V8 Allocations Profiler

Александра Михайлова, 444
mikhaylova.alexandra.a@gmail.com

научный руководитель: Юрий Семихатский, Google

V8 & Chrome DevTools

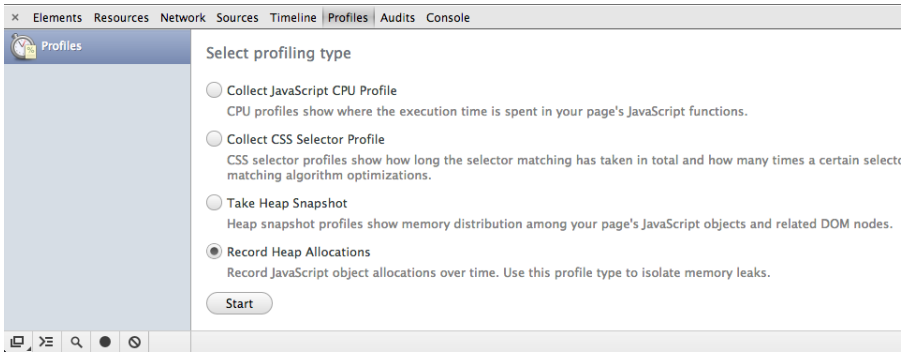
- Developer Tools -- инструменты веб-разработчика в браузере
- https://code.google.com/p/v8/
- JavaScript-движок в Chromium
  - генерация и исполнение машинного кода по JS
  - работа с памятью и сборка мусора

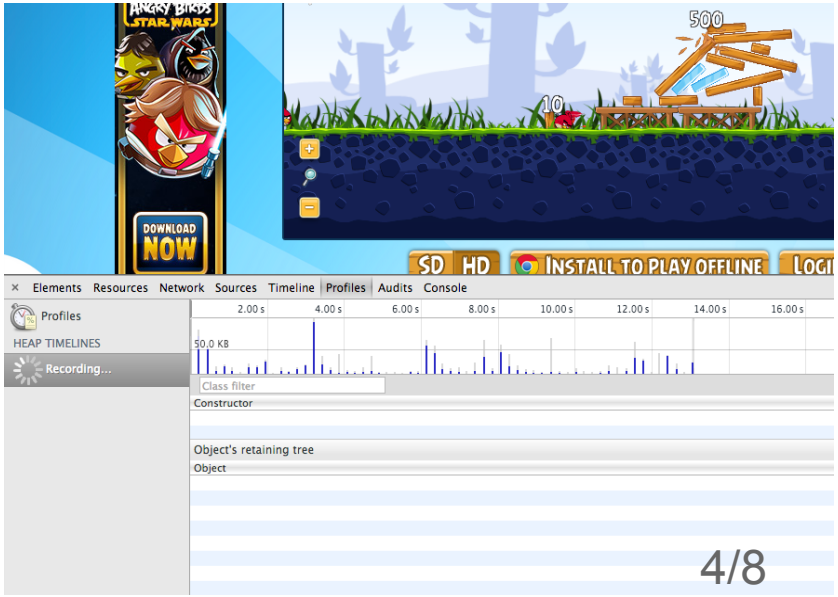
V8: профилирование памяти

- V8 Heap, HeapProfiler
- **JS allocations tracking**
  - тип, размер объекта
  - стек вызова функции при создании объекта
  - график зависимости от времени
  - важно для определения (предсказания) утечек памяти!

V8 до: мониторинг объектов

- Обход достижимых объектов кучи
- Минусы:
  - **Долго: + 50-70%!**
  - Информация неточна





Задачи курсовой работы

- **V8 Allocations Profiler**: “мгновенный” трекинг + больше информации
  - x64, внедрение
- Без падения производительности
  - выбор инструментов измерения
- Сравнить 2 подхода, выбрать лучший:
  - без перегенерации кода при переключении (jmp)
  - с перегенерацией

2 подхода к профилированию

- Без перегенерации кода
  - При загрузке страницы генерируется код
  - Внутри -- проверка, включен ли профайлер (cmp)
  - Если да -- запись, если нет -- jmp
- Перегенерация кода при включении/выключении профайлера
  - Проверка, включен ли профайлер, -- на этапе генерации кода

Результаты-1

- Реализованы оба подхода
- Сравнение -- система Octane
- С перегенерацией кода -- лучше (-0.1% vs -1%)
  - нет “прыжков” (jmp)
- Общее падение производительности:
  - << 1% при выключенном мониторинге
  - ~20-30% при включенном (раньше -- от 50%!)

Результаты-2

- Закоммитили в продукт
- Основа для дальнейших работ:
  - пропущенные объекты => мониторинг на уровне списка инструкций (виртуальный)
  - борьба с оптимизацией: folded allocations

    [if (a) then b else c, но выделяется a+b+c, распределяется потом]
  - предсказание утечек памяти по анализу работы приложения