ISBN 5-94052-127-4 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ. ПЕРЕСЛАВЛЬ-ЗАЛЕССКИЙ, 2006

УДК 004.75

Ф. А. Коряка, А. А. Московский, А. Ю. Первин

Создание испытательного полигона для Grid-приложений в городе Переславле-Залесском

Аннотация. В работе описывается разработка испытательного полигона для Grid-приложений в городе Переславле-Залесском. Затрагиваются основные принципы функционирования сети T-Grid и методы их реализации. Особое внимание уделено прототипу сети T-Grid созданному на базе нескольких машин и кластеров.

Ключевые слова и фразы: Grid, вычислительная сеть, виртуальные машины, UML, T-Система.

1. Введение

Благодаря развитой инфраструктуре городской компьютерной сети, Переславль-Залесский является удобным полигоном для разработки испытательного сегмента Grid. В городе расположен Институт программных систем Российской академии наук (порядка 30 вычислительных машин и 4 кластера), Университет г. Переславля (порядка 50 вычислительных машин), а также около 600 абонентских подключений к сети. Скорость передачи данных во внутренней сети г. Переславля-Залесского составляет от 10 до 1000 Mb/sec. В проекте T-Grid предполагается, что вычислительную сеть формируют добровольцы, которые устанавливают на свои машины программное обеспечение, предоставляемое организаторами сети T-Grid. При этом, часть ресурсов компьютера (процессорное время и дисковое пространство), подключенного к сети T-Grid оказывается в распоряжении администраторов сети.

2. Реализация

2.1. Основные принципы реализации. При подобной организации сети, наиболее острыми проблемами становятся ограничение ресурсов и управление конфигурацией. Решение было найдено с использованием технологии виртуальных машин, а именно, User Mode

Linux (UML). На машину, подключенную к сети, в полуавтоматическом режиме устанавливается образ ОС Redhat 9.0, который затем запускается как пользовательский процесс, предоставляя виртуальную Linux машину в распоряжение сети Grid. При этом владелец компьютера имеет все возможности по ограничению ресурсов, доступных из виртуальной машины. В то же время, виртуальная машина находится под полным контролем администрации сети, а процедуры обновления образа ОС виртуальной машины и управления пользователями T-Grid автоматизированы. Виртуальные машины объединяются в сеть, предоставляя возможности пользователям, зарегистрированным на головной машине сети запускать для выполнения программы. В сети установлены библиотеки MPI (MPI–PACX), позволяющие проводить параллельные вычисления. Как средство организации параллельных вычислений также используется T-система, позволяющая сравнительно быстро создавать параллельные программы с динамическим управлением нагрузкой и устойчивых к латентности коммуникационной сети. Мониторинг сети производится с помощью системы мониторинга Flame. На основе мониторинга создано решение, позволяющее запустить параллельную задачу на всех узлах сети, активных в данный момент времени.

2.2. Установка и функционирование программного обеспечения вычислительного узла. Для инсталляции программного обеспечения вычислительного узла необходимо загрузить инсталлятор tg-exec-install [1]

Запуск инсталлятора следует выполнять с суперпользовательскими привилегиями. Работа инсталлятора делится на несколько этапов:

1-ый этап — Тестирование системы. На данном этапе происходит тестирование системы с целью определения типа дистрибутива операционной системы и наличия необходимых компонентов. В настоящее время поддерживаются следующие типы дистрибутива операционных систем: Debian Linux, RedHat Linux. Необходимыми компонентами являются: wget — утилита для загрузки пакетов программного обеспечения вычислительного узла, iptables — утилита для настройки фильтрации пакетов и NAT. Также осуществляется проверка наличия следующих модулей ядра: iptables — модуль позволяющий

[1] `http://tgrid.botik.ru/download/tg-exec-install`

осуществлять фильтрацию пакетов и NAT, tun — модуль позволяющий создавать на физической машине виртуальные сетевые интерфейсы.

В случае, если инсталлятором не найден какой-либо из необходимых компонентов (кроме wget), производится их загрузка с узла tgrid.botik.ru и установка на машину пользователя.

2-ой этап — Загрузка необходимых компонентов. На данном этапе происходит загрузка компонентов программного обеспечения вычислительного узла с сервера tgrid.botik.ru В зависимости от дистрибутива Linux установленного на машине пользователя загружаются либо deb, либо rpm пакеты с последней версией UML и UML-utilites а также пакет tg-exec в котором содержатся необходимые для функционирования UML в среде T-Grid компоненты и образ виртуальной системы.

3-ий этап — Установка необходимых компонентов. На данном этапе происходит установка всех необходимых компонентов и предварительная настройка системы на физической машине.

4-ый этап — Конфигурация виртуальной сети. В большинстве случаев, на данном этапе можно использовать «базовую конфигурацию», в этом случае на машине пользователя будет запускаться одна виртуальная машина с IP-адресом 192.168.0.10 В случае, когда использование IP-адреса 192.168.0.10 затруднено или же необходим запуск нескольких виртуальных машин одновременно, рекомендуется использовать «ручную конфигурацию». В этом случае инсталлятор поможет сконфигурировать сеть желаемого вида и сообщит о ней такие сведения, как IP-адреса виртуальных узлов, порты по которым будет осуществляться доступ и т. д.

5-ый этап — Настройка системы. На данном этапе происходит окончательная настройка системы, настройка правил маршрутизации и т. д. После успешного завершения пятого этапа, система пользователя будет полностью готова к запуску на ней виртуальных узлов.

2.3. Конфигурация системы. При установке на машину UML ее необходимо соответствующим образом сконфигурировать. Это в первую очередь касается конфигурации iptables. Виртуальные машины при запуске получают приватный адрес из диапазона 192.168.0.10–192.168.0.99

Первая виртуальная машина получает адрес 192.168.0.10, вторая 192.168.0.11 и т. д. При этом, физическая и виртуальная машины могут обмениваться IP-пакетами без затруднений, в то время как доступ к виртуальной машине из внешней сети не возможен. Поскольку, для осуществления распределенных вычислений необходима коммуникация с виртуальной машиной по протоколу SSH и некоторым другим, то на физической машине необходимо задать несколько правил маршрутизации:

```
iptables -t nat -A PREROUTING -p tcp --dst PARENT_IP
        --dport 50010 -j DNAT --to 192.168.0.10:22
```

Данное правило указывает на необходимость переправлять все пакеты, пришедшие на 50010-ый порт физической машины на 22-ой порт виртуальной. Номер порта образуется прибавлением к 50000 номера виртуальной машины. Например, для машины 192.168.0.11 это будет 50011-ый порт.

Кроме 22-го порта необходимо сделать аналогичным образом возможность хождения пакетов на 31000-ый порт, который используется PACX-приложениями. Далее процесс функционирования PACX-приложений будет описан более подробно, здесь же отметим лишь тот факт, что хождение пакетов на 31000-ый порт необходимо лишь на первом виртуальном узле то есть 192.168.0.10. Поэтому, независимо от количества виртуальных узлов, правило касающееся 31000-ого порта всегда будет одно:

```
iptables -t nat -A PREROUTING -p tcp --dst PARENT_IP
        --dport 31000 -j DNAT --to 192.168.0.10:31000
```

Заметим, что переопределять порт на физической машине, как правило, не имеет смысла, поскольку вероятность запуска на ней PACX-приложений мала.

Однако, данные правила позволяют IP-пакетам проходить только в одностороннем порядке. Для полноценного хождения пакетов необходимо задать следующее правило:

```
iptables -t nat -A POSTROUTING -s 192.168.0.10
      --out-interface eth0 -j SNAT --to-source PARENT_IP
```

Во всех описанных правилах PARENT_IP соответствует IP адресу физической машины через который осуществляется выход во внешнюю сеть.

2.4. Запуск виртуальной машины. Для запуска виртуальной машины используется скрипт tg-exec из одноименного пакета. Формат запуска скрипта следующий:

```
tg-exec (start, stop, console, reboot, status, kill, mount, umount) [childnum]
```

start: Запуск новой виртуальной машины.

stop: Остановка виртуальной машины. Аналогично нажатию кнопки power на физической машине и может привести к сбою в работе системы.

reboot: Перезагрузка виртуальной машины. Аналогично нажатию кнопки reset на физической машине и может привести к сбою в системе.

cad: Нажатие Ctrl+Alt+Del на виртуальной машине.

console: Запуск утилиты mconsole, предназначенной для управления запущенной виртуальной системой.

status: Выдача текущего состояния.

kill: Посылка 9-ого сигнала процессу, в рамках которого работает виртуальная машина.

mount: Мантирование образа виртуальной системы.

du: Показ статистики использования диска.

childnum: Номер виртуальной машины, с которой производятся манипуляции. childnum должен лежать в диапазоне от 10 до 99 включительно и имеет значение по умолчанию 10.

Таким образом, для запуска одной виртуальной машины достаточно выполнить команду tg-exec start, а для запуска двух виртуальных машин потребуется выполнение команд tg-exec start 10, tg-exec start 11.

Все виртуальные машины используют один образ виртуальной системы, который называется rootfs. Образ используется в режиме «только для чтения». При этом, во время запуска виртуальной машины создается cow-файл в котором хранятся изменения эталонного rootfs применительно к данной виртуальной машине. cow-файлы имеют следующие имена: rootfs_N_cow, где N — номер виртуальной машины. Таким образом, в случае какого-либо сбоя на виртуальной машине, достаточно удалить cow-файл, принадлежащий данной машине, в этом случае запуск машины произойдет с эталонного образа.

В случае тестирования с запуском большого числа виртуальных машин, рекомендуется после окончания тестирования удалять ненужные cow-файлы, поскольку они занимают достаточно большой объем дискового пространства.

3. Взаимодействие с пользователями в сети T-Grid

3.1. Регистрация. Предварительная регистрация пользователей в сети T-Grid производится на сайте `http://tgrid.botik.ru/`.

После предварительной регистрации и ее подтверждения учетная запись будущего пользователя передается администратору сети T-Grid. Администратор сети T-Grid имеет набор утилит необходимых для удаления либо добавления пользователей в основную базу сети T-Grid.

Пользователи, прошедшие регистрацию и добавленные администратором сети T-Grid в основную базу пользователей, могут запускать на счет свои задачи в сети T-Grid.

Существует два вида пользователей сети T-Grid: локальные и глобальные. Глобальные пользователи при регистрации указывают свой публичный SSH-ключ и далее могут запускать задачи на счет с любой машины, на которой установлен секретный ключ соответствующий указанному публичному. Локальные пользователи не указывают при регистрации свой SSH-ключ, а запуск задач могут осуществлять только со специально предназначенной для этих целей виртуальной машины, расположенной на физической машине:

```
tgrid.botik.ru
```

На данной машине для всех локальных пользователей заводится учетная запись, а так же создается пара SSH-ключей. Доступ к машине осуществляется через специальный SSH-апплет находящийся на сайте `http://tgrid.botik.ru/`

3.2. Пакет программ T-Grid Users. Для управления пользователями в вычислительной сети T-Grid предназначен пакет программ T-Grid Users. Данный пакет состоит из двух основных частей:

(1) набор программ для управления БД пользователей;
(2) набор программ в образе виртуального вычислительного узла для синхронизации пользователей на всех вычислительных узлах.

Далее каждая из этих частей будет рассмотрена более подробно.

3.3. Набор программ для управления БД пользователей. Данный набор программ предназначен для заведения и удаления администратором сети T-Grid пользователей. Данные о всех пользователях хранятся в БД представляющей из себя плоский файл со строками следующего формата:

```
...
user_name     user_id      user_key
...
user_name     user_id      user_key
...
```

где user_name — имя пользователя в вычислительной сети T-Grid; user_id — уникальный идентификатор пользователя на всех вычислительных узлах сети T-Grid. На идентификатор введено ограничение: он не может быть меньше 5000. Данное ограничение обусловлено возможной в будущем проблемой создания пользователей на системах уже имеющих своих пользователей. В этом случае, идентификаторы T-Grid пользователей не будут конфликтовать с уже существующими. Конечно, данное правило верно, не для всех систем. Поле user_key — публичный ssh ключ пользователя, использующийся для беспарольного входа пользователя на все вычислительные узлы сети T-Grid.

Данная БД хранится на коммуникационных узлах сети T-Grid и доступна для загрузки по протоколу http.

Набор программ для управления БД пользователей состоит из следующих компонентов:

- tg-adduser
- tg-deluser

Оба данных компонента устанавливаются на машине администратора сети T-Grid и используются исключительно администратором

Программа tg-adduser предназначена для заведения пользователей в сети T-Grid. Данная программа представляет из себя скрипт на языке perl который запускается без параметров и работает в интерактивном режиме.

После запуска скрипта tg-adduser необходимо ответить на его вопросы, в часности:

- Имя пользователя UID — нажатие enter без указания идентификатора приведет к тому, что скрипт сам создаст идентификатор на основании уже использованных;
- Ключ — необходимо ввести публичный SSH-RSA ключ пользователя.

После получения всей необходимой информации, скрипт записывает в БД информацию о пользователе и оповещает все узлы о необходимости произвести синхронизацию.

Для удаления пользователей из сети T-Grid предназначен скрипт tg-deluser, который запускается с одним параметром: `имя пользователя` которого нужно удалить.

3.4. Набор программ в образе виртуального узла для синхронизации пользователей на всех вычислительных узлах. В состав вычислительного узла входит сервис tg-users-settings, который при загрузке узла:

- подключается к коммуникационному узлу;
- забирает с коммуникационного узла БД пользователей;
- создает и удаляет на своем узле пользователей в соответсвии с БД;
- осуществляет копирование публичных ключей для предоставления беспарольного доступа.

Кроме того, сервис tg-users-settings ожидает соединения на 50200 порт. Которое инициирует алгоритм синхронизации пользователей описанный выше.

4. Экспериментальный метакластер

4.1. Состав метакластера. В общем случае в вычислительной сети T-Grid возможно три типа вычислительных узлов:

(1) Однопроцессорный вычислительный узел с одним виртуальным хостом.
(2) Многопроцессорный вычислительный узел с двумя и более виртуальными хостами.
(3) Кластер состоящий из нескольких вычислительных узлов.

В качестве прототипа сети T-Grid на тестовых испытаниях был использован метакластер конфигурацией:

- brick — Celeron 1.70GHz, 256 Mb
- shura — Pentium III 1.0GHz, 512 Mb

- кластер «тестовый»:
  - фронтальная машина — 2x Pentium III 600 Mhz, 512 Mb
  - 4 узла — 2x Pentium III 600 Mhz, 512 Mb

Все узлы были связаны между собой сетью Ethernet.

Использование именно этих машин было обусловлено желанием попробовать в метакластере все возможные виды вычислительных узлов:

- shura — вычислительный узел первого типа.
- brick — вычислительный узел второго типа. [2]
- Кластер «тестовый» — вычислительный узел третьего типа.

4.2. Устройство виртуального узла. Программное обеспечение виртуального узла состоит из ОС Linux RedHat 7.2 с установленными на него gcc v 3.2 и glibc v2.3. Кроме того, на виртуальный хост установлены следующие дополнительные программные продукты: OpenTS, LAM, PACX. В процессе загрузки виртуальной системы инициализационный скрипт tg-settings производит настройку системы в соответствии с переданными ядру параметрами. В частности производится установка IP адреса, имени хоста для виртуальной машины. Далее скрипт проводит синхронизацию списка локальных пользователей с глобальным списком.

5. Результаты

Проведение тестовых испытаний показало целесообразность использования UML как средства построения вычислительной распределенной сети.

Одним из главных плюсов данного подхода является простота изменения настроек всех виртуальных узлов. Единожды сделав изменения на эталонном узле, нам достаточно распространить эталонный образ виртуальной системы на другие узлы. Так же использование UML позволяет достичь достаточно высокой отказоустойчивости. В случае поломки системы, будь то небольшой сбой или полное разрушение системы, достаточно перезагрузить виртуальную машину. Система так же удобна с точки зрения безопасности. В случае проникновения злоумышленника на вычислительный узел, он попадает

[2] Скорее имитация вычислительного узла второго типа, поскольку виртуальных узлов действительно было два в то время как процессор всего один.

на виртуальную машину, при этом физическая машина остается для него недоступной.

В качестве тестовых задач были использованы тесты: Bandwidth и NASA EP.

6. Благодарности

Работы проводятся в рамках проекта Президиума РАН «Функционально-ориентированные Т-суперструктуры как эффективное средство для построения высокопроизводительных распределенных приложений и сервисов».

Список литературы

[1] Сайт проекта T-Grid. — http://tgrid.botik.ru/.
[2] Сайт технологии UML. — http://user-mode-linux.sourceforge.net/.
[3] Ресурсы разработчиков OpenTS. — http://t-system2.polnet.botik.ru/.

Исследовательский центр мультипроцессорных систем ИПС РАН

P. A. Koryaka, A. A. Moskovsky, A. Y. Pervin. *Development of test area for Grid-applications in Pereslavl-Zalessky.* (in Russian.)

Abstract. This work describes development of a test area for Grid-applications in Pereslavl-Zalessky, deals with the main principles of functionality of T-Grid network and methods of its realization. Special attention is paid to a prototype T-Grid network consisting of several computers and clusters.