

Московский физико-технический институт (государственный университет)
Факультет радиотехники и кибернетики
Кафедра информатики и вычислительной техники

Выпускная квалификационная работа

Нелокальная оптимизация арифметических выражений

Выполнил: студент 918 гр. Земляков Дмитрий
Научный руководитель: к.т.н. Ермолицкий А.В.

Проблематика

- Наличие в программах избыточных арифметических вычислений является одной из причин неполного использования производительности вычислительных систем
- Классический вариант оптимизации по сбору общих подвыражений (Common Subexpression Elimination, CSE) не использует свойство ассоциативности арифметических операций

Простейший пример оптимизации по сбору общих подвыражений (CSE)

$$c = a + b$$

...

$$d = a + b$$



$$x = a + b$$

$$c = x$$

...

$$d = x$$

- Вычисление общего подвыражения $(a+b)$ выполняется только один раз
- В остальных случаях используется результат первого вычисления

Простейший пример невозможности применения оптимизации CSE

$$c = a + b$$

...

$$d = a + e + b$$

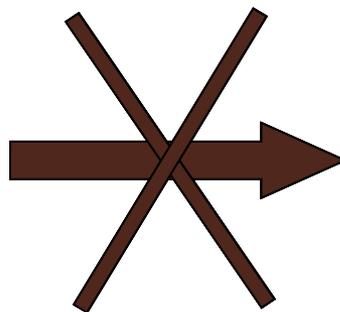
Избыточное сложение переменных a и b при вычислении d

$$c = a + b$$

...

$$x = a + e$$

$$d = x + b$$



Нет контекста для классического варианта оптимизации CSE

Постановка задачи

1. Реализовать новую оптимизацию, устраняющую избыточные арифметические вычисления в рамках целой процедуры, используя свойство ассоциативности операций
2. Внедрить оптимизацию в промышленный оптимизирующий компилятор для архитектуры “Эльбрус”
3. Провести замеры эффективности оптимизации на пакетах тестов производительности SPEC CPU

Реализация оптимизации для решения поставленной задачи

Реализованная оптимизация Common Arithmetic Subexpression Elimination (CASE) состоит из 3-х основных этапов:

- Анализ необходимости применения
- Анализ возможности и эффективности применения
- Применение

Анализ необходимости устранения избыточных арифметических вычислений



Пример работы анализа

$c = a + b$
 $d = a + i + b$
 $e = a + k + i + c + b + d$
 $a = tmp$
 $f = a + b$

$c = a + b$
 $x1 = a + i$
 $d = x1 + b$
 $x2 = a + k$
 $x3 = x2 + i$
 $x4 = x3 + c$
 $x5 = x4 + b$
 $e = x5 + d$
 $a = tmp$
 $f = a + b$

$c \mid a, b$

$d \mid a, b, i$

$e \mid a, b, i, c, d, k$

$a(3) \mid b(3), i(2)$

$b(3) \mid a(3), i(2)$

$i(2) \mid a(2), b(2)$

Анализ возможности и эффективности устранения избыточных вычислений



Пример работы анализа

$a(3) \mid b(3), i(2)$
 $b(3) \mid a(3), i(2)$
 $i(2) \mid a(2), b(2)$

$c \mid a, b$
 $d \mid a, b, i$
 $e \mid a, b, i, c, d, k$

$c = a + b$
 $d = c + i$
 $x1 = d + d$
 $x2 = x1 + c$
 $e = x2 + k$

$c \mid a, b$
 $d \mid c, i$
 $e \mid c, i, c, d, k$

$c \mid a, b$
 $d \mid c, i$
 $e \mid d, c, d, k$

$5 < 8$

Применение новой оптимизации по устранению избыточных вычислений

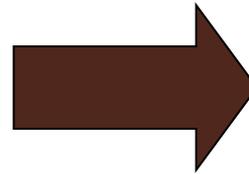


Пример устранения избыточных арифметических вычислений в процедуре

$$c = a + b$$

$$x1 = a + i$$
$$d = x1 + b$$

$$x2 = a + k$$
$$x3 = x2 + i$$
$$x4 = x3 + c$$
$$x5 = x4 + b$$
$$e = x5 + d$$

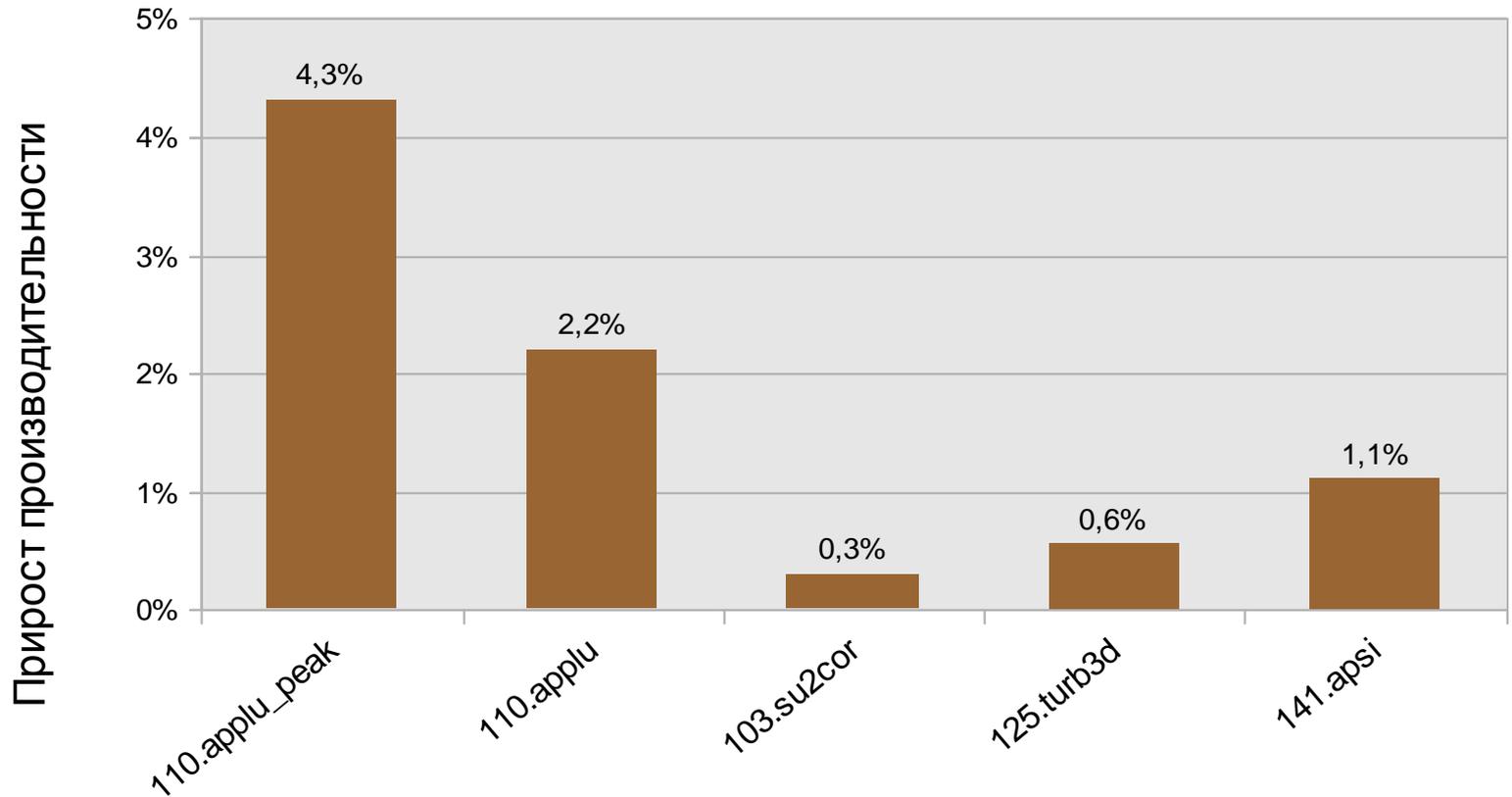


$$c = a + b$$

$$d = c + i$$

$$x1 = d + d$$
$$x2 = x1 + c$$
$$e = x2 + k$$

Экспериментальные результаты



Тесты из пакета SPEC CPU95

Результаты

- Реализована и интегрирована в оптимизирующий компилятор для архитектуры “Эльбрус” новая оптимизация по сбору общих арифметических подвыражений – CASE
- Оптимизация прошла проверку на внутренних пакетах тестирования, состоящих из отдельных задач пакетов тестирования SPEC и нескольких тысяч задач генератора случайных тестов
- Проведены замеры эффективности оптимизации на пакетах тестирования SPEC CPU95, показавшие прирост производительности до 4%