

Прямое исполнение

Курс «Программное моделирование вычислительных систем»

Григорий Речистов
grigory.rechistov@phystech.edu

13 марта 2016 г.

- 1 Прямое исполнение
- 2 Предпросмотр
- 3 Аппаратная поддержка
- 4 Коробка передач
- 5 Заключение

На прошлой лекции

- Интерпретаторы — медленная шутка
- Двоичная трансляция быстрее, потому что вычисляет «меньше»

Вопросы

- ЯВО \rightarrow маш. код — компиляция. Маш. код \rightarrow маш. код — ДТ. А что такое маш. код \rightarrow ЯВО?

Вопросы

- ЯВО \rightarrow маш. код — компиляция. Маш. код \rightarrow маш. код — ДТ. А что такое маш. код \rightarrow ЯВО? Декомпиляция

Вопросы

- ЯВО \rightarrow маш. код — компиляция. Маш. код \rightarrow маш. код — ДТ. А что такое маш. код \rightarrow ЯВО? Декомпиляция
- А ЯВО \rightarrow ЯВО?

Вопросы

- ЯВО \rightarrow маш. код — компиляция. Маш. код \rightarrow маш. код — ДТ. А что такое маш. код \rightarrow ЯВО? Декомпиляция
- А ЯВО \rightarrow ЯВО? Source-level компилятор

Вопросы

- ЯВО \rightarrow маш. код — компиляция. Маш. код \rightarrow маш. код — ДТ. А что такое маш. код \rightarrow ЯВО? Декомпиляция
- А ЯВО \rightarrow ЯВО? Source-level компилятор
- В каких случаях ДТ будет медленнее интерпретации на одной и той же гостевой программе?

Вопросы

- ЯВО \rightarrow маш. код — компиляция. Маш. код \rightarrow маш. код — ДТ. А что такое маш. код \rightarrow ЯВО? Декомпиляция
- А ЯВО \rightarrow ЯВО? Source-level компилятор
- В каких случаях ДТ будет медленнее интерпретации на одной и той же гостевой программе? Если часто происходят ретрансляции, например, программа полна SMC

Когда применимо прямое исполнение

Идея: не симулировать код вообще! Direct execution, DEX

- Когда гостевая ISA совпадает с хозяйской
- Ну или почти совпадает

Алгоритм

```
execute() {  
    save_host_ctx();  
    set_guest_ctx();  
    setjmp(back);  
    goto guest_start_ip;  
back: restore_host_ctx();  
}
```

Почему это не будет работать

- Не полностью совпадающие ISA
- Различное положение внешних ресурсов (устройств и памяти)
- Привилегированность некоторых инструкций
- Необходимость изоляции симулятора от обнаружения и разрушения гостем

Почему это не будет работать

```
add %r1, %r2  
mul $10, %r3  
sub %r11, %r1  
mov $16, %r13
```

OK

```
div %r4, %r5
```

Отсутствующая в хозяине инструкция

```
ld (0xa000), %r10  
st %r10, (%r11)
```

Другое расположение в памяти

```
mov %r13, %cr0  
trap $32
```

Привилегированные инструкции

Предпросмотр кода

- Инспектирование гостевого кода на предмет «опасных» инструкций перед исполнением
- Замена части инструкций на контролируемые — *инструментация*
- «Почти» двоичная трансляция

Заплаты и заглушки

Исходный код	Код после инструментации
<code>add %r1, %r2</code>	<code>add %r1, %r2</code>
<code>mul \$10, %r3</code>	<code>mul \$10, %r3</code>
<code>div %r4, %r5</code>	<code>trap \$255</code>
<code>ld (0xa000), %r10</code>	<code>ld (0xb000), %r10</code>
<code>st %r10, (%r11)</code>	<code>st %r10, (%r11)</code>
<code>sub %r11, %r1</code>	<code>sub %r11, %r1</code>
<code>mov \$16, %r13</code>	<code>mov \$16, %r13</code>
<code>mov %r13, %cr0</code>	<code>trap \$255</code>
<code>trap \$32</code>	<code>trap \$255</code>

`stub` — заглушка, `patch` — заплатка

Двоичная инструментация

Общее название для методик исследования и модификации приложений

- Pin <http://pintool.org>
- DynamoRIO <http://dynamorio.org>

Сложности DEX

- Необходимость предпросмотра негативно влияет на производительность симуляции
- Необходимость контролировать возможность самомодификации кода
- Переменная длина инструкций усложняет установку заплаток и заглушек
- Необходимость гарантировать, что управление вернётся из гостя в симулятор

Аппаратная поддержка прямого исполнения

- Набор интерфейсов аппаратуры хозяйской системы, предназначенных для упрощения DEX
- Аппаратная поддержка типичных операций: загрузка гостевого состояния, контроль за исполнением инструкций и доступов к ресурсам, обработка внешних и внутренних исключений
- Аппаратная виртуализация: тема отдельной лекции

Плюсы и минусы

- Прямое исполнение большинства инструкций \Rightarrow самый быстрый способ симуляции
- Упрощение модели, уменьшение объёма кода симулятора
- Необходима аппаратная поддержка
- Надо писать код для ядра ОС
- Медленное переключение между режимами
- Работает только при совпадении архитектур
- Не все режимы процессора могут быть смулированы

Предпочтительный подход: модуль ядра для частых операций и пользовательское приложение для всего остального

Спектр симуляционных подходов

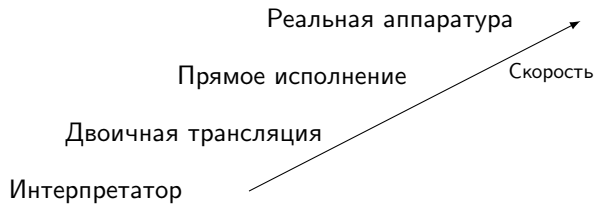
Реальная аппаратура

Прямое исполнение

Двоичная трансляция

Интерпретатор

Спектр симуляционных подходов



Спектр симуляционных подходов



Коробка передач






Динамическое переключение режимов

- Оптимальное использование лучших сторон каждого из подходов
- Необходимость разработки фактически нескольких симуляторов

Итоги

- Наивное прямое исполнение
- Заплатки и заглушки
- Аппаратная поддержка прямого исполнения
- Переключение режимов симуляции и условия на переходы между ними

Литература I

-  Kevin P. Lawton. Plex86: An 180x86 Virtual Machine - 2000
https://www.usenix.org/legacy/publications/library/proceedings/als00/2000papers/papers/full_papers/lawton/lawton.pdf
-  Pin - A Binary Instrumentation Tool - Papers
<https://software.intel.com/en-us/articles/pin-a-binary-instrumentation-tool-papers>
-  Heidi Pan, Krste Asanović, Robert Cohn and Chi-Keung Luk.
Controlling Program Execution through Binary Instrumentation <http://www.cs.berkeley.edu/~krste/papers/pin-wbia.pdf>

Литература II



F. Leung, G. Neiger, D. Rodgers, A. Santoni, R. Uhlig. Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization // Intel Technology Journal 10 (03) Aug 2006

<http://atakuu.doesntexist.org/public/archive/papers/Intel-VT-Hardware-Support-for-Efficient-Processor-Virtualization.pdf>



Программная симуляция микропроцессора. Коробка передач

<http://habrahabr.ru/company/intel/blog/202926/>

На следующей лекции

Симуляция периферийных устройств и полной платформы

Спасибо за внимание!

Слайды и материалы курса доступны по адресу

<http://is.gd/ivuboc>

Замечание: все торговые марки и логотипы, использованные в данном материале, являются собственностью их владельцев. Представленная здесь точка зрения отражает личное мнение автора, не выступающего от лица какой-либо организации.