Современные виртуальные машины

Кирилл Батузов

ИСП РАН

16 октября 2013

Введение

Всего про виртуальные машины будет две лекции:

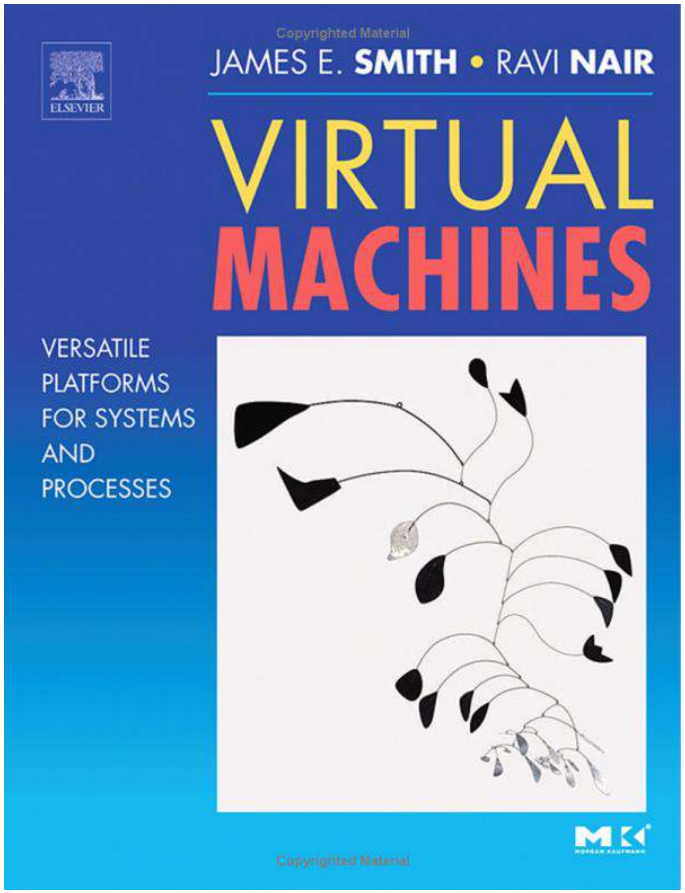
- введение в виртуальные машины,
- эмуляция системы команд.

Лекции будут обзорные и многие вопросы в них рассматриваться не будут, как например:

- виртуализация аппаратных ресурсов,
- обеспечение корректной работы исключений,
- управление кэшем трансляций
- и т.д.

Литература

James E. Smith, Ravi Nair. “Virtual Machines: Versatile Platforms for Systems and Processes”



Виртуальные машины

Виртуальные машины

Под словом «машина» будем подразумевать то, что выполняет программу.

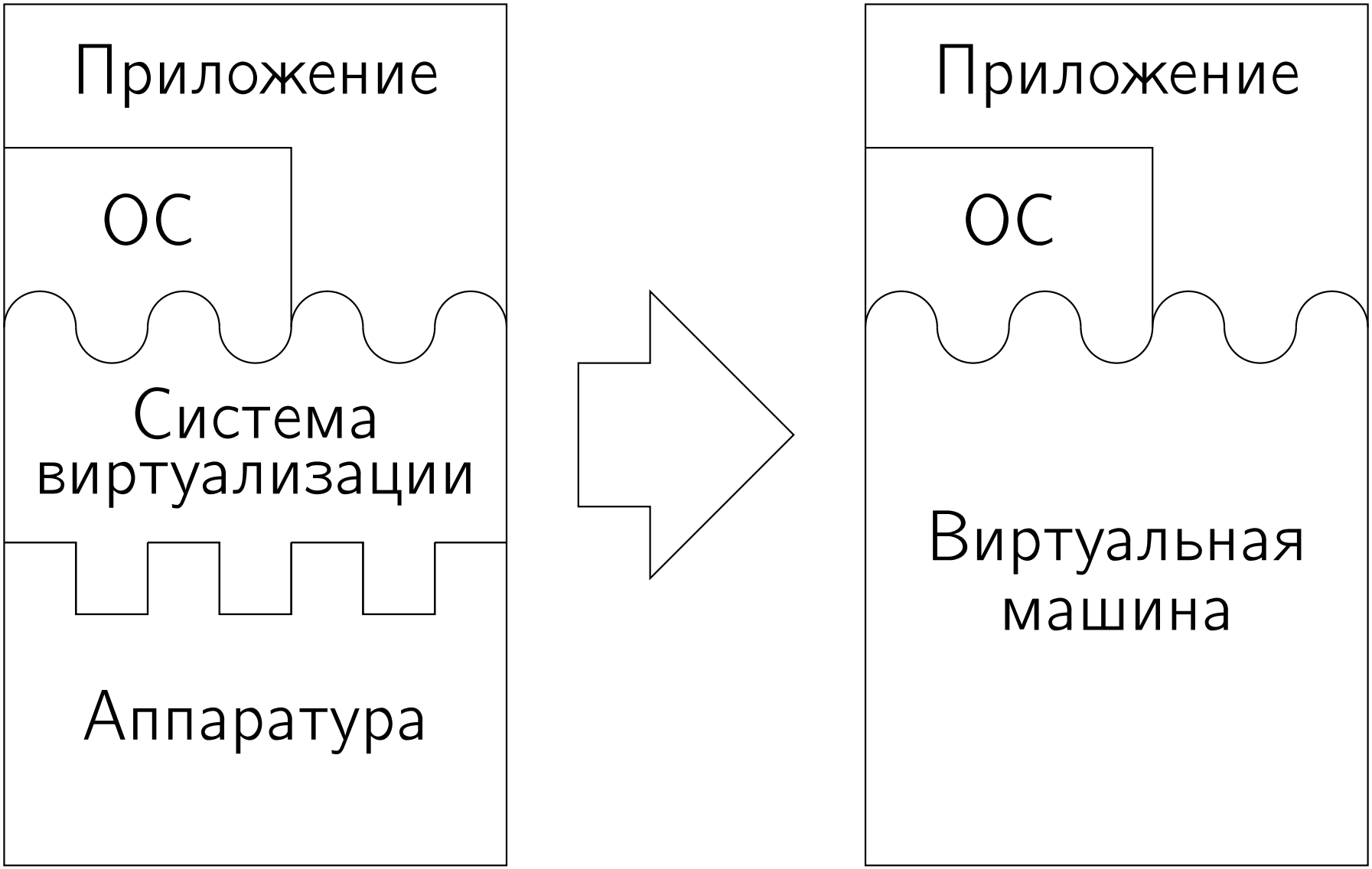
- С точки зрения операционной системы машина определяется ее аппаратной архитектурой (ISA).
- С точки зрения компилятора — ABI (прикладная ISA + системные вызовы).
- С точки зрения прикладной программы — API (прикладная ISA + набор библиотечных функций).

Виртуальные машины

Назовем эмуляцией процесс реализации интерфейса и функциональности одной системы или подсистемы поверх другой, имеющей отличный интерфейс. Полученная система или подсистема при этом будет называться виртуальной.

Виртуальные машины — частный случай таких эмулируемых систем. Существуют различные типы виртуальных машин в зависимости от того, какой интерфейс эмулируется (API, ABI или ISA).

Виртуальные машины



Преимущества виртуальных ресурсов

- Виртуальная память
  - совместное использование, защита, большое адресное пространство.
- Java Virtual Machine (JVM)
  - кросс-платформенность, защита.
- Виртуальные устройства ввода-вывода
  - гибкость, простота совместного использования, простота управления, независимость от реализации.
- Virtual Private Network (VPN)
  - безопасное соединение по незащищенному каналу.

Типы виртуальных машин

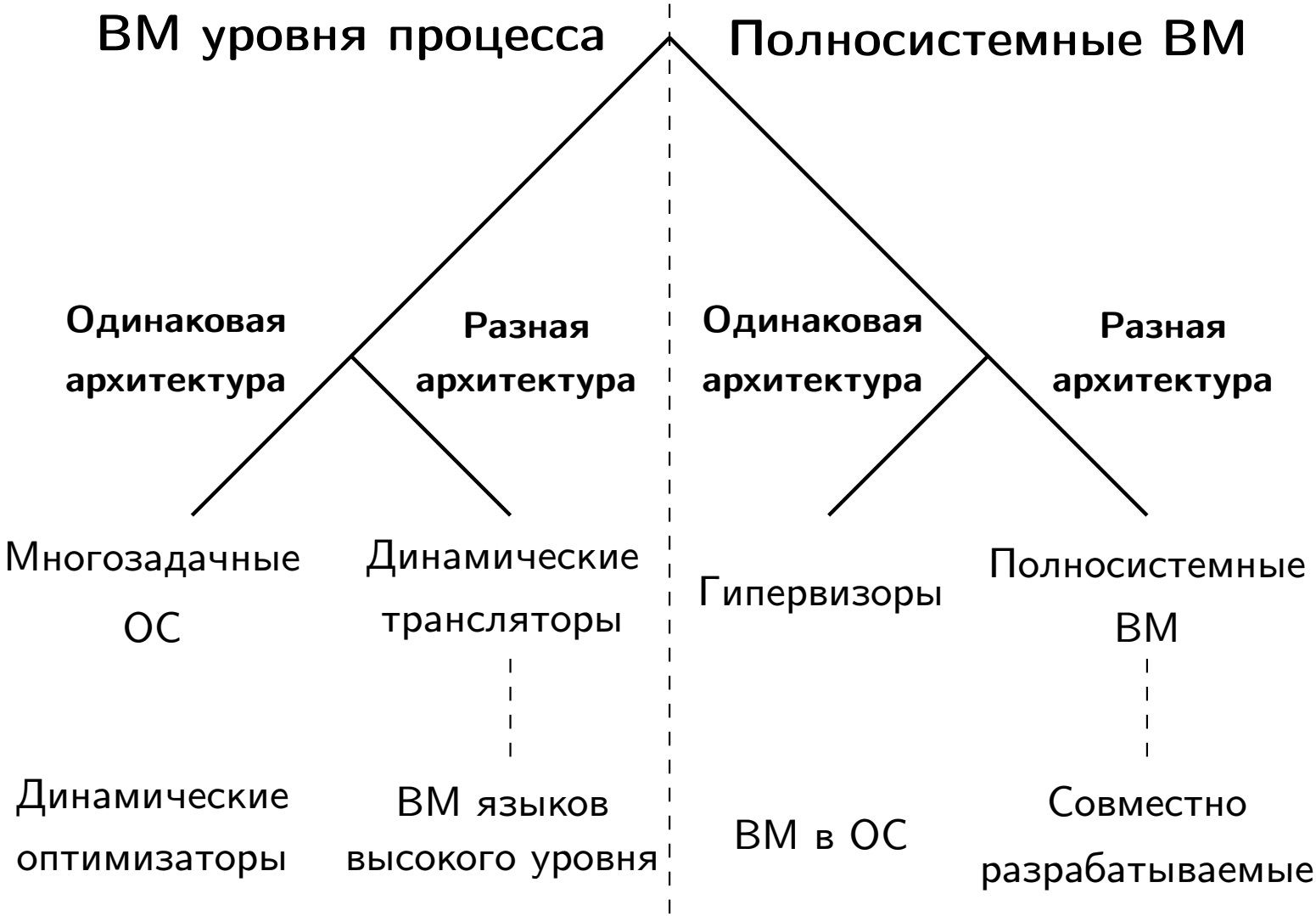
По тому, что они эмулируют, виртуальные машины можно разбить на два класса:

- Полносистемные виртуальные машины — эмулируют ISA.
- Виртуальные машины уровня процесса — эмулируют ABI.

С другой стороны, устройство виртуальной машины и ее быстродействие очень сильно зависят от гостевой архитектуры. Различают два случая:

- гостевая и основная архитектуры одинаковы,
- гостевая архитектура отлична от основной.

Классификация виртуальных машин



Многозадачные операционные системы

- Настолько распространены и привычны, что их как правило даже не считают за виртуальные машины.
- Одинаковая архитектура, одинаковая ОС.
- Изолирует процессы друг от друга, создавая иллюзию, что у каждого из них свой собственные компьютер, на котором он запущен.
- Примеры: GNU/Linux, MS Windows.

Динамические трансляторы и оптимизаторы

Динамические трансляторы.

- Одинаковая ОС, различная архитектура.
- Позволяют запускать программы, скомпилированные для одной архитектуры на другой.
- Производят интерпретацию кода, либо транслируют его в код основной архитектуры.
- Примеры: QEMU, IA-32 EL.

Динамические оптимизаторы.

- Одинаковая ОС, одинаковая архитектура.
- Пытаются увеличить производительность за счет оптимизации двоичного кода.
- Используют динамическую двоичную трансляцию.
- Пример: HP Dynamo.

Виртуальные машины языков высокого уровня

- Программа на языке высокого уровня компилируется в байт-код, который выполняется виртуальной машиной.
- Байт-код делается таким, чтобы виртуальная машина была максимально простой и эффективной.
- Позволяет добиться очень хорошей переносимости программ.
- Для переноса всех уже существующих программ на новую архитектуру достаточно написать виртуальную машину для этой архитектуры.
- Из недостатков — потеря производительности.
- Примеры: JVM, MS CLI.

Гипервизоры

- Одинаковая архитектура.
- Виртуальная машина работает непосредственно на аппаратуре.
- Виртуальная машина выполняется в привилегированном режиме.
- Самые эффективные из всех виртуальных машин.
- Виртуальная машина должна содержать драйвера нужных устройств.
- Пример: Xen, IBM VM/370.

ВМ, работающая под управлением ОС

- Одинаковая архитектура.
- Виртуальная машина работает как пользовательский процесс в ОС.
- Часть виртуальной машины может работать в привилегированном режиме в составе ядра ОС.
- Проще в установке и использовании, чем гипервизоры.
- Общается с устройствами через основную ОС и не содержит драйверов внутри себя.
- Высокие накладные расходы на операции с устройствами.
- Примеры: VMware Wk, VBox, KVM.

Полносистемные виртуальные машины

- Различная архитектура — необходимо программно эмулировать систему команд.
- Работает под управление ОС.
- Универсальность — подход позволяет эмулировать любую архитектуру и периферийные устройства.
- Очень низкая производительность по сравнению с предыдущими.
- Примеры: QEMU.

Совместно разрабатываемые ВМ

- Различная архитектура.
- Работает непосредственно на аппаратуре.
- Аппаратура и виртуальная машина разрабатываются совместно.
- Цель — получить возможность выполнять накопленное ПО для старой архитектуры на новом процессоре.
- Эффективнее предыдущего типа.
- Пример: Transmeta Code Morphing.

Применение виртуальных машин

Консолидация серверов и облачные вычисления

- Даже самый дешевый сервер стоит не так уж дешево и обладает избыточной для многих задач производительностью.
- Виртуальные машины позволяют разбить один мощный сервер на несколько более слабых. Удельная стоимость каждого такого сервера будет ниже.
- Консолидация серверов — объединение нескольких серверов в одном физическом компьютере.
- Облачные вычисления — множество физических серверов разбивается на еще большее множество виртуальных, которые обеспечивают выполнение различных сервисов.
  - Virtual Private Server (VPS)

Создание набора различных окружений и виртуальное аппаратное обеспечение

- Разработка и отладка переносимого ПО требует доступа к большому числу конфигураций для отладки. Делать их все реальными бывает слишком затратно.
- Разработка мобильных приложений.
- Разработка приложения для еще не поступивших в продаже платформ.
- Воссоздание конфигураций, на которых проявляется некорректное поведение программы.

Тестирование и обеспечение качества

- Нередки случаи, когда серверное ПО требует отдельную машину для запуска.
- Виртуальные машины позволяют обеспечить возможность отладки для всех разработчиков с незначительными затратами.
- Виртуальные машины позволяют автоматически выполнять тестовые сценарии и записывать результаты.
- «Горячая» замена — можно заменять текущую версию ПО на уже развернутую и, главное, протестированную новую версию.

Отладка

- Виртуальная машина имеет полный доступ к состоянию гостевой системы в любой момент времени.
- Полный доступ включает в себя в том числе и различные служебные регистры, считать значение которых на реальной аппаратуре может быть затруднительно.
- В виртуальной машине можно отлаживать не только прикладное ПО, но и системное, включая ядро операционной системы.
- Виртуальные машины широко используются для отладки мобильных приложений.
- Виртуальная машина может сохранять свое состояние и выполнять «откат» на это состояние позже.
  - Обратная отладка.

Контролируемое окружение

- Безопасный запуск сомнительных программ и операционных систем.
- Мониторинг действий пользователя и программ при работе с конфиденциальными данными.
- Ограничение возможностей пользователя и программ при работе с конфиденциальными данными.
- Расследование инцидентов с нарушением режима доступа к информации.
- Обнаружение атак.

Подведем итог

Виртуальные машины

Эмуляция — это процесс реализации интерфейса и функциональности одной системы или подсистемы поверх другой, имеющей отличный интерфейс. Полученная система или подсистема при этом называется виртуальной.

Виртуальные машины — частный случай таких эмулируемых систем. Существуют различные типы виртуальных машин в зависимости от того, какой интерфейс эмулируется (ISA или ABI).

Классификация виртуальных машин