УДК 004.75

КОМПЬЮТЕРНАЯ СЕТЬ С БЫСТРОЙ РАСПРЕДЕЛЕННОЙ ПЕРЕСТРОЙКОЙ СВОЕЙ СТРУКТУРЫ И ОБРАБОТКОЙ ДАННЫХ В ПРОЦЕССЕ ИХ ПЕРЕДАЧИ

Г.Г. Стецюра

Рассмотрена компьютерная сеть с расширенными функциональными возможностями. Сеть полносвязная, но формируются только связи, требуемые в текущий момент. Связи быстро перестраиваются, что позволяет многократно изменять их структуру в пределах решаемой задачи. Показано, что это существенно облегчает и ускоряет выполнение массовых операций с одновременным участием многих устройств системы. Сеть позволяет, используя только сетевые средства, выполнять ряд видов распределенных вычислений над данными непосредственно в процессе их передачи по сети. Приведены примеры применения сети и технической реализации ее основных компонентов.

Ключевые слова: беспроводная оптическая сеть, динамическая реконфигурация, вычисления средствами сети, распределенная синхронизация, барьерная синхронизация.

ВВЕДЕНИЕ

В статье рассмотрена разработанная в Институте проблем управления им. В.А. Трапезникова РАН компьютерная сеть, ориентированная на применение в компьютерных системах, компоненты которых разнесены на небольшие расстояния, например, в системах вычислительного центра. Свойства сети отличают ее от известных сетей, они перечислены далее в § 1. Остановимся на двух отличиях, влияющих не только на функционирование сети, но также на конструирование прикладных алгоритмов и программ, на управление системой. Одно из них состоит в том, что все связи в сети могут быть изменены одновременно за время выполнения типичной команды компьютера (в наносекундном диапазоне). Так же быстро сеть может быть разделена на произвольные, независимо работающие подсети. У создателей прикладных алгоритмов и программ появляется дополнительная степень свободы — они могут в динамике управлять структурой связей компьютерной системы, подстраивая ее под текущие требования решаемой задачи. В существующих универсальных компьютерных системах отсутствует возможность быстрой подстройки сети под требования задачи, и в алгоритме решения задачи необходимо дополнительно учитывать ограничения, налагаемые фиксированной сетью системы.

Другое отличие — совмещение передачи сообщений с их обработкой. В статье показано, что многие вычислительные операции и управление поведением системы могут выполняться непосредственно сетевыми средствами. При этом достигается существенное по сравнению с известной практикой ускорение таких операций.

Обычно функции средств компьютерной системы четко разделены. Основная задача сети состоит в передаче сообщений, используется режим коммутации сообщений и выполняются операции, необходимые только для передачи, обработка данных выполняется вне сети.

Совокупность предлагаемых возможностей влияет на функционирование использующей сеть компьютерной системы и позволяет сетевым средствам выполнять распределенные вычисления, уп-

рощает децентрализацию управления работой системы, ускоряет диагностику состояния сети и системы в целом.

Новые свойства сети не удается реализовать только электронными средствами, поэтому в сети применяются известные из литературы оптоэлектронные компоненты в новом их сочетании. Сведения об этих средствах приведены далее в § 6.

Цель статьи, не углубляясь в технические детали и ограничиваясь только самой необходимой информацией, показать возможности предлагаемой сети, ускоряющие обработку данных и управление работой распределенной компьютерной системы. Подстройка структуры сети под текущие требования решаемой задачи упрощает способы создания прикладных алгоритмов, программ и системного программного обеспечения.

В статье даны ссылки на публикации и патенты автора, содержащие технические и функциональные детали, включен ряд новых результатов. Приведены ссылки на публикации по техническим средствам других авторов. Публикации с аналогичными структурными и функциональными решениями других авторов не обнаружены, хотя вопросами реконфигурации сетевых связей в цифровых системах занимаются активно. Покажем различие в решениях на широко известном примере — Software-defined network (SDN). В сети SDN, ориентированной в основном на применение в Интернете, задачи передачи данных и управления коммутацией разделены, для более гибкой коммутации все управление концентрируется в центре управления и выполняется специальным программным обеспечением (см. подробный обзор в работе [1]).

Таким образом, сеть SDN и рассмотренная в настоящей статье сеть не только относятся к разным областям применения, но и организованы различно. В сети из статьи для достижения высокого быстродействия применено распределенное управление, в сети SDN — централизованное. Для формирования быстрых связей в статье невозможно использовать работу управляющих программ, и все связи могут быть установлены одновременно одной командой, сеть не программно-, а командно-управляема. Кроме этого, сетевые средства новой сети также осуществляют многие виды обработки данных и управления.

Структура статьи: в § 1 представлена полная сводка особенностей предлагаемой сети, в § 2—5 излагается организация сети и базовые операции, без которых реализация всего сказанного выше невозможна. Специфика этих операций — время выполнения операции не зависит от числа узлов сети, одновременно участвующих в операции. В § 6 уделено внимание реализации технических средств сети. Наконец, в § 7 рассмотрены примеры использования операций сети. В Заключении изложены наиболее важные результаты.

1. ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ СЕТИ

Представим полную сводку особенностей сети.

1. Сеть объединяет полносвязной структурой связей большое число объектов — устройств системы, выполняющих обработку и хранение данных.

2. Реализуются только те связи, которые необходимы в текущий момент времени. Средства коммутации находятся непосредственно в источнике или приемнике данных. Изменение структуры связей выполнимо в наносекундном диапазоне. Таким образом, объекты соединяются непосредственно, а структура связей способна изменяться не только от программы к программе, но и за время выполнения отдельной команды программы.

3. Непосредственное соединение объектов позволяет выполнять распределенную коммутацию каналов, закрепляя соединение между объектами на произвольный отрезок времени. Упрощение процесса соединения позволяет обмениваться короткими сообщениями.

4. Особенности п. 1—3 позволяют адаптировать структуру физических связей в системе под структуру виртуальных (логических) связей решаемой задачи, исключая появление длинных цепочек связей через звенья коммутации.

5. Выполняется быстрая синхронизация распределенных в сети источников сообщений, позволяющая приемнику получать от них сообщения одновременно или одно за другим без временных пауз между сообщениями.

6. Посылка сообщения одновременно группе приемников незначительно отличается по сложности и времени исполнения от посылки сообщения одному приемнику.

7. Сообщения могут конфликтовать только на входе в приемник; такие конфликты устраняются быстро.

8. При передаче сообщений сетевые средства могут над их содержимым выполнять вычисления без затраты дополнительного времени на проведение вычисления. Таким образом, в системе, использующей рассматриваемую сеть, нет полного разделения на коммутационные и вычислительные средства.

9. Имеются средства быстрого одновременного оповещения всех объектов об их текущем состоянии.

2. СТРУКТУРА СЕТЕВЫХ СВЯЗЕЙ И КОМПОНЕНТЫ ОПТИЧЕСКОЙ СЕТИ

Организация структуры оптических связей между объектами сети представлена на рис. 1. Здесь показаны компоненты сети: множество объектов и множество модулей связи, один из которых действует как системный информатор [2].

Объект выполняет внутренние действия (вычисления, хранение данных), требуемые решаемой объектом задачей. Он также выполняет описанные далее действия по организации взаимодействия узлов сети.

Каждому объекту ставится в соответствие модуль связи, которому объект непрерывно посылает оптический сигнал частоты f_1 (на рис. 1 показано сплошной линией). Таким образом, эта пара — объект и модуль — представляет собой единое устройство, содержащее два разнесенных в пространстве компонента.

Модуль связи получает сигналы от объекта и отражает без задержки каждый поступающий сигнал его источнику. Кроме этого, также без задержки модуль выполняет следующие действия. Объект — источник сообщения посылает в модуль связи приемника сообщение оптическими сигналами частоты f_2 (штриховая линия). Если за этим модулем наблюдает соответствующий модулю объект — приемник и другие объекты, также посылающие в модуль сигнал f_1, то модуль связи без задержки возвращает всем этим объектам их сигнал f_1, модулированный сигналами f_2 сообщения (штрихпунктир). Таким образом, модуль *не создает* новые сигналы, для связи между объектами он использует только сигналы объектов.

Объект посылает модулям связи оптические сигналы, с помощью содержащегося в объекте демультиплексора, который посылает сигналы любому модулю, одновременно группе модулей или всем модулям сети, информатору. Переключение направления передачи сигналов выполняется быстро, в наносекундном диапазоне.

Используются сигналы — импульсный, длящийся известное всем компонентам сети время, и непрерывный, длительность которого переменная и определяется источником сигнала.

Объект также посылает модулю оптический сигнал $*f$, наличие которого запретит возврат объектам сигнала f_1. Модуль имеет элемент памяти. Объект посылает модулю оптические сигналы $*f_1$ и $*f_2$ для перевода элемента памяти в состояния «включен/выключен» соответственно. В состоянии

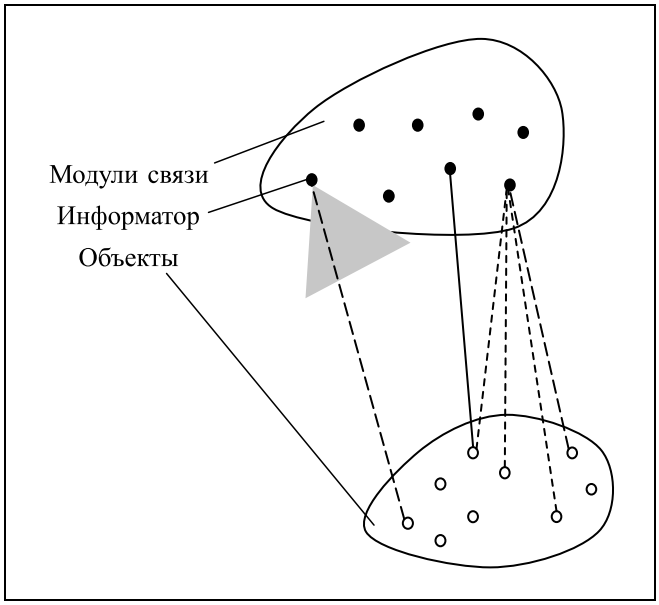

Рис. 1. Структура связей

«включен» модулю запрещен возврат полученного от объектов сигнала f_1.

Объект — приемник для передачи сообщения источнику должен действовать подобно источнику, но ему достаточно также посылать сообщение только своему модулю, за которым наблюдает источник.

Информатор отличается от модуля тем, что он при получении сигнала одного типа (выше это сигнал f_2) создает характерный только для информатора ненаправленный сигнал f_{si} и посылает его всем объектам сети (на рис. 1 это показано треугольником). При получении сигналов $*f$, $*f_1$ и $*f_2$ информатор не передает сигналы f_{si}.

Информатор позволяет изменять структуру связей посылкой соответствующей команды всем объектам одновременно. Имеются задачи, для решения которых полезно иметь несколько информаторов.

Показанное на рис. 1 установление непосредственной связи между объектами позволяет предоставлять им соединение на произвольный отрезок времени, т. е. *выполняется распределенная коммутация каналов*.

Здесь и далее будем обозначать объект с номером i как O_i, модуль связи с номером i обозначим как MS_i, информатор обозначим как SI. Системный информатор обладает многими возможностями модуля связи, и, если не потребуется учитывать особенности информатора SI, то будем говорить о нем как о модуле связи.

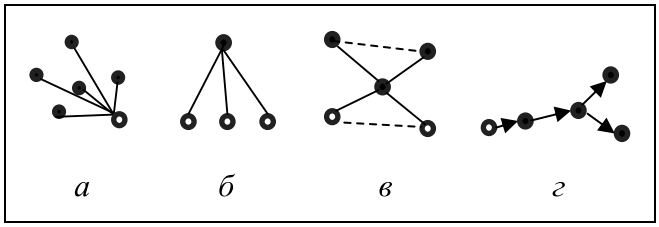

Рис. 2. Виды связей объектов сети (пояснения в тексте)

На рис. 2 приведены возможные виды связи объектов сети. Для упрощения показаны только ретрансляторы приемников (залитые кружки), которые должны быть размещены между источником и приемником.

На рис. 2, *а* источник (незалитый кружок) посылает сообщение произвольно выбранной группе приемников. На рис. 2, *б* произвольная группа источников посылает сообщения единственному приемнику. На рис. 2, *в* группа источников, используя ретранслятор объекта-посредника (или отдельный ретранслятор), посылает сообщение группе приемников. На рис. 2, *г* объект порождает цепочку или дерево связей между объектами. В цепочке логические соседи не обязательно будут физическими соседями, но связи между ними непосредственные.

Для связи (рис. 2, *а*) действует ограничение: к приведенной группе приемников может одновременно обращаться только один источник. Для связи (рис. 2, *б*) разрешено обращение к приемнику группы источников, существует способ разрешения конфликта доступа и синхронизация передач источников. Для связи (рис. 2, *в*) устраняется конфликт доступа источников к посреднику, после чего ему синхронно передаются сообщения источников. Приемники наблюдают за посредником и получают сообщения источников.

3. СИНХРОНИЗАЦИЯ ПОСЫЛКИ ИСТОЧНИКАМИ СИГНАЛОВ МОДУЛЮ

Приведенная здесь синхронизация в ответ на приходящий от модуля *MS* сигнал начала синхронизации обеспечивает приход сигналов разных источников в произвольный модуль *MS* одновременно [2, 3]. Обозначим через T_{ij} время прохождения сигнала в цепочке «объект O_i, любой модуль MS_j, объект O_j». Способы определения времени доставки сигнала от источника приемнику известны, не будем на них останавливаться и положим, что источники знают времена доставки сигнала каждому приемнику.

Для синхронизации произвольный объект O_i посылает свой сигнал в модуль MS_j с задержкой ${}^*T_i = T_{\max} - T_{ij}$, где $T_{\max} \geq \max T_{ij}$. Тогда сигналы всех объектов, действующих аналогично, поступят в модуль MS_j одновременно, с единой задержкой $T_{\max}$. Если передаются сообщения, то их одноименные разряды совместятся и представят собой единое сообщение.

Если требуется, чтобы сигналы или сообщения поступали в модуль *MS* одно за другим без временных пауз, как одно сообщение, то каждый объект O_i должен передать свое сообщение с задержкой $T_{\max} - T_{ij} + Q$, где Q — суммарная длительность сообщений, переданных объектами ранее объекта O_i.

Для медленных сетей можно выбрать задержку $T_{\max} + Q$, оставляя в силе изложенные в данной статье результаты, но в быстрых сетях это ведет к существенному уменьшению пропускной способности.

4. УСТРАНЕНИЕ КОНФЛИКТА ДОСТУПА ИСТОЧНИКОВ К МОДУЛЮ СВЯЗИ

Если источники посылают сигналы в модуль связи *MS*, не согласовав их отправку, то возникает конфликт на входе в модуль, и необходимы способы его устранения.

Способ фиксированной шкалы. В соответствии с этим способом источники используют момент обнаружения конфликта как приход синхронизующей команды от модуля *MS* и передают в модуль *MS* сообщение — логическую шкалу, представляющую собой последовательность двоичных разрядов. Каждому источнику, имеющему право посылать сообщение в данный модуль, поставлен в соответствие один из разрядов шкалы. Конфликтующий источник ставит в свой разряд единицу, остальные разряды содержат нули. Логические шкалы поступают в модуль *MS* одновременно, и объединенная шкала возвращается ко всем конфликтующим источникам. Источники определяют порядковый номер своей передачи и затем передают последовательно сообщения как единое сообщение без временных пауз между его частями. Конфликт устранен.

В ряде случаев следует отказаться от шкал с фиксированным числом разрядов. Например, к приемнику может обратиться заранее неизвестное число источников. В этих случаях применимы следующие способы использования шкалы с элементом случайности.

Приоритетный способ. Источникам присваиваются различающиеся между собой двоичные коды

приоритета. Источники формируют шкалу, в которой источник случайно выбирает разряд шкалы и вносит в него значение старшего разряда своего кода приоритета. Шкалы синхронно посылаются в модуль *MS* и возвращаются источникам. Далее применяется известный способ устранения конфликта [3]. Источники, пославшие в разряд значение «ноль», прекращают борьбу за право передачи сообщения, если в этом разряде обнаружен единичный сигнал, посланный другими источниками. Оставшиеся источники аналогично действуют со следующим разрядом кода приоритета и т. д., до исчерпания всех его разрядов. В результате в каждом из участвующих в борьбе разряде шкалы останется по одному источнику. Эти источники учтут состояние других разрядов шкалы и бесконфликтно передадут сообщения, как в способе фиксированной шкалы.

Случайный способ. Разрядность шкалы выбирается достаточно большой, чтобы была мала вероятность выбора одинакового разряда более чем одним источником. Источник случайно выбирает разряд шкалы и записывает в него единицу. С этой шкалой выполняются такие же действия, как в способе фиксированной шкалы. При указанном условии высока вероятность бесконфликтной передачи сообщений источников.

При объединении двух последних способов из случайной шкалы вначале создается сжатая шкала, в которой учитываются только единичные разряды, затем для этой шкалы применяется приоритетный способ.

5. РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ, ГРУППОВЫЕ КОМАНДЫ

5.1. Распределенные вычисления

Виды распределенных вычислений. Рассматриваются два вида распределенных вычислений, которые выполняют сетевые средства над содержимым передаваемых по сети сообщений.

Вид 1. Такие операции как логическая сумма, логическое произведение, определение максимума или минимума выполняются сетевыми средствами над сообщениями, передаваемыми в сеть большими группами объектов одновременно. Полученный результат доступен всем объектам также одновременно.

Вид 2. Вычисления над числом, находящимся в сообщении, которое проходит через цепочку последовательно соединенных объектов. При этом выполнение вычисления не вносит задержку в передачу сообщения. В каждом вычислении участвуют находящееся в сообщении число и число, хранящееся в объекте цепочки, к которому поступает сообщение. Выполняются все логические операции, нахождение max и min, арифметические сложение, вычитание и умножение.

Выполнение операций вида 1. Группа источников синхронно посылает сообщения в выбранный ими модуль *MS* так, что они накладываются друг на друга поразрядно, формируя общее сообщение. Разряды представлены активными сигналами, например, каждый разряд в сообщении представлен двумя битами — 10 для единицы и 01 для нуля. Все приемники, наблюдающие за этим модулем, получат отраженное им сообщение. Если выполняется логическое сложение, то полученные при наложении разрядов пары 10 и 11 считаются единицей. Для логического умножения единицей будет наложение разрядов пары 10 и 10.

При определении максимума или минимума группа источников посылает сообщения в модуль *MS*, как в предыдущем случае, но в сообщениях не требуется выделять для каждого разряда сообщения два бита. В числах, передаваемых в модуль *MS* источниками, каждый разряд выделяется для отдельной подгруппы из группы источников. Источники всех подгрупп одновременно находят числа с максимальным (минимальным) значением в подгруппе.

Для этого источники подгруппы посылают сообщения, содержащие числа, в модуль *MS* так, чтобы одноименные разряды чисел совместились. Вначале источники посылают в модуль *MS* старший разряд сравниваемых чисел. Модуль *MS* возвращает сигналы источникам, и если источник, пославший в модуль ноль, получает от него единицу, то он прекращает использовать далее этот разряд. Такая операция продолжается для всех остальных разрядов сравниваемых чисел. В итоге одновременно для всех подгрупп будут выявлены максимальные из посланных их источниками чисел. Приведенные действия не отличаются от действий со шкалой в приоритетном способе (см. § 4). При инверсии представлений сигналов «единица» и ноль аналогичным способом находится минимум.

Таким образом, результат операций вида 1 создается модулем *MS* без участия вычислительных средств объектов, а время выполнения операции не зависит от числа объектов, участвующих в операции. Результат операции поступает параллельно ко всем устройствам.

Расширение возможностей операций вида 1. Пусть объекты синхронно передают в один из модулей связи сообщения не одновременно, а одно за другим, но без временных пауз между сообщениями. Модуль возвращает сообщения объектам, которые выполняют их обработку доступным им способом.

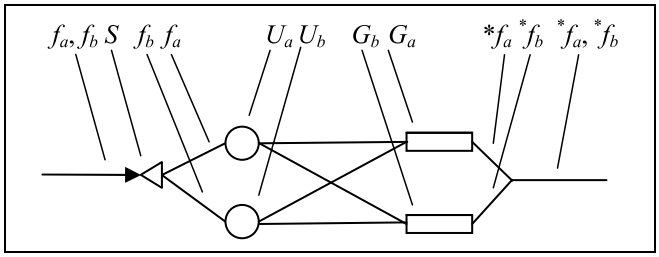

Рис. 3. Вычисление в цепочке объектов

Хотя здесь время доставки сообщений и зависит от числа объектов, но при коротких сообщениях в рассматриваемой распределенной системе время выполнения операции будет зависеть в основном не от числа сообщений, а от расстояния между объектами. Положительный эффект — снимается ограничение на состав операций вида 1.

Выполнение операций вида 2. Пусть группа объектов соединена в цепочку: первый объект направляет сообщение в *MS* второго объекта, второй объект, получив это сообщение, направляет его в модуль *MS* третьего объекта и т. д. Двоичные разряды чисел в передаваемом сообщении представлены сигналами двух видов: f_a для значения 1 и f_b для значения 0. Все дальнейшие действия выполняются без задержки поступивших в объект сигналов, как показано на рис. 3. Здесь в объект поступает сигнал f_a или f_b. Разделитель S направляет поступивший сигнал, в зависимости от его значения, по одному из двух указанных на рис. 3 путей. На выходе из этих путей расположены управляемые объектом переключатели U_a и U_b. Для выполнения вычисления объект *до прихода в сообщении очередного бита* обрабатываемого двоичного числа устанавливает переключатели в требуемое объекту состояние. Объект для этого использует только значение, хранящегося в нем числа, и информацию о виде операции. После этого поступает входной сигнал, который, пройдя переключатель, поступит на один из двух источников сигналов G_a, или G_b, создающих сигналы $^*f_a$ или $^*f_b$ соответственно. Сигнал, пришедший в источник, включает его, и созданный источником сигнал направляется следующему объекту в цепочке.

Объекты с помощью переключателей U_a и U_b выполняют четыре действия.

Действие M_1: при приходе сигнала f_a (f_b) он переводится на выходе в сигнал $^*f_b$ ($^*f_a$).

Действие M_2: сигналы f_a и f_b переводятся в сигнал $^*f_a$.

Действие M_3: сигналы f_a и f_b переводятся в сигналы $^*f_b$.

Действие M_4: сигналы f_a и f_b переводятся в сигналы $^*f_a$ и $^*f_b$ соответственно.

Эти действия не анализируют значение приходящего сигнала, и поэтому результат действия появляется на выходе объекта без временной задержки на анализ.

Действий M_1—M_4 достаточно для выполнения указанных операций над числом в сообщении и числом в объекте [2, 3]. Примеры выполнения распределенных вычислений без задержки приведены в работе [4].

Выдача сигналов $^*f_a$ и $^*f_b$ вместо сигналов f_a и соответственно f_b объясняется тем, что модуль *MS*, получая от источника сообщение сигналами одного типа (f_2), передаст его приемнику сигналами другого типа (f_1). Поэтому на рис. 3 на входе и выходе должны быть разные типы сигналов: $^*f_a$ вместо f_a и $^*f_b$ вместо f_b. Их создают источники сигналов G_a и G_b соответственно.

Подчеркнем, что использование пар сигналов f_a и f_b для вычислений в цепочке потребовало вместо указанных выше одиночных сигналов f_1 и f_2 ввести соответствующие им пары сигналов.

Приведем два простых примера, показывающих отсутствие задержки в операциях вида 2.

Пусть объектам цепочки требуется выполнить поразрядное логическое умножение. Каждый объект до начала операции анализирует значения разрядов хранящегося в нем числа и подготавливает следующие действия. Если разряд числа имеет значение 0, то следует выполнять действие M_3, иначе M_4. Затем передается сообщение, и время его передачи с выполнением в цепочке логических умножений не отличается от времени передачи без вычислений.

Теперь выполним сложение текущих разрядов двоичного числа, хранящегося в объекте и числа из приходящего в объект сообщения. Объект, с учетом значений переноса из предыдущего разряда и текущего разряда числа в объекте, выбирает следующие действия. Если объекту требуется выполнить сложение с нулем, то он выбирает действие M_4, иначе выбирается действие M_1. Так как каждый объект цепочки выбирает действия до прихода к нему разряда числа независимо от действий других объектов, то времена переключений в цепочке объектов не суммируются.

5.2. Групповые команды

В групповых командах используются вычисления по п. 5.1. Групповая команда (ГК) — это сообщение, которое перемещается через цепочку объ-

ектов и содержит данные и инструкции объектам по обработке находящихся в ГК данных, изменению содержащихся в ГК инструкций, выполнению локальных действий в объекте [3]. Для изменения в ГК ее содержимого объект, проанализировав часть проходящей через него ГК, заменяет оставшуюся часть ГК своей информацией *без задержки* ГК на это преобразование. В результате ГК, перемещаясь через цепочку объектов, собирает по пути информацию об объектах и изменяется сама. Ее влияние на объект зависит от действий предшественников объекта в цепочке. Сообщение может быть также групповой программой, состоящей из последовательности групповых команд, сформированной несколькими объектами.

При выполнении вычислительных операций вида 1 возможны групповые команды, которые как единая команда приходят ко многим объектам одновременно.

6. О РЕАЛИЗАЦИИ ТЕХНИЧЕСКИХ СРЕДСТВ СЕТИ

Для реализации предложенных в настоящей статье решений требуется разработка компонентов, выполняющих функции демультиплексора, ретранслятора и информатора. Анализ литературы показал наличие устройств, близких к требуемым. Рассмотрим эти устройства.

Демультиплексор — находится в каждом объекте и соединяет объект с любым модулем *MS* системы или одновременно с произвольной группой модулей. Требуемая организация такого демультиплексора на основе решетки управляемых лазеров описана в работах [5, 6]. В работах [7, 8] даны примеры реализации решеток лазеров.

Модуль связи. Основные составляющие модуля — ретрорефлектор, модуляторы света, фотоприемники [2, 9, 10]. В качестве примера работы, где применены все эти компоненты, приведем упрощенно результаты из статьи [11] (рис. 4).

Сигнал *1* фокусируется на зеркале 4 и возвращается к источнику. Сигналы *1* от других таких же источников попадут на другие участки зеркала и также возвратятся к источникам. На прямом и обратном пути каждый сигнал *1* проходит через соответствующий ему элемент *3*, используемый как фотодетектор при приеме сигнала и как модулятор света при возврате сигнала источнику. На прямом пути сигнал *1* несет сообщение источника, принимаемое фотодетектором *3*. После передачи сообщения источник, используя *1*, передает непрерывный сигнал *1*. Этот сигнал приемник модулирует электрическими сигналами, действующими на элемент *3* — модулятор, и такое сообщение возвращается источнику. Каждому источнику выделен отдельный элемент *3*, и устройство, показанное на рис. 4, имеет независимо работающие каналы для связи с каждым источником сигналов *1*. В настоящее время появились близкие по подходу работы других авторов.

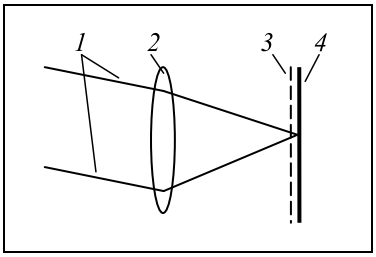

Рис. 4. Модуль связи с ретрорефлектором: *1* — световой сигнал от удаленного лазерного источника; *2* — линза; *3* — плоскость с многими модуляторами/фотоприемниками; *4* — фокальная плоскость линзы, в которой расположено зеркало; элементы *2* и *4* составляют ретрорефлектор типа «кошачий глаз»

В предлагаемом модуле связи требуются аналогичные компоненты с некоторыми изменениями:

— упрощение: модулятор/фотоприемник *3* должен быть общим для всех объектов;

— усложнение: модулятор должен быть избирательным по частоте. Однако можно ограничиться средствами из работы [11]: устройство (см. рис. 4) позволяет разделить поток после линзы на два потока и направить их на два комплекта компонентов *3* и *4*.

Информатор. В информаторе *SI* новым по сравнению с компонентами рассмотренных устройств является источник модулированных ненаправленных оптических сигналов. Известны появившиеся в последнее время разработки таких устройств, передающих сигналы с частотами модуляции свыше 10 ГГц [12].

Отметим, что появились публикации, посвященные созданию систем на кристалле с применением оптических беспроводных соединений устройств системы [13]. Некоторые из рассмотренных решений применимы и в таких системах.

В настоящей статье изложение построено в предположении, что сигналы передаются в пределах помещения вычислительного центра. Однако, подобно работе [13], для обмена сигналами можно создать пылезащитную конструкцию, в которую помещены оптические компоненты демультиплексоров и модулей связи.

7. ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ОПЕРАЦИЙ СЕТИ

Барьерная синхронизация. Эта обычно длительная операция быстро решается с использованием изложенных выше результатов. Рассмотрим один

из ее вариантов. Пусть все источники группы P должны передать сообщения группе приемников после того, как все приемники будут готовы к приему сообщений. Для синхронизации в группе P источников выделяется один ее представитель O_p. Его модуль MS_p становится известным всем участвующим в барьерной синхронизации источникам и приемникам, которые начинают следить за модулем MS_p, посылая ему сигналы f_1 (в качестве модуля MS_p может быть взят свободный модуль, не связанный с объектом). После этого источник O_p посылает всем объектам, наблюдающим за модулем MS_p, сообщение о начале барьерной синхронизации, в ответ на которое все приемники синхронно посылают в модуль MS_p сигнал $^*f_1$, запрещающий ему возврат сигналов f_1. Готовые к приему сообщения объекты через фиксированный интервал времени T_b пошлют в модуль MS_p сигнал $^*f_2$, и их запрет будет снят. Остальные приемники будут посылать сигнал $^*f_2$ по мере их готовности к приему сообщений. После готовности всех источников модуль MS_p начнет возвращать сигнал f_1, что будет признаком завершения подготовки к передаче сообщений. Получив сигнал f_1, объекты передадут сообщения синхронно с задержками $^*T_i$ в модуль MS_p, и все приемники получат его как единое сообщение.

Использование сигналов $^*f_1$ и $^*f_2$ вместо сигнала $^*f$, (см. п. 2.1) избавляет приемник от необходимости непрерывной передачи сигнала в модуль MS_p.

Интервал T_b вспомогательный, он гарантирует источнику, что все приемники участвуют в барьерной синхронизации (квитирование).

Аналогично определяется готовность источников к передаче сообщения.

Сетевые коллективные взаимодействия в MPI. Эту важную тему затронем кратко. Коллективные взаимодействия, в которых одновременно участвуют группы процессов, широко применяются в интерфейсе передачи сообщений MPI, но они требуют значительных затрат времени. Здесь покажем, что предлагаемые сетевые решения существенно ускоряют эти взаимодействия.

В MPI коллективные взаимодействия выполняют три вида функций. Первый вид — функции, выполняющие коллективный обмен сообщениями и синхронизирующие взаимодействие процессов. Примеры таких функций: MPI_BCAST — передача сообщения от одного всем; MPI_ALLTOALL — передача от всех всем; MPI_BARRIER — барьерная синхронизация. Второй вид — функции редукции, выполняющие распределенные вычисления с участием групп процессов. Это функции MPI: max, min, sum, maxloc и др. Третий вид — функции, отображающие виртуальную (логическую) структуру связей в задаче на реальную структуру системы, например, MPI_GRAPH_CREATE — создать граф связей.

Непосредственное применение изложенных выше операций синхронизации, барьерной синхронизации и двух видов вычислений (см. § 5) существенно ускоряют все три вида операций. В значительной степени действия выполняются с помощью аппаратных средств сетевых контроллеров объектов. По поводу операций третьего типа заметим, что при использовании рассматриваемой сети виртуальные соседи всегда соединяются короткими физическими связями. Установленная связь сохраняется в течение произвольного отрезка времени.

Эволюционные вычисления. Во многих эволюционных алгоритмах объекты группы, действуя параллельно, находят частные варианты решения, которые требуется сравнить между собой для выявления решения, имеющего максимальное значение. Часто после этого объектам требуется выделить частные решения, близкие к наилучшему решению, и сравнить структуры этих решений. Операции сравнения распределенных решений медленные. Рассмотренные в § 5 способы параллельных распределенных вычислений позволяют ускорить операции сравнения. Так как на получение частных решений разным объектам может потребоваться разное время, то начало операций сравнения требуется синхронизовать. Для синхронизации применяется предложенная барьерная синхронизация.

Борьба с повреждениями сети и системы. Рассмотрены два вопроса: устранение повреждений сети, нарушающих ее целостность и информирование о количестве и расположении неисправных объектов системы. Единственный вид компонентов сети, повреждения которых влияют на ее целостность, это *MS*. Объект — приемник, обнаружив отказ используемого им модуля, занимает запасной модуль, проводя борьбу за модуль с другими объектами. Если отказов больше, чем имеется запасных модулей *MS*, то объект будет подключен к модулю, уже занятому другими объектами, и будет использовать модуль совместно с ними. Таким образом, при наличии хотя бы одного исправного *MS* объекты сохраняют возможность взаимодействия.

При выявлении неисправных объектов предполагается, что они проводят диагностические тесты и требуется выяснить, есть ли отрицательные результаты тестов и если они есть, то указать количество и места отказов. Для обнаружения отказов объектам достаточно на проверяющий запрос синхронно ответить совмещенными во времени сооб-

щениями, состоящими из двух битов: 10, если тест прошел, и 01, если не прошел.

Количество и места отказов находятся при помощи шкал (см. § 4) или вычислением в цепочке нумерованных объектов. Каждый работоспособный объект цепочки принимает номер из цепочки и посылает свой номер далее в цепочку. Если принятый номер отличается более чем на единицу от номера объекта, то объект его запоминает. Затем посылается групповая команда, собирающая места всех неисправностей: в нее объекты помещают свой номер и дефектный номер, полученный из цепочки.

Сетевой закон Амдала в сети, совмещающей передачу и обработку данных. Сетевой закон Амдала записывается в форме

$$T = T_s + T_p/n + T_{nw}, \qquad (1)$$

где T, T_s и T_p — времена выполнения всей программы, ее последовательной и параллельной частей соответственно, n — число объектов (с ростом n вклад второго слагаемого уменьшается), T_{nw} — затраты времени на обмены в сети. Обычно это выражение преобразуется далее, но ограничимся формой (1).

Для сети, совмещающей передачу и обработку данных, время выполнения программы в сетевом законе Амдала выглядит иначе:

$$^*T = k_1 T_s + k_2 T_p/n + T_{pnw} + k_3 T_{nw}, \qquad (2)$$

где k_1, k_2 и k_3 — числовые коэффициенты, T_{pnw} — время выполнения в сети части вычислений объектов. Из-за появления величин T_{pnw} значения слагаемых в выражении (2) могут измениться, что учитывается коэффициентами.

Времена T_s и T_p могут уменьшиться из-за передачи части обработки данных в сеть. Значение T_{nw} может уменьшиться из-за появления слагаемого T_{pnw}, и значение $k_3 T_{nw}$ станет меньше T_{nw}. Покажем возможность $^*T < T$ на примерах операций статьи.

Для выполнения группой объектов поразрядных логического умножения и сложения требуется только одна одновременная пересылка всех операндов (см. п. 5.1). Эта пересылка совмещена с процессом вычисления, операция выполняется только сетевыми средствами, исключены все последовательные операции. В результате коэффициенты k_1, k_2 и k_3 равны нулю, и $^*T = T_{pnw}$. Время выполнения операции не зависит от числа выполняющих ее объектов. Ускорение вычисления и независимость от числа участников операции достигается также при нахождении max, min, вычислениях на цепочке объектов, барьерной синхронизации, при реализации многих функций MPI, эволюционных вычислениях. При этом изменяется организация процесса вычисления — операции, обычно длительные, применять которые стремились избегать, становятся быстрыми.

Таким образом, при соответствующем выборе алгоритма решения задачи сеть, совмещающая передачу и обработку данных, дает существенный выигрыш.

ЗАКЛЮЧЕНИЕ

Из полученных результатов выделим наиболее отличающие предложенную сеть от известных сетей.

- Структура связей предлагаемой беспроводной оптической сети может быть изменена за время выполнения отдельной команды программы.
- Сообщения, посылаемые произвольно расположенными в сети источниками, доставляются приемнику (или группе приемников) как единое сообщение без временных пауз между отдельными сообщениями.
- Сетевые средства над содержимым передаваемых сообщений могут выполнять вычисления без затрат дополнительного времени.
- Средства сети позволяют существенно ускорить реализацию ряда сложных функций (показано на примере MPI), иначе строить прикладные алгоритмы.
- Возможности сети изменяют оценку сложности выполнения задач на компьютерной системе и позволяют выбирать пути ее решения, которые обычно считаются неэффективными или невозможными.
- В последнее время часто считается, что в сложных компьютерных системах, например, в суперкомпьютерах, придется отказаться от универсальной структуры связей и создавать проблемно ориентированные системы с соответствующим задаче набором устройств и объединяющей их специализированной сетевой структурой. Предложенная сеть дает дополнительную возможность получения проблемной ориентации — быструю адаптацию *универсальной* структуры связей под *текущие требования* каждой отдельной задачи и алгоритма ее решения.

При этом алгоритм и технические средства могут взаимно влиять на их организацию, но нельзя изменять внутренние связи между компонентами задачи, которые, однако, влияют на все составляющие решения задачи. Поэтому желательно связи в различных задачах классифицировать с учетом их дальнейшего решения на компьютерах. Возможно, классификации может способствовать то,

что значительную часть решаемых на компьютерах задач составляют задачи моделирования явлений природы и поведения сложных артефактов.

Сетевые свойства таких объектов широко изучаются, в последние годы учитывается динамика их поведения. Полезно установить связь этих исследований с исследованиями компьютерных систем.

ЛИТЕРАТУРА

1. *Zilberman N., Watts P.M., Romsos C., Moore A.W.* Reconfigurable Network Systems and Software-Defined Networking // Proc. of the IEEE. — 2015. — Vol. 103, N 7. — P. 1102—1124.
2. *Стецюра Г.Г.* Быстрые способы выполнения параллельных алгоритмов в цифровых системах с динамически формируемой сетевой структурой связей // Управление большими системами. — 2015. — Вып. 57. — С. 53—75. — URL: http://ubs.mtas.ru/upload/library/UBS5703.pdf (дата обращения: 15.05.16).
3. *Стецюра Г.Г.* Базовые механизмы взаимодействия активных объектов цифровых систем и возможные способы их технической реализации // Проблемы управления. — 2013. — № 5. — С. 39—53. — URL: http://pu.mtas.ru/archive/Stetsyura_13.pdf (дата обращения: 15.05.16). — doi: 10.1134/S000511791504013X (in English).
4. *Стецюра Г.Г.* Совмещение вычислений и передачи данных в системах с коммутаторами // Автоматика и телемеханика. — 2008. — № 5. — С. 170—179. — URL: http://www.mathnet.ru/links/88b780543febe14c50f605248e58d92f/at664.pdf (дата обращения: 15.05.2016. — Рус./Engl.).
5. *Стецюра Г.Г.* Средства для расширения функций коммутируемых непосредственных оптических связей в цифровых системах // Управление большими системами. — 2015. — Вып. 56. — С. 211—223. — URL: http://ubs.mtas.ru/upload/library/UBS5610.pdf (дата обращения: 15.05.16).
6. *Пат.* 2580667 РФ. Способ лазерной беспроводной ретрорефлекторной распределенной оптической коммутации и система для его реализации / Г.Г. Стецюра. Опубл.: 10.04.2016 // Бюл. — 2016. — № 10.
7. *Малеев Н.А., Кузьменков А.Г., Шуленков А.С.* и др. Матрицы вертикально излучающих лазеров спектрального диапазона 960 нм // Физика и техника полупроводников. — 2011. — Т. 45, вып. 6. — С. 836—839. — URL: http://journals.ioffe.ru/ftp/2011/06/p836-839.pdf (дата обращения: 15.05.16).
8. *Bardinal V., Camps T., Reig B.,* at al. Collective Micro-Optics Technologies for VCSEL Photonic Integration // Advances in Optical Technologies. — 2011. — doi:10.1155/2011/609643.
9. *Стецюра Г.Г.* Организация коммутируемых непосредственных соединений активных объектов сложных цифровых систем // Управление большими системами. — 2014. — Вып. 49. — С. 148—165. — URL: http://ubs.mtas.ru/upload/library/UBS4906.pdf (дата обращения: 15.05.16). — doi: 10.1134/S0005117916030139 (in English).
10. *Пат.* 2538314 РФ. Способ повышения отказоустойчивости распределенной оптической коммутации и реализующий его бесконфликтный беспроводной ретрофлекторный коммутатор / Г.Г. Стецюра. Опубл.: 10.01.2015 / Бюл. — 2015. — № 1.
11. *Rabinovich W.S., Goetz P.G., Mahon R.,* at al. 45-Mbit/s cat's-eye modulating retroreflectors // Optical Engineering. — 2007. — Vol. 46, N 10. — P. 1— 8.
12. *Gomez A., Kai Shi, Quintana C., Sato M.,* at al. Beyond 100-Gb/s Indoor Wide Field-of-View Optical Wireless Communications // IEEE Photonics Technology Letters. — 2015. — Vol. 27, iss. 4. — P. 367—370.
13. *Savidis I., Ciftcioglu B., Xue J.,* at al. Heterogeneous 3-D circuits: Integrating free-space optics with CMOS // Microelectronics Journal. — April 2016. — Vol. 50. — P. 66—75.

Статья представлена к публикации членом редколлегии В.М. Вишневским.

Стецюра Геннадий Георгиевич — д-р техн. наук, гл. науч. сотрудник, Институт проблем управления им. В.А. Трапезникова РАН, г. Москва, ✉ gstetsura@mail.ru.

10-Я ВСЕРОССИЙСКАЯ МУЛЬТИКОНФЕРЕНЦИЯ ПО ПРОБЛЕМАМ УПРАВЛЕНИЯ (МКПУ-2017)

25—30 сентября 2017 г.

с. Дивногорское, Геленджикский р-н, Краснодарский край, Россия

Мультиконференция включает в себя три локальные научно-технические конференции:

- **Модели, методы и технологии интеллектуального управления (ИУ—2017)**, председатель — академик С.Н. Васильев;
- **Робототехника и мехатроника (РиМ—2017),** председатель — академик Ф.Л. Черноусько;
- **Управление в распределенных и сетевых системах (УРСС—2017)**, председатель — академик И.А. Каляев.

Заявки на участие и тезисы докладов должны поступить в Оргкомитет Мультиконференции (НИИ МВС ЮФУ) по e-mail**:** mail@niimvs.ru **не позднее 15.05.2017 г.**

Подробности см. по адресу: http://www.ipu.ru/node/40279.