ВИРТУАЛИЗАЦИЯ СЕРВЕРОВ НА БАЗЕ VMWARE ESXI

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Картошников Д.Н.

Королев А.И. – к.т.н.,доцент

Быстрое развитие рынка технологий виртуализации за последние несколько лет произошло во многом благодаря увеличению мощностей аппаратного обеспечения, позволившего создавать по-настоящему эффективные платформы виртуализации, как для серверных систем, так и для настольных компьютеров. Технологии виртуализации позволяют запускать на одном физическом компьютере (хосте) несколько виртуальных экземпляров операционных систем (гостевых ОС) в целях обеспечения их независимости от аппаратной платформы и сосредоточения нескольких виртуальных машин на одной физической.

В наши дни виртуализация на платформах Windows принимает одну из двух форм: тип 2 и гибридная (hybrid). Все начинается с базовой ОС, то есть с ОС, которая устанавливается непосредственно на физическое оборудование. Поверх базовой ОС работает монитор виртуальных машин (Virtual Machine Monitor, VMM), в задачу которого входит создание виртуальных машин и управление ими, распределение ресурсов между машинами, обеспечение изоляции машин друг от друга. Иными словами, в данном сценарии VMM играет роль уровня виртуализации (virtualization layer). Затем поверх VMM работают уже гостевые приложения. Ее производительность не велика, поскольку приложениям на пути к оборудованию приходится проходить как через VMM, так и через базовую ОС. [1]

В IT-среде более распространена гибридная виртуализация. Здесь непосредственно с оборудованием общаются как базовая ОС, так и VMM (хотя к различным аппаратным компонентам они имеют разный доступ), а гостевые ОС работают поверх уровня виртуализации. Точее, в этой конфигурации VMM также должен проходить через базовую ОС, чтобы получить доступ к оборудованию. Однако, как базовая ОС, так и VMM работают в режиме ядра и потому, по сути, конкурируют за обладание ресурсами ЦП. Базовой системе циклы процессора выделяются по мере надобности в ее контексте, затем циклы передаются VMM, а VMM передает циклы гостевым ОС. Процесс повторяется снова. Гибридная форма работает быстрее формы типа 2, поскольку в первом случае VMM работает в режиме ядра, а во втором – в пользовательском режиме

Имеется и третий тип технологии виртуализации – VMM типа 1 или технология гипервизора. Гипервизор (hypervisor) – это программный уровень, расположенный непосредственно над оборудованием и под одной или несколькими ОС. Его основное назначение – организовать изолированные среды выполнения, называемые разделами (partition), внутри которых будут работать виртуальные машины с гостевыми ОС. Каждому разделу выделяется собственный набор аппаратных ресурсов, в который входят память, процессорное время и устройства, а гипервизор отвечает за организацию доступа к реальному оборудованию.[2]

Можно сравнить два варианта VMM типа 1: монолитный и микроядерный.

В монолитной (monolithic) модели гипервизор использует для доступа к оборудованию собственные драйверы Гостевые ОС работают на виртуальных машинах поверх гипервизора. Когда гостевой системе нужен доступ к оборудованию, она должна пройти через гипервизор и его модель драйверов. Обычно одна из гостевых ОС играет роль ад-министратора или консоли, в которой запускаются компоненты для предоставления ресурсов, управления и мониторинга всех гостевых ОС, работающих на компьютере. Модель монолитного гипервизора обеспечивает прекрасную производительность, но уязвима с точки зрения защищенности и устойчивости. Это связано с тем, что она по своей сути обладает более широким фронтом нападения и подвергает систему большему потенциальному риску, поскольку разрешает работу драйверов (а иногда даже программ сторонних производителей) в очень чувствительной области.

Альтернативу монолитному подходу составляет микроядерная (microkernelized) модель. В ней можно говорить о «тонком гипервизоре» – в этом случае в нем совсем нет драйверов. Вместо этого драйверы работают в каждом индивидуальном разделе, чтобы любая гостевая ОС имела возможность получить через гипервизор доступ к оборудованию. При такой расстановке сил каждая виртуальная машина занимает совершенно обособленный раздел, что положительно сказывается на защищенности и надежности. Родительский раздел, является также корневым (root), поскольку он создается первым и владеет всеми ресурсами, не принадлежащими гипервизору. Обладание всеми аппаратными ресурсами означает среди прочего, что именно корневой (то есть, родительский) раздел управляет питанием, подключением самонастраивающихся устройств, ведает вопросами аппаратных сбоев и даже управляет загрузкой гипервизора. В родительском разделе содержится стек виртуализации – набор программных компонентов, расположенных поверх гипервизора и совместно с ним обеспечивающих работу виртуальных машин. Стек виртуализации обменивается данными с

гипервизором и выполняет все функции по виртуализации, не поддерживаемые непосредственно гипервизором. Большая часть этих функций связана с созданием дочерних разделов и управлением ими и необходимыми им ресурсами (центральный процессор, память, устройства). Стек виртуализации также обеспечивает доступ к интерфейсу управления.

Преимущество микроядерного подхода, примененного в Windows Server, по сравнению с монолитным подходом состоит в том, что драйверы, которые должны располагаться между родительским разделом и физическим сервером, не требуют внесения никаких изменений в модель драйверов. Иными словами, в системе можно просто применять существующие драйверы. В Microsoft этот подход избрали, поскольку необходимость разработки новых драйверов сильно затормозила бы развитие системы. Что же касается гостевых ОС, они будут работать с эмуляторами или синтетическими устройствами. С другой стороны, микроядерная модель может несколько проигрывать монолитной модели в производительности. Однако главным приоритетом стала безопасность, поэтому для большинства компаний вполне приемлема потеря пары процентов в производительности ради сокращения фронта нападения и повышения устойчивости.[3]

Список использованных источников:

1. Преимущества виртуализации [Электронный ресурс]. – Режим доступа: https://technet.microsoft.com/ru-ru/gg715011
2. Архитектура Hyper-V. Глубокое погружение [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/98580/
3. Михеев, М. Администрирование VMware vSphere 5 / М. Михеев. – СПб.: ДМК Пресс, 2012. – 508 с