

# АВТОМАТИЧЕСКОЕ РАЗВЕРТЫВАНИЕ СИСТЕМНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Миллер А.П.

ФГУП НИИАА им. Семенихина, Москва, Россия.

Время от времени (в период этапа начальной загрузки и настройки, в ходе обновления при выпуске новых версий программного обеспечения (ПО), при расширении/масштабировании комплекса средств автоматизации (КСА) и т.д.) сотрудники службы эксплуатации сталкиваются с необходимостью переустановки операционной системы (ОС) на некоторых или даже всех компьютерах, входящих в КСА. При этом помимо непосредственно самой ОС необходимо установить и настроить различное ПО, загрузить на жесткие диски разнообразную информацию (например, важные наработанные данные из резервных копий, файлы конфигурации) и тому подобное. Для того чтобы сделать все это на одном вычислительном блоке, рядовому специалисту необходимо от 1 до 3 часов. В сочетании с относительно медленными по скорости передачи данных носителями (а именно дискеты (гмд), магнитооптические и cd/dvd диски, использующиеся в настоящее время) и, порой, не малым количеством различающихся по архитектуре вычислительных средств процесс установки и начальной настройки «съедает» не менее 80% времени, отведенного на развертывание ПО, при этом задачи выполняются однотипные.

Как вариант, можно использовать установку по сети, т.е. выделить т.н. сервер инсталляции, к которому в ходе установки обращаются за дистрибутивами системного и прикладного ПО компьютеры-клиенты. Тем не менее, для запуска ОС на «голом» вычислительном блоке все равно необходимо загрузить с носителя ядро ОС. Сервер инсталляции намного сократит время загрузки, но не решит основной проблемы – для загрузки каждой машины требуется физическое присутствие специалиста.

Между тем, сегодня есть технологии, которые с использованием Ethernet-сетей позволяют разворачивать и настраивать ОС и любое ПО очень быстро, причем одновременно на произвольном количестве вычислительных блоков и, что самое важное, полностью неинтерактивно, т.е. автоматически. Использование данных технологий позволит значительно облегчить и ускорить процессы загрузки, настройки и обновления ПО КСА, тем самым освобождая время профессиональных сотрудников для решения других важных задач, как, например, оптимизации, тонкой настройки ПО, отладки и т.д.

Помимо удобства начальной загрузки, система развертывания позволит в кратчайшие сроки перенастроить или полностью заменить сбойный или вышедший из строя вычислительный блок, изменить его назначение в качестве сервера или рабочего места, обновить или откатить версию конкретного ПО.

Загрузка по сети - старая идея. Основная мысль состоит в том, что компьютер содержит некоторый код начальной загрузки в энергонезависимой памяти, такой как микросхема ПЗУ, что позволяет соединиться с сервером и получить системные файлы по сети.

Для загрузки по сети, компьютер должен:

1. Инициализировать сетевой контроллер.
2. Пройти идентификацию и зарегистрироваться в сети, транслировав аппаратный адрес в ip-адрес.
3. Получить образ ядра ОС.
4. Как правило, получить рабочую файловую систему.

Для инициализации драйвер сетевого контроллера должен быть загружен на ранней стадии загрузки компьютера, т.е. во время или сразу после инициализации BIOS. Наиболее успешным и распространенным решением этой задачи является стандарт PXE (Preboot Execution Environment) и порожденные им открытые GPL-проекты Etherboot/gPXE (GPL PXE) и Netboot. PXE-код, прописанный в ПЗУ сетевого контроллера, BIOS материнской платы, или, на худой конец, загруженный с дискеты (в случае, если оборудование вычислительного блока не позволяет использовать pxe), получает загрузчик из сети, после чего передает ему управление. Большинство современных материнских плат со встроенными сетевыми контроллерами имеют встроенный pxe-загрузчик.

Для преобразования аппаратного адреса в ip-адрес используются широко распространенные протоколы BOOTP (Bootstrap Protocol) и выросший из него DHCP (Dynamic Host Configuration Protocol).

Передача образа осуществляется посредством протокола TFTP (Trivial File Transfer Protocol - простой протокол передачи файлов). TFTP подобен обрезанной версии FTP - тут нет идентификации, и он работает поверх UDP (User Datagram Protocol), а не поверх TCP (Transmission Control Protocol). UDP был выбран вместо TCP для упрощения. Реализация UDP может быть небольшой, поэтому код легко помещается в ПЗУ BIOS или сетевого контроллера.

Компьютером с загруженным ядром ОС и настроенной рабочей сетью уже возможно полностью управлять удаленно. Далее, для последующего файлового обмена может быть использован любой поддерживающий аутентификацию метод передачи данных в сети, начиная от относительно простых NFS (Network File System) и FTP (File Transfer Protocol), стандартного для архитектуры Win32 SMB (Server Message Block) в основе протокола совместного использования файлов Microsoft/3Com, и заканчивая прогрессивными P2P (Peer-to-Peer – клиент-клиент), такими, как, например, BitTorrent.

Итак, удаленный компьютер загрузил ядро-агент установки и готов к обслуживанию. Теперь необходимо развернуть ОС и прикладное ПО. Есть несколько вариантов осуществления этого процесса:

1. Последовательно произвести штатную установку. Установить из сетевых дистрибутивов операционную систему, настроить ее, затем установить программное обеспечение, произвести и его настройку.
2. Использовать клонирование. После осуществления одной последовательной установки, при условии идентичности оборудования вычислительного блока, произвести посекторное клонирование жесткого диска с установленным и настроенным ПО (в том числе информации о разбиении жесткого диска на разделы) одного вычислительного блока на остальные. Метод подходит для развертывания компонент КСА, выполняющих одинаковые задачи.
3. Использовать частичное клонирование. К примеру, несколько однотипных машин, имеющих разное назначение, но использующие одну платформу. Производится полное посекторное клонирование, скажем, на этапе установленной и настроенной ОС, а прикладное ПО устанавливается автоматически из дистрибутивов.

Для автоматизации первого варианта, как правило, используют варианты, основанные на сценариях установки. Примером таких систем являются неинтерактивный режим утилиты Sysinstall, входящей в ОС FreeBSD, RedHat Enterprise Linux Kickstart, Microsoft RIS (Remote Installation Services).

Условно это выглядит так:

- Определить оборудование удаленного ПК
- Выбрать соответствующее ядро и драйверы
- Определить параметры жесткого диска
- Разбить жесткий диск, отформатировать разделы
- Передать и установить на разделы базовые системные файлы ОС
- Определить назначение вычислительного блока в соответствии с MAC или ip-адресом.
- Произвести конфигурацию ОС в соответствии с назначением
- Отобрать, передать и установить прикладное ПО в соответствии с назначением
- Настроить ПО
- Скопировать необходимые для работы данные или восстановить из резервной копии

Методы исполнения второго и третьего вариантов сводятся к осуществлению ядром-агентом посекторной копии – образа жесткого диска настроенного по первому варианту вычислительного блока, с последующей передачей образа на сервер развертывания или работающий в связке с сервером развертывания специальный файловый сервер. Далее агенты на остальных вычислительных блоках забирают образ и копируют на локальный диск. В случае использования третьего варианта после клонирования вызывается описанный выше сценарий установки, например, с этапа конфигурации ОС или установки ПО.

Некоторые современные системы (такие, например, как Microsoft ADS (Automated Deployment Services – Службы Автоматического Развертывания) или Symantec ADS (Altiris Deployment Solution) вместо предварительной сценарной установки на одну из машин сами генерируют образ диска непосредственно после получения информации об оборудовании удаленного вычислительного блока.

К сожалению, проприетарные системы автоматизации платформозависимы, т.е. к примеру, ADS позволяет разворачивать только ОС семейства Windows. Системы, оперирующие образами дисков, не актуальны при наличии вычислительных блоков, различающихся по оборудованию.

Дальнейшее развитие темы автоматизации развертывания КСА приводит к необходимости реализации комплекса программных средств, позволяющих централизованно удаленно разворачивать системное и прикладное ПО в условиях различных платформ, произвольной топологии сети, независимо от оборудования вычислительных блоков. В зависимости от конкретных условий система должна определять наиболее оптимальной метод передачи файлов дистрибутивов, комбинировать метод сценарной установка и клонирования. И, в конце концов, необходима возможность удаленного управления и визуального контроля.

#### Литература.

1. Документы RFC (Request for Comments), технические спецификации и стандарты (<http://www.rfc-editor.org/rfc.html>).
2. PXE specification - The Preboot Execution Environment - спецификация v2.1, опубликовано Intel и Systemsoft (<http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf>).
3. Х. Остерлох. «TCP/IP. Семейство протоколов передачи данных в сетях компьютеров», Глава 16 «Простейший протокол передачи файлов (TFTP)». ISBN 5-93772-039-3.
4. Материалы сайта OpenNET – Открытые системы ([opennet.ru](http://opennet.ru)).
5. Материалы сайта xgu.ru – точка обмена знаниями по UNIX/Linux-системам, системам с открытым исходным кодом.