Организация памяти в виртуальных машинах

Целью данного доклада является формирование небольшого представления о том, как организована память в ВМ.

***1. История***

Представьте начало 60-х годов 20 века. 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.

Первой операционной системой с поддержкой time-sharing стала Multics, предшественник ОС семейства Unix. Она нашла себе применение, но была далека от совершенства: медленной, нестабильной, небезопасной. 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 гипервизор

***4. Гипервизор***

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

***4.1. Виды***

- аппаратный

- программный

***4.1.1. 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.

***4.1.2. 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.

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 .

Чтобы перевести процессор в режим виртуализации, платформа виртуализации должна передать управление гипервизору, который запускает виртуальную гостевую систему.

***4.2. 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.

**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**

Если говорить про недостатки, то 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.