

## АКТИВНЫЕ СЕТИ И СЕТИ С ГРУППОВЫМИ ПРОГРАММАМИ (СХОДСТВО И РАЗЛИЧИЯ)

Г.Г.Стецюра

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

Россия, 117997, Москва, Профсоюзная ул., 65

E-mail: [stetsura@ipu.ru](mailto:stetsura@ipu.ru)

В докладе рассматриваются механизмы управления взаимодействием компонентов в двух видах сетей – активных сетях и сетях с групповыми программами. Эти механизмы позволяют включать в передаваемые по сети сообщения команды, но разными способами. Проведено сравнение указанных механизмов управления и показано, что в сетях с групповыми программами по сравнению с активными сетями существенно повышается реактивная способность, достигающая уровня, характерного для вычислительных кластеров.

**ACTIVE NETWORKS AND NETWORKS WITH GROUP PROGRAMS (SIMILARITY AND DISTINCTIONS) / G. Stetsyura** (Institute of Control Sciences RAS, 65 Profsoyuznaya, Moscow, 117997, Russia). In this article we consider the control for active networks and networks with group programs. The control in all these networks allows to include in messages the commands by various ways. The control in second networks works so fast as the control for computers interaction in multicomputers.

## Введение

В наше время все больший вклад в обработку данных и управление вносят распределенные системы с сетевой организацией взаимодействия компонентов. Такие системы нас будут интересовать, прежде всего, с точки зрения возможности их использования в системах управления жесткого реального времени. Причем будем рассматривать механизмы взаимодействия компонентов сетей, обеспечивающие высокие скорости реакции системы. Будут рассмотрены два вида механизмов взаимодействия – команды активных сетей (далее АС) и групповые программы (далее ГП). Оба механизма приближают характер взаимодействия в сетях к механизмам взаимодействия в вычислительных комплексах. Активные сети разрабатывались для использования в сетях с удаленными связями (например, в Интернете), а групповое программирование изначально ориентировалось на локальные сети или сети с дальними связями, узлы которых представляют собой локальные сети.

Вначале кратко рассмотрим наиболее известные сетевые технологии и новые механизмы взаимодействия компонентов сетевой системы, которые ориентированы на ускорение указанного взаимодействия и повышения его гибкости.

### Облачные вычисления, grid computing, локальные управляющие сети, кластеры

Облачные вычисления – Cloud Computing (далее СС) – это вид распределенных вычислений в значительной степени аналогичный вычислениям, проводимым в GRID Computing, но имеющий несколько другой потребительский аспект. Основная идея СС – предоставление удаленного сервиса, не требующего от потребителя больших начальных финансовых вложений.

Типичные виды предоставляемых сервисов:

- предоставление удаленно работающей программы – "Software as a Service" (SaaS);
- предоставление удаленной вычислительной платформы заданной конфигурации – "Platform as a Service" (PaaS);

- доступ к удаленным архивам данных – "Data as a Service" (DaaS);
- предоставление удаленной инфраструктуры – "Infrastructure as a Service" (IaaS);
- предоставление удаленного приложения – "Application as a Service" (AaaS) и др.

Потребители отдельных видов сервиса могут существенно отличаться друг от друга. Например, типичный потребитель AaaS – это частное лицо, использующее интерактивное телевидение, получившее право чтения электронной книги и т.п. Потребителем IaaS может быть компания, постоянно арендующая удаленную инфраструктуру с гарантированным временем доступа и качественным обслуживанием вместо создания собственной технической системы и найма обслуживающего ее персонала и т.п.

В целом, это направление позволяет избавиться от затрат на сложную технику и обслуживающий персонал, если гарантировано качественное удаленное обслуживание. Одна из неприятных особенностей СС – сложность защиты от враждебных воздействий.

Сейчас доступ к СС обеспечивает ряд фирм, например, Amazon (EC2), Google (Cloud Print, на выходе Chrome OS), VMware.

Надо отметить, что подход к вычислениям, предлагаемый СС, весьма распространен в других сферах деятельности человека. Например, при ремонте квартиры редко все работы делает владелец квартиры: из разных обслуживающих организаций вызываются электрики, водопроводчики, маляры, так же поступают во многих других случаях.

Наиболее характерное различие между СС и Grid Computing (GC) состоит в том, что СС пытается освободить потребителя от всех затрат по покупке и поддержке технического и программного обеспечения, а GC пытается увеличить имеющуюся на местах мощность по обработке данных за счет привлечения удаленных, свободных в данный момент ресурсов. В обоих случаях речь идет об использовании удаленных ресурсов, но к собственным ресурсам, как видно, отношение принципиально разное.

Локальные управляющие сети (ЛУС) используются для организации взаимодействия автоматов в рамках предприятия или более сосредоточенного объекта, например, самолета или современного автомобиля, в котором используется группа вычислительных устройств, датчиков и исполнительных механизмов. В отличие от рассмотренных выше СС и GC в ЛУС появляются чрезвычайно жесткие временные требования к работе системы, что выражается в использовании более простых и быстрых протоколов, чем в приведенных выше сетях.

Обратимся к вычислительным кластерам. Они могут использоваться в двух основных режимах. Так, на кластере может решаться одна сложная задача, и разработчик предварительно планирует использование задач ресурсов кластера. Отклонения от плана возникают только из-за непредсказуемости, порождаемой ветвлениями в вычислении.

В другом режиме, характерном для больших ВЦ общественного пользования, загрузку кластера для группы не связанных между собой задач приходится назначать в динамике. Этот режим имеет сходство с сетевым режимом работы при распределенном планировании загрузки. Однако, как правило, планирование выполняется централизованно.

В сетях основным механизмом взаимодействия компонентов сети является обмен сообщениями, с помощью которых в соответствии с задействованным протоколом передаются данные. В кластерах взаимодействие также в основном выполняется сообщениями, содержащими достаточно большие массивы данных.

Однако имеются и другие возможности – передавать сообщение-программу, содержащую как данные, так и программу, взаимодействующую с данными и программами адресатов (АС и ГП). Эти возможности позволяют сетям повысить гибкость и быстроту реакции на события. Переходим к описанию этих возможностей.

### Активные сети

Активными названы сети, в которых узлы могут быть запрограммированы для выполнения индивидуальной обработки, в частности маршрутизации, каждого отдельного сообщения, проходящего через узел. Для этого сообщение снабжается дополнительной управляю-

щей информацией, которая воспринимается как команда или программа (группа команд), определяющая действия узла по отношению к сообщению. Такой подход позволяет учитывать индивидуальные особенности каждого сообщения. Помимо гибкости в маршрутизации АС позволяют легко и быстро модифицировать используемые протоколы управления передачей сообщений с учетом индивидуальных требований участников обмена сообщениями. Это достигается за счет того, что только те узлы, которые могут обнаружить встроенную в сообщение управляющую информацию, реагируют на нее. Для остальных узлов такое сообщение не отличается от традиционного. Как следствие, АС совместимы с существующими протоколами и сообщения АС могут без ограничения передаваться через сети, на поддерживаемые возможности АС. Поэтому АС позволяет очень быстро и гибко вводить и опробовать инновации в существующие сетевые инфраструктуры.

Пакет передаваемого активного сообщения в АС состоит из трех частей. Начало пакета содержит обычный IP заголовок, позволяющий любому стандартному устройству принять и направить далее пакет в соответствии с информацией, указанной в заголовке.

АС-устройство анализирует следующую часть пакета – IP опцию. Опция содержит код команды и ее параметры. Пользуясь командой, АС-устройство требуемым способом обрабатывает сообщение. Наконец, третья часть – это данные пользователя.

Как видно, АС встраивается в структуру сети иерархически: любой узел сети воспринимает основной протокол сети, часть узлов обязана воспринимать основной протокол и дополнительно протокол АС.

Таким образом, основное назначение АС – обеспечить гибкое, легко трансформируемое управление транспортировкой сообщений в сетевой структуре. Помимо этого обеспечивается и некоторое ускорение обработки сообщений, так как эти решения могут приниматься непосредственно в сетевой аппаратуре, без необходимости привлечения основного вычислительного оборудования узлов системы. Из перечисленных выше облачных вычислений, grid computing и локальных управляющих сетей механизмы АС ориентированы на применение в первых двух видах сетей.

Работы над АС начаты в 1994 г при поддержке DARPA [1, 2], и активно развиваются в настоящее время [3].

### Групповые программы

В ИПУ РАН в течение длительного времени проводятся исследования по системам управления, имеющим сетевую структуру [4, 5,6]. При этом ставилась цель разработать принципы построения цифровых распределенных средств управления, обладающих быстрой реакцией на возникающие события и гибкостью структуры, упрощающей борьбу с нестандартными ситуациями в работе системы.

Для этого было решено полностью децентрализовать управление работой сетевой системы и максимально сократить временные задержки передаваемых сообщений в узлах. При этом требуется сочетать различные скоростные требования к указанным задержкам, возникающие на разных уровнях одной и той же системы. Рассмотрим пример.

Допустим, имеется распределенная система управления, состоящая из взаимно удаленных кластеров. В пределах кластера требуется обеспечить максимально быструю реакцию на события и очень малую задержку передаваемых между компонентами кластера сообщений. Сами эти сообщения могут иметь малую разрядность. С другой стороны, обмен сообщениями между удаленными кластерами не требует быстрой реакции, и здесь сообщения обычно имеют большую длину. Последний вид обмена обычно происходит через глобальные сети, например, Интернет, что означает необходимость следования общепринятым стандартам.

Для функционирования в столь различных условиях, требуется ввести иерархию форматов передаваемых сообщений и быстрые способы создания форматов и работы с ними. Этот вопрос будет рассмотрен ниже, но чтобы было ясно, что значит "быстро" рассмотрим организацию децентрализованного управления на самом быстром уровне иерархии – уровне об-

менов в кластерах (далее УОК) и разработанные для работы на нем групповые команды и групповые программы.

**Групповая программа уровня УОК.** На уровне УОК обмен сообщениями можно выполнять как распределенный совмещенный процесс передачи данных и их обработки в процессе передачи, причем без задержки сообщений в узлах [4, 5, 6]. Без задержки выполняются логические операции, арифметические: сложение, вычитание, умножение и операции нахождения *max*, *min*. Этих операций достаточно для выполнения практически всех задач, возникающих при распределенном управлении работой системы. Кроме этого с таким набором операций можно решать многие вычислительные задачи [7].

Указанные операции выполняются следующим способом. Вначале надо сформировать последовательное соединение требуемых узлов. Сообщение, в котором надо провести вычисление содержит один операнд и перемещается через цепочку узлов, точнее через сетевые процессоры узлов. Вторые операнды находятся в каждом из сетевых процессоров. При прохождении сообщения через узел без задержки сообщения выполняется требуемая операция над двумя операндами. *Ключевое свойство* совмещенных операций – результат вычисления заменяет первый операнд в сообщении. После прохождения сообщения через цепочку узлов в нем вместо первого операнда будет находиться результат действия всех узлов.

Чтобы указать узлу, как он должен обрабатывать операнд в сообщении, перед операндом помещается код групповой операции, понимаемой всей группой узлов. Последовательность из кода групповой операции и операнда составляет простейшую групповую команду. В общем случае код групповой команды может содержать сложные указания по обработке содержимого сообщения, определяющие места сообщения (операнды), которые следует обрабатывать, возможно, различным способом. То есть код групповой операции должен описать узлам структуру сообщения.

Более того, действия групповой команды могут распространяться не только на сообщение, содержащее эту команду, но и на определенную в команде группу последующих сообщений. Групповая команда может также изменять состояние конкретных узлов системы, в которые она поступает, тем самым в динамике реконфигурируя систему, включая изменение прав отдельных узлов. Например, так можно задавать ветвление вычислений, передачу от одного узла другому прав лидера – узла, имеющего право посылать групповые команды.

Следующий шаг в построении групповых взаимодействий – определение групповой программы. Групповая программа – это сообщение, в разных частях тела которого размещаются определенные выше групповые команды. Чтобы структуру групповой программы можно было формировать произвольным образом, будем код каждой групповой операции отмечать специальной меткой. На рис. 1 показан пример сообщения – групповой программы.



Рис. 1 Сообщение – групповая программа

Здесь серым цветом обозначены коды групповых команд, без окраски – данные групповых команд.

Надо также отметить, что групповая программа не обязательно должна проходить через цепочку узлов. Она может быть направлена средствами коммутации системы в конкретный узел, где и произведет изменение состояния этого узла.

#### **Иерархия сообщений, содержащих групповые программы.**

Изложенная организация групповых программ ориентирована на самые быстрые взаимодействия узлов (УОК). Теперь вернемся к использованию групповых программ в смешанных системах, использующих также удаленные связи, например, соединяющие кластеры через интернет при проведении облачных вычислений или в *grid computing*.

Удаленную передачу будем проводить с помощью сообщений, удовлетворяющих международным стандартам, например, протоколу TCP/IP. В это сообщение требуется поместить групповую команду или групповую программу, адресованную удаленным узлам УОК. Формирование сообщения в формате TCP/IP выполняется следующим образом. Узел УОК создает сообщение – групповую программу, содержащую в своем теле групповую программу для удаленных узлов УОК. Напомним, что если такое сообщение проходит через цепочку узлов, то групповая программа для удаленных узлов может без задержки формироваться всеми узлами цепочки. Это сообщение направляется узлу-передатчику, который способен организовать удаленную связь. Передатчик сохраняет у себя полученную групповую программу, создает сообщение TCP/IP, упаковывает в него в качестве данных групповую команду для удаленных узлов УОК, и направляет это сообщение удаленному приемнику сообщений через сеть общего пользования. Приемник извлекает из принятого сообщения групповую программу и направляет ее удаленным узлам УОК.

Передача с переходом на более высокие уровни иерархии в принципе возможна без Указанной задержки. Пусть узел УОК посылает групповую команду, как и ранее, узлу-передатчику, способному отправить сообщение по сети с TCP/IP. Но в тело этой команды узел помещает сообщение, полностью сформированное в формате TCP/IP. Передатчик принимает групповую команду, определяет свои действия, и, выполнив соответствующие приготовления, передает остаток сообщений на следующий верхний уровень. Однако, часто требуется вводить задержку как из-за более низкой скорости передачи данных на верхнем уровне, так и из-за специфики протокола верхнего уровня. Например, может потребоваться работа с квитанциями и придется либо ограничиться на нижнем уровне короткими сообщениями, либо сохранить в узле-передатчике длинное сообщение нижнего уровня, и далее передатчик должен самостоятельно выполнить все действия, требуемые протоколом верхнего уровня.

Действия приемника зависят от требований протокола верхнего уровня. Если таким протоколом, например, является TCP/IP, то приемник должен принять сообщение (точнее группу пакетов), подтвердить источнику корректность каждого полученного пакета, и после этого собрать из пакетов сообщение и в сохраненном сообщении выделить часть, которую следует передать далее в виде групповой программы УОК.

Если же верхний уровень иерархии позволяет передавать сообщения в виде групповых программ, скорость передачи которых совпадает со скоростью на УОК, то приемник может без задержки выделить часть сообщения – групповую программу для узлов УОК и отправить ее далее в сетевую структуру УОК. Головную часть сообщения-команды, адресованную приемнику, последний просто не пропустит в сеть нижнего уровня. Альтернативный вариант – сообщение верхнего уровня приемник должен доставить без предварительной обработки многим узлам. Это может потребоваться, если такие узлы ретранслируют сообщение в связанные с ними подсети узлов, и головная часть сообщения из верхнего уровня иерархии одинаково определяет действия всех узлов-ретрансляторов. В этом случае приемник пропускает сообщение "как есть". Выбор действий приемника осуществит отдельное сообщение-команда, посланное в приемник предварительно.

### **Сравнение взаимодействия в АС, GRID и сетях с групповыми программами.**

В активных сетях и GRID взаимодействие узлов сетей осуществляется по схеме "клиент – сервер": клиенты, которым требуются результаты некоторой работы, находят серверы, которые выполняют эту работу и отправляют результат клиенту. В этом взаимодействии сообщения по назначению различны, хотя и имеют одинаковый формат. Клиент посылает серверу команду, сервер возвращает клиенту данные. В локальных сетях и кластерах, особенно предназначенных для систем управления, узлы в значительной степени равноправны, при этом преобладающим видом сообщения является команда, определяющая характер взаимодействия узлов и работ, выполняемых узлами. Поэтому в управляющих системах, особенно в тех, где требуется быстрое время реакции на события, целесообразно все сообщения оформ-

лять в виде групповых команд или групповых программ. Весь материал последнего раздела направлен на подтверждение этого тезиса.

В итоге приходим к следующим выводам.

1. В сетевых взаимодействиях, по крайней мере, одно из сообщений в сеансе обмена сообщениями содержит команду.

2. Сообщение АС, оформляют как обычное сообщение, передающее данные, и только внутри него специальной меткой отмечается наличие команды. Обнаружение команды не обязательно доступно всем приемникам сообщения. Такая структура вызвана необходимостью соответствовать требованиям ТСР/IP или другим общепринятым сетевым протоколам.

3. Если нет необходимости соответствовать существующим протоколам, то целесообразно оформлять сообщения в виде групповой команды или программы, что позволяет достигнуть большей гибкости взаимодействия узлов и быстроты реакции системы на события.

4. Групповые программы по сравнению с командами в АС позволяют организовать распределенные вычисления, которые во многих случаях выполняются быстрее, чем на многопроцессорных и многомашинных комплексах со связями, организованными традиционно.

5. Несколько слов об управлении коммутацией – основной задачей АС. Групповая программа может управлять многозвенной коммутацией, в каждой точке выбора маршрута без задержки выбирая нужное направление передачи сообщения – программы. Головная часть сообщения используется в узле для выбора дальнейшего маршрута, а когда узел придет остаток сообщения, то он начнет перемещаться в подготовленном узлом направлении. Может также потребоваться не задерживать начало сообщения в узле. Это возможно, головная часть сообщения будет далее перемещаться в сети как "мусор", не мешая работе других узлов. Допустимо также при помощи групповой команды дублировать передачу сообщения по многим направлениям.

Дополнительно отметим, что групповые команды служат средством формирования элементов самоорганизации систем при устранении конфликтов доступа узлов к общим ресурсам [8]. Пусть возникает ситуация, при которой отдельные узлы обнаруживают указанный конфликт из-за соперничества с другими узлами, и узлы не имеют никакой информации о соперниках: об их количестве, составе, внутреннем состоянии.

В такой системе с помощью групповых команд удастся децентрализованно выделить узел-лидер, который создаст новые связи между узлами и проведет обмен информацией между узлами, устраняющий конфликт.

### Заключение

Предложен иерархический способ построения сообщений с групповыми программами, позволяющий эффективно их использовать на разных уровнях иерархии сетей.

Показано, что для распределенных систем управления способ ввода управляющей информации в сетевые сообщения с помощью групповых команд и групповых программ дает существенные преимущества по сравнению с решением, принятым в активных сетях: повышается скорость и гибкость распределенного управления.

### Литература

1. Sixto Ortiz Jr. Active Networks: The Programmable Pipeline// Computer. – 1998. vol. 31, no. 8. – pp 19–21.
2. Tennenhouse D.L. at alias. A Survey of Active Network Research// IEEE Communications Magazine. – 1997. vol. 35, no.1. – pp 80–86.
3. Hussain S. A. Active and programmable networks for adaptive architectures and services. Auerbach Publ. 2007. P. 329.
4. Прангишвили И.В., Стецюра Г.Г. Микропроцессорные системы М.: Наука, 1980. 237 с.

5. Прангишвили И.В., Подлазов В.С., Стецюра Г.Г. (1984). Локальные микропроцессорные вычислительные сети, С. 176. Наука, Москва.
6. Стецюра Г.Г. (2005). Методы совмещения вычислений и передачи данных в мультипроцессорных системах и локальных сетях, С. 86. Ин-т пробл. упр., Москва.  
<http://www.ipu.ru/labs/lab31kom/ggs.zip>
7. Стецюра Г.Г. (2008). Совмещение вычислений и передачи данных в системах с коммутаторами. *Автоматика и телемеханика*, № 5. С. 170–179.
8. Стецюра Г.Г. (2010а) Быстрые децентрализованные алгоритмы устранения конфликтов и тупиков при доступе к ресурсам в системах обработки данных и управления. *Автоматика и телемеханика*, № 4. С. 181–190.