Исследование влияния разбиения кода на блоки трансляции на производительность динамической двоичной трансляции

Шляпин Илья Игоревич

Студент

Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия

E-mail: shliapin.ilia@yandex.ru

Научный руководитель *— Батузов Кирилл Андреевич*

Зачастую возникает необходимость запуска программы на аппаратной платформе, отличающейся от той, под которую исходная программа была разработана. Для осуществления этого используются динамические двоичные трансляторы.

При разработке динамического двоичного транслятора возникает множество вопросов, решение которых значительно влияет на эффективность работы транслятора. Одним из таких является организация единиц трансляции.

В качестве единиц трансляции могут служить динамические базовые блоки [1] и состоящие из них линейные (трассы, суперблоки [2]) и нелинейные (деревья) структуры [3]. Различные форматы единиц трансляции поддерживают различные наборы применимых оптимизаций.

Данная работа рассматривает факторы, влияющие на производительность динамического двоичного транслятора, на основе существующего транслятора x86_64 на ARM, использующего в качестве единиц трансляции трассы.

Сложность эффективной трансляции кода x86_64 на другую архитектуру связана в том числе со значительным числом инструкций, генерирующих флаги состояния процессора. При трансляции трассы работа с флагами может быть значительно оптимизирована. В случае перехода из одной трассы в другую пространство для возможных оптимизаций существенно уже, поскольку зачастую заранее нет возможности предугадать, будут ли флаги использованы, и каким образом. Помимо этого, переходы между различными трассами сами по себе менее эффективны, чем переходы между блоками в рамках одной единицы трансляции. Отсюда основными целями алгоритма набора базовых блоков в трассы являются сокращение числа переходов между трассами и накладных расходов на такие переходы. Сокращения накладных расходов можно добиться, объединяя блоки, использующие сгенерированные ранее флаги, в одну трассу

с блоками, генерирующими такие флаги. Сокращения числа переходов между трассами можно добиться, объединяя в одну трассу блоки, переход между которыми наиболее вероятен.

В рамках данной работы алгоритм набора трасс дополнялся различными эвристиками по выбору ветви условного перехода и определения свойств базовых блоков. Для исследования был выбран набор тестов, проверяющих эффективность как оттранслированного кода, так и процесса инициализации трансляции. К тестам первой категории относятся 400.perlbench, 403.gcc и 456.hmmer из SPECINT2006 [4]. Тесты второй категории заключаются в запуске таких программ, как firefox и GIMP. На представленных тестах была собрана статистика влияния проверяемых эвристик на эффективность работы транслятора.

Среди исследуемых эвристик можно выделить: связанные с особой обработкой условных переходов по обратной дуге; определяющие вероятность перехода на блок в зависимости от его размера и содержания. Первая группа исходит из того, что условный переход по обратной дуге чаще всего наблюдается в контексте цикла. Вторая группа исходит из стремления держать переходы, находящиеся в одной цепочке, по возможности ближе друг к другу. На основе полученных экспериментальных данных эвристики второй категории показали себя эффективными, т.е. заметно снижающими как общее время работы динамического двоичного транслятора, так и число генерируемых в процессе трасс. В то же время эвристики первой категории оказались неэффективными, зачастую заметно увеличивая время работы транслятора.

Литература

1. Aho, Alfred V. Compilers: principles, techniques and tools (for Anna University), 2/e / Alfred V Aho. — Pearson Education India, 2003
2. The superblock: An effective technique for VLIW and superscalar compilation / Wen- Mei W Hwu, Scott A Mahlke, William Y Chen et al. // Instruction-Level Parallelism. — Springer, 1993. — Pp. 229–248
3. Dynamic binary translation and optimization / Kemal Ebcioglu, Erik Altman, Michael Gschwind, Sumedh Sathaye // IEEE Transactions on Computers. — 2001. — Vol. 50, no. 6. — Pp. 529–548
4. Официальный сайт SPEC CPU 2006: `https://spec.org/cpu2006/`