

Глава 7

Виртуализация и облако

Бывает, что в организации имеется система из нескольких компьютеров, которая ей фактически не нужна. Типичным примером может послужить наличие в компании почтового сервера, веб-сервера и FTP-сервера, нескольких серверов электронной торговли и других серверов. И все они запускаются на разных компьютерах в единой стойке оборудования, соединены высокоскоростной сетью, то есть, иначе говоря, составляют мультикомпьютер. Одной из причин запуска всех этих серверов на отдельных машинах может послужить то, что одна машина не может справиться с нагрузкой, а другая отличается особой надежностью: управление компании просто не верит в то, что операционная система может работать без сбоев 24 часа в сутки 365 или 366 дней в году. А если каждая служба помещена на отдельный компьютер, то сбой одного из серверов по крайней мере не повлияет на работу остальных серверов. Также при этом легче решаются вопросы безопасности. Даже если злоумышленник взломает веб-сервер, то одновременно с этим он не получит доступ к конфиденциальной почте — иногда такое свойство называют **использованием песочницы**. Хотя благодаря этому достигаются изоляция и устойчивость к сбоям, такое решение является весьма дорогостоящим и трудно управляемым, поскольку в нем задействовано множество машин.

Имейте в виду, что для использования отдельных машин есть всего лишь две причины. Например, организации в повседневной работе часто зависят от более чем одной операционной системы: веб-сервер работает на Linux, почтовый сервер — на Windows, сервер электронной торговли для клиентов — на OS X, а ряд других серверов запущены под несколькими разновидностями UNIX. При всей своей работоспособности такое решение обходится недешево.

Что же делать? Возможным (и весьма популярным) решением является использование технологии виртуальных машин, которая при всей новизне и стильности звучания базируется на довольно старой идее, датированной далекими 1960-ми годами. Но даже при этом способ, используемый в наши дни, конечно же, новый. Основная идея заключается в том, что **монитор виртуальных машин** (Virtual Machine Monitor (**VMM**)) создает иллюзию присутствия нескольких (виртуальных) машин на одном и том же физическом оборудовании. VMM известен также как **гипервизор**. В разделе 1.7.5 мы уже определили разницу между гипервизором первого типа, запускаемым непосредственно на оборудовании, и гипервизором второго типа, который может воспользоваться всеми полезными службами и абстракциями, предлагаемыми основной операционной системой. Так или иначе, виртуализация позволяет одному компьютеру стать базой для нескольких виртуальных машин, на каждой из которых потенциально может быть запущена совершенно другая операционная система.

Преимуществом такого подхода является то, что авария одной виртуальной машины не приводит к аварии любой другой машины. На виртуализированной системе разные серверы могут запускаться на разных виртуальных машинах, поддерживая тем самым

модель частичных отказов, характерную для мультикомпьютеров, но при меньшей стоимости и более простом обслуживании. Более того, теперь на одном и том же оборудовании можно запускать несколько разных операционных систем, получая преимущества изолированности виртуальных машин при угрозе хакерских атак и другие преимущества.

Разумеется, подобное объединение серверов сродни укладыванию всех яиц в одну корзину. При падении сервера, на котором запущены все эти виртуальные машины, результат будет еще катастрофичнее падения отдельного выделенного сервера. Но смысл связываться с виртуализацией заключается в том, что подавляющая часть выходов служб из строя происходит не по вине оборудования, а из-за недочетов в разработке, ненадежности, наличия ошибок и плохой настройки программного обеспечения, включая, и это следует особо подчеркнуть, операционные системы. При использовании технологии виртуальных машин единственным программным обеспечением, запущенным в режиме наивысших привилегий, является гипервизор, у которого имеется на два порядка меньше строк кода, чем у всей операционной системы, а следовательно, и на два порядка меньше потенциальных ошибок. Гипервизор проще операционной системы, поскольку он занимается только одним — эмулированием нескольких копий оборудования (чаще всего с архитектурой Intel x86).

Запуск программного обеспечения кроме строгой изолированности имеет и другие дополнительные преимущества. Одно из них заключается в меньшем числе физических машин, что позволяет экономить средства на оборудовании и энергопотреблении и использовать меньшее пространство для аппаратных стоек. Для таких компаний, как Amazon или Microsoft, у которых в каждом центре обработки данных могут находиться сотни тысяч серверов, выполняющих великое множество разнообразных задач, сокращение физических потребностей в их дата-центрах выльется в гигантскую экономию средств. Фактически серверные компании зачастую размещают свои дата-центры в любых местах, лишь бы они были недалеко от гидроэлектростанций (и от дешевой электроэнергии). Виртуализация также содействует проверке жизнеспособности новых идей. Обычно в крупных компаниях отдельные подразделения или группы занимаются проработкой интересных идей, а затем идут на затраты, приобретая сервер для их реализации. Если идея получает популярность и ей необходимы сотни или тысячи серверов, дата-центр корпорации расширяется. Зачастую перемещение программного обеспечения на уже существующие машины дается нелегко, поскольку каждому приложению часто требуется другая версия операционной системы, его собственные библиотеки, конфигурационные файлы и многое другое. При использовании виртуальных машин каждое приложение может взять с собой все свое окружение.

Еще одним преимуществом виртуальных машин является то, что установка контрольных точек и миграция этих виртуальных машин (например, для выравнивания баланса загрузки нескольких серверов) даются намного легче, чем миграция процессов, запущенных на обычной операционной системе. В последнем случае в таблицах операционной системы хранится изрядное количество важной информации о состоянии каждого процесса, в том числе информация, относящаяся к открытым файлам, аварийным сигналам, обработчикам сигналов и многому другому. При миграции виртуальной машины все, что должно перемещаться, касается только памяти и образов дисков, поскольку с ними перемещаются также и все таблицы операционной системы.

Еще одним примером использования виртуальных машин является запуск устаревших приложений на операционных системах (или версиях операционных систем), которые

больше не поддерживаются или не работают на существующем оборудовании. Они могут работать в то же время и на том же оборудовании, что и текущие приложения. Фактически возможность запуска в одно и то же время приложений, использующих разные операционные системы, является существенным аргументом в пользу виртуальных машин.

Еще одним важным аспектом использования виртуальных машин является разработка программного обеспечения. Программист, желающий убедиться в работоспособности своей программы под Windows 7, Windows 8, несколькими версиями Linux, FreeBSD, OpenBSD, NetBSD и OS X, а также под управлением других систем, теперь не нуждается в десятке компьютеров и в установке операционных систем на все эти компьютеры. Вместо этого он просто создает десять виртуальных машин на одном компьютере и устанавливает на каждую из них разные операционные системы. Разумеется, он может разбить жесткий диск на разделы и установить в каждый из разделов другую операционную систему, но этот подход дается намного сложнее. Во-первых, на стандартном персональном компьютере независимо от объема диска поддерживаются только четыре первичных дисковых раздела. Во-вторых, хотя в загрузочный блок можно установить программу мультизагрузки, для работы компьютера под управлением новой операционной системы его придется перезагрузить. При использовании виртуальных машин все они могут работать одновременно, поскольку на самом деле все они являются всего лишь возведенными на более высокий уровень процессами.

Возможно, наиболее важным и соответствующим времени случаем использования виртуализации является **облако** (cloud). Ключевая идея облака довольно проста: передать ваши потребности в вычислениях или хранении данных в высокоорганизованный дата-центр, запущенный компанией, специализирующейся на подобных услугах и укомплектованной специалистами в данной области. Поскольку дата-центр обычно принадлежит кому-нибудь другому, вам, скорее всего, придется платить за использование ресурсов, но при этом по крайней мере не придется волноваться за физические машины, энергопотребление, охлаждение и обслуживание. Поскольку изолированность обеспечивается виртуализацией, поставщики облачных услуг могут позволить нескольким клиентам, даже конкурирующим друг с другом, пользоваться общей физической машиной. Каждый клиент получает свой кусок пирога. Рискуя растянуть метафору облака, следует упомянуть, что на ранних этапах критики утверждали, будто этот пирог находится только на небесах и что настоящие организации не захотят помещать свои конфиденциальные данные и вычисления на какие-то другие ресурсы. Но теперь виртуализированные машины в облаках используются несметным числом организаций для не поддающегося подсчету количества приложений, и хотя это, может быть, не все организации и не все данные, не возникает никаких сомнений, что облачные вычисления пользуются успехом.

7.1. История

Во всей этой шумихе, окружающей виртуализацию в последние годы, мы иногда забываем, что по меркам Интернета виртуальные машины являются весьма древними устройствами. Их истоки теряются в 60-х годах прошлого столетия. В IBM проводились эксперименты даже не с одним, а с двумя разработанными независимо друг от друга гипервизорами: **SIMMON** и **CP-40**. Хотя CP-40 был исследовательским проектом, он был переработан в **CP-67** и сформирован в виде программы управления для **CP/CMS**, операционной системы виртуальных машин для IBM System/360 Model 67. Позже он

был снова переработан и реализован в виде **VM/370** для серии машин System/370, выпущенной в 1972 году. Линейка машин System/370 в 1990-х годах была заменена компанией IBM линейкой System/390. В основном изменилось только название, поскольку базовая архитектура из соображений обратной совместимости осталась прежней. Разумеется, аппаратные технологии стали совершеннее и более новые машины были больше и быстрее старых, но что касается виртуализации, ничего не изменилось. В 2000 году IBM выпустила z-серии, поддерживающие 64-разрядное виртуальное адресное пространство при сохранении обратной совместимости с System/360. Все эти системы поддерживали виртуализацию на десятилетия раньше того момента, когда она приобрела популярность на машинах семейства x86.

В 1974 году двое ученых из Калифорнийского университета (Лос-Анджелес), работающих в компьютерной сфере, Геральд Попек (Gerald Popek) и Роберт Голдберг (Robert Goldberg), опубликовали основополагающую статью («Formal Requirements for Virtualizable Third Generation Architectures»), в которой дали точный перечень тех условий, которым должна отвечать компьютерная архитектура, чтобы иметь возможность эффективно поддерживать виртуализацию (Popek and Goldberg, 1974). Написать главу о виртуализации без ссылки на их работу и терминологию просто невозможно. Примечательно, что широко известная архитектура x86, которая также берет начало в 1970-х годах, десятилетиями не отвечала этим требованиям. Но это было еще не все. Практически каждая архитектура со времен универсальных машин (мейнфреймов) также проваливала тест. 1970-е годы были весьма продуктивными, они увидели рождение UNIX, Ethernet, Cray-1, Microsoft и Apple, — следовательно, несмотря на то что могли бы сказать ваши родители, в 1970-е на планете гремел не только стиль диско!

Фактически настоящая революция **Disco** грянула в 1990-е годы, когда исследователи из Стэнфордского университета разработали новый гипервизор с таким именем и стали основателями **VMware**, гиганта виртуализации, предлагающего гипервизоры типа 1 и типа 2 и теперь гребущего миллиарды долларов дохода (Bugnion et al., 1997; Bugnion et al., 2012). Кстати, разница между гипервизорами типа 1 и типа 2 также пришла из 1970-х (Goldberg, 1972). VMware представила свое первое решение по виртуализации для x86 в 1999 году. Затем последовали и другие продукты: **Xen**, **KVM**, **VirtualBox**, **Hyper-V**, **Parallels** и многие другие. Похоже, время для виртуализации как раз подошло, даже при том, что теорию застолбили еще в 1974 году, а IBM десятилетиями продавала компьютеры, поддерживающие и активно использующие виртуализацию. В 1999 году виртуализация приобрела широкую популярность, но несмотря на внезапно возросший к ней массовый интерес, новинкой она не была.

7.2. Требования, применяемые к виртуализации

Важно понимать, что виртуальные машины работают так же, как и реальные. В частности, у них должна быть возможность начальной загрузки, как на реальных машинах, и установки на них произвольных операционных систем, точно так же, как это может быть сделано на реальном оборудовании. Предоставление этой иллюзии с обеспечением достаточной эффективности является задачей гипервизора. Несомненно, гипервизоры должны хорошо проявлять себя по трем направлениям:

1. **Безопасность** — у гипервизора должно быть полное управление виртуализированными ресурсами.