

55 Системы виртуализации ОС. Основные понятия, паравиртуализация, аппаратная виртуализация, гипервизор. Примеры применения.

<http://ru.wikipedia.org/wiki/%D0%92%D0%B8%D1%80%D1%82%D1%83%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F> - виртуализация

Виртуализация — предоставление набора [вычислительных ресурсов](#) или их логического объединения, абстрагированное от [аппаратной реализации](#), и обеспечивающее при этом логическую изоляцию вычислительных процессов, выполняемых на одном физическом ресурсе.

Примером использования виртуализации является возможность запуска нескольких [операционных систем](#) на одном компьютере, при этом каждый из экземпляров таких гостевых операционных систем работает со своим набором логических ресурсов (процессорных, оперативной памяти, устройств хранения), предоставлением которых из общего пула, доступного на уровне оборудования, управляет хостовая операционная система или гипервизор.

Виртуализацию можно разделить на два типа: программная, аппаратная и виртуализация на уровне ОС.

К программной виртуализации относятся динамическая трансляция и паравиртуализация.

Динамическая трансляция. При использовании данного вида виртуализации происходит перехват проблемных команд гостевой ОС гипервизором. После этого осуществляется замена проблемных команд на безопасные, то есть возвращается управление гостевой ОС. Динамическую трансляцию еще называют бинарной трансляцией.

Паравиртуализация — представляет собой технику виртуализации, при которой операционная система подготавливается для начала активности в виртуализированной среде. При этом ядро операционной системы подвергается незначительной модификации (изменению). При использовании такого вида виртуализации, как паравиртуализация, происходит взаимодействие операционной системы с программой Гипервизора. Функция Гипервизора заключается в том, что он предоставляет операционной системе гостевой API вместо использования такого масштабного ресурса, как таблица страниц памяти. Код, касающийся виртуализации, локализуется непосредственно в операционную систему. Паравиртуализация таким образом требует, чтобы гостевая операционная система была изменена для гипервизора, и это является недостатком метода, так как подобное изменение возможно лишь в случае, если гостевые ОС имеют открытые исходные коды, которые можно модифицировать согласно лицензии. Но зато паравиртуализация предлагает производительность почти как у реальной не виртуализированной системы. Как и при полной виртуализации, одновременно могут поддерживаться многочисленные различные операционные системы. Метод паравиртуализации позволяет добиться более высокой производительности, чем метод [динамической трансляции](#).

Паравиртуализация не эмулирует "железо", а вместо этого просто координирует доступ операционных систем и программ к настоящим аппаратным ресурсам. Этот подход менее тяжеловесен - меньше потери производительности на виртуализацию, к тому же драйвера устройств используются оригинальные, из состава операционных систем.

Платформы, использующие паравиртуализацию:

- Xen,
- системы виртуализации [Vmware](#) (частично),
- L4,
- Trango,
- XtratuM.

Аппаратная виртуализация — [виртуализация](#) с поддержкой специальной процессорной архитектуры. В отличие от программной техники, с помощью аппаратной виртуализации возможно получение изолированных гостевых систем, управляемых гипервизором напрямую. Такой подход может обеспечить простоту реализации платформы виртуализации и увеличить надежность платформы с несколькими одновременно запущенными гостевыми системами, при этом нет потерь производительности на обслуживание хостовой системы. Такая модель позволит приблизить производительность гостевых систем к реальным и сократить затраты производительности на поддержание хостовой платформы.

Аппаратная виртуализация обеспечивает производительность, сравнимую с производительностью неvirtуализованной машины, что дает виртуализации возможность практического использования и влечет её широкое распространение. Наиболее распространены технологии виртуализации [Intel](#)-VT и [AMD](#)-V.

Платформы, использующие аппаратную виртуализацию:

- IBM [LPAR](#)
- [VMware](#)
- [Hyper-V](#)
- [Xen](#)
- [KVM](#)

Виртуализация на уровне операционной системы позволяет запускать изолированные и безопасные виртуальные машины на одном физическом узле, но не позволяет запускать операционные системы с ядрами, отличными от типа ядра базовой операционной системы. При виртуализации на уровне операционной системы не существует отдельного слоя гипервизора. Вместо этого сама хостовая операционная система отвечает за разделение аппаратных ресурсов между несколькими виртуальными машинами и поддержку их независимости друг от друга. Среди реализаций:

- [Solaris Containers/Zones](#)
- [FreeBSD Jail](#)
- [Linux-VServer](#)^[en]
- [LXC \(Linux Containers\)](#)
- [FreeVPS](#)^[en]
- [OpenVZ](#)
- [Virtuozzo](#)
- [iCore Virtual Accounts](#)

Гипервизор (Hypervisor) (или **Монитор виртуальных машин**) — [программа](#) или аппаратная схема, обеспечивающая или позволяющая одновременное, параллельное выполнение нескольких или даже многих [операционных систем](#) на одном и том же хост-компьютере. Гипервизор также обеспечивает изоляцию операционных систем друг от друга, защиту и безопасность, разделение ресурсов между различными запущенными [ОС](#) и управление ресурсами.

Гипервизор также может (но не обязан) предоставлять работающим под его управлением на одном хост-компьютере ОС средства связи и взаимодействия между собой (например, через обмен файлами или сетевые соединения) так, как если бы эти ОС выполнялись на разных физических компьютерах.

Гипервизор сам по себе в некотором роде является минимальной операционной системой ([микроядром](#) или [наноядром](#)). Он предоставляет запущенным под его управлением операционным системам сервис [виртуальной машины](#), виртуализируя или эмулируя реальное (физическое) аппаратное обеспечение конкретной машины. И управляет этими виртуальными машинами, выделением и освобождением ресурсов для них. Гипервизор позволяет независимое «включение», перезагрузку, «выключение» любой из виртуальных машин с той или иной ОС. При этом операционная система, работающая в виртуальной машине под управлением гипервизора, может, но не обязана «знать», что она выполняется в виртуальной машине, а не на реальном аппаратном обеспечении.

Типы гипервизора

-Автономный гипервизор - имеет свои встроенные драйверы устройств, модели драйверов и планировщик и поэтому не зависит от базовой ОС. Так как автономный гипервизор работает непосредственно в окружении усечённого ядра, то он более производителен, но проигрывает в производительности виртуализации на уровне ОС и паравиртуализации. Xen может запускать виртуальные машины в паравиртуальном режиме (зависит от ОС).

Примеры: VMware ESX , Citrix, XenServer.

-На основе базовой ОС - это 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.

Примеры: Microsoft Virtual PC, VMware Workstation, QEMU, Parallels, VirtualBox.

-Гибридный гипервизор - 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.

Примеры: Microsoft Virtual Server, Sun Logical Domains, Xen, Citrix XenServer, Microsoft Hyper-V.

Примеры применения

1. Консолидация серверов.

В данный момент приложения, работающие на серверах в IT-инфраструктуре компаний, создают небольшую нагрузку на аппаратные ресурсы серверов (в среднем 5-15 процентов). Виртуализация позволяет мигрировать с этих физических серверов на виртуальные и разместить их все на одном физическом сервере, увеличив его загрузку до 60-80 процентов и повысив тем самым коэффициент использования аппаратуры, что позволяет существенно сэкономить на аппаратуре, обслуживании и электроэнергии.

2. Разработка и тестирование приложений.

Множество продуктов виртуализации позволяют запускать несколько различных операционных систем одновременно, позволяя тем самым разработчикам и тестерам программного обеспечения тестировать их приложения на различных платформах и конфигурациях. Также удобные средства по созданию «снимков» текущего состояния системы одним кликом мыши и такого же просто восстановления из этого состояния, позволяют создавать тестовые окружения для различных конфигураций, что существенно повышает скорость и качество разработки.

3. Использование виртуальных рабочих станций.

С приходом эры виртуальных машин бессмысленно будет делать себе рабочую станцию с привязкой ее к аппаратуре. Теперь создав однажды виртуальную машину со своей рабочей или домашней средой, можно будет использовать её на любом другом компьютере. Также можно использовать готовые шаблоны виртуальных машин (Virtual Appliances), которые решают определенную задачу (например, сервер приложений).