Глава 5.
Установка, настройка и оптимизация системного программного обеспечения

От вопросов, связанных с аппаратной составляющей кластера, перейдем к базовому и специализированному программному обеспечению. Заметим сразу, что в задачи данного раздела не входит обучение установке стандартных вариантов операционных систем. Это уже описано во множестве книг и предполагается, что читатель обладает необходимыми минимальными навыками и знаниями. Сосредоточимся на “кластерных” особенностях, позволяющих множеству независимых компьютеров согласованно работать в рамках единого комплекса.

Стандартом de-facto **операционной системы** для вычислительных кластеров в настоящее время является Linux. Какой дистрибутив предпочесть? Иногда это решает сам системный администратор, который будет поддерживать работу кластера, в некоторых случаях выбор однозначно определяется прикладным программным обеспечением, оптимизированным под конкретную версию ОС. С точки зрения собственно построения кластера большой разницы в дистрибутивах нет, однако специализированные прикладные пакеты могут оказаться к ним чувствительными. Это же касается и драйверов ко всему набору аппаратного обеспечения, особенно к новым моделям или нестандартному оборудованию.

В последнее время все большую популярность приобретает система Windows Compute Cluster Server 2003, разработанная компанией Microsoft для поддержки кластерных платформ. Вариант интересный, особенно если учесть большой объем прикладного ПО, работающего именно под MS Windows. Его миграция под кластерный вариант этого семейства операционных систем, безусловно, является лишь вопросом

времени. Однако к настоящему моменту во всем мире опыта использования Windows Compute Cluster Server 2003 не много, система только недавно анонсирована, поэтому в данной работе мы будем предполагать, что выбрана ОС семейства Linux.

Существует несколько продуктов, позволяющих провести быструю установку и самую начальную базовую настройку целого кластера. В качестве примеров можно назвать достаточно популярные на практике пакеты Rocks (`http://rocksclusters.org/`) или OSCAR (`http://oscar.openclustergroup.org/`). С их помощью можно установить на сервер и кластерные узлы готовые образы Linux, которые в дальнейшем дополняются необходимыми пакетами и настраиваются на работу в кластере. Стоит отметить дистрибутив Parallel Knoppix (`http://idea.uab.es/mcreel/ParallelKnoppix/`), который нет необходимости устанавливать ни на сервер, ни на узлы, а достаточно просто загрузиться с CD и задать конфигурацию кластера.

Рассматривая возможность использования продуктов подобного рода, нужно четко представлять возможные последствия этого решения. С одной стороны, необходимо понижать трудоемкость сопровождения и администрирования кластерных систем за счет автоматизации рутинных и предписанных регламентом процессов – это правильно, в таком направлении как раз и идет развитие данной области, в частности, для снижения стоимости владения сложными компьютерными системами. Но с другой стороны, кластерная система с самого начала должна быть максимально эффективной по отношению к задачам и оставаться такой все время своего существования. Значительным недостатком упомянутых выше продуктов является то, что в качестве основы коммуникационной среды везде предполагается Ethernet и взаимодействие через TCP/IP, что может самым печальным образом сказаться на эффективности работы кластерной системы в целом. Если для работы в таком режиме система и ставилась, если есть уверенность, что все остается под контролем, или есть время для накопления опыта и экспериментов, то вполне можно

воспользоваться и таким путем.

Установив и настроив обычный вариант операционной системы на головном узле кластера, проведем аналогичную операцию на файловом сервере. Настоятельно рекомендуем использовать Logical Volume Manager для раздела, на котором будут располагаться данные пользователей кластера. Это позволит легко проводить дальнейшее расширение хранилища и безболезненно переносить логический раздел с данными пользователей в будущем.

На головную машину операционная система ставится обычным образом. А вот для того, чтобы **установить ОС на вычислительные узлы кластера**, как правило, требуются дополнительные усилия. С одной стороны, на вычислительных узлах не принято устанавливать привычные в такой ситуации CD или Floppy-приводы, а с другой – установить и настроить ОС на десятках узлов само по себе является занятием долгим и утомительным.

Для упрощения процесса можно воспользоваться виртуальными приводами, если они поддерживаются сервисной сетью. Если нет, то достаточно подключить через USB или напрямую к одному из узлов CD-привод и провести установку операционной системы на этом узле.

Проведя начальную установку и стандартную настройку узла, нужно сделать еще несколько шагов для его дальнейшей работы в составе кластера. Убедитесь, что на узле установлен сервер `ssh`, и что пользователю `root` разрешен удаленный вход. Настройте беспарольный вход на узел для пользователя `root`, используя авторизацию по ключу, для чего воспользуйтесь командой `ssh-keygen`. Необходимо иметь возможность заходить на этот узел привилегированным пользователем `root` с головного узла – это значительно облегчит жизнь в дальнейшем.

Настройте на файловом сервере сетевой каталог, разрешив доступ к нему с вычислительных узлов и головного узла. Это можно сделать, прописав соответствующую строку в файл `/etc/exports` и запустив

сервер NFS. Убедитесь, что сервер NFS стартует автоматически при загрузке файл-сервера.

Крайне желательно добавить в строку экспорта в файле `/etc/exports` опцию `no_root_squash`. По умолчанию NFS отменяет права суперпользователя для удаленных хостов, а указанная опция не позволяет этого сделать.

На вычислительном и головном узлах создайте каталог с одинаковым именем (например, `/common` или даже `/home`) и настройте автоматическое монтирование в него каталога с файл-сервера. Это можно сделать, прописав соответствующую строку в файл `/etc/fstab`. Убедитесь, что каталог монтируется без проблем, до того как будете перезагружать узлы!

Обратите внимание на то, что многие современные дистрибутивы (такие как SuSE, RedHat, Fedora Core и другие) делают жесткую привязку настроек сетевого интерфейса к MAC-адресу сетевой карты. Если просто скопировать такие настройки на другой узел, сетевой интерфейс просто не заработает. Для того чтобы настройки можно было безболезненно перенести на любой узел, необходимо убрать привязку к MAC-адресу из файла `/etc/sysconfig/network/ifcfg-eth-XXXXXX`, если она там есть, переименовать этот файл в `ifcfg-eth0` или `ifcfg-eth1`. Точно также необходимо убрать явные переименования сетевых интерфейсов в подсистеме `udev`. Чтобы найти остальные не столь очевидные привязки к MAC-адресу, поищите все его упоминания командой '`grep -ri MAC /etc`', где `MAC` замените реальным значением MAC-адреса. Узнать его можно командой `ifconfig`.

Синхронизация времени в системе. Она осуществляется с помощью пакета `xntp` или его аналогов. При существенном расхождении часов на различных узлах могут наблюдаться сбои в работе параллельных программ. Необходимо настроить ntp-сервер на головном узле, а на всех остальных узлах – клиентов, которые будут с ним синхронизироваться. При старте все узлы должны явно синхронизироваться с головным узлом

командой `ntpdate` (она обычно входит в состав пакета `xntp`), что необходимо, поскольку при большом расхождении часов клиент `ntp` коррекцию времени может и не выполнить.

Отслеживание состояния UPS. Большинство современных источников бесперебойного питания способны сообщать о своем текущем состоянии через COM-порт, USB или по сети через SNMP. Некоторые производители включают программы под Linux для отслеживания состояния UPS в поставку, но, увы, делают это далеко не все. Существует свободный пакет для мониторинга состояния UPS, который называется NUT. Он поддерживает большое число моделей UPS, но перед покупкой оборудования лучше проверить, поддерживается ли конкретная модель. NUT включает в себя сервер, снимающий данные с UPS, и клиентов. Сервер работает на головном узле, к нему же должен быть подключен управляющий кабель UPS. Клиенты должны быть установлены на все узлы. В случае отключения питания клиенты дадут команду на выключение узлов. Таким образом, можно избежать потери данных и сохранить работоспособность кластера. Если UPS достаточно мощный и может поддерживать работу кластера в течение какого-то времени, то команда выключения узлов может быть настроена на отсроченное выключение. В этом случае, если электропитание восстановится, то выключение будет отменено.

Синхронизация системных файлов (`passwd`, `shadow`, `hosts`, ...). Для того, чтобы системные изменения затрагивали не только головной узел, но весь кластер сразу, можно применять различные схемы. NIS – это одно из самых простых решений, которое, однако, чревато проблемами в случае сетевых сбоев. Следует специально отметить, что, несмотря на “простоту”, хорошая настройка этой схемы может оказаться весьма нетривиальным делом для неискушенного администратора. Использование `rsync` является более простым решением, требующим лишь включения нужного сервиса на всех узлах и начальной настройки на головном узле. Третий вариант можно условно назвать “ручным” копированием, что

предполагает использование `scp` в скрипте для автоматического дублирования всех нужных файлов на узлы. Каждый из перечисленных методов требует явного вызова определенной команды после изменения системных файлов. Есть и другие методы, но все они, в целом, аналогичны `rsync`.

Следующим шагом в настройке программного обеспечения кластерной системы является тиражирование установленной ОС на все остальные узлы. Для этого можно воспользоваться тремя схемами.

- Берем любой доступный компьютер. Вставляем в него жесткий диск из вычислительного узла с уже настроенной ОС и один (или несколько) дисков из других узлов. Затем с помощью команды `dd` копируем на чистые диски содержимое первого диска.

- Можно воспользоваться программой типа Acronis True Image или Norton Ghost для клонирования диска первого узла на диски остальных узлов.

- Можно создать образ установленной ОС с помощью архиватора `tar` или `cpio` (исключите при архивировании содержимое файловых систем `proc`, `sysfs`, `devfs`, `usbfs` и им подобных). Установите `syslinux`. С помощью пакета `syslinux` и серверов `dhcpd` настройте сетевую загрузку. Далее нужно создать сетевой NFS-диск, на котором будет создана минимальная система, достаточная для подготовки жесткого диска (разбиение и форматирование), для разворачивания архива с образом ОС и установки загрузчика. Убедитесь, что ядро данной минимальной системы поддерживает корневой каталог на NFS. Затем скопируйте в каталог сетевого диска образ ОС и в качестве стартового сделайте скрипт, который автоматически установит это образ. После этого произведите сетевую загрузку всех узлов, где операционная система еще не установлена.

Третий способ, конечно, сложнее, но он более универсален и позволяет в дальнейшем быстро добавлять новые узлы и восстанавливать

испорченные простой перезагрузкой (с указанием “грузиться по сети”). Для подготовки минимальной системы, которая будет грузиться по сети и устанавливать ОС на узлы, можно воспользоваться любым мини-дистрибутивом из сети Интернет или подмножеством программ из уже имеющегося дистрибутива.

В минимальной системе обязательно должны присутствовать: `bash`, `tar` (или `cpio`), `sfdisk`, `mke2fs` (`mkreiserfs` или иное в зависимости от выбранной файловой системы), полный пакет `grub` (или `lilo`) и набор библиотек с динамическим линкером, необходимые для работы этих программ.

С помощью программы `sfdisk` можно записать в файл разметку жесткого диска с первого узла и в стартовом скрипте использовать ее для разметки жестких дисков чистых узлов.

О безопасности кластера нужно позаботиться заранее. Не стоит полагаться на соображения типа: “Да кому нужно взламывать наш кластер?”. Будьте уверены, что желающих найдется много. Совсем не обязательно их целью будет помешать вашей работе. Скорее всего, задачей станет использовать взломанные компьютеры как плацдарм для будущих хакерских действий или рассылки спама.

О том, как повысить безопасность Linux-сервера, написано немало книг и статей, желательно с ними ознакомиться. Приведем лишь несколько основных советов.

- Для удаленного входа на кластер и удаленной передачи файлов используйте `ssh`. Под Windows есть немало программ, реализующих этот протокол, например, свободно распространяемая программа `putty`. Не используйте для этих целей `telnet`, `ftp`, `nfs` или `samba` (windows share).
- Отключите все ненужные сервисы.
- Включите файрволл, продумайте политику его использования.

- По возможности, дополнительно ограничьте доступ пользователей. Это можно сделать, задав список адресов, с которых разрешено заходить на головной узел, либо запретив авторизацию по паролю и обязав пользователей использовать для авторизации ключи.
- Включите "устаревание" паролей пользователей.
- Включите проверку сложности паролей.
- Установите одну из программ проверки целостности системы (такую, как `tiger`, `ossec`, `tripware`).
- Регулярно проверяйте журналы на головном узле, воспользуйтесь пакетом `logcheck` или `logwatch`.
- Время от времени проверяйте систему на наличие "закладок" программами типа `chkrootkit`, `rkhunter`. Не храните эти программы в доступном с головного узла каталоге, лучше запускайте их с USB-Flash или с дискеты.

О том, как произвести такие настройки, можно прочесть в man-страничках `chage`, `sshd`, `sshd_config`, `pam`, а также в документации к упомянутым пакетам. Подчеркнем еще раз: обеспечение безопасности – это очень важный вопрос. Сразу уделите безопасности особое внимание, поскольку решение этих проблем после обнаружения факта взлома уже может быть сопряжено с потерями.

В данном разделе мы сразу предположили использование NFS в качестве сетевой файловой системы кластера. Она хорошо известна, у многих есть опыт ее использования, поэтому именно на NFS чаще всего останавливается выбор администраторов. Но хочется предостеречь сразу: этот вариант, во-первых, не единственный и, во-вторых, совсем не идеальный. Основное слабое место связано с плохой масштабируемостью NFS и очень сильным падением характеристик ее работы при увеличении числа узлов. Какова цель кластерного проекта и чего хотелось бы достичь?

Максимум процессоров, максимум “флопсов”? Бывает и такое. В частности, именно такие требования выдвигают некоторые масштабные задачи физики высоких энергий, для которых чем больше в компьютерной системе вычислительных узлов, тем лучше. Если говорить про общий случай, то каждое приложение устанавливает некоторый порог, соотношение между вычислительными способностями кластера и характеристиками подсистем ввода/вывода, за которым выполнение приложения перестает быть эффективным, а использование кластера становится полностью нецелесообразным. Для параллельных приложений порог может меняться в зависимости от числа используемых процессоров, что создает дополнительные трудности в его определении на практике. Но сделать это необходимо, причем сделать на этапе проектирования архитектуры, чтобы вместо сбалансированной кластерной системы не получить однобоко развитого монстра. Здесь уместно вспомнить определение суперкомпьютера, приведенное в предисловии данной книги, согласно которому в системах подобного класса “проблема вычислений сводится к проблеме ввода/вывода”...

Альтернативные варианты файловых систем, к которым стоит приглядеться: Panasas File System, Lustre, Terragrid, Parallel Virtual File System (PVFS2), General Parallel File System (GPFS). Данные файловые системы изначально предназначались для параллельных компьютеров, они активно развиваются и реально используются на многих больших кластерных системах. Имеет смысл подумать об этих вариантах. Сделать “как все” не означает принять оптимальное для себя решение: не исключено, что именно данные файловые системы лучше всего подойдут для решения задач проекта.

Итак, на все узлы кластера установлена операционная система, проведена начальная настройка, кластер “задышал”. Следующим шагом будет переход к содержательной работе кластера и запуск параллельных программ. Для этого нам нужен набор компиляторов, одна или несколько

сред параллельного программирования, дополнительные библиотеки и пакеты, специализированные прикладные системы.