

УДК 004.75
ББК 32.81 max

БЫСТРЫЕ СПОСОБЫ ВЫПОЛНЕНИЯ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ В ЦИФРОВЫХ СИСТЕМАХ С ДИНАМИЧЕСКИ ФОРМИРУЕМОЙ СЕТЕВОЙ СТРУКТУРОЙ СВЯЗЕЙ

Стецюра Г. Г.¹

(ФГБУН Институт проблем управления РАН, Москва)

Предложены способы ускорения выполнения параллельных алгоритмов с использованием средств быстрого непрерывного формирования структур связей в цифровой системе и быстрых распределенных вычислений. Используются беспроводные оптические соединения.

Ключевые слова: параллельные алгоритмы, распределенные вычисления, сетевые структуры, беспроводные оптические связи.

1. Введение

В статьях [3, 4] предложены технические средства, расширяющие функциональные возможности цифровых систем и повышающие скорость выполнения параллельных алгоритмов. Также было предложено изменение структуры алгоритмов и организации их выполнения по сравнению с обычной практикой. Хотя в [3, 4] даны примеры таких изменений, технические подробности отодвигали этот вопрос на задний план. В настоящей статье отсутствуют технические детали системы, но подробно изложены сообщения-команды, которыми обмениваются устройства системы для выполнения ими операций

¹ Геннадий Георгиевич Стецюра, г.н.с., доктор технических наук, профессор (gstetsura@mail.ru).

формирования структуры связей системы и выполнения распределенных вычислений.

Рассматриваемая система имеет сетевую организацию взаимодействия активных цифровых объектов системы, связывающихся по собственной инициативе с другими устройствами для разрешения собственных или общесистемных потребностей (если контекст позволяет, будем активные объекты называть просто объектами). Активный объект – это любое программируемое цифровое устройство или комплекс таких устройств. Большинство рассматриваемых в статье видов связей – множественные, одновременно объединяющие группы объектов.

Использованные в статье оптические средства не выполняют вычисления, это делают объекты системы – обычные электронные устройства. Оптоэлектроника используется только для получения новых видов взаимодействия между объектами. Однако взаимодействия построены так, чтобы существенно влиять на структуру и скорость выполнения распределенных параллельных вычислений, что отличает подход статьи от чисто коммуникационных решений и от оптических вычислений, выполняемых оптическими вычислительными средствами.

В разделе 2 статьи кратко представлена структура системы; в разделе 3 рассмотрены посылаемые объектам сообщения-команды; в разделах 2 и 3 рассмотрены выполняемые по команде операции, включены новые результаты по выполнению распределенных вычислений. В разделе 4 даны примеры областей применения решений статьи; в разделе 5 приведены сведения из литературы о разработках оптоэлектронных устройств, которые могут быть взяты за основу при создании наиболее сложных компонентов системы – ретранслятора и демультимплексора, и показывают техническую осуществимость предлагаемой системы.

2. Структура системы

Структура предлагаемой системы показана на рис. 1.

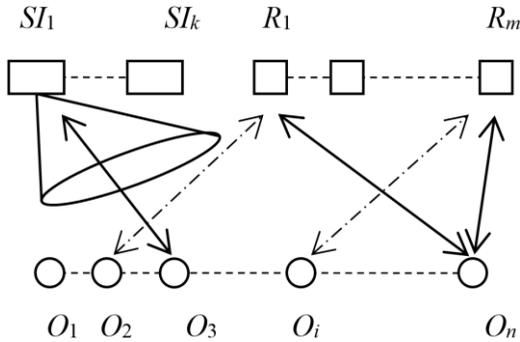


Рис. 1. Структура системы

Здесь O – активные объекты, R – ретрансляторы сигналов, SI – системные информаторы. Для упрощения текста, где возможно, будем называть R и SI просто модулями связи (MS). Перечисленные устройства обмениваются оптическими сигналами, посылая их в свободном пространстве. Объекты располагаются так, чтобы между каждым объектом и всеми MS была прямая видимость. Только объекты O_j имеют специальные устройства – демультиплексоры, которые выбирают конкретный MS для отправки ему сигналов.

Ретранслятор новой системы существенно отличается от обычно применяемых ретрансляторов. Он содержит оптический ретрорефлектор – типа «катафот» или «кошачий глаз» (раздел 5 статьи) и модуляторы оптических сигналов, но в нем нет источника оптических сигналов. Сигналы, приходящие в ретрорефлектор, отражаются им в направлении источника этих сигналов. Сигналы, поступающие одновременно от многих источников не создают взаимных искажений возвращаемых сигналов. В целом система обладает новизной [9].

Организация связей. Чтобы получать от ретранслятора сигналы источника, приемники должны посылать в R непрерывный сигнал, который возвращается в приемник, модулированный сигналами источника. От ретранслятора таким же способом одновременно, не конфликтуя между собой, может получать сигналы любая группа объектов. Если несколько O_i одновременно пытаются воспользоваться для передачи сообще-

ний одним и тем же MS , то действует алгоритм обнаружения и быстрого устранения возникающего конфликта доступа [5, 6]. Информатор SI действует подобно R , но, в отличие от R , получив сигналы, генерирует и передает такие же сигналы *одновременно* всем объектам системы, но на другой частоте, характерной для данного SI . При обращении к SI объект действует, как с R , но объекты также могут посылать в SI ненаправленные сигналы, подобно тому, как действует SI . Получив такой сигнал, SI в ответ посылает всем объектам подобный сигнал, но на другой частоте. Такие действия полезны для связи с периферийными устройствами, так как не требуют ориентации на SI и не нужен демультимплексор (раздел 3, группа команд 10).

На рис. 1, например, объект O_n посылает в свободном пространстве оптические сигналы ретрансляторам R_1 и R_m (сплошные линии), и эти сигналы получают объекты O_2 и O_i . Объект O_3 посылает сигналы в SI_1 и последний посылает такие же сигналы на другой частоте одновременно всем объектам.

Ретрансляторы и системные информаторы располагаются регулярно, например, в виде решетчатых структур. Каждому объекту, имеющемуся в системе или подключаемому вновь, выделяется фиксированное место размещения в системе. При размещении объекта для попадания лучей его демультимплексора на заданные ретрансляторы требуется дополнительная ориентация, соответствующая месту расположения объекта.

Устройства MS имеют дополнительную возможность. По крайней мере для части поступающих в MS сигналов возврат их источнику управляем: в MS посылается управляющий оптический сигнал, который включает элемент памяти, запрещающий возврат сигнала источнику. Запрет сохраняется до перевода памяти также оптическим сигналом в исходное состояние. Для SI запрет действует и на посылку из SI широковещательного оптического сигнала.

Эти действия позволяют, например, одновременно проверять состояние объектов, которым предоставлен данный MS , не обращаясь непосредственно к самим объектам.

В разделе 5 статьи даются сведения о способах реализации демультимплексоров и ретрансляторов системы.

Отметим наиболее важные возможности, вытекающие из рассмотренной организации системы.

1. Сигнал, посылаемый объектом в *MS*, возвращается ему и всем следящим за *MS* объектам. При этом возвращаемый объектам сигнал *MS* модулируется входящими в него от объектов импульсными сигналами. Возвращается сигнал источнику или нет, зависит от состояния элемента памяти в *MS*, удаленно управляемого объектами. Для *SI* запрет действует на передачу им широковещательного сигнала.

2. Объект может послать сигналы одного и того же сообщения как одному, так одновременно и группе ретрансляторов, указав для этого своему демультиплектору группу адресов ретрансляторов. Демультиплексор направит копии сообщения по указанным адресам.

3. Хотя для взаимодействия объектов в системе задаются адреса *MS*, допустимы и ассоциативные связи между объектами. Так, если передает широковещательное сообщение *SI*, то в сообщении можно указать набор признаков, которыми должен обладать объект. Таким объектам достаточно приписать новый, обобщающий результаты поиска признак, и далее этот признак используется широковещательно. Такая обработка проведена ассоциативно, без использования адресов объектов и *MS*.

Ассоциативный поиск выполняется и при обращении к *R*, подробности в разделе 4 (ассоциативные операции).

4. Объекты могут одновременно следить за состоянием группы модулей *MS* и, тем самым, за состоянием объектов, связанных с этими модулями. Слежение за состоянием *MS* сводится к проверке наличия возвращаемого модулем сигнала.

5. Посылка сигналов объектами, произвольно расположенными относительно любого ретранслятора, синхронизируется так, чтобы они поступили в ретранслятор в заданные моменты времени.

Синхронизация требует пояснения. Между демультиплексором объекта и ретранслятором или системным информатором отсутствуют задерживающие или преобразующие сигнал устройства. Это позволяет известными способами определить время прохождения сигнала T_{ij} между каждым объектом O_i и

любым MS . Для синхронизации в ответ на входящий от MS_j сигнал объекты должны послать свои сигналы так, чтобы они поступили в MS_j одновременно. Так как расстояния объектов от MS_j в общем случае различны, то сигнал от MS_j поступит к объектам в разное время и немедленно отправленные объектами сообщения поступят в MS_j в различные моменты времени. Для устранения разброса во времени 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 так же, поступят в MS_j одновременно, с единой задержкой T_{max} . Если передаются сообщения, то одноименные разряды сообщений объектов совместятся и представят собой единое сообщение.

Чтобы сообщения поступали в MS одно за другим без временных пробелов, как одно сообщение, 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 .

3. Команды управления взаимодействием объектов

Посылаемые объектам команды запускают операции, в том числе выполняющие взаимодействие частей распределенных алгоритмов. Команды направляются объектам в виде сообщения, которое состоит из одной команды или последовательности команд (команды группы 6). Некоторые группы команд представлены единственной командой, но их можно добавлять. Причем группа 6 позволяет разработчику алгоритма пополнять группы, создавая сложные сообщения – последовательности команд, с привлечением для этого команды из разных групп.

После рассмотрения команд перечислены операции, выполняющие все требуемые командами взаимодействия объектов. Детальное описание организации операций, в отличие от описания команд, требует обращения к деталям технической реализации. Поэтому в разделе 2 и этом разделе статьи описан только принцип действия операций, а более подробное описание требует обращения к цитируемым статьям.

Все команды разделим на следующие группы.

1. Команды анализа и изменения состояния модулей связи.
2. Команды передачи сообщений.
3. Команды синхронизации взаимодействия объектов.
4. Команды устранения конфликтов доступа.
5. Команды формирования системы.
6. Команды формирования цепочек связей.
7. Команды выполнения распределенных вычислений.
8. Групповые команды.
9. Команды поддержки работоспособности системы.
10. Команды связи с простыми объектами (интеллектуальными датчиками)

Рассмотрим каждую группу команд.

Группа 1 «Команды анализа и изменения состояния модулей связи». Используются описанные выше операции запрета и разрешения возврата модулем MS сигнала, поступившего в модуль от объекта. Если MS имеет несколько так управляемых каналов, то каждому соответствует характерная только для него частота управляемого сигнала.

Команды позволяют группе объектов следить за состоянием других групп объектов, бесконфликтно считывая состояние их MS . Проверяется отсутствие всех возвращаемых группой сигналов или присутствие хотя бы одного такого сигнала.

Группы 2 и 3 «Команды передачи сообщений» и «Команды синхронизации взаимодействия объектов». Пока предположим, что нет конфликта доступа объектов-источников к ретранслятору. Для передачи сообщения в MS источник использует оптические непрерывные сигналы двух частот f_1 и f_2 . Источник модулирует сигнал f_1 битами передаваемого сообщения, MS возвращает сигнал f_2 источнику, модулированным сигналами его сообщения, принимаемого ретранслятором на частоте f_1 . Любое количество объектов может одновременно посылать сигнал f_2 в ретранслятор, что не мешает передаче сообщения на частоте f_1 . Каждый объект, пославший f_2 в ретранслятор, получает его модулированным сигналами принятого ретранслятором сообщения. Таким способом осуществляется информирование группы объектов о текущем состоянии ре-

транслятора, и все эти объекты, включая приемники сообщения, получают передаваемое сообщение. Ссылки на подробности технической реализации этой схемы действий даны в разделе 5.

Синхронизация. Покажем примеры, использующие возможность демультиплексора одновременно посылать сигналы группе *MS* (раздел 5 и [4]). Первый пример: передача сообщения одного источника группе ретрансляторов. Для этого демультиплексор объекта просто *одновременно* посылает сообщение группе ретрансляторов, используемых приемниками [4].

Второй пример: передача сообщений группой источников одному ретранслятору в виде единого сообщения без временных пробелов между составляющими его сообщениями отдельных источников. Один объект из группы объектов, взаимодействующих через указанный ретранслятор, посылает всей группе одновременно синхронизирующую команду с указанием упорядоченного списка имен объектов, которые должны передать сообщения известной длины. Указанные объекты, получив команду, используют, как показано выше, свои $*T_i$ и формируют единое сообщение без временных пауз между сообщениями объектов.

Третий пример: группа источников также должна передать свои сообщения приемникам, ожидающим эти сообщения, но только после того, как эти источники сформируют свои сообщения, на что им требуется разное время. В группе источников выделяется один представитель всей группы. Его ретранслятор известен всем источникам и приемникам.

Подготовив сообщение для передачи, каждый источник группы в своем *MS* запрещает возврат сигнала модулем *MS*. Все участвующие в этих действиях объекты одновременно следят за состоянием всех модулей *MS* источников группы [4]. Если все модули *MS* источников не возвращают сигнал, то все источники сформировали свои сообщения, что является моментом синхронизации источников, которые передадут сообщения синхронно с задержками $*T_i$ в ретранслятор представителя группы и все приемники его получат.

Группа 4 «Команды устранения конфликтов доступа». Выше предполагалось, что источник передает сообщение при-

емнику без конфликта с другими источниками. Но конфликты доступа в сложных системах, особенно в системах управления, избежать нельзя. Сущность быстрого способа устранения конфликта в системе (рис. 1) заключается в следующем [5, 6].

Объект обнаружит конфликт доступа, если посланное им и возвращенное из *MS* сообщение будет искажено. После этого требуется выяснить, какие источники участвуют в конфликте, и упорядочить их передачи сообщений так, чтобы они были переданы как единое сообщение без временных пауз между частными сообщениями источников.

Для этого источники используют момент обнаружения ошибки как приход синхронизирующей команды от *MS* и передают в *MS* сообщение – логическую шкалу. В шкале 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 всем конфликтующим источникам. Источники определяют порядковый номер своей передачи и передают последовательно сообщения как единое сообщение. Конфликт устранен.

Группа 5 «Команды формирования системы». Если объектам присвоены порядковые номера (адреса), то, как показано выше, они позволяют формировать и изменять связи в системе с высокой скоростью. Покажем, как присваивать порядковые номера.

Присвоение новым объектам порядковых номеров. Пусть в произвольные моменты времени к системе подключаются новые объекты и им надо присвоить новые адреса. Присваивается объекту номер того свободного места в системе, в которое будет установлен объект. Эти места заранее выделены, и помещение в них объекта должно гарантировать поступление сигналов с демультимплексора объекта на *MS* модули системы.

Присвоение порядковых номеров в очереди к модулю *MS*. Потребность в этой операции возникает при работе со шкалами в операциях синхронизации передач сообщений и устранения конфликтов доступа. При обращении к конкретному модулю *MS*, как правило, участвует незначительная часть объ-

ектов системы, и достаточно применять шкалы с количеством разрядов значительно меньшим количества объектов в системе. Поэтому надо вводить отдельную нумерацию объектов, которым разрешено в текущей ситуации обращаться к модулю *MS*.

Приведем нумерацию с использованием способа децентрализованного приоритетного управления (ДПУ) [2, 7]. Суть простейшей формы ДПУ – «децентрализованного кодового управления» (ДКУ) – в рассматриваемом случае такая. Объектам уже приписаны номера занимаемых ими мест в системе. Начиная со старшего разряда номера, объекты последовательно посылают в *MS* сигналы на частоте f_1 , если значение очередного разряда их номера равно единице. От модуля *MS* объекты получают на частоте f_2 сигнал в ответ на каждый указанный сигнал, поступивший на частоте f_1 . Если текущий разряд порядкового номера конкретного объекта равен нулю, а сигнал f_2 сообщает объекту, что есть другие претенденты со значением единица этого разряда, то объект прекращает передачу остальных разрядов. При передаче всех разрядов объекту-победителю предоставляется первый разряд в шкале. Затем проводится аналогичный процесс для оставшихся объектов. Победитель получает второй разряд шкалы и т.д.

Группа 6 «Команды создания цепочек связей». Пусть имеется группа объектов, например, показанная на рис. 2 группа из четырех объектов *A*, *B*, *C*, *D*. Требуется так организовать связи между ними, чтобы сообщение, полученное объектом *A*, было им послано в *B*, от него в *C* и затем в *D*.

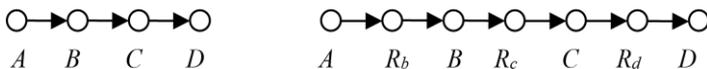


Рис. 2. Цепочка связей

Решение показано на этом же рисунке: объект *A* посылает сообщение в *R_b* – ретранслятор объекта *B* и т.д. Передача сообщения совмещается с его обработкой, часто выполняемой без задержки сообщения (группа команд 7 и 8).

В системе может быть создано несколько цепочек. Если объекты хранят имена цепочек и инструкции по их созданию, то

посылка единственной команды через *SI* запустит процесс одновременного формирования множества цепочек связей.

Группа 7 «Команды выполнения распределенных вычислений». Первый вид распределенных вычислений: по цепочке объектов передается сообщение, содержащее число, над которым объекты без задержки сообщения поочередно выполняют вычисления, используя свои локальные данные.

Ранее [2] для электронных проводных связей было показано, как выполнять над сообщением без его задержки двухместные логические операции, нахождение *max* и *min*, арифметические сложение, вычитание и умножение, и при этом не создавать сигнал заново в каждом объекте. Позднее такое решение было получено для волоконно-оптических связей [7].

Ниже дано решение, применимое для беспроводных оптических связей и канала с единственной волоконно-оптической линией. При вычислениях придется в объекте создавать новые оптические сигналы за счет энергии объекта, но задержка сигналов отсутствует, так как не требуется предварительный анализ значения приходящего в объект оптического сигнала.

Заметим, что в модулях *MS* сигналы не преобразуются и достаточно рассмотреть только действия в объектах.

Ограничимся примером выполнения вычислений с привлечением фотоприемников и лазерных источников. Пусть двоичные разряды чисел, над которыми выполняются вычисления, передаются в свободном пространстве по общему каналу оптическими сигналами (или по линии) на двух частотах: f_a для значения 1 и f_b для значения 0. Сигналы поступают в устройство, показанное на рис. 3. Пассивными оптическими средствами сигнал направляется в группу фотоприемников $P_{1,4}$ (нумерация на рис. 3 – сверху вниз), причем сигнал f_a поступает на $P_{1,2}$, сигнал f_b поступает на $P_{3,4}$. Электрические сигналы с выходов $P_{1,3}$ поступают на лазерный источник T_a , генерирующий при этом оптический сигнал f_a . Сигналы от $P_{2,4}$ поступают на лазерный источник T_b , генерирующий сигнал f_b . Сигналы с выходов T_a и T_b объединяются и передаются в свободном пространстве в требуемый модуль *MS*.

Объект до прихода f_a или f_b в устройство подает управляющие сигналы C_1, \dots, C_4 из группы сигналов C_{1-4} на P_1, \dots, P_4 соответственно. Если поступают управляющие сигналы на $P_{2,3}$, то значение пришедшего оптического сигнала без анализа его значения и без задержки будет инвертировано; $P_{1,3}$ формирует из любого входного оптического сигнала оптический сигнал f_a ; $P_{2,4}$ формирует сигнал f_b ; $P_{1,4}$ не изменяет значение входного сигнала. Этих сигналов достаточно для выполнения указанных операций над числом в сообщении и числом в объекте [7]. Примеры выполнения распределенных вычислений приведены в [8].

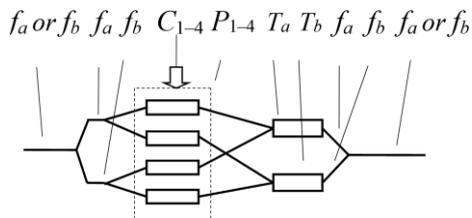


Рис. 3. Вычислитель

Второй вид распределенных вычислений не требует построения цепочек объектов. Он применим только для логических операций и нахождения max и min . Объект, используя SI , посылает объектам системы команду выполнить одну из таких операций. В ответ объекты синхронизируют в SI посылку каждого своего сигнала (команды группы 3) и далее действуют с учетом специфики конкретной команды. Так, для нахождения max объекты синхронно посылают старший разряд своего числа. Модуль SI ретранслирует эти сигналы всем их источникам. Если объекты послали в SI сигнал f_b и получают от SI сигналы f_a и f_b , то их числа не являются наибольшими и эти объекты прекращают участие в нахождении max . Оставшиеся объекты поступают аналогично с остальными разрядами чисел, что приводит к нахождению max . Легко построить аналогичные действия для остальных операций. Ретрансляторы используются подобно SI .

Группа 8. «Групповые команды». Эффективное средство воздействия на процессы, выполняемые в системе с оптически связями в свободном пространстве – групповые команды [7]. Групповая команда (ГК) – это сообщение, которое перемещается через цепочку объектов и содержит данные и инструкции объектам по выполнению указанных в ГК действий: обработке находящихся в ГК данных, изменению содержащихся в ГК инструкций, локальным действиям в объекте. Выполнение действий ГК не должно задерживать ее передачу. Изменение инструкций в ГК означает следующее. Объект, проанализировав часть проходящей через него ГК, заменяет оставшуюся часть ГК своей информацией *без задержки* на это преобразование. В результате ГК, перемещаясь через цепочку объектов, собирает по пути информацию об объектах и изменяется сама – ее влияние на объекты зависит от действий предшественников объекта в цепочке.

При посылке ГК группе объектов они могут распространять ее древовидно по многим цепочкам объектов.

Системные информаторы *SI* используются для посылки ГК объектам, имеющим указанные в команде отличительные признаки.

Хотя выше говорится о групповой команде, сообщение может быть групповой программой, состоящей из последовательности групповых команд, сформированной несколькими объектами. Такую последовательность желательно начинать с маркера начала (МН), в котором указана длина сообщения – место, где располагается маркер конца сообщения (МК). Если объекту требуется добавить в конец сообщения свою информацию, то он, не задерживая сообщение, будет пытаться заменить код МК кодом маркера продолжения сообщения (МП). Если при этом объект обнаруживает на этом месте код МК, то этот код заменяется кодом МП с указанием места расположения нового МК. Если же обнаружен код МП, то новая запись кода МП не портит предшествующий МП, но объект должен продолжать поиск МК. В МН следует указывать также максимально допустимую длину сообщения.

Группа 9. «Команды наблюдения за состоянием системы и поддержки ее работоспособности». Во многих ответственных задачах автору алгоритмов полезно иметь простой доступ к средствам проверки состояния системы, выполняющей его задачу. Это облегчают описанные выше команды.

– Достаточно послать только одну команду через *MS*, чтобы одновременно запустить тесты во многих объектах системы. В команде задаются детальные требования к тестам. Объекты могут проверять также работоспособность модулей *MS*.

– Если сформированы цепочки связей (группа 6) и в перемещающейся по цепочке групповой команде (группа 8) задано условие проверки, то за время перемещения сообщения по цепочке будет определено количество объектов, удовлетворяющих заданному условию, и указаны их адреса (группа 7) [7].

– Смена объектом модуля *MS* позволяет просто повышать отказоустойчивость средств связи: достаточно иметь количество дополнительных модулей *MS* для замены, равное только ожидаемому количеству отказавших модулей.

Один из возможных вариантов замены отказавшего модуля: объект, обнаружив отказ используемого им модуля, занимает запасной модуль, проводя борьбу за модуль с другими объектами подобно способу устранения конфликта доступа (группа 4). Если отказов больше, чем имеется запасных модулей *MS*, то объект будет подключен к модулю, уже занятому объектами, и будет использовать модуль совместно с ними.

Группа 10. Команды связи с простыми объектами (интеллектуальными датчиками). Для таких объектов желательно упрощать средства сетевого взаимодействия. Исключим в них демультимплексоры. Простые объекты посылают в *SI* ненаправленный сигнал или сигнал лазерного источника, постоянно направленный на *SI*.

Приведем три примера. Пусть простые объекты разделены на группы, которым присвоены имена. Произвольный объект системы через *SI* посылает группе команду передать свои данные (например, показания прибора). Объекты группы упорядочены, и, следуя порядку, они последовательно передают через *SI*

свои данные, которые ретранслируются всем объектам, включая «заказчика».

Пусть теперь группе посылается команда: передать данные только от тех объектов, у которых произошло изменение их состояния: простые объекты также обращаются к *SI*, применяют команды групп 3 и 4, определяют моменты начала передачи своих данных и передают их как одно общее сообщение.

Наконец, пусть простым объектам разрешено передавать данные по собственной инициативе. Тогда они действуют как обычные активные объекты, но обращаться могут только к *SI*.

Все такие упрощенные действия с применением *SI* подобны взаимодействиям через общую шину.

О базовых операциях. Как следует из описания групп команд, они требуют исполнения следующих базовых операций: запрета и разрешения модулю *MS* возвращать сигнал; бесконфликтного считывания состояния *MS*; передачи сообщения одного источника одному приемнику без конфликта; такой же передачи, но группе приемников; синхронизации передач сообщений группы источников; устранения конфликтов доступа; присвоения порядковых номеров объектам системы; то же для объектов в очереди к *MS*; создания цепочек объектов; распределенных вычислений.

Замечание к разделу 3. В разделе не приведены форматы команд, так как цель статьи – только дать способы формирования групп команд, однако для каждой из приведенных команд указан способ ее реализации. Группы команд могут пополняться новыми командами, которые потребуют решаемые задачи. Это особенно относится к командам групп 7 и 8.

4. Примеры областей применения предложенных команд и операций

Этот раздел носит иллюстративный характер, приложения операций рассмотрены кратко, основная цель – продемонстрировать новизну и полезность решений статьи для ряда областей применения.

Ассоциативные операции. Под ассоциативными операциями понимаются операции по поиску и преобразованию информации, которые обращаются к множеству ее носителей не по их адресам, а по набору признаков, характеризующих сам носитель или его информацию. Обычно речь идет об обращении к специально организованному ассоциативному запоминающему устройству (АЗУ) или к ассоциативному процессору (АП), в котором в процессе поиска допускается изменять содержимое его АЗУ и использовать полученный результат в последующих ассоциативных операциях. Часть предложенных в настоящей статье операций являются ассоциативными. Типично ассоциативными являются операции, использующие обращение к системному информатору *SI*, позволяющие обращаться к объектам по набору характерных для объекта признаков. К ассоциативным операциям относятся групповые операции, обращающиеся по набору признаков к цепочке объектов. Перемещаясь от объекта к объекту, они передают объектам новые признаки и получают от них признаки для следующих объектов. Такие команды или программы имеют переменную длину и без задержки постепенно пополняются новым содержимым, а также разделяются на части, перемещающиеся по разным путям.

Можно рассмотреть и другие варианты ассоциативности, но важно выделить, что принципиально нового дает описываемое решение. В настоящей статье и ее двух предшественниках [3, 4], где рассмотрены технические детали оптоэлектронной организации связей, ассоциативные операции действуют не на отдельном устройстве, а на множестве распределенных программируемых устройств с сетевой организацией связи между ними. При этом ассоциативные операции *вливают на всю структуру взаимодействия системы устройств.*

Активные объекты, участвующие в ассоциативных взаимодействиях, не обязательно должны иметь АЗУ или АП, но их наличие ускоряет действия объектов.

Ассоциативные операции полезны также в приведенных ниже трех областях.

Управление потоком данных (Dataflow architecture). В таких архитектурах продвижение процесса обработки данных

управляется готовностью промежуточных результатов, что требует быстрых действий по информированию о готовности. Типичные действия: рассылка результатов потребителям по их адресам или потребителям, имеющим заданный набор свойств (признаков); проверка потребителем готовности всех результатов групп источников.

Команды и операции из раздела 3 позволяют быстро выполнять приведенные выше и другие близкие по структуре действия. Рассылка источниками результатов потребителям, имеющим заданный набор признаков или адресов, выполняется с помощью модулей *MS*.

Более детально. Источники записывают сведения о готовности результатов в элементы памяти *MS*, используя операции группы 1. Группы приемников, не конфликтуя между собой и также используя операции группы 1, одновременно обращаются к группе модулей *MS* и проверяют готовность всех результатов в источниках, подключенных к этим модулям.

Команды статьи ускоряют еще одно полезное действие – формируют группы потребителей результата (приемников). Группа источников, используя *MS*, в общем сообщении посылает приказ приемникам, имеющим требуемые признаки или требуемые адреса, подключиться к общему для них *MS* (создать группу).

Системы с переменной структурой. Работы в этой области направлены на быстрое изменение связей в компьютере под решаемую задачу. Способы статьи направлены на это же: переключение отдельных связей, групп связей или всех связей – практически одинаково быстрые операции, сравнимые по скорости с обращением к ячейке оперативной памяти; перестройка связей в системе проводится не в специально выделенные интервалы времени, а непрерывно по мере потребности. Структура системы формируется непрерывно и быстро.

Приведем примеры. Пусть объект, используя *SI*, посылает всем объектам системы команду выполнить настройку своих связей в системе. Объекты хранят инструкции по настройке связей с *MS*. В ответ на единственную команду выполнится требуемая перестройка связей одновременно всеми объектами.

Пусть теперь объект, используя команды групп 2 и 3, адресно, через ретрансляторы, обращается к группе объектов с аналогичной командой. Скорость перестройки такая же.

Пусть, наконец, объекты используют для перестройки групповые команды, они дают дополнительные средства для перестройки связей, учитывающие сведения о состоянии других объектов, помещаемые объектами в групповую команду.

Эволюционные алгоритмы. Во многих эволюционных алгоритмах группы объектов, действуя параллельно, многократно находят частные варианты решения, которые затем также многократно сравниваются между собой для выявления имеющих максимальное или минимальное значение. Команды группы 7, вычисляют *max* или *min* без задержки, существенно быстрее обычно применяемых способов.

Общее замечание. Применение команд из раздела 3 в рассмотренных областях приводит к очень близким действиям, хотя в обычной практике они существенно различаются.

5. О технической реализуемости устройств системы

Так как статья не рассматривает технические вопросы, приведем только известные примеры, показывающие реализуемость двух центральных компонентов системы – ретранслятора и демультиплексора.

Ретранслятор. Основные составляющие ретранслятора – ретрорефлектор, модуляторы света, фотоприемники. В качестве примера работы, где рассмотрены все эти компоненты, приведем упрощенно (с показом только необходимых для данной статьи деталей) результат U.S. Naval Research Laboratory (NRL) [13], рис. 4.

На рис. 4 введены: 1 – световой сигнал, идущий от удаленного лазерного источника; 2 – линза; 3 – плоскость с модуляторами/фотоприемниками, созданными с использованием вертикальных множественных квантовых ям (MQW Vertical Modulator/Photoreceiver); 4 – фокальная плоскость линзы, в которой расположено зеркало. Элементы 2 и 4 составляют ретрорефлектор типа «кошачий глаз».

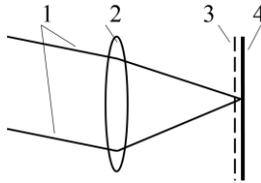


Рис. 4. Устройство с ретрорефлектором

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, которое принимает фотодетектор 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. 4, имеет независимо работающие каналы для связи с каждым источником сигналов 1. Передача сообщений выполнялась на 7 км в обычных атмосферных условиях со скоростью 45 Мбит/сек.

В нашей статье ретранслятор имеет общие для всех объектов ретрорефлектор, модуляторы и фотоприемники.

Демультимплексор. В статье [1] приведена решетка из тридцати пяти лазерных источников, выполненных на вертикально излучающих лазерах (ВИЛ, VCSEL). Лазер выбирается адресно и генерирует сигналы в диапазоне 958–962 нм со скоростью ~ 10 Гбит/сек (позднее получено ~ 40 Гбит/сек).

В Sandia National Laboratories был проделан цикл работ по созданию большого размера решеток адресуемых VCSEL – до 16К лазеров [11, 12].

Рассмотрим один из возможных путей создания демультимплексоров для объектов рассматриваемой системы. Демультимплексор объекта содержит решетку лазеров;

дешифратор, позволяющий обратиться к любому лазеру решетки; регистр, хранящий историю обращения дешифратора к лазерам. При обращении дешифратора или регистра к лазеру последний генерирует непрерывный сигнал, который пассивными отклоняющими средствами направляется в требуемом направлении. Дополнительные средства задают частоту сигнала лазера и модулируют его сигналами сообщения. Наличие регистра не обязательно, но ускоряет обращение к группам объектов.

Известны другие способы управления направлением светового луча, которые также потенциально могут быть использованы в демультимплексоре, некоторые из них приведены в [3].

Демультимплексоры и ретрансляторы не участвуют в преобразовании данных, и, как указано во введении, система не выполняет оптические вычисления [10].

6. Заключение

Суммируем основные результаты статьи.

– Все предложенные в статье команды и операции ориентированы на цифровую систему, в которой активные объекты и модули связи объединены оптическими каналами связи, формируемые в свободном пространстве.

– Времена формирования отдельной связи, одновременного формирования группы связей или всех связей в системе близки к времени обращения к ячейке оперативного запоминающего устройства. Интервал времени между началом передачи сообщения источником и началом приема его приемником равен времени прохождения в свободном пространстве оптического сигнала между указанными объектами.

– Объекты системы объединяются в цепочки, на которых выполняются распределенные вычисления над данными, находящимися в передаваемом по цепочке сообщении и в объектах. Такие вычисления выполняются без задержки сообщения.

– В цепочках объектов распределенными вычислениями управляют специальные сообщения – групповые команды, объекты могут вносить в них изменения, влияя на действия

остальных объектов цепочки. Все эти действия выполняются без задержки сообщения.

– Объекты быстро устраняют конфликты при доступе к общим ресурсам.

– Объекты синхронизируют передачу своих сообщений, и они поступают к приемнику одно за другим без временных пробелов, как одно сообщение.

– Объектом системы может быть отдельное программируемое устройство или комплекс таких устройств. Во всех таких объектах естественно использовать как оптические связи, так и электронные. Поэтому количество оптических связей в системе может быть существенно меньше количества коммутируемых устройств.

– Приведенные в разделе 5 статьи ссылки показывают, что технические устройства, необходимые для предлагаемых способов коммутации и обработки данных, являются предметом многих успешных исследований.

– Рассмотренная система имеет высокую однородность, в нее легко вносить изменения, влияющие на производительность алгоритмов. Так, незначительное усложнение модуля связи – включение в него элемента памяти, управляющего возвратом светового луча объекту, – существенно расширило возможности системы.

– Сетевые структуры вычислительных и управляющих систем представляют также интерес как объекты управления с очень жесткими требованиями к гибкости и скорости управления формированием сетевых связей и к скорости протекающих в системе процессов. Приведенные в статье способы позволяют дополнительно улучшить эти параметры систем.

Литература

1. 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, вып. 6. – С. 836–839. – [Электронный ресурс]. – URL: <http://journals.ioffe.ru/ftp/2011/06/p836-839.pdf> (дата обращения: 03.05.2015).

2. ПРАНГИШВИЛИ И.В., ПОДЛАЗОВ В.С., СТЕЦЮРА Г.Г. *Локальные микропроцессорные вычислительные сети.* – М.: НАУКА, 1984. – 176 с.
3. 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/archive/search_results_new.php?publication_id=19284 (дата обращения: 03.02.2015). (Перевод на английский: 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).
4. 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: 03.08.15).
5. СТЕЦЮРА Г.Г. *Способ устранения конфликтов доступа* // Журнал радиоэлектроники. – 2012. – №5. – [Электронный ресурс]. – URL: <http://jre.cplire.ru/jre/may12/7/text.pdf> (дата обращения: 03.05.15).
6. СТЕЦЮРА Г.Г. *Исключение задержки в передаче сообщений при устранении конфликтов доступа* // Журнал радиоэлектроники. – 2012. – №8. – [Электронный ресурс]. – URL: <http://jre.cplire.ru/jre/aug12/9/text.pdf> (дата обращения: 03.05.15).
7. СТЕЦЮРА Г.Г. *Базовые механизмы взаимодействия активных объектов цифровых систем и возможные способы их технической реализации* // Проблемы управления. – 2013. – №5. – С. 39–53. – [Электронный ресурс]. – URL: http://pu.mtas.ru/archive/Stetsyura_13.pdf (дата обращения: 03.05.2015). ('Online First' on SpringerLink – URL: <http://link.springer.com/article/10.1134/S000511791504013X>).
8. 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: 03.05.15).

9. СТЕЦЮРА Г.Г. *Патент РФ* № 2538314. – 2015.
10. AMBS P. *Optical Computing: A 60-Year Adventure* // *Advances in Optical Technologies*. – 2010. – Vol. 2010. Article ID 372652. – 15 p. – [Электронный ресурс]. – URL: <http://www.hindawi.com/journals/aot/2010/372652/> (дата обращения: 23.05.15).
11. *Final Report on LDRD Project: Heterogeneous Integration of Optoelectronic Arrays and Microelectronics* // Sandia National Laboratories. – 2003. – 97 p. – [Электронный ресурс]. – URL: <http://prod.sandia.gov/techlib/access-control.cgi/2003/030411.pdf> (дата обращения: 03.05.15).
12. GEIB K.M., CHOQUETTE K.D., SERKLAND D.K., ALLERMAN A.A. *Fabrication and performance of two-dimensional matrix addressable arrays of integrated vertical-cavity lasers and resonant cavity photodetectors* // *IEEE Journal of Selected Topics in Quantum Electronics*. – 2002. – Vol. 8. – No. 4. – P. 943 – 947.
13. RABINOVICH W.S., GOETZ P.G., MAHON R. et al. *45-Mbit/s cat's-eye modulating retroreflectors* // *Optical Engineering*. – 2007. – Vol. 46. – No. 10. – P. 104001-1–104001-8.

FAST EXECUTION OF PARALLEL ALGORITHMS ON DIGITAL SYSTEMS WITH DYNAMICALLY FORMED NETWORK STRUCTURES

Gennady Stetsyura, Institute of Control Sciences of RAS, Moscow, Doctor of Science, professor (65, Profsoyuznaya str., Moscow, Russia, phone (495)334-78-31).

Abstract: Wireless optical connections are suggested to accelerate execution of parallel algorithms by supporting fast formation of connection structures and fast distributed computations in a digital system.

Keywords: parallel algorithms, distributed computing, network structures, wireless optical communications.

Статья представлена к публикации членом редакционной коллегии М.В. Губко

Поступила в редакцию 13.05.2015.

Опубликована 30.09.2015.