УДК 004.4'422

Нелокальная оптимизация арифметических выражений

Д.А. Земляков

ЗАО «МЦСТ»

Московский физико-технический институт (государственный университет)

Dmitry.A.Zemliakov@mcst.ru

Одной из причин неполного использования производительности вычислительных систем является наличие в программах избыточных вычислений. Если независимо транслировать каждую конструкцию высокоуровневого языка программирования в машинный код, то это может приводить к существенным накладным расходам времени выполнения программы.

Оптимизации, выполняемые компилятором, должны сохранять семантику исходной программы. За исключением очень редких случаев, если программист выбрал и реализовал конкретный алгоритм, компилятор не в состоянии достаточно хорошо разобраться в программе, чтобы заменить его совершенно иным более эффективным алгоритмом.

В типичной программе имеется масса различных избыточных операций. Иногда избыточность проявляется на уровне исходного текста программы. Часто избыточность оказывается побочным действием написания программ на языке программирования высокого уровня. Еще одним возможным источником избыточных операций может являться сам компилятор.

Проблема наличия избыточностей приобретает особенную актуальность применительно к архитектурам, рассчитанным на достижение высших показателей производительности, к которым относится архитектура отечественных микропроцессоров серии «Эльбрус», предназначенных для разработки крупномасштабных иформационно-вычислительных систем стратегического значения.

Поэтому актуальной задачей является реализация новой оптимизации по удалению общих подвыражений промежуточного представления программы в промышленном оптимизирующем компиляторе для архитектуры «Эльбрус». Алгоритм данной оптимизации, используя основные структуры данных оптимизирующего компилятора, такие как граф потока управления и Def-Use граф, а также отношение доминирования линейных участков, находит общие арифметические подвыражения.

Определим избыточные вычисления, как второе и последующие выполнения одной и той же арифметической операции над одними и теми же переменными, значения

которых не успевают измениться между рассматриваемыми вычислениями. Классический алгоритм удаления общих подвыражений (CSE) не использует такое свойство ряда арифметических операций как ассоциативность.

Рассмотрим простейший пример, демонстрирующий контекст, содержащий избыточное повторное сложение переменных *a* и *b*, недоступное для удаления в результате применения оптимизации CSE:

$$x = a + b$$
$$t = a + c$$
$$y = t + b$$

Ассоциативные избыточные операции образуют контекст в промежуточном представлении для реализации новой оптимизации по удалению общих подвыражений. Реализация оптимизации включает в себя следующие основные этапы:

- *Анализ необходимости применения оптимизации* – проводится последовательный обход всех операций исходного промежуточного представления, выполняются проверки на наличие подходящих арифметических операций и на наличие контекста для оптимизации. На основе результатов анализа принимается решение о продолжении работы оптимизации.
- *Анализ возможности и эффективности применения оптимизации* – проводится проверка применения оптимизации и ее эффективности. Последовательно выбираются пары наиболее используемых переменных; моделируется создание новой арифметической операции и замена всех остальных вхождений пары данных переменных на одну новую переменную. Итогом данного этапа является решение о возможности, корректности и эффективности применения оптимизации. В случае положительного итога формируется модель, в соответствии с которой в дальнейшем проводится преобразование операций в процедуре.
- *Применение оптимизации* в случае положительного результата предыдущих этапов. Удаляются все избыточные операции, и производится корректировка основных структур оптимизирующего компилятора.

Были проведены замеры эффективности применения оптимизации на пакетах тестов производительности SPEC CPU95. Замеры производились на восьмиядерных машинах архитектуры «Эльбрус» с тактовой частотой 480 MHz под управлением ОС Linux 2.6.33. Прирост производительности составил до 4%.

Литература:

1. *Альфред Ахо, Моника Лам, Рави Сети, Джеффри Ульман* Компиляторы: принципы, технологии и инструментарий. – 2-е издание. – М.: «Вильямс», 2008. – 1184 с.

2. *Steven S. Muchnick* Advanced Compiler Design Implementation. – 5-е издание. – San Francisco: Morgan Kaufmann Publishers, 1997. – 856 с.
3. *Ким А.К., Перекатов В.И., Ермаков С.Г.* Микропроцессоры и вычислительные комплексы семейства «Эльбрус». – СПб.: Питер, 2013. – 272 с.
4. Standart Perfomance Evaluation Corporation. The Spec Benchmark Suites. CPUintensive benchmark suite. [Electronic resource]. – 1995-2000. http://www.spec.org/cpu