ИСП РАН

МЕТОДЫ СТАТИЧЕСКОГО И ДИНАМИЧЕСКОГО АНАЛИЗА ПРОГРАММ ДЛЯ ОБЕСПЕЧЕНИЯ ЭФФЕКТИВНОСТИ И БЕЗОПАСНОСТИ ПРОГРАММНО-АППАРАТНЫХ ПЛАТФОРМ. СОВРЕМЕННОЕ СОСТОЯНИЕ И ПЕРСПЕКТИВЫ

д.ф.м.н. Арутюн Аветисян {arut@ispras.ru}

24 января 2014

Тенденции развития отрасли ИТ

- Сервис-ориентированные центры обработки данных («облачные» инфраструктуры)
  - ✓ инфраструктуры, ориентированные на конкретные классы приложений
- Массовое внедрение мобильных платформ
- Массовое внедрение встроенных систем в физические объекты и их объединение в единую сеть – «Интернет вещей»

Тенденции развития аппаратуры и ПО

ИСП РАН

- Аппаратура:
  - ✓ обновление каждые несколько месяцев, в том числе, многочисленные варианты в рамках одного семейства архитектур
  - ✓ многоуровневый параллелизм, акселераторы, SoC-процессоры, распределенность
  - ✓ переход к открытым стандартам и использованию серийных компонент, доступных на рынке (commodity)
- Программное обеспечение:
  - ✓ Рост размера программных систем (десятки-сотни миллионов строк кода)
  - ✓ Рост популярности динамических языков и платформ на них (JavaScript – Tizen, Python – OpenStack)
  - ✓ Свободное ПО – существенное повышение производительности труда

ИСП РАН

Оптимизация – языки C/C++

Современные компиляторы, в частности GCC, LLVM:

- ✓ поддерживают генерацию кода для различных платформ (x86, ARM, …)
- ✓ содержат миллионы строк кода
- ✓ включают сотни различных оптимизаций (эвристик)
- ✓ обеспечивают приемлемую скорость компиляции

➢ Разработка методов оптимизации

- ✓ ускорение в среднем 1-5% для набора программ (SPEC CPU, тесты заказчика), до 10% для отдельных программ
- ✓ возможность оптимизации сверхбольших программ
- ✓ более 100 патчей включены в основную ветвь разработки GCC и LLVM

➢ Автоматическая настройка компилятора (TACT)

- ✓ на основе генетических алгоритмов
- ✓ ускорение в среднем 10% (6 тестов SPEC), до 30% на отдельных приложениях

Оптимизация – динамические языки

Необходимость оптимизаций во время выполнения (динамические типы и объекты)

- Разработка методов оптимизации для SFX JavaScript
  - ✓ Ускорение на стандартных тестовых пакетах (SunSpider, V8, Kraken, BrowserMark) 4-10% (до 30% на отдельных тестах)
  - ✓ Более 10 патчей включены в основную ветвь SFX
- Исследовательский проект с участием Rice University. Ускорение запуска JavaScript-приложений и затруднение анализа их исходного кода. На первом этапе обеспечено:
  - ✓ Ускорение 4% на тех же тестовых пакетах
  - ✓ Сокрытие исходных текстов JavaScript
  - ✓ ***Будет включена в следующую версию Tizen***

Эффективность – параллелизм

ИСП РАН

Программирование на так называемом «ассемблерном уровне», стандарты MPI, OpenMP, Cuda, OpenACC, OpenCL

- Инструменты, поддерживающие поиск циклов, выгодных для переноса на акселераторы (GPU) и анализ трассы для выявления ситуаций ложного разделения кэша (борьба за кэш между потоками выполнения)
- Разработка библиотеки поддержки вычислений с разреженными матрицами на GPU (совместно с Cambridge University)
  - ✓ Собственный формат матрицы – ускорение 30-40% в среднем и до 2х раз на тестах по сравнению с библиотекой CUSP
  - ✓ Автоматическая настройка (тип матриц, аппаратура)
- Разработка (MPI + Multi-GPU)-версии OpenFOAM
- Совместно с MIT исследования по созданию языковых расширений и компиляторной инфраструктуры для конкретного класса приложений

Безопасность. Статический анализ исходного кода

Одно из важных требований к современному ПО – отсутствие дефектов, трудно обнаруживаемых ситуаций в исходном и/или объектном коде, которые могут приводить к уязвимостям защиты (переполнение буфера, форматная строка и т.п.) и потере стабильности работы программы

Разработан ряд стандартов: *CWE, CWE/SANS Top 25, CERT, OWASP, DISA STIG, MISRA*

Активные исследования и разработки по созданию инструментов статического анализа (без выполнения кода) исходного кода, обеспечивающих автоматическое обнаружение дефектов в больших объемах кода (десятки миллионов строк) ведутся с середины 2000-х гг.

Svace – автоматическое обнаружение дефектов

ИСП РАН

- Находит более 100 различных видов дефектов
  (переполнение буфера, разыменование нулевого указателя, проблемы синхронизации, проблемы работы с памятью и др.)
- Нет ограничений на размер анализируемой программы:
  линейная масштабируемость – полный анализ Андроида (около 5 миллионов строк кода) за 3 часа
- Качество анализа и производительность сопоставимы с лучшими инструментами существующими на рынке:
  Coverity SAVE и **Klockwork Insight**
- В настоящее время поддерживаются языки Си и Си++, поддержка языка Java находится в разработке (окончание 2014)
- Пользовательский интерфейс поддерживает распределенную работу с историей результатов анализа, как из командной строки, так и среды разработки Eclipse, в том числе, через web

Анализ бинарного кода

ИСП РАН

Цель:

- Восстановление алгоритмов, структур данных, протоколов
- Нахождение недокументированных возможностей

Актуальность:

- Используется ПО, имеющееся только в бинарном коде, и/или «недостоверная» среда сборки
- Агрессивная оптимизация может вносить в код дефекты

Средства:

- Статический анализ (трудно применять к программам, снабженным средствами защиты от анализа, например, самомодифицирующийся код)
- Динамический или комбинированный анализ

Динамический анализ бинарного кода – Avalanche

ИСП РАН

- Avalanche – технология автоматического динамического анализа (C/C++, Java), основанная на обходе возможных трасс выполнения и поиске ошибок на этих трассах:
  - ✓ Инструментация кода для получения трасс выполнения в виде набора ограничений (C/C++ - Valgrind, Java – BCEL)
  - ✓ Разрешение ограничений с помощью солвера (используется солвер на основе свободного ПО STP)
  - ✓ Построение новых наборов входных данных
  - ✓ Поиск ошибок, в том числе критических:
    - NPD, деление на ноль найдены в **mono, llvm, libjpeg, swftools, xmllint** и др.;
    - состояния гонки – в **Browser и Mms (Android)**
- Достоинство – генерация входных данных, на которых проявляется дефект
- Недостаток – ресурсоемкость

Trex: среда комбинированного анализа бинарного кода (I)

ИСП РАН

- Базируется на **модели** анализируемого бинарного кода
- Обеспечивает:
  - ✓ сбор и систематический анализ трасс выполнения (объединение трасс, динамический слайсинг и др.)
  - ✓ интеграцию с результатами статического анализа (Ida PRO)
- Позволяет автоматизировать процесс восстановления интересующих аналитика:
  - ✓ алгоритмов
  - ✓ структур входных и выходных данных
  - ✓ протоколов обмена и др.

Trex: среда комбинированного анализа бинарного кода (II)

- Восстановлен алгоритм проверки лицензии SmartPlant License Manager 2010
- Восстановлен алгоритм распаковки кода и формат файла, в котором этот код содержится
- Выявлены механизмы внедрения вредоносного кода в операционную систему (вирус/руткит Trojan.Win32.Tdss.ajfl)

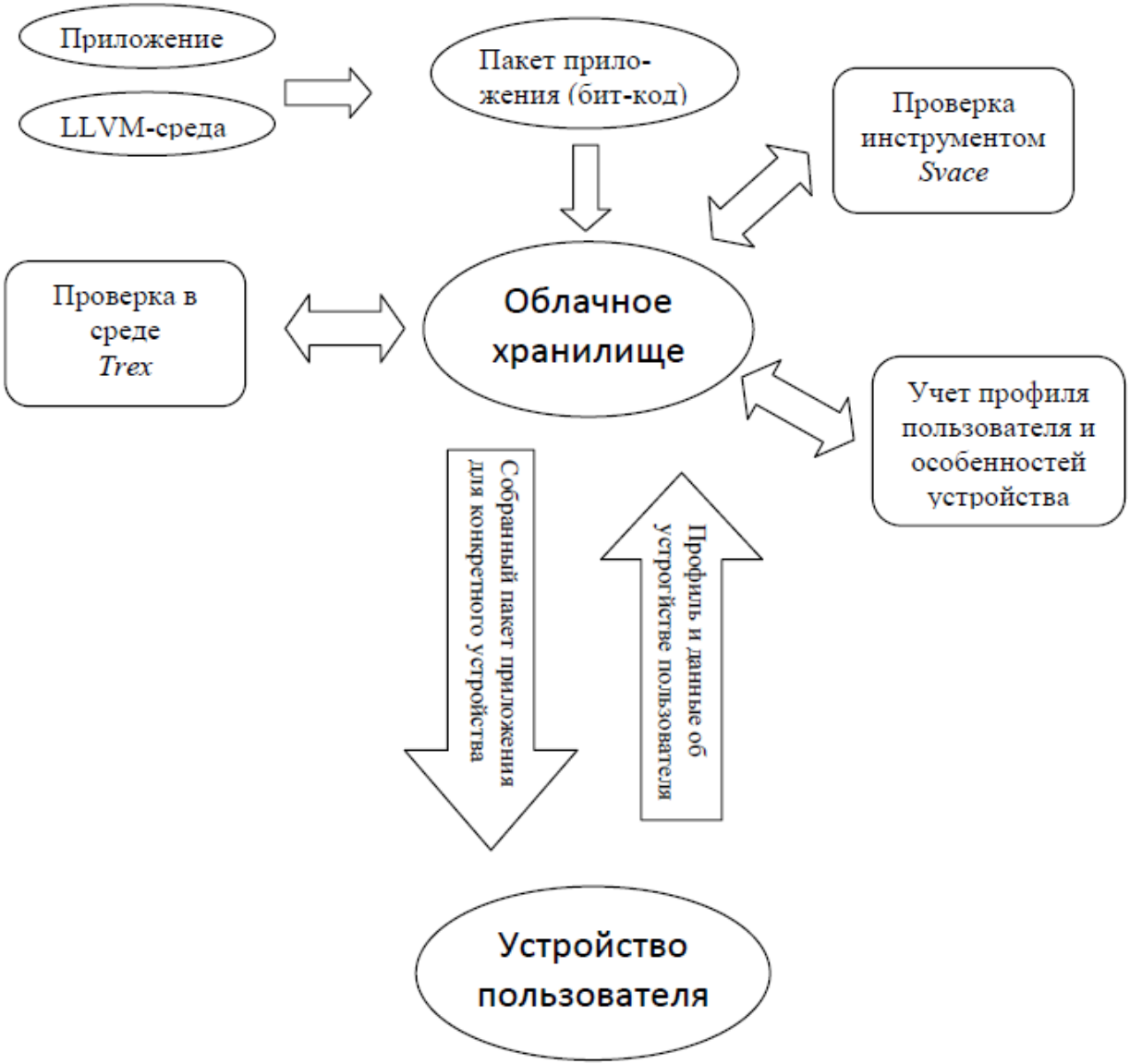
Пример выделения инструкций требуемого алгоритма:

	Трасса, содержащая код анализируемой программы в пользовательском режиме			Число инструкций после обработки
	Размер, МБ	Число шагов	Число инструкций в листинге	
VB v6.0	42	575 392	26 620	356
VB .NET	35	484 248	62 726	143

Переносимость C/C++-приложений с сохранением эффективности

- Двухэтапная компиляция на основе LLVM:
  - ✓ распространение программы в биткоде LLVM
  - ✓ машинно-независимые оптимизации на стороне разработчика
  - ✓ машинно-зависимые оптимизации и оптимизации по профилю:
    в «облаке» (индивидуально для устройства)
    или статически/динамически на устройстве
  - ✓ возможность дополнительного статического анализа в облаке (выявление дефектов, обфускация, …)
- Используется в Tizen App Store

«Облачное хранилище» приложений нового поколения



UniHUB - технологическая платформа программы «Университетский кластер»

- Создана технологическая платформа, реализующая в полном объеме концепцию web-ориентированных научно-производственных центров
- Платформа полностью базируется на свободном ПО («облако» на основе OpenStack)
- Платформа развернута в ЦОД ВЦ им. А.А. Дородницына РАН
- Пользователи могут создавать и разворачивать свои собственные «облачные» сервисы в рамках единой платформы, в том числе создавать web-ориентированные научно-производственные центры в конкретных предметных областях
- Web-лаборатории:
  1) Механика сплошных сред (МСС)
  2) Системное программирование
  3) ГИС