удаленного узла (без подтверждения; блокирующая/неблокирующая);

– чтение из памяти удаленного узла (без оповещения);

– атомарные операции в памяти удаленного узла (без возврата значения, без подтверждения).

Так как возможны блокировки вида запрос/ответ, записи могут выполняться по двум виртуальным каналам: блокирующему (vc1) и неблокирующему (vc0). Чтения и атомарные операции выполняются только по блокирующему каналу, ответы на чтения посылаются только по неблокирующему каналу.

Операции при инжекции их в сеть разбиваются на пакеты (сообщения), каждый из которых маршрутизируется независимо (и в случае адаптивной маршрутизации маршруты у пакетов различны). Помимо сообщений, созданных в рамках выполнения операций, могут использоваться служебные сообщения.

Пакеты передаются по сети посредством связей (линков) между узлами (передача от одного узла соседнему по линку называется хопом или переходом) согласно правилам маршрутизации [2, 3]. Пакеты могут передаваться как детерминированно, так и адаптивно.

Варианты реализации барьерных операций

Барьерная синхронизация (наряду с иными коллективными операциями) является важным аспектом работы сети именно ввиду нетривиальности эффективности своей реализации. Существует большое число работ, посвященных эффективной реализации синхронизационных операций [4–6]. Среди них можно выделить следующие подходы.

1) *Синхронизации с использованием операций с подтверждением*. В этом случае узлы сначала дожидаются (программно или аппаратно) подтверждений о доставке всех пакетов, посланных до вызова операции барьерной синхронизации, после чего необходимо распространить информацию о выполнении синхронизационной гарантии. Это может быть сделано следующими способами:

а) двухфазная синхронизация с выделенным мастером;

б) синхронизация на дереве;

в) синхронизация на гиперкубе.

Подробно данные виды синхронизации рассмотрены, например, в [7] (случай с малым размером данных).

2) *Синхронизация, использующая подсчет пакетов*. В этом случае разделение счетчиков по узлам можно делать только для отправляемых пакетов, что позволяет исключить подобное разделение на принимающей стороне (которое в этом случае пришлось бы реализовывать исключительно аппаратно).

3) *Синхронизация с использованием односторонних операций*, основанная на порядке доставки пакетов.

а) односторонняя синхронизация «все–всем». Тривиальный способ, основанный на последовательной доставке пакетов от одного узла к другому.

б) многофазная синхронизация.

Далее более подробно рассматриваются способы синхронизации, использующие детерминированный порядок передачи пакетов.

Односторонняя синхронизация «все-всем». Идея данного варианта реализации барьера состоит в том, что:

1) при детерминированной маршрутизации пакеты, посланные от одного узла другому, не обгоняют друг друга;

2) односторонние RDMA записи используют гораздо меньше ресурсов, чем записи с подтверждением (не надо хранить информацию об outstanding запросах и посылать подтверждения по сети).

Как следствие, если все узлы обменяются пакетами со всеми, то у каждого узла будет информация о том, что до него дошли барьерные пакеты от всех других узлов и, как следствие детерминированной передачи, все пакеты, посланные от данных узлов до начала барьера.

К несомненным преимуществам данного метода можно отнести его простоту как с алгоритмической точки зрения, так и с точки зрения аппаратной поддержки (при наличии поддержки детерминированного порядка передачи от аппаратуры более не требуется ничего). Однако очевидна его плохая масштабируемость (для полубарьера требуется $N^2 - N$ пакетов, где N – число процессов-участников синхронизационной операции), которая делает данный способ барьерной синхронизации малопригодным для использования при числе участников синхронизационной операции в несколько тысяч и более.

Многофазная синхронизация. Барьер выполняется для группы узлов и проходит в K фаз, где K должно равняться диаметру группы (диаметр множества узлов – максимальное число переходов между узлами сети, требуемое для передачи сообщения между какими-либо двумя узлами, принадлежащими множеству). Для синхронизации используются специальные сообщения, для которых гарантируется, что они не могут обгонять пакеты, посланные ранее по линку (например, это верно для пакетов, которые

доставляются последовательно в рамках одного виртуального канала) и при пересылке с линка на линк внутри маршрутизатора.

Первая фаза ($i = 0$) начинается как только узел входит в барьер. На фазе i маршрутизатор выполняет следующие операции:

1) разослать по одному специальному пакету каждому непосредственному соседу;

2) дождаться по специальному пакету от каждого непосредственного соседа;

3) $i = i + 1$, если $i = K$ — выйти из барьера, иначе перейти на новую фазу.

Лемма 1. Фазы соседних узлов отличаются не больше, чем на 1.

По определению:

1) если узел находится в фазе i, то все его соседи находятся в фазе, не меньшей $i - 1$ (иначе узел не смог бы перейти в данную фазу);

2) если узел находится в фазе i, то все его соседи находятся в фазе, не меньшей $i + 1$ (они не могут перейти в фазу $i + 2$ и более как минимум из-за данного узла).

Отсюда по индукции фазы узлов на расстоянии n хопов отличаются не более, чем на n. Как следствие, если узел находится в фазе i, то это значит, что в радиусе i все узлы вышли на барьер (так как номер фазы у соседних узлов не может отличаться более, чем на 1).

Лемма 2. Пакет, посланный из некоторого узла до барьерной синхронизации, на n-м хопе окажется в узле в фазе не больше n.

Данный вид барьера может использоваться как при адаптивной минимальной маршру- тизации пакетов, так и при детерминированной, поскольку, как сказано выше, при нахождении узла в фазе i все узлы в радиусе i вышли на барьер, следовательно, ни по одному из путей в радиусе i не может пройти пакет, инжектированный внутри него.

Гарантии после K фаз:

1) узел не выйдет из барьера раньше, чем к нему дойдут все пакеты ему (из леммы 2);

2) узел не выйдет из барьера раньше, чем все узлы в барьер войдут (из следствия из леммы 1);

3) как следствие, все пакеты, посланные после барьера, будут доставлены строго после пакетов, посланных до начала синхронизационной операции.

Отсюда получаем, что данный вид синхронизационной операции предоставляет гарантии полубарьера после выполнения K фаз.

Для получения гарантии доставки всех пакетов, посланных с данного узла, нужно сделать $2K$ фаз.

Оценка производительности

Как было указано при наличии подсчета пакетов или подтверждений задача барьерной синхронизации сводится к распространению информации о достижении барьера (по получению всех подтверждений или распространение информации о числе пакетов, необходимых для доставки). Временные оценки для этих операций есть, например, в [7]. В связи с этим далее проводится временная оценка вариантов реализации барьерной синхронизации для односторонних сообщений.

Примерное время работы барьера (на пустой сети с числом узлов) можно оценить (снизу) как

$$\max\left(rate_{link} \max_{i \in links}\left(pkts_{link}(i)\right),\ rate_{inj}(N-1),\ rate_{ej}(N-1)\right) + lat_{hop}K + lat_{inj} + lat_{ej}, \quad (1)$$

где K – диаметр сети; lat_{hop} – задержка на передачу пакета между узлами; lat_{inj} – задержка при инжекции; lat_{ej} – задержка при эжекции; $rate_{link}$ – величина, обратная темпу передачи пакетов по линку; $rate_{inj}$ – величина, обратная темпу инжекции; $rate_{ej}$ – величина, обратная темпу эжекции; $pkts_{link}(i)$ – число пакетов для передачи по линку i; $links$ – множество линков сети.

Односторонняя синхронизация «все–всем». Если предположить, что $N = n \times n \times n$ (случай трехмерного равностороннего тора), то формулу можно переписать следующим образом:

$$\max\left(rate_{link} \times \max_{i \in links}\left(pkts_{link}(i)\right), rate_{inj} \times \left(n^3 - 1\right), rate_{ej} \times \left(n^3 - 1\right)\right) + lat_{hop} \times \frac{3}{2} \times n + lat_{inj} + lat_{ej}. \quad (2)$$

При этом при использовании детерминированного порядка направлений наибольшее число пакетов будет пересылаться по линкам в первом направлении (по всем – одинаковое число вследствие симметрии), которое можно оценить как $\frac{n^4}{4}$.

Следовательно, если предположить, что $rate_{inj} = rate_{ej}$, и ввести коэффициент $r = rate_{link}/rate_{inj}$, то можно оценить, что при $n > r \times (n^3 - 1)/n^3 \approx r$ время работы барьера «все–всем» определяется временем передачи пакетов по линкам и примерно оценивается снизу как $n^4/4\ rate_{link} + 3/2\ lat_{hop}\ n + lat_{inj} + lat_{ej}$ на больших сетях и $rate_{inj}\ (n^3 - 1) + 3/2\ lat_{hop}\ n + lat_{inj} + lat_{ej}$ на малых. При этом по сети пересылается $N(N-1) = n^6 - n^3$ пакетов со средней длиной маршрута $\frac{3}{4}n$.

Аналогично, если предположить, что $N = n \times n \times n \times n$ (случай четырехмерного тора), то оценку можно записать как $n^5/4\ rate_{link} + 2\ lat_{hop}\ n + lat_{inj} + lat_{ej}$ на больших сетях и $rate_{inj}\ (n^4 - 1) + 2\ lat_{hop}\ n + lat_{inj} + lat_{ej}$ для малых. При этом по сети пересылается $N(N-1) = n^8 - n^4$ пакетов со средней длиной маршрута n.

Многофазная синхронизация. Время работы многофазного барьера ограничено задержкой (так как на каждой фазе по одному линку пересылается ровно один пакет), отсюда можно оценить время работы многофазного барьера на пустой сети как $K(lat_{hop} + lat_{inc}) + lat_{inj} + lat_{ej}$, где lat_{inj} – задержка на инкрементирование фазы; dim – число измерений.

Можно оценить, в каких случаях использование многофазного барьера имеет смысл:

$$dim \times n \times \left(lat_{hop} + lat_{inc}\right) + lat_{inj} + lat_{ej} < \max\left(\frac{n^{dim+1}}{4} rate_{link}, rate_{inj}\left(n^{dim} - 1\right)\right) + \frac{dim}{2} lat_{hop} n + lat_{inj} + lat_{ej}, \quad (3)$$

$$dim \times n \times lat_{hop} + dim \times n \times lat_{inc} < \max\left(\frac{n^{dim+1}}{4} rate_{link}, rate_{inj}\left(n^{dim} - 1\right)\right) + \frac{dim}{2} lat_{hop} n, \quad (4)$$

$$\frac{dim}{2} \times n \times lat_{hop} + dim \times n \times lat_{inc} < \max\left(\frac{n^{dim+1}}{4} rate_{link}, rate_{inj}\left(n^{dim} - 1\right)\right), \quad (5)$$

$$dim\left(\frac{1}{2} lat_{hop} + lat_{inc}\right) < \max\left(\frac{n^{dim}}{4} rate_{link}, rate_{inj}\left(n^{dim} - 1\right)\right). \quad (6)$$

В (6) приведено соотношение, позволяющее на основе информации о характеристиках сети (темпе инжекции, эжекции, прохождении пакетов по линку, размере сети и числа измерений тора, а также задержках на хоп и смену фазы) принять решение об оптимальности возможности использования одного или другого варианта реализации барьерной синхронизации.

Заключение

Итак, рассмотрены некоторые варианты реализации барьерной синхронизации. Для проведения их сравнения были определены понятия синхронизационных гарантий и для синхронизационных операций указаны наборы гарантий, ими предоставляемые. Также были классифицированы возможные варианты реализации барьерной синхронизации по способу получения гарантии по доставке пакетов. Для варианта с односторонними операциями без подтверждений доставки и без подсчета пакетов были рассмотрены два варианта реализации барьерной синхронизации, дана оценка их временной сложности и проведено сравнение возможной применимости того или иного варианта в зависимости от размера сети.

Литература

1. *Alverson R., Roweth D., Kaplan L.* The Gamini System Interconnect, 18th // IEEE Symposium on High Performance Interconnects. 2010.
2. *Макагон Д.В., Сыромятников Е.Л.* Сети для суперкомпьютеров // Открытые системы. СУБД. Сентябрь 2011. № 7.
3. *Корж А.А., Макагон Д.В., Жабин И.А., Сыромятников Е.Л.* и др. Отечественная коммуникационная сеть 3D-тор с поддержкой глобально адресуемой памяти для суперкомпьютеров транспетафлопсного уровня производительности // Параллельные вычислительные технологии (ПаВТ'2010): Труды Междунар. научн. конф. (Уфа, 29 марта — 2 апреля2010 г.), http://omega.sp.susu.ac.ru/books/conference/PaVT2010/full/134.pdf. Челябинск: Издательский центр ЮУрГУ. 2010. С. 227–237.

4. *Ramachandra Nanjegowda, Oscar Hernandez, Barbara Chapman, Haoqiang H. Jin* Scalability Evaluation of Barrier Algorithms for OpenMP // IWOMP 2009. LNCS 5568. Springer-Verlag Berlin Heidelberg. 2009. P. 42–52.
5. *John Sartori, Rakesh Kumar* Low-Overhead, High-Speed Multi-core Barrier Synchronization // HiPEAC 2010. LNCS 5952. Springer-Verlag Berlin Heidelberg. 2010. P. 18–34.
6. *Hoefler T.* A survey of barrier algorithms for coarse grained supercomputers. Chemnitzer Informatik-Berichte. 2004.
7. *Kayhan M. Imre, Cesur Baransel, Harun Artuner* Efficient and Scalable Routing Algorithms for Collective Communication Operations on 2D All-Port Torus Networks // Int J Parallel Prog. Springer Science+Business Media, LLC. 2011.
8. Vijay *Moorthy, Dhabaleswar K. Panda, and P. Sadayappan* Fast Collective Communication Algorithms for Reflective Memory Network Clusters, CANPC 2000 // Lecture Notes in Computer Science 1797. Springer-Verlag Berlin Heidelberg New York. 2000. P. 100–114.

Implementation possibilities of barrier synchronization on high-speed interconnection networks with multi-dimensional torus topology

D. V. Makagon, E. L. Syromyatnikov

This article focuses on several approaches to barrier synchronization hardware support in high-speed interconnection networks with multi-dimensional torus topology.
The barrier synchronization is a key element of a wide range of parallel algorithms; it is the efficiency of its implementation that very often restricts the scalability of applications when running on a large number of compute nodes.
The basic definitions of a process, a task, a communication operation, a synchronization operation and synchronization guarantees are given in the first section of the article.
In the next two sections the basic types of synchronization guarantees and operations are specified.
The following sections describe the barrier and half-barrier synchronization for one-sided communications and non-blocking operations, as well as a multiphase barrier algorithm, designed to overcome the constraints and shortcomings of other methods.
The implementation possibilities, analytical computations and performance evaluation are presented, based on the generalized model of an interconnection network with multi-dimensional torus topology and compute nodes with x86-based processors and PCI Express network interface cards.
In summary the article gives a comprehensive insight into the problem of implementation of barrier synchronization for high-speed interconnection networks with multi-dimensional torus topology and proposes an algorithm, that provides a topology-aware traffic minimization and is suitable for bothone-sided and two-sided communication models.