Имитационное моделирование в исследовании и разработке информационных систем

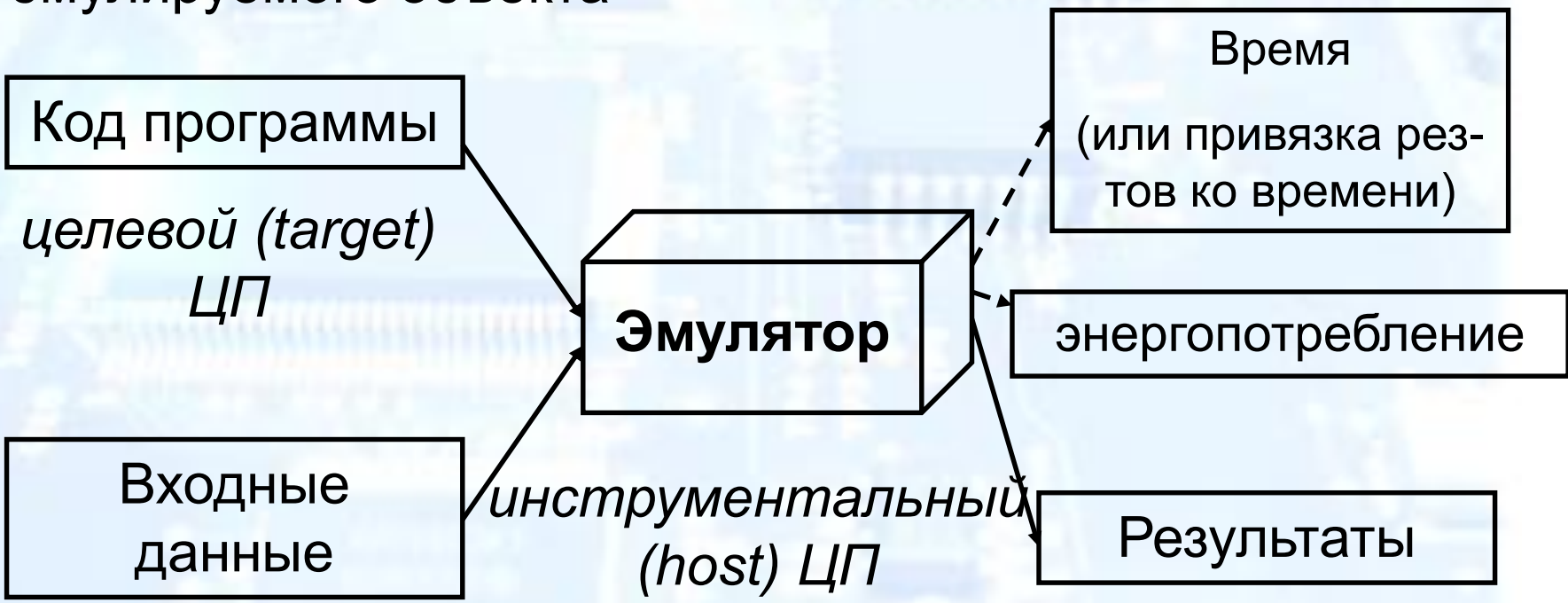
Лекция 7

Эмуляторы процессоров

Эмулятор

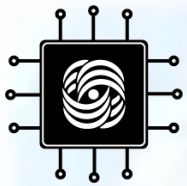
(вставить определение из словаря)

Акцент не на изучение, а на использование взамен эмулируемого объекта



Характеристики

- Архитектуры host и target
  - Для одной целевой архитектуры
  - Для различных архитектур (retargetable)
- Точность и детальность
  - Без учёта времени выполнения
  - С потактовой точностью времени (cycle clock accurate)
  - С потактовой точностью (cycle accurate)
- Скорость работы



Интерпретирующий эмулятор

```
while (!stop())
{
    декодировать;
    выполнить;
    обновить_состояние;
    выбрать следующую команду;
}
```

Simplescalar

- www.simplescalar.com
- доступен бесплатно для учебных и исследовательских целей (на сайте последняя версия 3.0 от 2003 г.)
- Наборы команд:
  - Alpha; PISA (учебная машина); в версии 4.0 заявлено ARM и x86
  - Возможность задавать свой набор команд
- модели кэш-памяти
- ? системные вызовы

Язык для описания набора команд (LISA)

```
01:   RESOURCE {
02:     PROGRAM_MEMORY byte8 prog_mem[0x0..0x1000];
03:     REGISTER word32 R[1..15];
04:   }
05:   OPERATION ADD {
06:     DECLARE { GROUP dst,src1,src2 = {Register}}
07:     CODING { 0b01011 0b0000 src1 src2 dst}
08:     SYNTAX { "ADD" dst "," src1 "," src2 }
09:     BEHAVIOR{ dst = src1 + src2}
10:   }
11:   OPERATION Register {
12:     DECLARE { LABEL index; }
13:     CODING { index=0bx[4]}
14:     SYNTAX { "R" index }
15:     EXPRESSION { R[index]}
16:   }
```

Эмуляторы с двоичной трансляцией

Статическая трансляция

- в C/C++, затем в машинный код для хоста;
- Напрямую в команды хоста;
- В команды абстрактной машины.

- Ускорение – до сотен раз

Динамическая двоичная трансляция

- Во время выполнения программы
- Кэш для результатов трансляции
- Единица трансляции – как правило, линейный участок
- Вопросы оптимизации, в т.ч. отображение виртуальных регистров на регистры хост-машины
- (рекорд?) 1,6 раз медленнее исполнения на натурной цели [2]

Эмулятор QEMU

- www.qemu.org
  - «full system emulator»
  - «user-space emulator»
- одна из первых статей – 2005 г.
- В составе:
  - эмулятор ЦП;
  - эмуляторы периферийных устройств
- x86, PowerPC, ColdFire (m68k), SPARC, MIPS, ARM …

Двоичная трансляция в QEMU

TCG (Tiny Code Generator):

- Команда целевого ЦП разделяется на микрооперации;
- Микрооперации реализуются на Си и компилируются для хоста;
- Двоичный транслятор склеивает нужные реализации микроопераций для очередного линейного участка
- Имеет место оптимизация

Усовершенствование двоичной трансляции

- Использовать профессиональный набор средств для построения компиляторов, в частности – LLVM
- Применено в эмуляторах:
- Rapido
- llvm-qemu
- SimSoc

LLVM

- www.llvm.org
- Промежуточное представление

(машинные операции, языково-независимая система типов, модульность …);

- Средства глобальной оптимизации;
- Средства межпроцедурной оптимизации;
- C, C++, … “front-ends” (на базе GCC);
- Многоцелевой генератор машинного кода;
- JIT генератор кода;
- …

Сравнение методов динамической трансляции

SimSoc

DT0: чистая интерпретация;

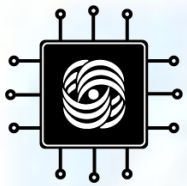
DT1 «простая динамическая трансляция»

DT2: инструкция –> функция (статически);
п-сть вызовов – динамически;

DT3: генерация внутреннего представления LLVM на основе “домашних заготовок”

DT3 вдвое быстрее DT2

DT2 в 5-10 раз быстрее DT0



Учёт времени выполнения при быстрой эмуляции (1)

Частный случай:

- входной язык C/C++;
- выполняем программу на хосте, на цели не выполняем (считаем рез-$т_{хост}$==рез-$т_{цель}$);
- нужна оценка времени именно для выполнения на цели

Статико-динамический подход (1) (ЛВК)

- Программа разбивается на фрагменты (как правило, линейные участки)
- Получаем ассемблерный текст фрагментов для целевого процессора
- Размечаем исходный текст (обновление счётчика времени по прохождении фрагмента)
- Получаем **статически** (до прогона) оценки времени выполнения фрагментов
- В ходе выполнения, **динамически**, рассчитываем оценку времени

Статико-динамический подход (ЛВК)(2)

- 1998 г. Цель: Motorola DSP96002
- Хост: microSPARC 33 МГц, i486dx40 (для эмулятора)
- Погрешность: 0%;
- Выигрыш во времени: 10^3

Модель NM6403 (ЛВК)

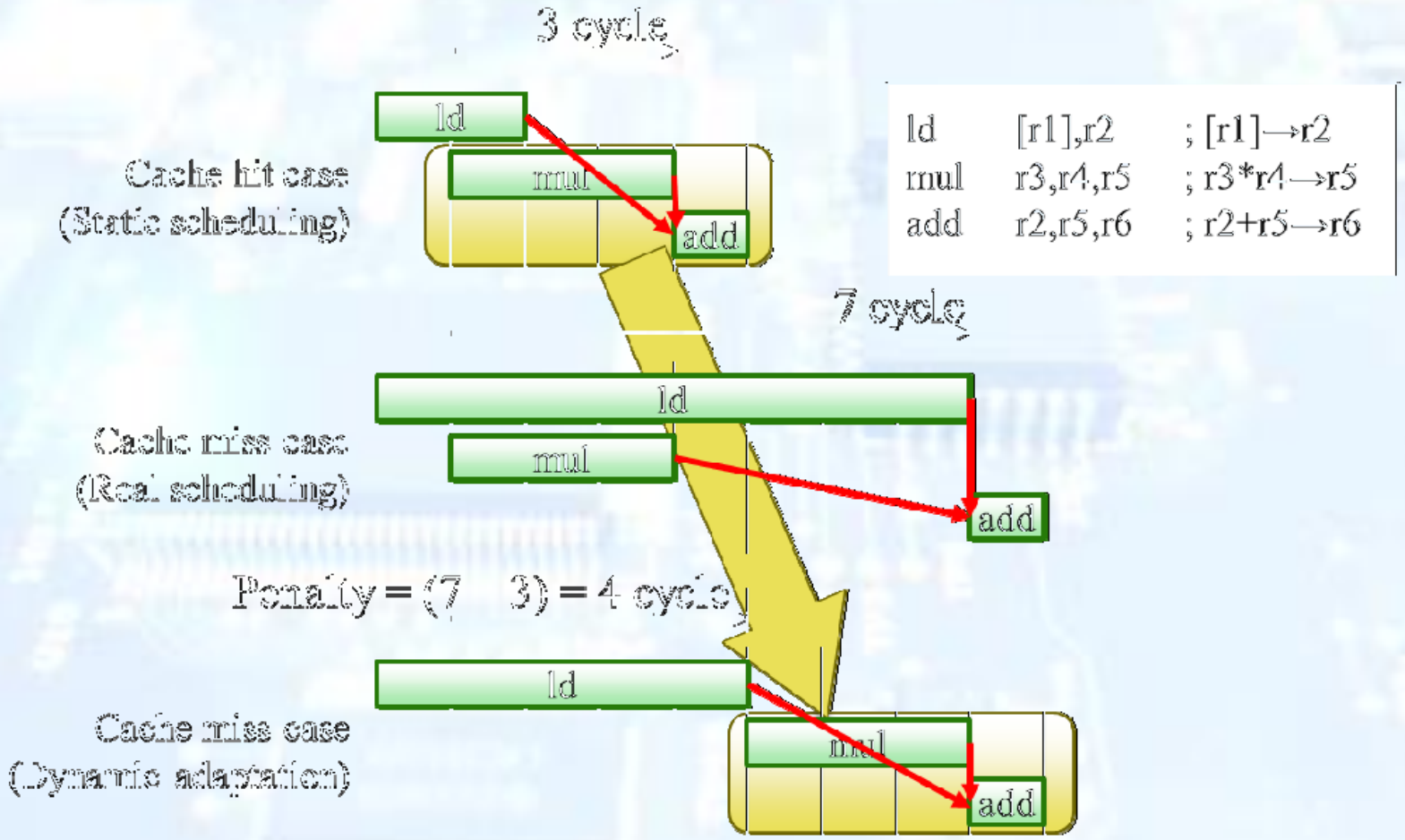
Тест	Модель NM	Emurun (без учета времени)	Temu (время с точностью до такта)
quicksort.cpp	7,4 с	16 с	893 с
primes.cpp	4,3 с	11 с	538 с
svertka.asm	28,6 с	7 с	110 с

Для текстов на asm – статическая компилируемая эмуляция (трансляция на C++). Хост Pentium IV 1,4 (1,7) ГГц

Эмулятор ARM от Fujitsu Lab

- Основан на QEMU
- Добавлена модель конвейера команд и кэша 1 и 2 уровня
- При трансляции линейного участка делается оценка времени выполнения при попадании в кэш и правильном предсказании перехода
- При выполнении, при необходимости, выполняется корректировка.

Статико-динамический подход Fujitsu



Характеристики

- Погрешность оценки времени не более 10%
- Время эмуляции в среднем в 3,37 раза больше по сравнению с исходным QEMU
- Время эмуляции без коррекции в среднем в 1,69 раза больше по сравнению с исходным QEMU

Эмулятор с потактовой точностью на основе QEMU и SystemC

- Статья [6]
- В транслированный код вставляются вспомогательные функции, сообщающие о выборке и записи в память, для взаимодействия с компонентами на SystemC
- 0% погрешность в числе тактов
- 17 минут – загрузка ядра Linux

(хост Core 2 Duo T7300 2ГГц)

В промышленности

http://arm.com/files/pdf/SDD_Fast_Models.pdf

ARM Fast Models

High performance, high fidelity Virtual Platform

Ссылки
(поправить!)

1. Fast Instruction Set Simulation Using LLVM-based Dynamic Translation 2011
2. An Ultra-Fast Instruction Set Simulator 2002
3. Shade: A Fast Instruction-Set Simulator for Execution Profiling 1993
4. Fast Cycle Estimation Methodology for Instruction-Level Emulator 2012, Fujitsu Lab.

Ссылки (2)

5. A Retargetable Framework for Instruction-Set Architecture Simulation 2006
6. A fast cycle-accurate instruction set simulator based on QEMU and SystemC for SoC development 2010

Спасибо за внимание!