Организация памяти в виртуальных машинах

Целью данного доклада является формирование небольшое представление о том, что такое виртуальная память и как она организована.

1. История

Представьте на секунду: начало 60-х годов 20 века, крупный научный институт, несколько сот сотрудников. На весь институт одна вычислительная машина. Если учёному нужно что-то посчитать, он пишет программу и несёт человеку, который работает с ЭВМ — оператору. Тот запускает машину, а когда обработка заканчивается, отдаёт программисту результат.

Вычислительные машины в те годы были редкими, медленными и очень дорогими. У самих программистов доступа к ним не было, потому что «индивидуальная работа» считалась неэффективной: пока человек вводил данные или думал, машина простаивала, а это недопустимо. Чтобы простоя не было, разработчиков и ЭВМ разделяли.

Со временем учёные догадались, что если допустить к компьютеру не одного, а нескольких пользователей, эффективность повысится. Пока один будет вводить данные, компьютер будет решать задачи других. Паузы заполнятся, простоя не будет. Эту идею назвали концепцией разделения времени.

Концепция разделения времени или time-sharing — это когда вычислительные ресурсы распределяют между многими пользователями. Она появилась в начале 1960-х годов и привела ко многим революционным изменениям, в том числе, появлению виртуализации.

Сначала идею разделения времени реализовали буквально: процессор переключался между задачами только на время операций ввода-вывода. Пока один пользователь думал или вводил данные, компьютер обрабатывал задачи другого.

Оборудование становилось всё мощнее, задач нескольких пользователей ему уже не хватало. Тогда процессор научили переключаться между задачами чаще. Каждая задача получала «квант» времени, когда процессор занимался её обработкой. Если не успевал завершить, переходил к другой, а первая задача ждала следующего кванта. Переключения происходили так быстро, что каждому пользователю казалось, будто он использует машину единолично.

Впервые концепция разделения времени была реализована в Compatible Time-Sharing System в Массачусетском технологическом институте где-то в начале 1960-х. Она была совместима с Fortran Monitor System и работала на майнфрейме IBM 7090.

Первой операционной системой с поддержкой time-sharing стала Multics, предшественник ОС семейства Unix. И Multics, и система из Дармутского колледжа нашли себе применение, но были далеки от совершенства: медленные, нестабильные, небезопасные. Ученые хотели и знали, как сделать лучше, только возможности оборудования были ограничены. Нужна была поддержка производителей, и вскоре те подключились к работе. Так и зародилась виртуализация.

2. Определения

ВМ – программа, эмулирующая аппаратное обеспечение ПК и исполняющая программы гостевой-платформы на хост-платформе.

Хост-платформа – ПК, на котором стоит ВМ.

Гостевая платформа – ПК, реализованный внутри ВМ.

3. Первое впечатление

Запустив ВМ с несколькими гостевыми платформами на ней у вас создаться впечатление, что каждая из платформ имеет свой собственный процессор, ОП и т.д. На самом деле виртуальная машина не имеет доступа к физическим ресурсам реального компьютера. Работа с ними возложена на упоминавшийся ранее гипервизор

4. Гипервизор

**Гипервизор** – это программа, которая управляет физическими ресурсами вычислительной машины и распределяет эти ресурсы между несколькими различными операционными системами, позволяя запускать их одновременно.

4.1. Месторасположение

1 тип

Работают на «голом железе» и не требуют установки какой-либо операционной системы в качестве прослойки. Их также называют аппаратными гипервизорами.

Примеры: VMware ESXi, Hyper-V, KVM.

Ориентированы на бизнес-задачи и использование в корпоративных средах. С их помощью можно виртуализировать не только серверные ОС. Гипервизоры первого типа позволяют также виртуализировать десктопные операционные системы и интегрировать VDI. Благодаря этому компании получают возможность развернуть полноценную инфраструктуру виртуальных рабочих столов и предоставить сотрудникам безопасную среду для решения задач вне зависимости от их физического нахождения. VDI могут использовать не только те, кто работает в офисе, но также удаленные и полевые сотрудники.

Администратор дата-центра создает виртуальное рабочее место для пользователя с необходимым набором документов, файлов, программ, устанавливает права доступа и подключает его виртуальный компьютер к серверу через мини-ПК — «тонкий клиент». Он поддерживает работу протоколов передачи данных, работает со всеми ОС, поддерживает локальные сети и удаленный доступ.

Пользователь подключается к своему виртуальному рабочему месту с помощью интернета через специальное программное обеспечение или «тонкого клиента». После подключения он видит свой привычный рабочий стол с папками, документами и ярлыками, которые хранятся на ВМ, и может полноценно работать из любой геолокации в любое время.

Преимущества

Благодаря прямому доступу к железу, гипервизоры I типа обладают высокой производительностью. Кроме того, они достаточно безопасны, поскольку между ПО и процессором нет никакой прослойки, которая может подвергнуться атаке и последующему взлому.

Недостатки

Нередко нужна отдельная виртуальная машина для управления аппаратным обеспечением и администрирования определенных ВМ.

2 тип

Запускаются поверх операционной системы, установленной на хосте.

Примеры: VMware Workstation, Oracle VirtualBox.

Гипервизоры второго типа ближе к «домашним» решениям и для корпоративного сегмента не подходят.

Преимущества

Позволяют быстро получить доступ к необходимой операционной системе, которая работает параллельно с основной. Наиболее оптимальный вариант для пользователей, которым нужна альтернативная ОС для запуска определенных программ и инструментов.

Недостатки

Из-за того, что гипервизоры второго типа обращаются к ресурсам через основную ОС, установленную на машине-хосте, может увеличиваться время отклика. К тому же потенциально снижается уровень безопасности. Если киберпреступник получит доступ к основной операционной системе, он сможет управлять и гостевой ОС. В силу этих и некоторых других факторов гипервизоры II типа не используются в корпоративной среде.

Стоит также отметить, что аппаратная виртуализация потенциально несет в себе не только положительные моменты. Возможность управления гостевыми системами посредством гипервизора и простота написания платформы виртуализации с использованием аппаратных техник дают возможность разрабатывать вредоносное программное обеспечение, которое после получения контроля на хостовой операционной системой, виртуализует ее и осуществляет все действия за ее пределами.

В начале 2006 года в лабораториях Microsoft Research был создан руткит под кодовым названием SubVirt, поражающий хостовые системы Windows и Linux и делающий свое присутствие практически не обнаруживаемым. Принцип действия этого руткита заключался в следующем:

Через одну из уязвимостей в операционной системе компьютера вредоносное программное обеспечение получает административный доступ.

После этого, руткит начинает процедуру миграции физической платформы на виртуальную, по окончании которой происходит запуск виртуализованной платформы посредством гипервизора. При этом для пользователя ничего не меняется, все продолжает работать, как и раньше, а все средства и службы, необходимые для доступа к гипервизору извне (например, терминального доступа), находятся за пределами виртуализованной системы.

Антивирусное программное обеспечение после осуществления процедуры миграции не может обнаружить вредоносный код, поскольку он находится за пределами виртуализованной системы.

Наглядно эта процедура выглядит так: (слайд)

4.2. Виды

- аппаратный

- программный

4.2.1. Программный

Начнем с него, так как он появился раньше. Связано с тем, что для его разработки не нужно менять архитектуру ПК, нужно просто написать программу.

Программный гипервизор — это эмулятор, программа, которая запускается внутри операционной системы и имитирует работу виртуального компьютера. Всё, что она делает, — это выделяет немного ресурсов и следит за тем, чтобы гостевая система не вышла за их пределы. Иногда такой гипервизор позволяет обращаться к секторам диска или регистрам процессора напрямую, но так бывает не всегда.

Программные гипервизоры проще в установке и могут работать почти на любом железе. Но из-за того, что все команды и разрешения на доступ к железу проходит через них, то реальная скорость работы системы на таком гипервизоре ниже, чем на таком же, но настоящем железе.

4.2.2. Аппаратный

Необходимость поддержки аппаратной виртуализации заставила производителей процессоров несколько изменить их архитектуру за счет введения дополнительных инструкций для предоставления прямого доступа к ресурсам процессора из гостевых систем. Этот набор дополнительных инструкций носит название Virtual Machine Extensions (VMX). VMX предоставляет следующие инструкции: VMPTRLD, VMPTRST, VMCLEAR, VMREAD, VMREAD, VMWRITE, VMCALL, VMLAUNCH, VMRESUME, VMXON и VMXOFF.

Процессор с поддержкой виртуализации может работать в двух режимах root operation и non-root operation. В режиме root operation работает специальное программное обеспечение, являющееся «легковесной» прослойкой между гостевыми операционными системами и оборудованием — монитор виртуальных машин (Virtual Machine Monitor, VMM), носящий также название гипервизор (hypervisor). Слово «гипервизор» появилось интересным образом: когда-то очень давно, операционная система носила название «supervisor», а программное обеспечение, находящееся «под супервизором», получило название «гипервизор».

Чтобы перевести процессор в режим виртуализации, платформа виртуализации должна вызвать инструкцию VMXON и передать управление гипервизору, который запускает виртуальную гостевую систему инструкцией VMLAUNCH и VMRESUME (точки входа в виртуальную машину). Virtual Machine Monitor может выйти из режима виртуализации процессора, вызвав инструкцию VMXOFF.