О расширении функциональных возможностей суперкомпьютера

Г.Г. Стецюра

ФГБУН Институт проблем управления им. В. А. Трапезникова РАН

Предложено к средствам коммутации суперкомпьютера добавить систему оптических беспроводных связей, образующую сеть с расширенными функциональными возможностями. Структура связей узлов сети (устройств компьютера) – полный граф, в котором реализуются только связи, необходимые в текущий момент времени. Средства коммутации находятся непосредственно в источниках или приемниках данных. Структура связей всей сети может изменяться быстро, за время выполнения команды программы. Над данными в передаваемом сообщении проводятся вычисления без затраты дополнительного времени по сравнению со временем передачи сообщения без вычисления.

Ключевые слова: беспроводная оптическая сеть, ретрорефлектор, динамическая реконфигурация, распределенная синхронизация, барьерная синхронизация, распределенные вычисления, отказоустойчивость.

1. Введение

Возможности суперкомпьютеров (СК), содержащих большое количество взаимодействующих устройств, в значительной степени определяются системой коммутации этих устройств, часто это сеть с фиксированной топологией связей. Функции средств СК четко разделены. Система коммутации транспортирует данные, используя режим коммутации сообщений, выполняет маршрутизацию сообщений, устраняет конфликты транспортировки, выполняет промежуточное хранение передаваемых данных. Обработка данных выполняется вне сети.

В докладе с целью упростить и ускорить массовое взаимодействие компонентов СК предлагается дополнить средства коммутации СК беспроводной оптической сетью, использующей оптоэлектронные компоненты. В сети частично снято четкое разделение на коммутирующие и вычислительные средства. Новая сетевая структура использует технические устройства, реализация которых известна из литературы. Оптоэлектроника использована потому, что не удается получить новые сетевые возможности, применяя только электронные средства.

Основные возможности предлагаемой сети следующие.

1. Беспроводная оптическая сеть объединяет большое количество узлов – устройств системы полносвязной структурой связей (полный граф).

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

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

4. Возможности п.п. 1 – 3 позволяют адаптировать структуру физических связей в системе под структуру виртуальных 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 над их содержимым могут выполнять вычисления без затраты дополнительного времени на проведение вычисления. Таким образом, в СК, использующем предлагаемую сеть, не будет полного разделения на коммутационные и вычислительные средства.

9. Имеются средства быстрого одновременного оповещения всех узлов сети о текущем состоянии сети.

Совокупность этих возможностей не только позволяет сети более гибко и быстро выполнять обмен сообщениями, но влияет и на другие функции СК. Сетевые средства выполняют распределенные вычисления, упрощают децентрализацию управления работой системы, ускоряют диагностику состояния сети и системы в целом.

Адаптация структуры сети под требования программы за время выполнения команды программы, поддержка быстрых массовых взаимодействий в СК, выполнение распределенных вычислений, совмещенных с передачей сообщений, также расширяют возможности конструирования прикладных алгоритмов и программ.

В докладе рассмотрены все эти вопросы.

2. Компоненты оптической сети и виды сетевых связей

2.1 Узлы сети

В сети имеются три вида узлов: объект, ретранслятор, информатор [1]. Здесь и далее будем обозначать объект с номером i как O_i, ретранслятор с номером j как R_j, информатор как SI. Если не потребуется различать R и SI, то будем их обозначать как модули связи MS.

Объект выполняет внутренние действия (вычисления, хранение данных), требуемые решаемой объектом задачей. Он также выполняет описанные ниже действия по организации взаимодействия узлов сети. Объект посылает модулям связи и получает от них оптические сигналы. Сигналы могут быть нескольких типов, различающихся качественно, например, частотой. Узел не различает поступающие к нему одновременно сигналы одного и того же типа.

Объект посылает узлам сообщения – специальным образом организованные последовательности оптических сигналов. Используются сигналы – импульсный, длящийся известное всем компонентам сети время, и непрерывный, длительность которого переменная и определяется источником сигнала.

Ретранслятор получает сигналы от объекта и, используя ретрорефлктор, отражает без задержки каждый поступающий сигнал его источнику. Ретранслятор использует поступающие к нему импульсные или постоянные сигналы одного типа для модуляции ими сигналов другого типа. Так, пусть группа объектов посылает в конкретный ретранслятор непрерывные сигналы типа f_1, и один из объектов посылает дополнительно сообщение сигналами f_2. Пусть ретранслятору разрешено модулировать сигналами f_2 сигналы f_1, копируя последними поступающее сообщение. Тогда это сообщение получат все объекты группы. Таким образом, ретранслятор *не создает* новые сигналы, для связи между объектами он использует только сигналы объектов.

Объект, использует демультиплексор, который посылает сигналы любому ретранслятору сети, выбранной им в текущий момент группе ретрансляторов или всем ретрансляторам сети, информатору.

Объект посылает ретранслятору оптический сигнал $*f$, наличие которого запретит возврат объектам сигнала f_1. Ретранслятор имеет элемент памяти. Объект посылает ретранслятору оптические сигналы $*f_1$ и $*f_2$ для перевода элемента памяти в состояние «включен / выключен» соответственно. В состоянии «включен» ретранслятору запрещен возврат полученного от объектов сигнала f_1.

Информатор отличается от ретранслятора только тем, что он при получении сигнала одного типа (в приведенном выше примере сигнала f_2) создает характерный только для информатора ненаправленный сигнал f_{si} и посылает его всем объектам сети.

2.2 Связи в сети (взаимодействие узлов)

Организация связей между объектами сети специфична. Объект – источник сигналов не посылает сигналы непосредственно приемнику. Вместо этого выполняется следующая процедура. Объект – приемник сигналов выбирает ретранслятор, через который приемник будет получать предназначенные ему сигналы. Этому ретранслятору приемник посылает непрерывный сигнал f_1. Объект – источник посылает выбранному приемником ретранслятору непрерывный сигнал f_1 и сообщение сигналами f_2. Ретранслятор пересылает сигналы источника приемнику, модулируя сигналами f_2 сигнал f_1 приемника. Объект, как указано выше, посылает сигналы конкретному ретранслятору, одновременно их группе, одновременно всем модулям связи сети. Приемник для передачи сообщения источнику должен действовать подобно источнику, но ему достаточно также посылать сообщение только своему *MS*, за которым наблюдает источник.

На рис.1 приведены виды связи объектов сети. Для упрощения показаны только ретрансляторы приемников (черные круги), которые должны быть размещены между источником и приемником.

На рис. 1a объект-источник (не закрашен) посылает сообщение произвольно выбранной группе объектов-приемников. На рис. 1b произвольная группа объектов-источников посылает сообщения единственному объекту-приемнику. На рис. 1c группа источников, используя ретранслятор объекта-посредника (или отдельный ретранслятор), посылает сообщение группе приемников.

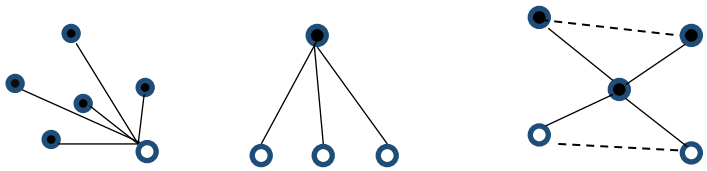

Рис. 1. Виды связей объектов сети

Для 1a действует ограничение: к приведенной группе приемников может одновременно обращаться только один источник. Для 1b разрешено обращение к приемнику группы источников, существует способ разрешения конфликта доступа и синхронизация передач источников. Для 1c устраняется конфликт доступа источников к посреднику, после чего синхронизуется передача сообщений источников. Приемники наблюдают за посредником и упорядоченно получают сообщения источников.

Топология связей изменяется посылкой соответствующей команды конкретному объекту, группе объектов или всем объектам одновременно, используя информатор. В разделе 7 показано, что существующие технические средства позволяют изменять топологию быстро.

Установление непосредственной связи между объектами позволяет предоставлять им соединение на произвольный отрезок времени, т.е. *возможна канальная коммутация*.

3. Синхронизация посылки источниками сигналов ретранслятору

Приведенная в разделе синхронизация обеспечивает приход сигналов разных источников в произвольный *MS* одновременно в ответ на приходящий от *MS* сигнал начала синхронизации (см. п. 4) [1, 2]. Время прохождения сигнала между каждым объектом O_i и любым MS_j и обратно обозначим через T_{ij}. 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 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 MS_j одновременно, с единой задержкой T_{max}. Если передаются сообщения, то одноименные разряды сообщений объектов совместятся и представят собой единое сообщение.

Если требуется, чтобы сигналы или сообщения поступали в *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 $T_{max} + Q$, сохраняя результаты доклада, но в быстрых сетях это ведет к существенному уменьшению пропускной способности.

4. Устранение конфликта доступа источников к модулю связи

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

– **Способ фиксированной шкалы.** Основа этого способа следующая [1]. Источники используют момент обнаружения конфликта как приход синхронизующей команды от *MS*, и передают в *MS* сообщение – логическую шкалу, представляющую собой последовательность двоичных разрядов. Каждому источнику, имеющему право посылать сообщение в данный *MS*, поставлен в соответствие один из разрядов шкалы. Конфликтующий источник ставит в свой разряд единицу. Логическая шкала поступает в *MS* и 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 количеством разрядов. Например, к приемнику может обратиться заранее неизвестное количество 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.

Шкала посылается в *MS* и возвращается источникам. Далее используется следующий известный способ устранения конфликта [2]. 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 первом из способов.

– **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 первом способе. При указанном условии высока вероятность бесконфликтной передачи сообщений источников.

5. Распределенные вычисления, групповые команды

5.1 Распределенные вычисления

В разделе рассмотрены распределенные вычисления, которые выполняют сетевые средства над содержимым передаваемых по сети сообщений. Рассматриваются два вида вычислений.

– Вид 1: такие операции как логическая сумма, логическое произведение, определение максимума или минимума выполняются с одновременным участием больших групп объектов. Результат вычисления появляется вне объектов и доступен всем объектам.

– Вид 2: вычисления над числами, находящимися в сообщении, проходящем через цепочку последовательно соединенных объектов. При этом проведение вычисления не вносит задержку в передачу сообщения. В каждом вычислении участвует находящееся в сообщении число и число в объекте цепочки, к которому поступает сообщение. Выполняются все логические операции, нахождение *max, 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, формируя общее сообщение. Каждый разряд в сообщении представлен двумя битами – 10 для единицы и 01 для нуля. Все приемники, наблюдающие за этим *MS* получат отраженное им сообщение и при выполнении логического сложения, полученные при наложении разрядов пары 10 и 11, будут считать единицей. Для логического умножения единицей будет наложение разрядов пары 10 и 10.

При определении максимума или минимума группа источников посылает сообщения в *MS* как в предыдущем случае, но в сообщениях не требуется выделять для каждого разряда сообщения два двоичных разряда. В числах, передаваемых в *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 числа в *MS* так, чтобы одноименные разряды чисел совместились. Вначале источники посылают в *MS* старший разряд сравниваемых чисел. *MS* возвращает сигналы источникам, и если источник, пославший в *MS* ноль, получает от *MS* 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 (раздел 4). При инверсии представлений сигналов единица и ноль аналогичным способом находится минимум.

Результат операций вида 1 создается *MS* без участия логических устройств с очень небольшой загрузкой сетевых средств. Например, пусть требуется сравнить N чисел, распределенных в группе из N устройств системы. Обычно требуется зависящее от N число пересылок этих чисел для получения результата и дополнительно пересылки результата каждому устройству. В нашем случае требуется одновременная поразрядная посылка N чисел в *MS*, и результат доступен одновременно всем устройствам.

Выполнение операций вида 2. Пусть группа объектов соединена в цепочку: первый объект направляет сообщение в *R* второго объекта, второй объект, получив это сообщение, направляет его в *R* 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. 2.

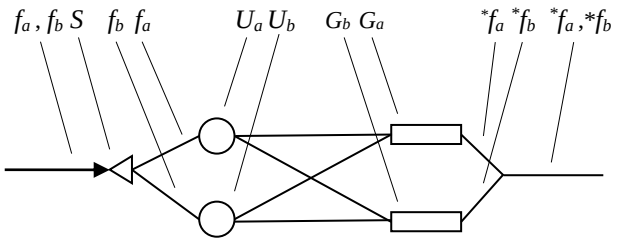

Рис. 2. Вычисление в цепочке объектов

На рис. 2 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. 2 путей. На выходе из этих путем 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 вид 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 выполняют следующие четыре действия, используя U_a и U_b.

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 $*f_a$. Действие M_3: сигнал f_a и f_b переводятся в $*f_b$. Действие M_4: сигналы f_a и f_b переводятся в $*f_a$ 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 [1, 2]. Примеры выполнения распределенных вычислений без задержки приведены в [3].

Выдача $*f_a$ вместо f_a и $*f_b$ вместо f_b объясняется тем, что *MS*, получая от источника сообщение сигналами одного типа (f_2), передаст его приемнику сигналами другого типа (f_1). Поэтому на рис. 2 на входе и выходе должны быть разные типы сигналов: $*f_a$ вместо f_a и $*f_b$ вместо f_b. Их создают G_a и G_b соответственно.

Подчеркнем, что использование пар сигналов f_a и f_b для вычислений в цепочке требует вместо указанных выше сигналов f_1 и f_2 использовать 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 своего 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, приходящего в объект. 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.

Замечания к операциям вида 2. Для выполнения циклов последний объект цепочки быть подключается к первому объекту. Для выполнения ветвления требуется перестройка структуры связей, которую организует инициатор вычислений, посылая команду ветвления. Некоторые ветвления могут выполняться объектами цепочки локально, изменением выполняемых ими действий с учетом предыдущих результатов.

5.2 Групповые команды

В групповых командах используются вычисления по п. 5.1. Групповая команда (ГК) – это сообщение, которое перемещается через цепочку объектов и содержит данные и инструкции объектам по обработке находящихся в ГК данных, изменению содержащихся в ГК инструкций, выполнению локальных действий в объекте [2]. 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 применении вычислительных операций вида 1 групповые команды приходят ко многим объектам одновременно с наложением разрядов как единая команда.

6. Примеры использования возможностей сети

6.1 Барьерная синхронизация

Эта обычно длительная операция быстро решается с использованием изложенных выше результатов. Рассмотрим следующий ее вариант. Пусть все источники группы P должны завершить работу и после этого передать сообщения – результаты своей работы приемникам этой группы, ожидающим сообщения. На завершение работы источникам требуется разное время. Завершив работы, источники должны передать сообщения приемникам как единое сообщение без временных пауз между отдельными сообщениями.

Для синхронизации в группе источников выделяется один представитель группы P. Его ретранслятор MS_p известен всем источникам и приемникам, которые следят за MS_p., посылая ему сигнал f_1. (В качестве MS_p может быть взят свободный модуль, не связанный с объектом.)

При подготовке сообщения источники передают в MS_p непрерывный сигнал $*f_3$, запрещающий возврат сигналов f_1. Подготовив сообщение, источник снимает запрещающий сигнал. После готовности всех источников MS_p начнет возвращать сигнал f_1, что будет признаком синхронизации. Получив f_1 объекты передадут сообщения синхронно с задержками $*T_i$ в ретранслятор MS_p и все приемники получат его как единое сообщение.

6.2 Синхронизация посылки сообщения источника группе приемников

Источник сообщения посылает приемникам заявку на принятие сообщения. В ответ каждый приемник посылает в свой модуль MS сигнал $*f_1$, запрещающий ему возврат сигналов f_1. При готовности к приему сообщения приемник пошлет в MS сигнал $*f_2$ и запрет будет снят.

Источник следит за MS всех приемников, посылая им одновременно сигналы f_1, и исчезновение, а затем появление f_1 служит синхросигналом возможности передачи.

Использование сигналов $*f_1$ и $*f_2$ вместо $*f$, как в разделе 6.1, избавляет приемник от необходимости непрерывной передачи сигнала в MS.

6.3 Сетевые взаимодействия в MPI

Эта важная тема затронута кратко. Вначале дадим примеры влияния свойств сети на реализацию функций MPI, затем отметим возможность создания новых функций.

1. Применение канальной коммутации (раздел 2) закрепляет связь между источником и приемником на требуемое время, что упрощает реализацию ряда функций MPI.

2. Пусть группа процессов, каждый из которых расположен на отдельном объекте сети, с помощью функции MPI_ALLTOALL должна передать через сеть сообщение всем процессам – приемникам, также расположенным на отдельных объектах группы. Реализация этой функций требует выполнения длительных программных действий. С применением барьерной синхронизации раздела 6.1 она быстро выполняется в основном аппаратными действиями сетевого контроллера объекта.

3. Приемники размещены как в пункте 2. Источник выполняет функцию MPI_BCAST для рассылки копий сообщения группе приемников, предварительно выяснив их готовность к приему. Для проверки готовности применяется быстрая синхронизация из раздела 6.2. Копии сообщения рассылаются одновременно.

4. В MPI функция MPI_GRAPH_CREATE отображает виртуальную топологию связей задачи на реальную топологию системы. Непосредственные соседи в виртуальной топологии в реальной системе могут быть соединены через длинную цепочку связей сети.

В предлагаемой сети каждой виртуальной связи соответствует реальная связь между устройствами системы. Связи системы в реальном масштабе времени формируются под задачу с целью устранения длинных цепочек связей. Установленная связь сохраняется в течение произвольного отрезка времени.

О создании новых функций MPI. Приведенные примеры касаются только реализации функций MPI, но сеть также позволяет разрабатывать новые функции MPI. Эту возможность

дают новые быстрые действия в сети: изменение топологии связей, канальная коммутация, синхронизация, способ разрешения конфликтов, распределенные вычисления.

6.4 Эволюционные вычисления

Во многих эволюционных алгоритмах группы объектов, действуя параллельно, многократно находят частные варианты решения, которые также многократно требуется сравнить между собой для выявления имеющих максимальное или минимальное значение. Рассмотренные в разделе 5.1 способы вычисление *max* или *min* выполняются существенно быстрее обычно применяемых способов. Один из вариантов действий такой. Вначале все объекты проводят цикл вычислений, среди которых надо найти наилучшее решение. На это объектам может потребоваться разное время. Для определения момента времени, когда очередной цикл таких вычислений будет всеми завершен, используется предложенная барьерная синхронизация. После этого находится *max* выполнением операции вида 1 или вида 2.

6.5 Борьба с повреждениями сети и системы

Рассмотрим два вопроса: устранение сетевых повреждений, нарушающих целостность системы и быстрое информирование о количестве и расположении неисправных объектов системы. Единственный вид компонентов сети, повреждения которых влияют на целостность системы, это *MS*. Объект – приемник, обнаружив отказ используемого им модуля, занимает запасной модуль, проводя борьбу за модуль с другими объектами подобно способу устранения конфликта доступа. 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* работоспособность системы сохраняется.

Для определения количества неисправностей в системе объекты объединяются в цепочку, по которой посылается групповая команда, выдающая суммарное количество неисправностей.

Для идентификации неисправных объектов в цепочке каждый исправный объект заносит в групповую команду свое имя или координату в системе.

7. О технической реализации средств сети

Для реализации предложений доклада требуется разработка компонентов, выполняющих функции демультиплексора, ретранслятора и информатора. Анализ литературы показал наличие устройств, близких к требуемым. Сошлемся, в частности, на следующие результаты.

Демультиплексор – находится в каждом объекте и соединяет объект с любым *MS* системы или одновременно с произвольной группой *MS*. Требуемая организация такого демультиплексора описана в [4, 5], где использована решетка управляемых лазеров. В [6, 7] даны примеры реализации решеток лазеров.

Ретранслятор. Основные составляющие ретранслятора – ретрорефлектор, модуляторы света, фотоприемники [1, 8, 9]. В качестве примера работы, где использованы все эти компоненты, приведем упрощенно результаты из [10].

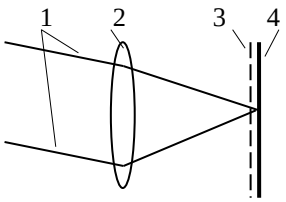

Рис. 3. Ретранслятор с ретрорефлектором

На рис. 3: 1 – световой сигнал, идущий 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».

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 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. 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. 3, имеет независимо работающие каналы для связи с каждым источником сигналов 1. В настоящее время появились близкие по подходу работы других авторов.

Ретранслятору в предлагаемом докладе требуются аналогичные компоненты со следующими изменениями.

– Упрощение: модулятор/фотоприемник (3) должен быть общим для всех объектов.

– Усложнение: модулятор должен быть избирательным по частоте. Однако можно ограничиться средствами [10]: устройство по рис. 3 позволяет разделить поток после линзы на два потока и направить их на два комплекта (3, 4).

Информатор. В *SI* новым по сравнению с компонентами приведенных устройств является источник модулированных ненаправленных оптических сигналов. Известны появившиеся в последнее время разработки таких устройств, передающих сигналы со скоростями модуляции свыше 10 гигагерц [11].

В заключение раздела отметим, что появились публикации по созданию систем на кристалле с применением оптических беспроводных соединений узлов системы [12]. Некоторые из рассмотренных в докладе решений применимы и в таких системах. Подобно решению в [12] в нашем случае беспроводные обмены можно выполнять в пылезащищенной конструкции, содержащей только оптоэлектронные компоненты ретрансляторов и модулей связи.

8. Заключение

Из полученных результатов доклада 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.

– Над содержимым передаваемых сообщений сетевые средства могут выполнять вычисления, не затрачивая на вычисление дополнительного времени.

– Средства сети позволяют существенно ускорить реализацию ряда сложных функций (на примере MPI), иначе строить некоторые прикладные алгоритмы.

Один из создателей современной теории сложных сетей A.L. Barabási в известной статье [13] подчеркивает, что в XXI веке сети стали центральным объектом всех областей исследований.

Это справедливо и для суперкомпьютеров, где сеть объединяет отдельные устройства в единую сложную систему. Доклад показывает, что в этой области влияние сетей может выходить за рамки их использования, как средства транспортировки сообщений.

Литература

1. Стецюра Г.Г. Быстрые способы выполнения параллельных алгоритмов в цифровых системах с динамически формируемой сетевой структурой связей // Управление большими системами. 2015. Выпуск 57. С. 53-75. URL: http://ubs.mtas.ru/upload/library/UBS5703.pdf (дата обращения: 15.05.16).

2. 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); (in English **DOI**: 10.1134/S000511791504013X)
3. Стецюра Г.Г. Совмещение вычислений и передачи данных в системах с коммутаторами // Автоматика и телемеханика. 2008. № 5. С. 170-179. (Русский/English) URL: http://www.mathnet.ru/links/88b780543febe14c50f605248e58d92f/at664.pdf (дата обращения: 15.05.16).
4. Стецюра Г.Г. Средства для расширения функций коммутируемых непосредственных оптических связей в цифровых системах // Управление большими системами. 2015. Выпуск 56. С. 211-223. URL: http://ubs.mtas.ru/upload/library/UBS5610.pdf (дата обращения: 15.05.16).
5. Патент на изобретение № RU 2580667 C1 от 10.01.2015 Стецюра Г.Г.
6. Малеев Н.А., Кузьменков А.Г., Шуленков А.С. и др. Матрицы вертикально излучающих лазеров спектрального диапазона 960 нм // Физика и техника полупроводников. 2011. Том 45. Выпуск 6. С. 836-839. URL: http://journals.ioffe.ru/ftp/2011/06/p836-839.pdf (дата обращения: 15.05.16).
7. V. Bardinal, T. Camps, B. Reig, at al. Collective Micro-Optics Technologies for VCSEL Photonic Integration // Advances in Optical Technologies. 2011. **DOI**:10.1155/2011/609643
8. Стецюра Г.Г. Организация коммутируемых непосредственных соединений активных объектов сложных цифровых систем // Управление большими системами. М.: ИПУ РАН, 2014. вып. 49. С. 148-165. URL: http://ubs.mtas.ru/upload/library/UBS4906.pdf (дата обращения: 15.05.16). (in English **DOI:** 10.1134/S0005117916030139)
9. Патент на изобретение № RU 2538314 C1 от 10.04.2016 Стецюра Г.Г.
10. Rabinovich W.S., Goetz P.G., Mahon *R.* at al. 45-Mbit/s cat's-eye modulating retroreflectors // Optical Engineering. 2007. Vol. 46. No. 10. P. 1-8.
11. Gomez A., Kai Shi, Quintana C., Sato M. at al. Beyond 100-Gb/s Indoor Wide Field-of-View Optical Wireless Communications // Photonics Technology Letters. IEEE. 2015.Vol. 27. Issue 4. P. 367-370.
12. Savidis I., Ciftcioglu B., Jie Xu, at alias. Heterogeneous 3-D circuits: Integrating free-space optics with CMOS. // Microelectronics Journal. April 2016. Vol. 50. P. 66-75.
13. Barabási A.L. The Network Takeover//Nature Physics. Jan 2012. Vol. 8. P.14-16.

Addition for supercomputer functionality

G.G. Stetsyura

V. A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences

It is proposed add the optical wireless switching network with advanced functionalities to the supercomputer system. The structure of links of nodes (computer devices) - complete graph in which only links are realized, necessary at the current time. The switching units are located in the sources and receivers only. The structure of the network links can be changed quickly during the execution of the single program instruction. The calculations may be executed for the data in the message without additional time for these calculations.

Keywords: wireless optical network, retroreflector, dynamical reconfiguration, distributed synchronization, barrier synchronization, distributed computing, fault tolerance.

References

1. Stetsyura G. Fast Execution of Parallel Algorithm on Digital System with Dynamically Formed Network Structures // Large-scale System Control. M.: Institute of Control Sciences of Russian Academy of Sciences. Issue. 57. 2015. P. 53-75. (in Russian) URL: http://ubs.mtas.ru/upload/library/UBS5703.pdf (accessed: 15.05.2016).
2. Stetsyura G. Basic Interaction Mechanisms of Active Objects in Digital Systems and Possible Methods of Their Technical Realization// Control Sciences. 2013. №5. С. 39-53. DOI: 10.1134/S000511791504013X. (in Russian URL: http://pu.mtas.ru/archive/Stetsyura_13.pdf accessed: 15.05.2016).
3. Stetsyura G. Combining Computation and Data Transmission in the Systems with Switches // Automation and Remote Control. May 2008. Volume 69. Issue 5. pp 891-899 (English/Russian) URL:http://www.mathnet.ru/php/archive.phtml?wshow=paper&jrnid=at&paperid=664&option_lang=rus (accessed: 15.05.2016).
4. Stetsyura G. Extending Functions of Switched Direct Optical Connections in Digital Systems // Large-scale Systems Control. M: Institute of Control Sciences of Russian Academy of Sciences. Issue. 56. 2015. С. 211-223. (in Russian) URL:http://ubs.mtas.ru/upload/library/UBS5610.pdf (accessed: 15.05.2016).
5. Patent RU 2580667 C1 10.01.2015 Stetsyura G.
6. Maleev N.A., Kuzmenkov A.G., Shulenkov A.S. at alias. Matrix of 960 nm Vertical-cavity Surface-cmitting Lasers// Semiconductors. Vol. 45. n. 6. 2011. P. 836-839. URL: http://journals.ioffe.ru/ftp/2011/06/p836-839.pdf (in Russian) (accessed: 15.05.2016).
7. 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
8. Stetsyura G.G. Organization of Switched Direct Connections of Active Objects in Complex Digital Systems// Automation and Remote Control, 2016, Vol. 77, No. 3, P. 523–532. DOI:10.1155/2011/609643. (in Russian URL: http://ubs.mtas.ru/upload/library/UBS4906.pdf accessed: 15.05.2016)
9. Patent RU 2538314 C1 10.04.2016 Stetsyura G.
10. Rabinovich W.S., Goetz P.G., Mahon R. et al., 45-Mbit/s Cat's-Eye modulating Retroreflectors // Optical Engineering. Vol. 46. n.10. 2007. P. 1-8.
11. Gomez A., Kai Shi, Quintana C., Sato M. at alias. Beyond 100-Gb/s Indoor Wide Field-of-View Optical Wireless Communications // Photonics Technology Letters. IEEE. 2015. Vol.27. Issue 4. P. 367-370.
12. Savidis I., Ciftcioglu B., Jie Xu, at alias. Heterogeneous 3-D circuits: Integrating free-space optics with CMOS. Microelectronics Journal April 2016. Vol. 50. P. 66-75.
13. Barabási A.L. The Network Takeover//Nature Physics. Jan 2012. Vol. 8. P.14-16.