

УДК 004.7

А.В. Коуров,
г. Шадринск

Современные гипервизоры как основа инфраструктуры образовательной организации

В статье раскрывается и конкретизируется понятие "гипервизор", а также исследуется архитектура современных гипервизоров. Вместе с тем описываются технологии виртуализации применяемые в совершенствовании и оптимизации информационной среды вуза.

Технологии виртуализации, гипервизоры, классификация гипервизоров, информатизация образовательной организации

А.В. Kourov,
Shadrinsk

Modern hypervisors as the basis of the educational infrastructure of the organization

The article describes and specifies the concept of a "hypervisor", and explores the architecture of modern hypervisors. However describes the virtualization technology used in the improvement and optimization of the information environment of the University.

Key words: *virtualization, hypervisors, classification of hypervisors, the Informatization of the educational organization*

Сегодня виртуализация из вспомогательного средства превращается в базовую инфраструктурную технологию, и, следовательно, выбор той или иной платформы представляет собой стратегическое решение на многие годы вперед. Текущие возможности различных платформ виртуализации, их потенциал развития, во многом определяются архитектурными принципами, заложенными на уровне гипервизоров. Как отмечалось ранее, современные подходы к классификациям технологий виртуализации неразрывно связаны с возможностями программ-гипервизоров. Термин "гипервизор" был впервые применен компанией IBM в 1972 г. для того, чтобы подчеркнуть отличие от термина супервизор, которым традиционно называли ядро ОС. В сущности, гипервизор является расширением и обобщением понятия супервизора: подобно тому, как супервизор в ядре ОС обеспечивает изоляцию пользовательских программ друг от друга, выделение и освобождение ресурсов для пользовательских процессов, так и гипервизор обеспечивает изоляцию и управление ресурсами для самих ОС как целого, вместе с их пользователями и процессами.

Отметим, что термин гипервизор (hypervisor) зачастую используется в двух понятиях, которые следует различать. Первое, более широкое, включает все виды технологий поддержки исполнения виртуальных машин (ВМ). Более узкое – один из вариантов таких решений, основанный на отсутствии хостовой ОС.

М. Тим Джонс отмечает, что гипервизор – это специальная операционная система (в случае с гипервизорами первого типа), работающая непосредственно на аппаратном обеспечении. Гипервизор создает абстракцию нижележащей аппаратной платформы, таким образом, чтобы она могла использоваться одной или несколькими виртуальными машинами (ВМ), при этом ВМ не знают, что совместно используют одну и ту же платформу. В данном контексте виртуальная машина – это просто контейнер для операционной системы и приложений. Интересное преимущество данного подхода состоит в том, что виртуальная машина изолируется от других виртуальных машин, запущенных на этом же гипервизоре, что позволяет иметь

несколько операционных систем или несколько конфигураций одной операционной системы.

Схожее определение понятия гипервизор мы находим и в свободной энциклопедии, которая определяет его как программу или аппаратную схему, обеспечивающую или позволяющую одновременное параллельное выполнение нескольких операционных систем на одном и том же хост-компьютере. Гипервизор обеспечивает изоляцию ОС друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными ОС и управление ресурсами. В этом смысле гипервизор обозначает целый класс ПО, которое отвечает за процесс исполнения ВМ, и отделяет эту категорию продуктов от других компонентов системы виртуализационного ПО.

Все гипервизоры, обеспечивающие работу виртуальных машин и приложений, делятся на два типа. Большинство современных гипервизоров относятся к Type 2, что подразумевает установку гипервизора в основную клиентскую операционную систему. Гипервизоры, которые относятся к Type 1, интегрируются с аппаратной составляющей ВС, а клиентская операционная система работает поверх этой аппаратуры и гипервизора.

По мнению Тима Джонса [3] в реализации технологий ВМ выделяются три основных подхода (рис. 1).

- Гипервизор первого типа (автономный, тонкий, исполняемый на “голом железе” — Type 1, native, bare-metal) — программа, исполняемая непосредственно на аппаратном уровне компьютера и выполняющая функции эмуляции физического аппаратного обеспечения и управления аппаратными средствами и гостевыми ОС. То есть такой гипервизор сам по себе в некотором роде является минимальной операционной системой.

- Гипервизор второго типа (хостовый, монитор виртуальных машин — hosted, Type-2, V) — специальный дополнительный программный слой, расположенный поверх основной хостовой ОС, который в основном выполняет функции управления гостевыми ОС, а эмуляцию и управление аппаратурой берет на себя хостовая ОС.

- Гипервизор гибридный (Hybrid, Type-1+) — объединенный вариант первых двух, в котором функции управления аппаратными средствами выполняются тонким гипервизором и специальной депривилегированной сервисной ОС, работающей под управлением тонкого гипервизора. Обычно гипервизор управляет напрямую процессором и памятью компьютера, а через сервисную ОС гостевые ОС работают с остальными аппаратными компонентами.

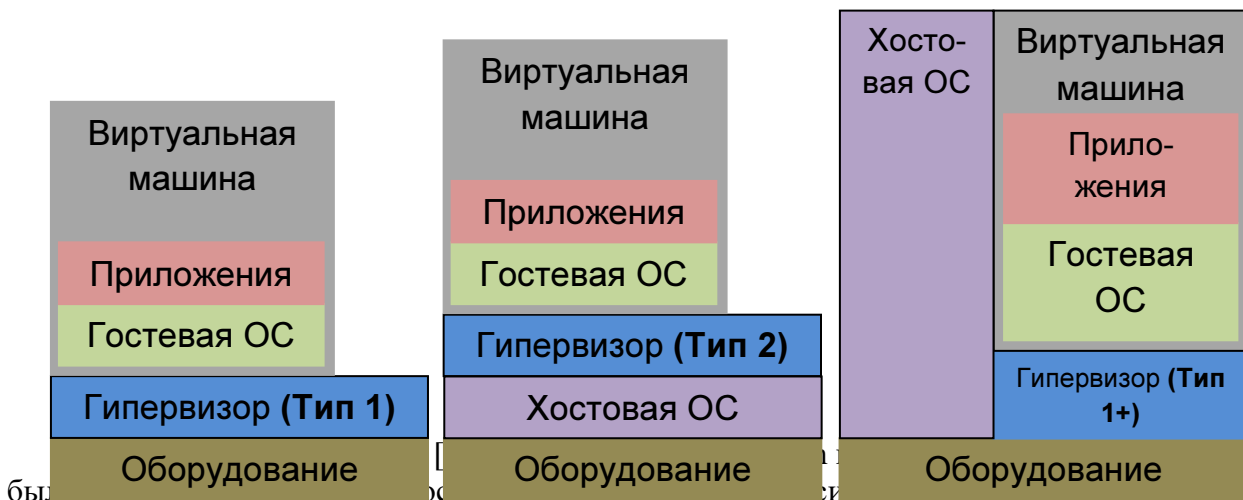
Итак, в широком смысле гипервизор - это все три перечисленные выше вида, а в узком - это гипервизор первого типа. Гипервизор первого типа считается классическим вариантом реализации архитектуры ВМ, как раз он был впервые реализован компанией IBM в 1960 г. в виде системы CP/CMS.

Виртуализация же систем x86 начиналась в конце 90-х с использования второго (хостового) варианта, который долгое время доминировал на рынке и тогда назывался просто “ПО виртуализации”. Термин гипервизор стал применяться применительно к x86-компьютерам лишь 4-5 лет назад в связи с началом широкого применения первого варианта другими разработчиками для создания собственных продуктов и чтобы отличить его от традиционных хостовых решений. В 2008 г. тип 1 гипервизора стал уже доминирующим на рынке, и потому данный термин теперь используется для обозначения всего ПО исполнения виртуальных машин [4].

В 2004 - 2005 гг. Intel и AMD внесли коррективы в архитектуру своих процессоров, реализовав в них технологии соответственно Intel VT и AMD SVM. Это существенно упростило создание виртуализационных средств, что способствовало расширению числа разработок в этой области. Но здесь следует отметить, что многие появившиеся в последнее время гипервизоры (например, Microsoft Hyper-V) изначально ориентированы на работу только с современными процессорами с поддержкой архитектуры Intel-VT или AMD SVM.

При многих технологических различиях архитектурные подходы Microsoft Hyper-V и Citrix XenServer в целом очень близки, что и определяет их более тесный на рынке альянс (Рис. 1).

Рис. 1. Архитектура гипервизоров



(Тип 1) была только у VMware - ESX Server. У Microsoft был только Virtual Server, в котором управление виртуальными машинами (VM) реализовано с использованием хостовой ОС (Тип 2), что для производственного применения не подходит вследствие низкой скорости выполнения. XenServer в этот период принадлежал небольшой компании, веса которой было недостаточно для серьезного присутствия на корпоративном рынке. Решающее конкурентное преимущество у VMware исчезло с появлением у Microsoft гипервизора Hyper-V, а XenServer перешел компании Citrix. Гипервизорных решений стало достаточно много, к тому же они быстро перешли в разряд бесплатных, как следствие возросшей конкуренции, и акцент борьбы за заказчика переместился на уровень выше - в сферу управления виртуальными средами.

Развивая технологии виртуализации отдельных серверов, в последние годы, все чаще стали переходить к решению задачи комплексного управления виртуальной ИТ-инфраструктурой (По мнению компании IDC данный этап – Виртуализация 2.0). Вопрос управления виртуальной ИТ-инфраструктурой выходит сейчас на передний план, но это не значит, что вся проблема выбора той или иной платформы сводится к наличию или отсутствию возможности “живой миграции”, как это может показаться из анализа публикаций по данной теме последних месяцев. Всё же возможности самой среды исполнения виртуальных машин тоже важны.

В настоящее время системы виртуализации серверов на базе хостового гипервизора (тип 2) уходят с рынка, и продолжают использоваться в узких специализированных областях. В качестве примера отметим VMware Server, Oracle VirtualBox, Microsoft Virtual Server и Parallels Server for Mac. При этом мы видим, что данные программные решения сегодня, в виде гипервизоров второго типа, не существуют. Современные их реализации в определенной степени поддерживают

технологии аппаратной виртуализации, являясь гибридными, т.е. могут работать как с поддержкой аппаратной виртуализации, так и без нее в зависимости от выбора настроек.

А. Колесов отмечает, что доминирующее положение сегодня занимают виртуализационные решения, реализованные на базе гипервизора (тип 1). Хотя на самом деле вопрос о том, к какому типу гипервизора относится то или иное решение, является спорным. Википедия причисляет к типу 1 почти все имеющиеся на рынке продукты - VMware ESX Server, Microsoft Hyper-V, Citrix XenServer, Oracle VM, Sun Logical Domains Hypervisor и др. Однако более внимательное изучение этого вопроса говорит о том, что “настоящий” тип 1 - это только VMware ESX, а остальные - это, скорее, гипервизоры смешанного типа [4].

VMware - классический вариант полностью автономного гипервизора, который относится к категории архитектуры монолитного ядра. Он содержит всё необходимое для работы ВМ и по сути является автономной ОС, включающей в том числе и драйверы для работы с оборудованием. Именно на эти моменты как на архитектурные преимущества делает акцент VMware, подчеркивая, что такой подход обеспечивает наиболее полную изоляцию ВМ, а следовательно, и более высокую надежность системы в целом. В целом ее ESX Server состоит из собственно гипервизора, среды исполнения (ESXi) и сервисной консоли на базе ядра Linux, при этом ESXi может использоваться автономно, но в этом случае у него ограничены возможности управления системой.

Критики данного варианта указывают на сложность решения задачи реализации собственной модели драйверов, доводка которой до нужного уровня является многолетней задачей. Отмечается также, что вся архитектура ESX - закрытая, проприетарная (на это делает акцент даже Microsoft).

Microsoft придерживается иного подхода, который называют архитектурой микроядра, т.е. разделения функций по разным модулям и уровням. В сущности данный подход вариант гипервизора смешанного типа (Тип 1+), когда сам гипервизор выполняет только функции управления памятью и процессором, а для взаимодействия с внешними устройствами и управления служит привилегированная родительская ВМ на базе ядра Windows (в случае Citrix — Linux).

По мнению разработчиков Microsoft, такой подход более эффективен при высокой вычислительной нагрузке, когда используется только “тонкий” гипервизор, который в случае Microsoft занимает всего порядка 100 Кб оперативной памяти.

Для ускорения же работы на уровне драйверов Microsoft использует два механизма взаимодействия прикладных ВМ с родительской. В одном случае применяется специальный внутренний интерфейс VMBus, который позволяет общаться ВМ между собой напрямую. Он доступен для ВМ, реализованных на базе Windows, а также Xen, но только для тех разработчиков, с кем у Microsoft есть соответствующий уровень сотрудничества. Для всех остальных ОС используется второй вариант полной эмуляции драйверов.

Как свое преимущество Microsoft также подчеркивает наличие в ее Hyper-V проверенной модели драйверов, которая развивается в рамках Windows Server в целом на протяжении ряда лет.

Итак, автономный гипервизор (Тип 1) имеет свои встроенные драйверы устройств, модели драйверов и планировщик и поэтому не зависит от базовой ОС. Так как автономный гипервизор работает непосредственно в окружении усеченного ядра, то он более производительен, но проигрывает в производительности виртуализации на

уровне ОС и паравиртуализации. Примерами таких технологий выступают: VMware ESX, Citrix, XenServer.

Гипервизор на основе базовой ОС (Тип 2, V) представляет собой компонент, работающий в одном кольце с ядром основной ОС (кольцо 0). Гостевой код может выполняться прямо на физическом процессоре, но доступ к устройствам ввода-вывода компьютера из гостевой ОС осуществляется через второй компонент, обычный процесс основной ОС — монитор уровня пользователя. Примерами таких гипервизоров выступают: VirtualBox, VMware Workstation, QEMU, Parallels.

Гибридный гипервизор (Тип 1+) состоит из двух частей: из тонкого гипервизора, контролирующего процессор и память, а также работающей под его управлением специальной сервисной ОС в кольце пониженного уровня. Через сервисную ОС гостевые ОС получают доступ к физическому оборудованию. Примерами данного вида гипервизоров выступают: Sun Logical Domains, Xen, Citrix XenServer, Microsoft Hyper-V.

На основе сказанного выше, отметим, что гипервизор (Монитор виртуальных машин) — это программа или аппаратная схема, обеспечивающая или позволяющая одновременное, параллельное выполнение нескольких, или даже многих операционных систем, на одном и том же хост компьютере.

В обязанности гипервизора входит: изоляция операционных систем друг от друга, разделение ресурсов между операционными системами, управление ресурсами, обеспечение защиты и безопасности операционных систем [1].

Гипервизор — предоставляет операционным системам, работающим под его управлением виртуализацию и эмуляцию реального аппаратного обеспечения, управляет этими виртуальными операционными системами и выделяет и освобождает ресурсы для них, так же предоставляет возможности независимого запуска, перезагрузки и останова каждой из них. Работа для операционной системы под управлением гипервизора ничем не отличается от работы на реальном аппаратном обеспечении.

Многие из существующих классификаций технологий серверной виртуализации неразрывно связаны с возможностями программ-гипервизоров. Основываясь на типах гипервизоров, можно определить следующие виды виртуализации: полная виртуализация, паравиртуализация, виртуализация уровня ОС и различные виды эмуляции. При этом, под гипервизором понимают программу или аппаратную схему, обеспечивающую одновременное, параллельное выполнение нескольких операционных систем, на одном и том же хост компьютере. Существуют автономные гипервизоры, на основе базовой операционной системы и гибридные.

В настоящее время, существует несколько лидирующих систем виртуализации. Среди всех систем особо выделяется openVZ, популярность её использования обеспечивается высоким функционалом, большой степенью надежности и изоляции ресурсов и поддержкой «живой» миграции, отсутствующей у конкурентов.

Применяя технологии виртуализации в совершенствовании и оптимизации информационной среды вуза в конфигурации OpenVZ (базовая), Hyper-V и Xen (вспомогательные), можно эффективно обеспечивать студентов и преподавателей круглосуточно функционирующими гостевыми машинами с возможностью доступа в Интернет, и осуществлять централизованный контроль над ресурсами локальной сети.

Решением проблем оптимизации информационной инфраструктуры организации, связанных со сложностью, безопасностью, надёжностью и дороговизной компонентов информационной системы, может служить повсеместное внедрение технологий виртуализации. Основываясь на многолетнем опыте и потребностях

ШГПИ, оптимальным явилось решение о внедрении в работу Вычислительного Центра, обслуживающего большинство локальных сетей вуза, системы виртуализации OpenVZ. Выбор системы виртуализации основывается на:

1. Возможностях передачи в индивидуальное пользование студентам и преподавателям вуза заранее подготовленных виртуальных машин, высокая скорость подготовки таких машин, минимальные усилия по вводу и выводу их из эксплуатации.
2. Наличие централизованного контроля за состоянием и содержимым виртуальных машин, возможность административного доступа к виртуальным машинам администратору хостовой системы.
3. Низкий уровень требований к мощности физического узла, высокая плотность размещения виртуальных машин, минимальный объем шаблонов и резервных копий виртуальных машин,
4. Быстрая миграция виртуальных машин с одного физического узла на другой.

ЛИТЕРАТУРА

1. Виртуализация для хостинга: тупик или прорыв? [Электронный ресурс] // Администрирование серверов. Обслуживание компьютеров : [web-сайт]. – Режим доступа: <http://ha-systems.ru/virtualizacija-dlja-hostinga>. – 26.04.2014.
2. Гипервизор [Электронный ресурс] // Википедия : свобод. энцикл. – Режим доступа : <http://ru.wikipedia.org/wiki/Гипервизор>. – 17.04.14.
3. Джонс, Тим М. Виртуализация для встроенных систем. Архитектура и возможности гипервизоров для встраиваемых устройств [Электронный ресурс] // DeveloperWorks®. - Режим доступа: <http://www.ibm.com/developerworks/ru/library/l-embedded-virtualization/>. – 31.01.2012.
4. Колесов, А. Вернемся к нашим гипервизорам [Электронный ресурс] / А. Колесов // PC Week/RE : ИТ-сервисы. Облака. – Режим доступа: [http://www.pcweek.ru/its/article /detail.php?ID=118886](http://www.pcweek.ru/its/article/detail.php?ID=118886). – 18.05.2009.
5. Попов, А.И. Виртуальные машины и сети [Текст] / А.И. Попов. – СПб. : БХВ-Петербург, 2012. – 477с.

