Виртуализация аппаратного обеспечения

Николай Иготти

Богомол

Идеологические моменты

- Можно реализовывать конкретную архитектуру в ВМ
- И исполнять программы и ОС для этой архитектуры в ВМ
- Что позволяет
  - *прототипировать новые архитектуры*
  - *гибко управлять ресурсами существующих*
  - *использовать программы для устаревших*
- Сложность в приемлемой производительности полученной реализации
- Весь дальнейший курс – о сложностях и их преодолении

Немного истории

- 1964 – CP/40, IBM (14 виртуальных машин одновременно) – полная виртуализация
- 1966 - M44/44X, IBM - паравиртуализация, виртуальная память
- 1972 – VM/370, IBM – полноценный гипервизор, используется до сих пор
- 1974 – формальные критерии виртуализирумости Попека-Голдберга
- 1995 – основание Transmeta, процессоры с бинарным транслятором
- 1998 – VMWare – первый виртуализатор x86 с бинарным транслятором
- 2005 – Intel/AMD – расширения x86 для аппаратной виртуализации

Основные проблемы

- Производительность (исполнение инструкций)
- Производительность (доступ к памяти)
- Производительность (выполнение ввода-вывода)
- Производительность (время отклика системы)
- Безопасность, стабильность
- Точная поведенческая эмуляция, особенно для сложных устройств ввода-вывода

Подходы к решению

- Программное моделирование целевой архитектуры
- Интерпретация
- Динамическая трансляция
- Непосредственное исполнение инструкций
- Непосредственное использование MMU
- Прозрачная обработка исключений реальной машины
- Программные модели оборудования
- Непосредственный (безопасный) доступ к оборудованию

Критерии Попека-Голдберга

- Критерии полезной виртуализуемости архитектуры для систем непосредственного исполнения (Java?)
- Необходимы
  - *Эквивалентность*
  - *Контроль VMM над ресурсами*
  - *Эффективность*
- В наборе инструкций
  - *Привилегированные инструкции (P)*
  - *Чувствительные (по управлению, по поведению) инструкции (S)*
  - *Все инструкции (A)*
- От архитектуры требуются

$$S \subseteq P \qquad P \underset{\substack{стат.\\знач}}{\subset} A$$

Попек-Голдберг и x86

- X86 не удовлетворяет критериям ПГ
- Из-за наличия чувствительных непривилегированных инструкций (S[GIL]DT, SMWS, PUSHF, LAR, POP [CS]S, STR)
- Тем не менее, можно создать эффективный VMM
- Используя механизм бинарной трансляции
- Или расширения архитектуры виртуализирующие привилегированное состояние (VT-x, AMD-V)

Вопросы

- Какие аппаратные компоненты критичны для реализации VMM?
- В чём отличие механизма прерываний от устройств ввода/вывода в виртуализированном окружении
- Доступ к памяти – привилегированная операция или нет?
- Как проверить, удовлетворяет ли архитектура XYZ критериям ПГ?