

ОПТИМИЗАЦИЯ КОДА И ПРОИЗВОДИТЕЛЬНОСТИ JUST-IN-TIME КОМПИЛЯТОРА ДЛЯ МОБИЛЬНОГО JAVASCRIPT

Ёрш А.О., Байданов А.М.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Дик С.К. – канд. физ.-мат. наук, доцент

Для полноценного просмотра веб-страниц на смартфонах требуется высокопроизводительный движок JavaScript, поскольку выполнение JavaScript с мобильным процессором происходит медленно. Таким образом, в мобильных JavaScript-движках используется компилятор JIT, который преобразует код JavaScript в машинный код во время выполнения. Одна из проблем заключается в том, что, поскольку мобильные телефоны страдают от жестких ограничений памяти, компилятору JIT необходимо сохранять небольшой объем памяти, генерируя машинный код небольшого размера. В этой статье описывается генерация и оптимизация кода для мобильного JIT компилятора JavaScript в SquirrelFish Extreme (SFX) Webkit для ARM Thumb2. Главной целью является генерация как можно большего количества 16-битных инструкций и сокращение области данных. Экспериментальные результаты показывают, что мы можем уменьшить размер кода на 29% при снижении производительности на 3,5% по сравнению с ARM-версией SFX.

Экспериментальное оборудование представляет собой плату с процессором Cortex-A8, основанном на архитектуре ARMv7, аналогичную используемой в смартфонах. Эксперименты проводились с автономным SFX. Использовался тест SunSpider, который состоит из 26 программ JavaScript. Для оценки измерялся общий размер сгенерированного JIT-кода и общее время выполнения.

Рисунок 1 показывает размер кода SFX на ARM как 100% (крайний левый столбец) и сравнивает SFX на Thumb2 с каждой оптимизацией в разделе 3, включенной в совокупности (правый столбец).

На рисунке 2 показано общее время работы SunSpider для конфигураций SFX на Thumb2 с SFX на ARM, равным 100%.

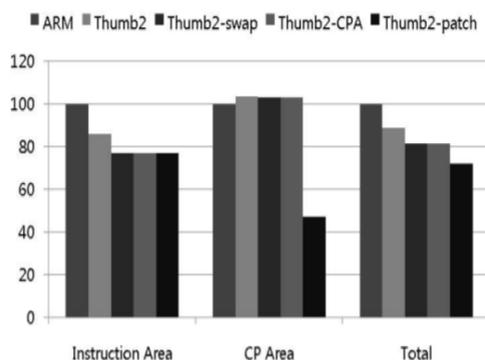


Рисунок 1 – Сравнение размера кода

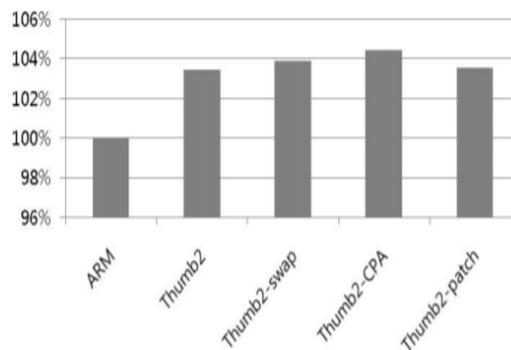


Рисунок 2 – Сравнение времени выполнения

График показывает результат области инструкций и области CP отдельно и их суммарный результат в конце. Для области инструкций, когда генерируются инструкции Thumb2 вместо инструкций ARM, размер кода уменьшается на 14% из-за его 16-битных инструкций. Когда меняются регистры r5 и r8, можно добиться дополнительного уменьшения размера на 9% за счет дополнительной замены 32-битных инструкций 16-битными.

В случае, когда генерируются инструкции Thumb2, время выполнения увеличивается примерно на 3,5%. Поскольку нужно заменить некоторые инструкции ARM несколькими инструкциями Thumb2, цикл должен быть увеличен.

Список использованных источников:

1. A. Gal et. al, Trace-based just-in-time type specialization for dynamic languages. In Proceedings of the 2009 ACM SIGPLAN conference on Programming language design and implementation (PLDI'09), Dublin, Ireland, June 2009.
2. M. Berndt et. al, Context Threading: A Flexible and Efficient Dispatch Technique for Virtual Machine Interpreters, Proceedings of the international symposium on Code generation and optimization, p.15-26, March 20-23, 2005.