УДК 004.032.24

МЕТОДЫ УСКОРЕНИЯ ВЫПОЛНЕНИЯ КОЛЛЕКТИВНЫХ ОПЕРАЦИЙ СТАНДАРТА MPI

Г. Г. Стецюра

Институт проблем управления им. В. А. Трапезникова, г. Москва

Рассмотрены быстрые способы выполнения операций стандарта MPI в системах, совмещающих процессы вычисления и обмена данными.

ВВЕДЕНИЕ

Интерфейс MPI (Message Passing Interface) широко распространен в мультипроцессорных и мультикомпьютерных системах. Он чаще всего реализуется довольно медленно программным путем [1, 2]. Но даже в системах, где для MPI создана соответствующая техническая поддержка, операции MPI также одни из самых медленных [3]. Это в основном относится к коллективным операциям, результат которых создается при коллективном взаимодействии процессоров — широковещательный обмен, сбор данных, рассылка, нахождение максимума, сумма, произведение, "логическое И" и др. Для ускорения коллективных операций MPI в данной статье предлагается использовать групповые операции (ГО) и групповые команды (ГК), разработанные в Институте проблем управления РАН. Для их реализации требуются специализированные технические средства, выполняющие вычисления непосредственно в процессе обмена данными в системе. Ниже кратко изложен принцип работы ГО и ГК и приведены примеры их возможного использования в MPI.

1. ГРУППОВЫЕ ОПЕРАЦИИ

Имеется много разновидностей ГО, они подробно рассмотрены в работах [4—6]. Здесь изложен лишь принцип действия ГО. На рис. 1 изображен канал, состоящий из двух линий, по которым могут распространяться слева направо сигналы. Сигнал от внешнего источника поступает на левый конец одной из линий. Если он поступает в верхнюю линию, то он интерпретируется как "1", иначе — как "0". Пунктиром выделены переключатели, через которые линии, входящие в переключатель, либо проходят без изменения, либо перекрещиваются, либо обе линии подключаются к одной из линий на выходе из переключателя. Левый переключатель переводит "1" в "0" и наоборот. В следующем переключателе любой сигнал превращается в "1", в следующем — в "0". В любом сечении канала сигнал находится только в одной линии. Сигналы асинхронные и один посылаемый в канал сигнал отделяется от другого паузой — отсутствием сигнала. В канале одновременно могут находиться несколько коротких сигналов или же весь канал может занять один длительный сигнал.

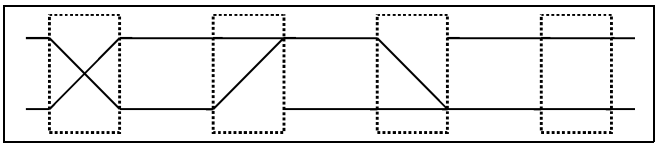

Рис. 1. Структура канала для выполнения групповой операции

При помощи указанных манипуляций с сигналом можно выполнить отрицание, логические операции, арифметические сложение, вычитание, умножение, а также операции max и min [4—6]. В этих операциях один операнд "x" поступает в переключатель из канала, а другой — "y" — находится в устройстве, управляющем состоянием линий в переключателе. Перестройка соединений *выполняется до прихода сигнала* в переключатель из канала, и поэтому выполнение операций над сигналами не требует дополнительной временной задержки. Следовательно, время выполнения ГО, которую выполняют устройства, управляющие переключателями, не зависит от их числа.

Приведем примеры выполнения ГО.

Операция "отрицание". В переключателе линии перекрещиваются, и значение x инвертируется.

Операция "исключающее ИЛИ". Если $y = 0$, то x пропускается далее в канал без изменений. Если $y = 1$, то значение x инвертируется.

Операция "логическое И". Если $y = 1$, то x пропускается в канал. Если $y = 0$, то вместо x далее в канал передается сигнал 0.

Операция "логическое ИЛИ" Если $y = 0$, то x пропускается в канал. Если $y = 1$, то вместо x в канал передается сигнал 1.

Счет. Каждый разряд числа обрабатывается операцией "исключающее ИЛИ". Разряды обрабатываются, начиная с младшего. Для проверки наличия переноса в старший разряд имеется время до прихода следующего разряда. Аналогично выполняется сложение и вычитание в дополнительном коде.

Обратим внимание на особенность рассмотренного метода. Для выдачи результата переключатель не должен создавать новый сигнал, используя энергию источника питания, как это делается в обычных логических элементах. Сигнал не генерируется, а изменяется путь его перемещения. Это ведет к существенному уменьшению потребления энергии в переключателе со всеми вытекающими последствиями.

2. ГРУППОВЫЕ КОМАНДЫ

Групповые операции позволяют создавать ГК и распределенные программы. Под распределенной программой будем понимать последовательность ГК, обеспечивающую вычисление общего для многих устройств результата и ветвление последовательности ГК, включая смену источника команд (мастера).

Структура ГК показана на рис. 2. В ГК имеется заголовок — код ГО, который определяет вид требуемых от устройств действий. Он определяет способ обработки операндов, место расположения данных, их значение, место расположения результата. Групповая команда содержит блоки хранения операндов и результатов. Групповые команды переменной длины должны завершаться признаком конца команды. На рис. 2 признак конца команды — пустой блок результата. Код операции ГК должен задавать способы передачи квитанции источнику и передачи прав мастера другим устройствам [4]. Устройство, получив из ГК код операции, выполняет последовательность обычных или групповых операций над операндами, указанными в ГК или (и) над операндами, хранящимися в устройстве.

Если результат вычисления устройство помещает в ГК, то это делается *без задержки* передаваемой по каналу ГК, как объяснено выше.

В блоке операндов располагаются значения операндов или адреса и имена, которые указывают путь к устройству и операнду в устройстве.

Результат, созданный одним устройством, для другого устройства служит операндом, который может быть замещен новым результатом без задержки команды.

Программе обычно необходимы средства ветвления, охватывающие группу устройств. Требуются ветвления следующих трех типов.

- Устройство-лидер (мастер), выполняющее программу, посылает ГК ветвления, которая передает управление другому, указанному в команде, устройству.
- Лидер оповещает группу устройств о требуемой работе. Если претендентов на выполнение этой работы одновременно несколько, то они должны выбрать одного из них с учетом приоритетов всех претендентов. Это задача приоритетного доступа к средствам связи. Ее решает описанная ниже операция "max". В ней в качестве сравниваемых чисел используются значения приоритетов. В результате либо выполняется работа без смены лидера, либо появляется новый лидер.
- Периодически (или постоянно, с использованием отдельного канала связи) лидер предоставляет право отнять у него лидерство другим устройствам с учетом их приоритетов.

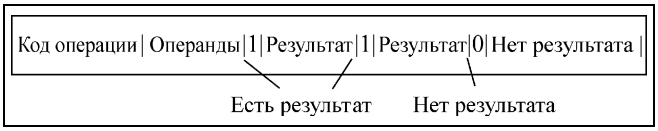

Рис. 2. Групповая команда

Изложенная структура ГК — это общая схема, которая допускает модификации в различных приложениях. Все ГК используют небольшое число следующих базовых действий:

— формирование многими устройствами общего результата и помещение его в ГК;
— отбор данных из ГК в устройства;
— загрузка в ГК группы результатов работы устройств;
— работа с ГК заранее неизвестной длины;
— смена лидера, имеющего право передачи команд распределенной программы.

3. КАНАЛ ТИПА "СКРЕПКА"

Канал "скрепка" (рис. 3) разработан для технической поддержки ГО [4, 6]. Источник посылает сообщение в участок *1* линии, оно переходит в участок *2* линии и затем в участок *3*. Используя участки *1* и *2*, все источники вносят изменения в сообщение. На участке *3* всем устройствам виден результат внесенных изменений.

Для работы с ГО были разработаны и другие структуры каналов, например, фрактальный канал [4, 5].

4. ОПЕРАЦИЯ НАХОЖДЕНИЯ МАКСИМУМА

Будем передавать в канал целые положительные двоичные числа, начиная со старшего разряда. Устройство, к которому из канала поступает старший разряд, совершает следующие действия. Если старший разряд числа в устройстве равен 0, то устройство лишь проверяет значение пришедшего из канала разряда. При этом, если пришла 1, то данное устройство прекращает участие в

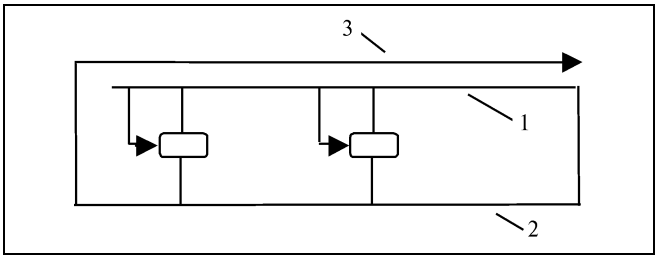

Рис. 3. Канал типа "скрепка"

операции, так как ясно, что его число меньше находящегося в ГК. Если же пришел 0, то устройство ожидает прихода следующего разряда и с ним начинает аналогичную проверку. Если в устройстве старший разряд его числа равен 1, то устройство, *не проверяя значение приходящего бита,* заменяет его своей единицей. Далее, если к устройству поступил 0, то оно заменяет остаток числа в команде на остаток своего числа. Если же поступила 1, то надо ждать прихода следующего разряда. Аналогично проверяются остальные разряды числа. Когда ГК обойдет все устройства, в ней будет находиться максимальное число из чисел, участвовавших в операции. Если применен канал "скрепка", то, наблюдая за участком *3* линии, все устройства увидят результат операции.

5. ПРИМЕНЕНИЕ ГРУППОВЫХ КОМАНД В MPI

Продемонстрируем на примере нескольких операций MPI применение в них групповых команд.

Рассылка данных. В ГК источник помещает данные, которые обходят все приемники, например, перемещаясь по каналу "скрепка".

Сбор данных. Используется ГК переменной длины. Группа источников один за другим помещают в команду без ее задержки свои данные.

Обмен данными в группе. В ГК посылают данные несколько источников. Каждая такая посылка начинается с указания имен (адресов) источника и приемников данного. Все приемники принимают из команды адресованные им данные.

Сдвиг. Источник принимает данные из канала и вместо них передает свои данные.

Счет. В ГК, в блок результата, каждое участвующее в операции устройство прибавляет к содержимому блока единицу, используя ГО-сложение. Операция *Счет* позволяет, например, проверить работоспособность всех устройств или установить факт получения всеми приемниками широковещательного сообщения.

Сложение целых двоичных чисел. Выполняется подобно операции *Счет*.

Другие операции редукции MPI (нахождение максимума (минимума), вычитание, умножение, логические операции) также выполняются при помощи ГО [4—6].

Не имеет смысла более подробно останавливаться на реализации операций MPI с помощью групповых операций, так как общая схема ясна, а детали здесь не существенны и могут быть реализованы разными способами. Вместо этого приведем три примера использования групповых операций в других приложениях, которые могут быть полезны, если потребуется расширить набор операций MPI.

Пример 1: *вычисление значения полинома.* Выполним в пределах единственной ГК сложное распределенное вычисление — нахождение значения полинома

$$y = a_0 + a_1x + a_2x^2 + ... + a_{n-1}x^{n-1} + a_nx^n.$$

Коэффициенты $a_1, a_2, ..., a_n$ распределены по устройствам, вычисляющим значение y.

Инициатор вычисления посылает ГК вида:

<Получить значение полинома
<φ = 1><*n*><*x*><*R*><R^+><*R*−> >.

В скобках находится код ГК, начинающийся с кода операции. Далее, также в скобках находятся операнды: флаг φ, степень полинома n, переменная x. В конце команды расположены три блока результата R, R^+ и R^-, содержащие одноименные переменные, которые имеют следующие начальные значения:

$R = 1$;

$R^+ = a_0$; $R^- = 0$, при $a_0 \geqslant 0$;

$R^+ = 0$; $R^- = |a_0|$, при $a_0 < 0$.

Команду выполняют устройства, у которых флаг φ равен 1.

Устройство, содержащее коэффициент a_i, должно получить произведение a_ix^i и до прихода к нему блоков R^+ и R^- решить, к какой из переменных R^+ или R^- надо прибавить это произведение. Для этого достаточно учесть значение i и знаки чисел x и a_i, которые до начала умножения известны.

Первое устройство с флагом φ = 1, находящееся на пути перемещения ГК, выполняет следующие действия.

- С помощью групповой операции без задержки x умножается на число R и результат помещается в блок R групповой команды. Теперь значение $R = x$.
- Внутренними средствами устройство выполняет умножение числа $R = x$ на хранящееся у него значение a_1. Умножение должно быть завершено до прихода из канала блоков R^+ и R^- групповой команды.
- Если $a_1x \geqslant 0$, то при помощи ГО это число прибавляется к числу R^+, иначе его модуль прибавляется к числу R^-.

Второе устройство с флагом φ = 1, находящееся на пути перемещения ГК, действует аналогично первому. Но к устройству поступает число $R = x$ и формируется $R = x^2$.

Так же будут действовать остальные устройства.

Если бы все коэффициенты a_i и значения x были положительны, то в конце вычисления будет получено $R^+ = y$. В общем случае части результата будут находиться в блоках R^+ и R^-. Окончательный результат будет получен алгебраическим сложением чисел R^+ и R^- внутренними средствами процессора.

Приведенный пример показывает, что сложные распределенные вычисления можно быстро выполнить с помощью единственной ГК.

Время выполнения изложенного вычисления не зависит от числа устройств.

Пример 2: *обнаружение искаженных широковещательных сообщений и команд.* Пусть каждое широковещательное сообщение завершается квитанцией, направленной источнику сообщения. Источник знает контрольную сумму посылаемого им сообщения, число получателей сообщения и умеет перемножать эти числа. Каждый приемник сообщения формирует контрольную сумму принятого им сообщения. Эти числа приемники с помощью ГО прибавляют к уже имеющемуся в квитанции

числу. Источник сравнивает значение полученной из квитанции суммы и вычисленной им, что позволяет проверить корректность приема сообщения всеми приемниками. Такой способ позволяет также обнаруживать появление несанкционированного сообщения от другого источника.

Пример 3: *устранение конфликтов на стороне приемника*. Пусть в системе устройства-заказчики работ распределяют эти работы среди устройств-исполнителей. При этом возможны конфликты — одному исполнителю будет адресовано несколько заявок. Заказчики выполняют следующую процедуру устранения конфликта. Перенумеруем исполнителей. Один из заказчиков получает право послать ГК для выбора исполнителя. В ней имеются пары блоков, содержащих имя исполнителя и соответствующий имени счетчик. Если какой-либо заказчик не обнаруживает в указанной команде имя требуемого ему исполнителя, то он создает блок с требуемым именем и блок-счетчик, в который заказчик записывает единицу. Если же требуемое имя обнаружено, то заказчик увеличивает на единицу число в счетчике. Заказчик запоминает сформированное им число — порядковый номер своего заказа. Эти номера определяют порядок обслуживания заказов. Задача обеспечения равномерной загрузки исполнителей, рассмотрена в работе [4].

ЗАКЛЮЧЕНИЕ

Добавим несколько слов о техническом обеспечении рассмотренных методов. Все необходимые средства могут быть выполнены в виде модулей, вставляемых, подобно картам Ethernet, в слоты материнских плат компьютеров. Сложность этих модулей сравнима со сложностью карт Ethernet. В отличие от Ethernet здесь надо применить другие протоколы, ориентированные на короткие сообщения и быстрое взаимодействие многих устройств.

ЛИТЕРАТУРА

1. *MPI-2* (пер. на рус. язык) http://www.cluster.bsu.by/download/MPI-2_ruc.zip
2. *Корнеев В. Д.* Параллельное программирование в MPI. — М.: Институт компьютерных исследований, 2003. — 303 с.
3. *An Overview* of the BlueGene/L Supercomputer The BlueGene/L http://sc-2002.org/paperpdfs/pap.pap207.pdf
4. *Стецюра Г. Г.* Методы совмещения вычислений и передачи данных в многопроцессорных системах и локальных сетях. — М.: ИПУ, 2005. — 86 с.
5. *Стецюра Г. Г.* Возможности применения фрактальных связей и групповых операций в многопроцессорных системах с перестраиваемой структурой для эволюционных вычислений //Автоматика и телемеханика. — 2003. — № 12. — С. 164—176.
6. *Прангишвили И. В., Подлазов В. С., Стецюра Г. Г.* Локальные микропроцессорные вычислительные сети. — М.: Наука, 1984. — 176 с.

☎ *(095) 334-78-31*

E-mail: stetsura@ipu.ru ❑

КОНФЕРЕНЦИЯ ВО ФЛОРИДЕ

С 6 по 8 апреля 2005 г. в университетском городке Гейнесвилле, США, штат Флорида, проходила международная конференция "Risk Management and Quantitative Approaches in Finance". Организатором выступил Университет штата Флорида. Сделанные доклады были посвящены последним достижениям финансовой математики и риск-менеджмента — методам оптимизации портфелей, управлению активами и пассивами, вопросам ценообразования производных финансовых инструментов.

Порадовало присутствие на конференции многих известных математиков, в частности Р. Калмана, Т. Рокафеллара и Б. Зиемба. Помимо автора заметки Россия была представлена бывшими соотечественниками, а ныне американскими профессорами С. Урясьевым, М. Забаранкиным и В. Холодным. Первый из названных является сейчас одним из признанных мировых авторитетов в области риск-менеджмента и фактическим организатором конференции.

Очень приятное впечатление оставили любезные и улыбчивые жители городка, а также его девственная природа. Особую пикантность кампусу придают плакаты типа "Alligator feeding is a criminal offence!", расположенные возле каждого из многочисленных прудов и озер..

Более подробную информацию о конференции можно получить на странице
http://www.ise.ufl.edu/rmfe/events/qf2005/
В следующем, 2006, году Университет Флориды
планирует проведение конференции по финансовой инженерии
(см. http://www.ise.ufl.edu/rmfe/events/qf2006/index.htm)

Д. Ю. Голембиовский

☎ **(095) 937-07-37, доб. 26-83**
E-mail: d.golembiovsky@zenit.ru