ОРГАНИЗАЦИЯ ОБЩЕСТВЕННЫХ GRID-СИСТЕМ НА БАЗЕ ПИРИНГОВЫХ СЕТЕЙ

А.В. Святцев
Беларусь, г. Минск

Введение

Общественные GRID-системы, построенные на базе объединения в GRID-систему различных ЭВМ её участников, являются альтернативой корпоративных GRID-систем, построенных на базе кластерных систем, с точки зрения архитектуры, но с точки зрения решаемых задач занимают свою нишу, дополняющую спектр задач, решаемых корпоративными GRID-системами.

Эту нишу составляют нерегулярные по времени появления задачи, требующие выделения большого объёма вычислительных ресурсов на короткое время и не критичные к срочности выполнения. Если вспомнить, что суть GRID-систем – это создание сети, предоставляющей надёжный, согласованный и недорогой доступ к вычислительным ресурсам, наподобие доступа к ресурсам генераторов электроэнергии, предоставляемого электрическими сетями, то такой класс задач является для них целевым и должен ими решаться [1].

Для построения общественных GRID-систем имеет смысл использовать близкие к ним по своей сути уже существующие и развёрнутые пиринговые сети (peer-to-peer network (англ.) – сеть взаимообмена информацией).

Организации общественных GRID-систем на базе пиринговой сети Bittorent и посвящена данная работа.

Задачи и общие принципы общественных GRID-систем

Общественной можно считать любую GRID-систему, не имеющую определённого владельца (в противовес корпоративным системам). Аппаратное и программное обеспечение узлов такой сети является собственностью их владельцев и приобретается ими самостоятельно, общими являются только сетевые протоколы. Это не даёт свести архитектуру такой системы к конкретному высокопроизводительному решению и вынуждает использовать глобальную сеть Internet. Однако общественные GRID-системы имеют и положительные стороны, дающие им право на жизнь:

1)Громадный ресурс потенциальных узлов. К общественной GRID-системе может быть подключен любой компьютер, имеющий доступ в Internet.

2)Экономия на оборудовании. Общественная GRID-система может быть создана на базе существующей аппаратно-программной инфраструктуры сети Internet. Более того, возможна их организация на базе уже развёрнутых высокоуровневых пиринговых сетей.

Организация общественной GRID-системы на базе пиринговой сети

Структуру уже существующей пиринговой сети можно рассматривать как совокупность протокола, набора программных агентов и узлов со стоящими на них программными агентами, обменивающихся по данному протоколу. Для создания на её базе GRID системы необходимо выполнить следующие действия:

1)Дополнить протокол возможностями, позволяющими проводить на его основе обмен служебной для GRID-системы информацией между узлами сети.

2)Добавить в существующие программные агенты поддержку расширенного протокола или разработать новый программный агент, имеющий его поддержку изначально.

3)Заинтересовать пользователей использовать агенты, обладающие поддержкой расширенного протокола.

Основные принципы работы BitTorrent

Протокол BitTorrent очень простой, но в то же время он позволил создать эффективную и популярную пиринговую сеть.

Основная идея протокола состоит в максимально равномерном распределении нагрузки между узлами сети, что делается разбиением передаваемой информации на блоки и передачей блоков теми узлами сети, которые ещё не получили всё, но имеют хотя бы один уже полученный блок. За счёт этого время загрузки информации при росте числа клиентов не растёт, а, наоборот, уменьшается [2].

При распределённой передаче данных обмен служебной информацией (в первую очередь, списками активных узлов), напротив централизованный и проводится через выделенный сервер называемый tracker-ом, который поддерживает виртуальную сеть для каждого ресурса. Публикатор ресурса формирует для него небольшой дескриптор, содержащий ссылку на tracker, небольшую служебную информацию, и хэш-коды для проверки целостности закачанной информации. Дескриптор публикуется

в Internet, и этим дескриптором пользуются программные агенты для получения/выдачи ресурса через пиринговую сеть [3].

Расширение протокола BitTorrent для построения на его базе общественной GRID-системы

При модификации автором протокола BitTorrent ставились следующие ограничения:

1)Модификация должна быть именно расширением, сохраняя все стандартные возможности протокола с таким расчетом, чтобы при попытке запуска дескриптора BitTorrent не происходило критических ошибок у клиентов без поддержки данного расширения.

2)Расширение не должно перекрывать основные свойства протокола BitTorrent, а, наоборот, должно их использовать.

Таблица 1

Обычный BitTorrent дескриптор (в левой колонке имена полей, в правой - значения)

announce	http://tracker.org	
info	name	Information.txt
	piece length	4096
	length	29876
	pieces	AAAAAAAAAAAAAA AAAAAAAAAAAAAA AAAAAAAAAAAAAA

Расширенный BitTorrent дескриптор (новые поля выделены цветом)

announce	http://tracker.org		
info	name	Calculation.task	
	piece length	4096	
	length	32768	
	dynamic data offset	10	
	executable data offset	30	
	pieces	AAAAAAAAAAAAAA BBBBBBBBBBBBBB CCCCCCCCCCCCCC	
	dependencies	1	0,0
		2	1,0,1
		3	2,2

Сам протокол BitTorrent, как и любой другой протокол пиринговой сети, предназначен для передачи данных и никаких встроенных возможностей работы с программным кодом не имеет. Однако это ограничение можно обойти разделением всего спектра данных на три класса:

1)Статические данные – входные данные для выполняющего вычисления программного кода. При стандартном использовании BitTorrent все данные статические.

2)Динамические данные - выходные данные для выполняющего вычисления программного кода. Эти же данные могут в свою очередь являться входными для другого программного кода, также выполняющего вычисления.

3)Исполняемые данные - программный код.

В результате такого разделения появляется надобность в трёх дополнительных служебных полях:

1)Смещение начала области динамических данных.

2)Смещение начала области исполняемых данных.

3)Список зависимостей.

Формат BitTorrent дескриптора, представляет собой bencoded строку (способ кодирования взят из языка программирования Python). А так как, во-первых, строки такого формата расширяемы путём добавления новых полей, а, во-вторых, все нестандартные поля просто игнорируются, то добавление этих полей не нарушает ни одно из поставленных ограничений. Хэш-коды статических и исполняемых данных вычисляются обычным образом, в то время как хэш-коды динамических данных заведомо инвалидные.

Проектирование и кодирование программ для выполнения в общественной GRID-системе на базе BitTorrent

В качестве программной платформы для общественной GRID-системы с разнородным характером аппаратного и программного обеспечения её узлов и необходимостью предоставить узлам гарантии того, что передаваемый им на выполнение код не причинит им вреда (эта надобность появляется только для общественных GRID-систем, в корпоративных такого нет и близко) больше всего подходят платформенно-независимые технологии с контролируемым исполняемым кодом. Одной из таких технологией является Java.

При проектировании программы она должна быть разбита на блоки, образующие направленный мультиграф зависимостей по данным. Вершинами мультиграфа выступают примитивные программные блоки, получающие некоторые входные данные и создающие некоторые выходные данные (второе обязательно, первое - нет), дугами – промежуточные данные. Затем к мультиграфу добавляются мнимые вершины для входных и выходных данных всей программы, соединяемые с реальными

блоками дугами, соответствующим входным и выходным данным всей программы.

Данный подход предназначен для решения задач статической размерности, когда размерность данных известна заранее или может быть определена или с небольшой погрешностью оценена по входным данным. Параллельный запуск блока в количестве n экземпляров означает, что для каждого экземпляра должна быть отдельная дуга, соответствующая его выходным данным.

Последовательный программный код

```
public double[][] multiply
    (double[][] a, double[][] b)
{
  double[][] result = new
double[N][N];

  for (int i=0; i<N; i++)
    for (int j=0; j<N; j++) {

      double cellValue = 0;
      for (int k=0; k<N; k++)
        cellValue += a[i][k] *
                     b[k][j];
      result[i][j] = cellValue;
    }

  return result;
}
```

Блоки программы для параллельного выполнения предложенной GRID-системой

Блок 0 *мнимый блок, поставляющий* входные данные

Блок 1 *код генерирующий для заданного* внутреннего номера выходной дуги outNum динамические данные

```
double result = 0;
for (int k=0; k<N; k++)
  result += a[outNum / N][k] *
            b[k][outNum % N];
```

Блок 2 *мнимый блок, принимающий* выходные данные

Мультиграф структуры программы для параллельного выполнения предложенной GRID-системой: каждая стрелка объединяет N^2 дуг

Рис. 1. Пример создания мультиграфа структуры программы для задачи перемножения 2-х квадратных матриц размерности n

Когда мультиграф со структурой программы задан, создаётся одна мнимая входная вершина с дугами, ведущими от неё ко всем мнимым вершинам. соответствующих входным данным, и с помощью алгоритма заливки выполняется нумерация вершин мультиграфа, исключая самую

первую. Полученные пронумерованные блоки, кроме мнимых, соответствуют исполняемым данным

Затем каждый блок должен быть закодирован в отдельное небольшое Java-приложение, и упакован в jar-архив. При таком подходе данный блок может быть запущен из любого клиента, реализованного не только на Java, в отличие от менее ресурсоёмкого подхода с кодированием каждого блока в отдельный Java-класс. Входные данные и внутренний номер выходных данных передаются с помощью параметров запуска.

Затем выполняется нумерация дуг, исходящих из мнимых вершин, с условием последовательной нумерации всех исходящих из блока дуг в его внутренней последовательности. Эти дуги соответствуют статическим данным.

Затем выполняется нумерация всех оставшихся дуг, с условием исключения номеров, занятых дугами, соответствующими статическим данным и таким же условием нумерации. Эти дуги соответствуют динамическим данным.

На базе полученной информации формируется список зависимостей (ключи в списке – номера дуг, соответствующих динамическим данным, значение формируется перечислением блока, являющегося началом дуги, и входящих в этот блок дуг) и собирается расширенный BitTorrent дескриптор. Дескриптор публикуется, исходные данные и блоки кода предоставляются инициатором вычислений через пиринговую сеть. Далее, при наличии пользователей с расширенными клиентами, начинается процесс вычислений.

Заключение

В данной работе рассмотрен класс общественных GRID-систем, показана возможность их развёртывания на базе существующих пиринговых сетей и на концептуальном уровне описаны архитектура и основные принципы работы общественной GRID-системы на базе пиринговой сети BitTorrent.

Литература

1. GRID-системы: Отчёт о НИР (заключительный): БГУ; Руководитель А.Н. Курбацкий.

2. Bram Cohen: Incentives Build Robustness in BitTorrent, 2003 - http //www.bittorrent.com/bittorrentecon.pdf

3. BitTorrent protocol specification - http://www.bittorrent.com/protocol.html

4. Буза М. К. Введение в архитектуру компьютеров / Учебное пособие. Мн.: БГУ, 2000. 253 с.