

Разработка алгоритма оптимизации синтеза фрагментированных программ и его реализация для системы программирования LuNA

Софронов Иван Викторович,
НГУ ФИТ гр.15221

Руководитель
Малышкин Виктор Эммануилович,
д.т.н., проф., ИВМиМГ СО РАН

Проблема создания параллельных программ (ПП)

Создание ПП требует:

- специальной подготовки;
- особого образа мышления;
- высокой квалификации программистов.

Средства создания ПП

- MPI
- OpenMP

Повышение уровня программирования

Ведутся работы по созданию инструмента, позволяющего автоматизировать процесс синтеза параллельной программы.

В таких инструментах:

- алгоритмы записываются на языках более высокого уровня;
- для определенных классов вычислителей создается исполнительная система (ИС);
- компилятор транслирует программу в конкретную ИС.

Шаги компиляции

1. Лексический анализ
2. Синтаксический анализ
3. Построение дерева программы и семантический анализ
4. Преобразования дерева программы
5. Генерация промежуточного кода

Шаги компиляции

1. Лексический анализ
2. Синтаксический анализ
3. Построение дерева программы и семантический анализ
4. Преобразования дерева программы
5. Генерация промежуточного кода

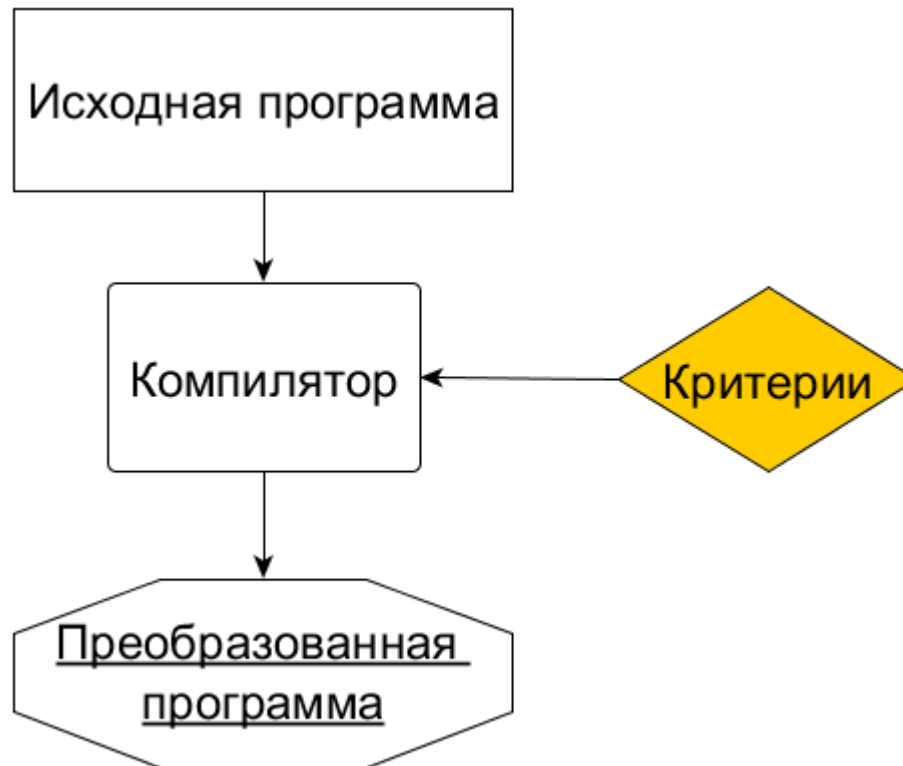
Оптимизации при компиляции

- Компилятор содержит множество различных оптимизирующих преобразований
- Различные комбинации преобразований формируют пространство оптимизаций
- Работая с этим пространством, компилятор способен синтезировать различные по свойствам программы

Проблема автоматического синтеза

- Результирующая программа может быть как *хорошей*, так и *плохой*
- Неизвестно какие свойства в программе нужно оптимизировать
- Заранее нельзя понять какая программа может получиться
- Не хватает информации о том, как оптимизации работают и как они сочетаются

Проблема организации работы компилятора



Существующие подходы

Возможные алгоритмы выбора оптимизации

- Генетический алгоритм
- Алгоритм случайного поиска
- Жадный конструктивный алгоритм
- Нетерпеливые алгоритмы
- На вычислительных моделях
- ...

Постановка задачи

Разработка оптимизирующего модуля, который будет осуществлять интеллектуальную оптимизацию программы с использованием вычислительных моделей в системе фрагментированного программирования LuNA.

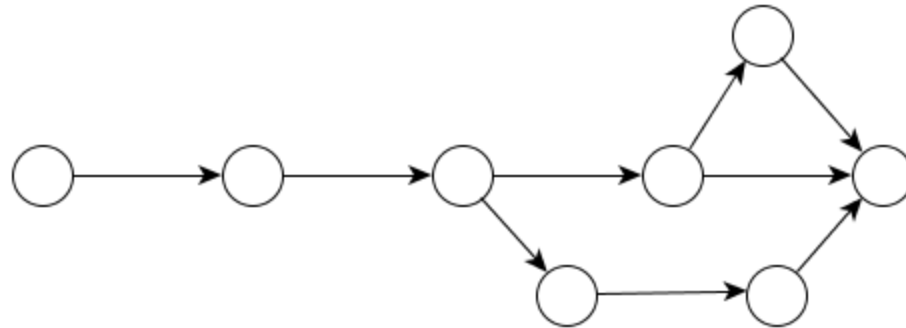
- снимает с программиста необходимость знания пространства оптимизаций
- предлагает ему набор критериев, которые можно применить для оптимизации программы

Постановка задачи

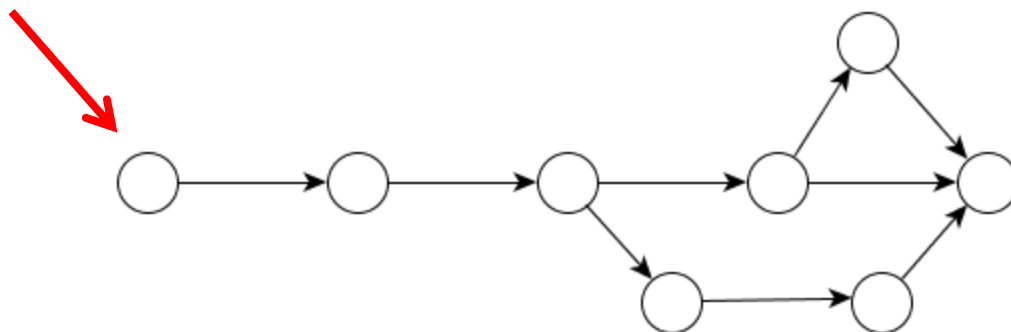
Требования к модулю:

- возможность добавления новых моделей и критериев;
- редактирования существующих моделей и критериев;
- независимость от конкретных компиляторов;
- результат должен быть удовлетворительным в сравнении с ручным заданием применяемых оптимизаций.

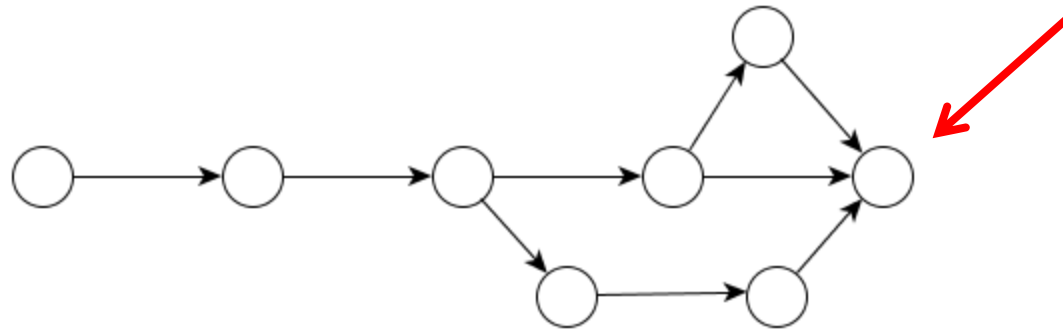
Работа оптимизирующего компилятора



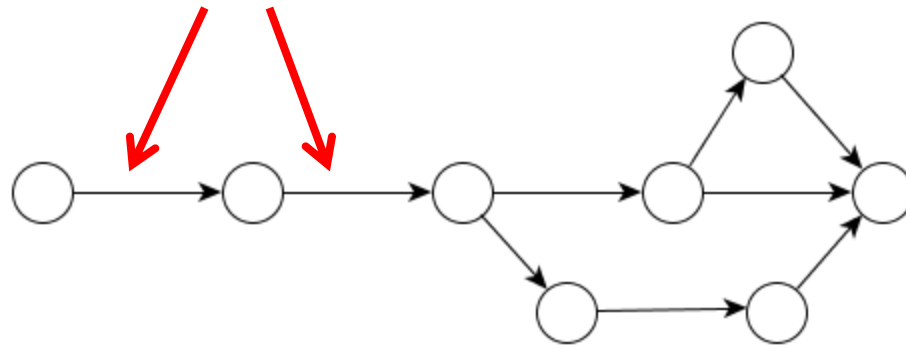
Работа оптимизирующего компилятора



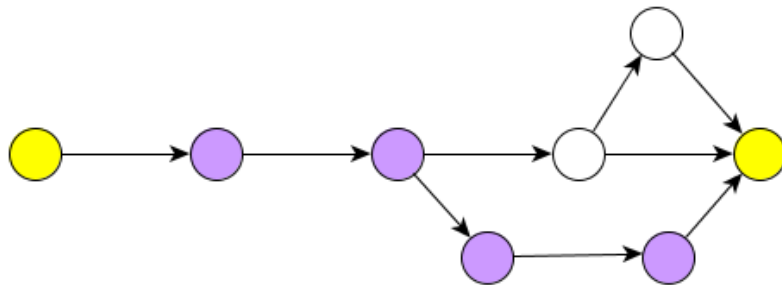
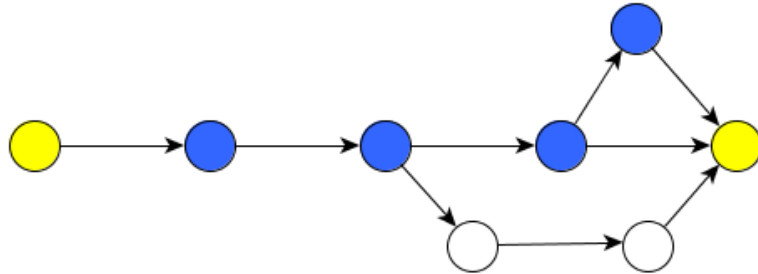
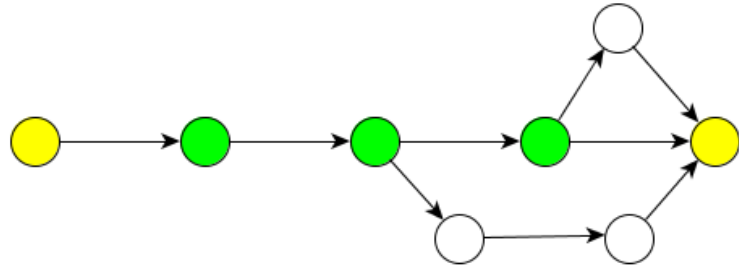
Работа оптимизирующего компилятора



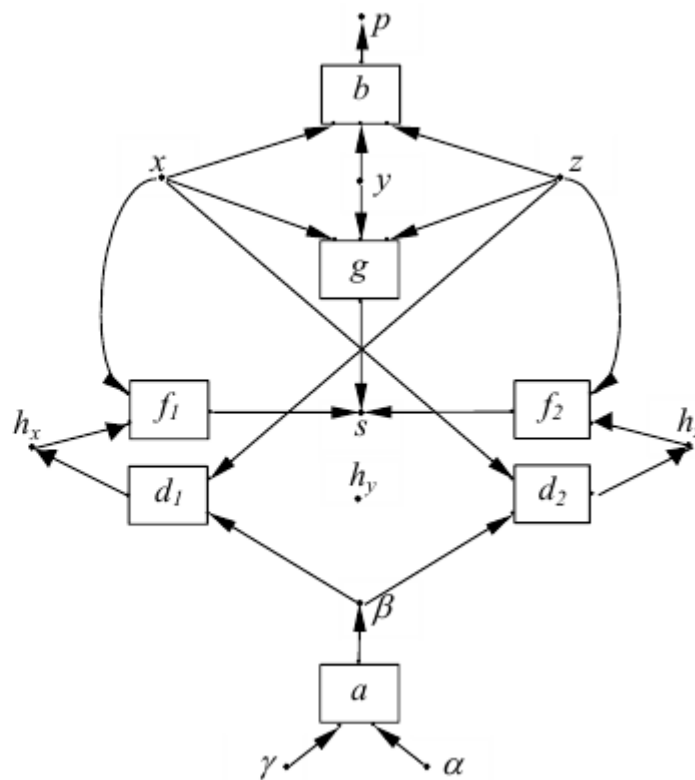
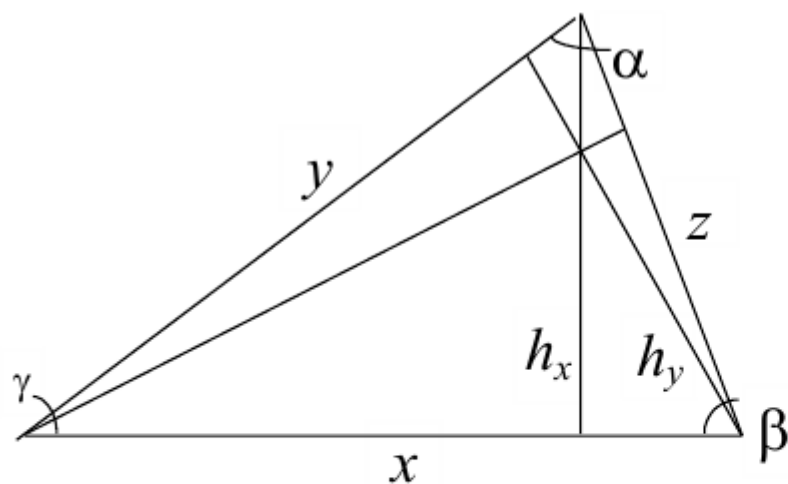
Работа оптимизирующего компилятора



Работа оптимизирующего компилятора



Простые вычислительные модели (ПВМ)



ПВМ оптимизирующего компилятора

- Множество переменных модели содержит результаты преобразования программы в различные моменты процесса применения оптимизаций
- Множество операционных символов содержит операции применения к программе некоторых оптимизаций

(V, W) -план

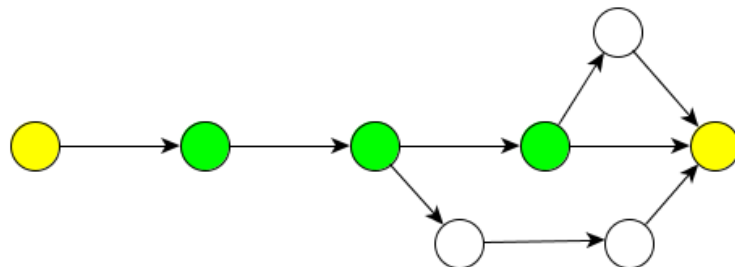
Вычисления на ПВМ представлены в виде (V, W) -плана

(V, W) -план – это терм, показывающий как вычислить из некоторого начального набора величин - V , некоторые конечные величины W

(V, W) -план

Вычисления на ПВМ представлены в виде (V, W) -плана

(V, W) -план – это терм, показывающий как вычислить из некоторого начального набора величин - V , некоторые конечные величины W



Описание алгоритма

Основа :

алгоритм планирования на простых вычислительных моделях

Работает с

множеством модулей компилятора, которое представляется как вычислительная модель.

Результат :

(V, W) -план содержащий список оптимизаций, применение которых позволит получить требуемую программу.

Проверка работы алгоритма

- Создание простой вычислительной модели, включающей существующие оптимизирующие модули компилятора системы LuNA;
- Реализация алгоритма синтеза оптимального (V,W) -плана;
- Оптимизирующими критериями будут параллелизм и накладные расходы;
- При решении одной и той же задачи, в зависимости от применяемых критериев, синтезируются разные программы.

Заключение

На текущий момент:

- Осуществлено знакомство с системой LuNA и модулями компиляции;
- Реализован алгоритм планирования на ПВМ;
- Сформулированы первые критерии оптимизации;
- Предложен вид вычислительной модели пространства оптимизаций.

В дальнейшем планируется:

- Довести алгоритм планирования до алгоритма синтеза оптимального плана на основе выбранных критериев;
- Проверить работу алгоритма на модельной задаче.

Спасибо за внимание!