

**Московский физико-технический институт
(государственный университет)
Физтех-школа радиотехники и компьютерных технологий
Кафедра информатики и вычислительной техники**

**Выпускная квалификационная работа
(бакалаврский диплом)**

Оптимизация раннего выхода из функции путем постановки вызовов функции под условие

**Выполнил: Лозновенко В. И., 903 гр.
Научный руководитель: к.т.н., Ермолицкий А. В.**

Конструкция раннего выхода

Ранний выход – конструкция при которой в начале функции стоит условие, которое может вызвать выход из функции.

```
1 bool foo( obj_t * ptr) {  
2     if( ptr == NULL )  
3         return false;  
4     ...  
5     // Some code  
6 }
```

Пример раннего выхода

Актуальность проблемы

- Функции с ранним выходом встречаются повсеместно.
- Существуют задачи, в которых ранний выход является часто исполняемым.
- Затраты на вызов функции являются высокими.

Пример функций с ранним выходом из задачи SPEC CPU 2017 511.povray:

Название функции	Количество вызовов	Вероятность раннего выхода
Destroy_Bbox_Tree	95586	25%
MessageReceiver	10963301	25%
Locate_Filename	53814096	25%
Copy_Textures	14319	29%

Актуальность проблемы

Приведены данные о том как часто в функциях встречается ранний выход.
Значения взяты для некоторых задач пакетов SPEC CPU 2006, 2017 :

Задача	Всего функций	Функций с ранним выходом	Доля функций с ранним выходом
473.astar	50	3	6%
511.povray	1287	117	9%
526.blender	33088	1604	5%

Цель работы

Исследование и реализация оптимизации выноса раннего выхода из функции в языковом оптимизирующем компиляторе Iss для архитектуры Эльбрус.

Задачи

- 1) Оценить ускорение на задаче 511.povgaу, на которой изначально был замечен горячий ранний выход.
- 2) Реализовать оптимизацию 'вынос раннего выхода из функции' в языковом оптимизирующем компиляторе Iss для архитектуры Эльбрус.
- 3) Отладить оптимизацию на задачах SPEC CPU 95, 2000, 2006, 2017.
- 4) Произвести измерение влияния внесённых в компилятор изменений на время исполнения задач SPEC CPU 2006, 2017.

Применимость оптимизации

Оценка ускорения на задаче 511.povray

- Применение оптимизации позволяет экономить до 16 тактов на 1 вызов.
- На задаче 511.povray всего вызовов функций с ранним выходом 341 млн, 6 млн из которых не могут быть оптимизированы из-за различных ограничений.
- Всего на исполнение задачи затрачено 30 млрд тактов.
- В итоге оптимизация может сэкономить 4,6%.

Алгоритм оптимизации

Реализованный алгоритм оптимизации состоит из следующих этапов:

- 1) Поиск функций с ранним выходом
- 2) Поиск вызовов функций с ранним выходом
- 3) Вынос раннего выхода в местах вызова функций
- 4) Удаление раннего выхода из функций с ранним выходом

Ограничение применимости:

- Рекурсивный вызов
- Вызов по указателю

Алгоритм оптимизации

Поиск функций с ранним выходом

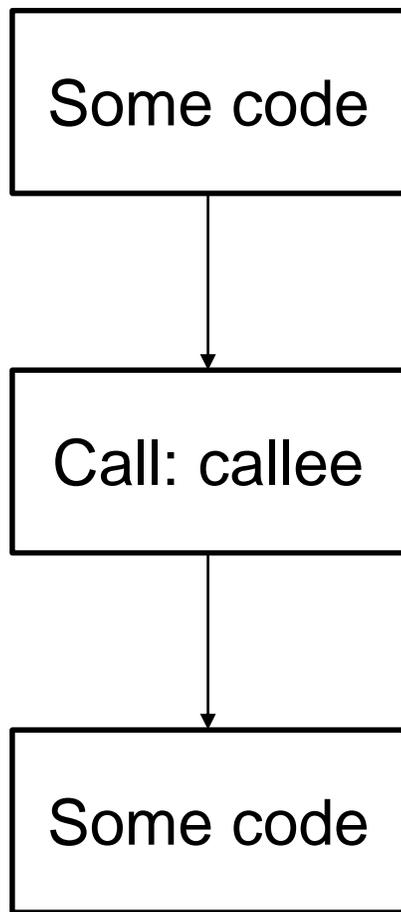
Условия того, что функция имеет ранний выход:

- 1) Первый узел после стартового – if-узел, одно из исходящих рёбер которого ведёт в return-узел.
- 2) В if-узле нет операций обращения к памяти и вызовов функций.
- 3) Количество операций в if-узле $\leq N$
Количество операций во всей функции $\geq M$
Параметры M и N подбираются эмпирически.

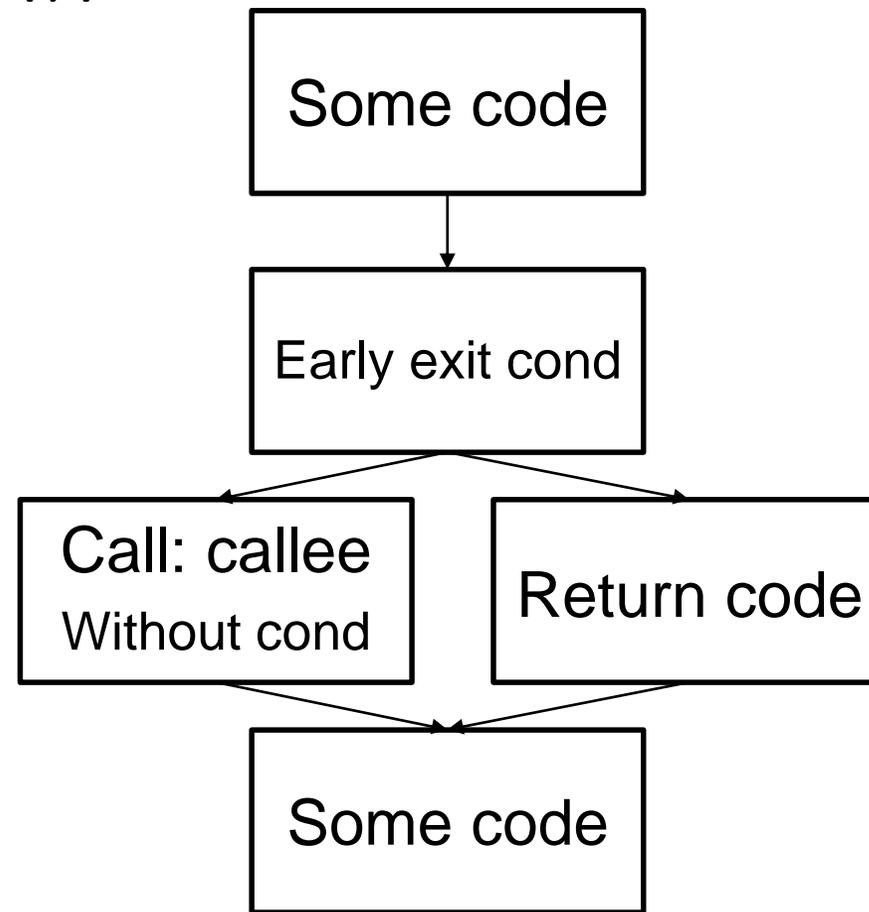
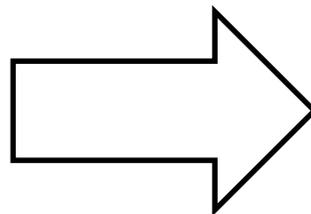
```
int foo( int num )
{
    if (num < 0)
    {
        return 0;
    }
    while ( num < 1000)
    {
        num++;
    }
    //...
    // Big code
    //...
    return num;
}
```

Алгоритм оптимизации

Пример выноса раннего выхода в местах
вызова функций



Граф потока управления до применения
оптимизации

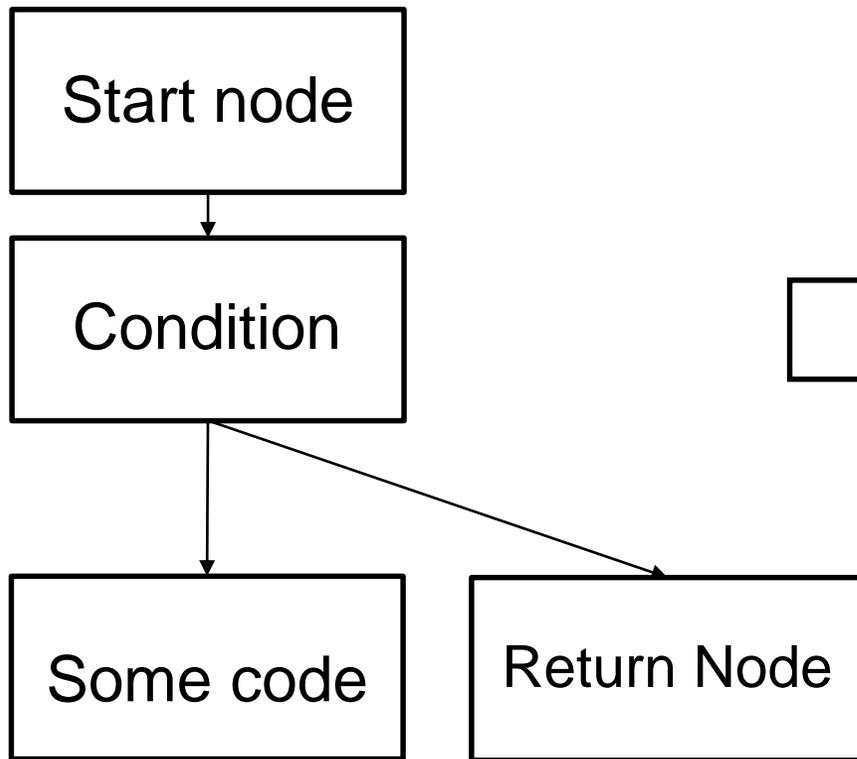


Граф потока управления после
применения оптимизации

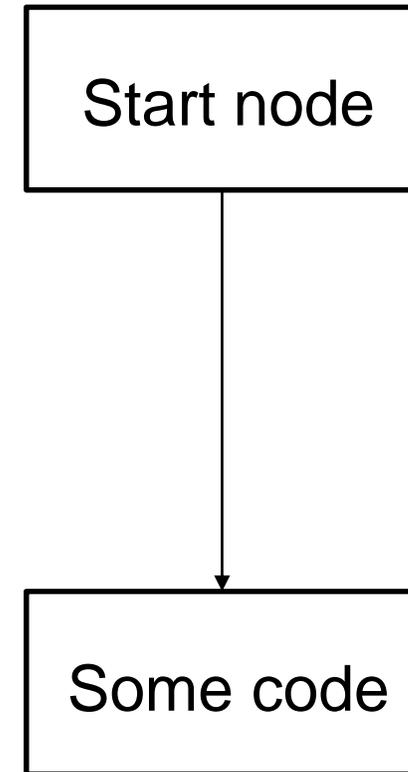
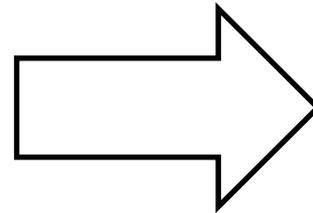
Алгоритм оптимизации

Пример удаления раннего выхода из вызываемой функции

Удаление раннего выхода возможно только при условии, что оптимизация была применена ко всем вызовам функции.



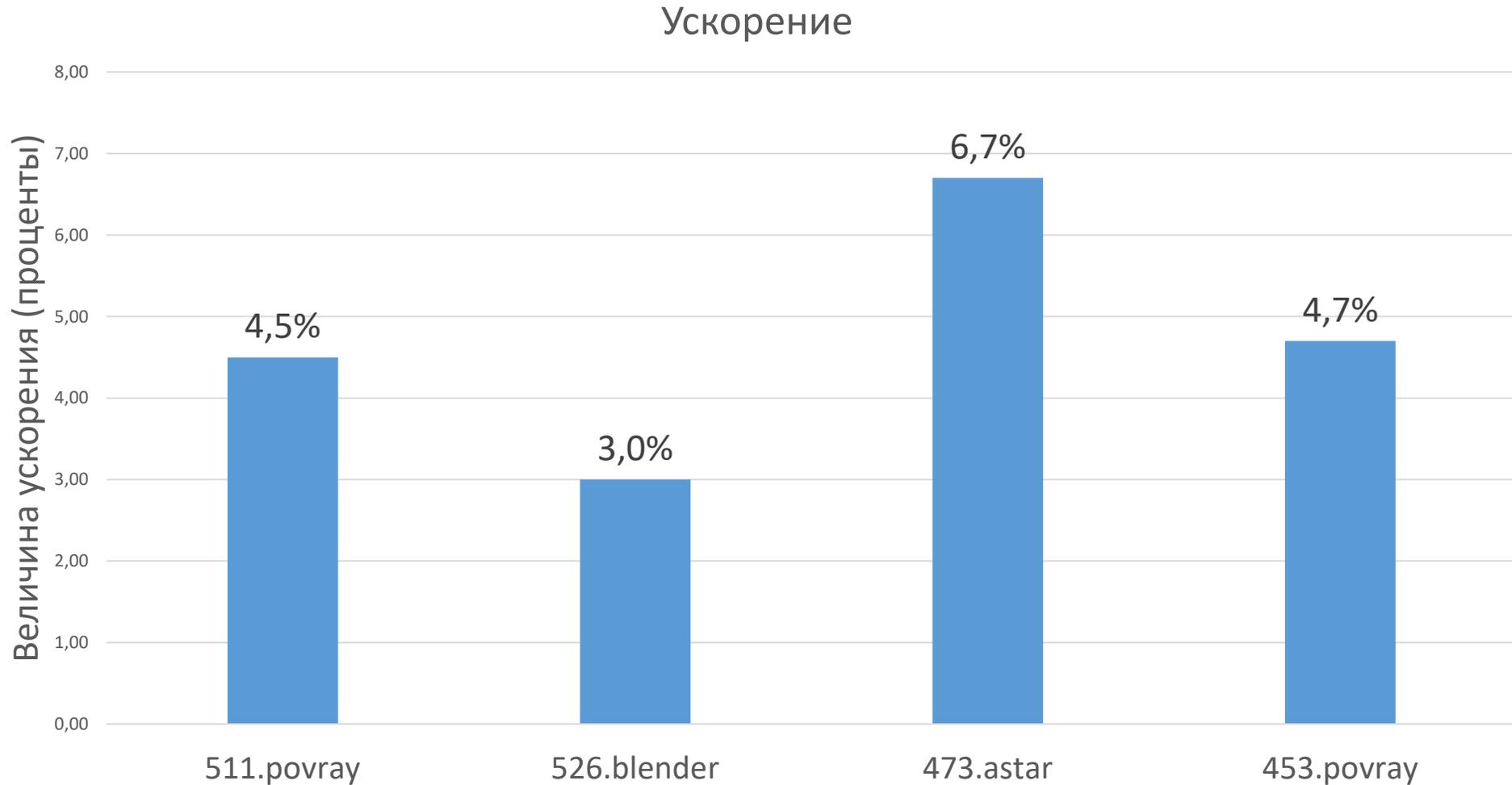
Граф потока управления до применения оптимизации



Граф потока управления после применения оптимизации

Применение оптимизации

Измерения проводились на на компьютере с процессором «Эльбрус-8СВ»



Результаты

- 1) Теоретически оценена применимость на задаче 511.povray.
- 2) Реализована оптимизация 'вынос раннего выхода из функции' в языковом оптимизирующем компиляторе Iss для архитектуры Эльбрус.
- 3) Оптимизация отлажена на задачах SPEC CPU 95, 2000, 2006, 2017.
- 4) Проведено исследование влияния предложенной оптимизации на скорость исполнения задач пакетов SPEC CPU 2006, 2017.

Ее применение позволило получить ускорение задач:

- 511.povray на 4,5%
- 473.astar на 6,7%
- 526.blender на 3,0%
- 453.povray на 4,7%

Время исполнения остальных задач не изменилось.

Замер накладных расходов

Проведён замер накладных расходов на пакете SPEC CPU 2017:

- Относительный прирост времени компиляции 0,06%
- Относительный прирост размера бинарных файлов 0,6%