



Распределение данных программ для многобанковых гетерогенных архитектур памяти

Д. П. Малютин, 646 группа

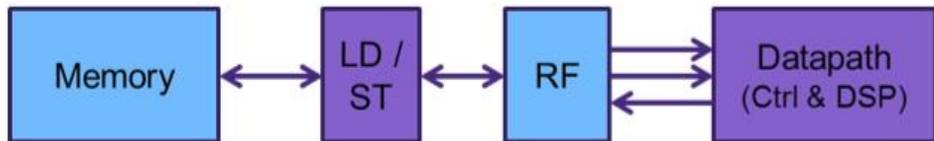
Научный руководитель: ст. преп. А. Н. Терехов

Консультант: тех. руководитель, Huawei, С. И. Якушкин

Рецензент: инженер-программист, Д. Ю. Антрушин

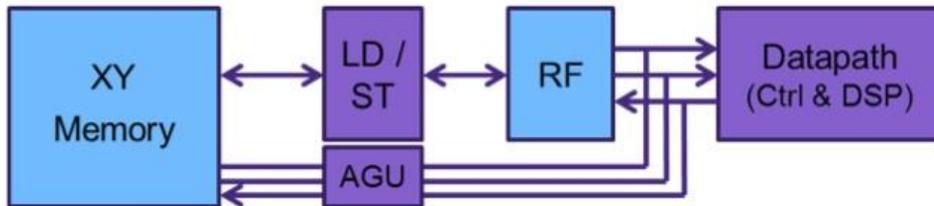


- Многие современные вычислительные системы используют высокоспециализированные гетерогенные архитектуры памяти
- Одним из распространенных решений является использование нескольких физически независимых банков памяти
- Разнообразии устройств, задач и областей применения процессоров приводит к необходимости быстро и эффективно адаптировать существующие приложения под новые конфигурации и структуру памяти



1/3 MAC/cycle

LD, LD, MAC,
LD, LD, MAC



1 MAC/cycle

XYMAC, XYMAC



- Правильное распределение данных может ускорить приложение более чем в 17 раз



- DesignWare ARC EM



Создание инструментов, позволяющих производить анализ использования данных на уровне всей программы, написанной на языках C и C++, с целью автоматического распределения статических данных для гетерогенной многобанковой архитектуры памяти.



Конфигурация памяти – множество доступных памятей (банков памяти) $B = \{b_1, \dots, b_n\}$, каждый из которых имеет определённый размер и стоимость операций загрузки/выгрузки данных из него.



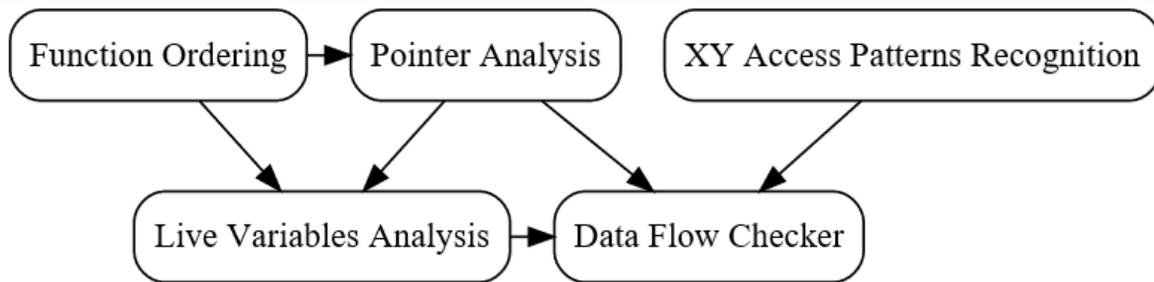
Граф конфликтов

- Вершины – переменные
- Рёбра – потенциальный параллельный доступ к переменным (конфликт)
- Веса – примерное количество раз данный конфликт реализуется

Задача: раскрасить граф так, чтобы сумма весов рёбер между вершинами одного цвета была минимальной.

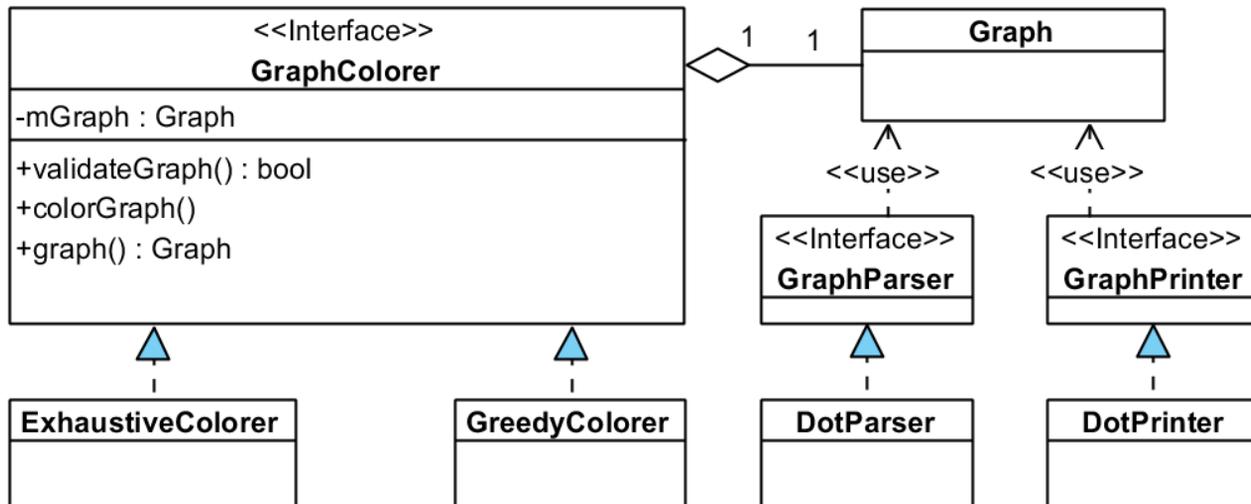


- Анализатор
 - Проводит анализ программы и строит граф конфликтов
- Аллокатор
 - Решает задачу минимизации стоимости доступов в память при ограничении по размерам банков памяти



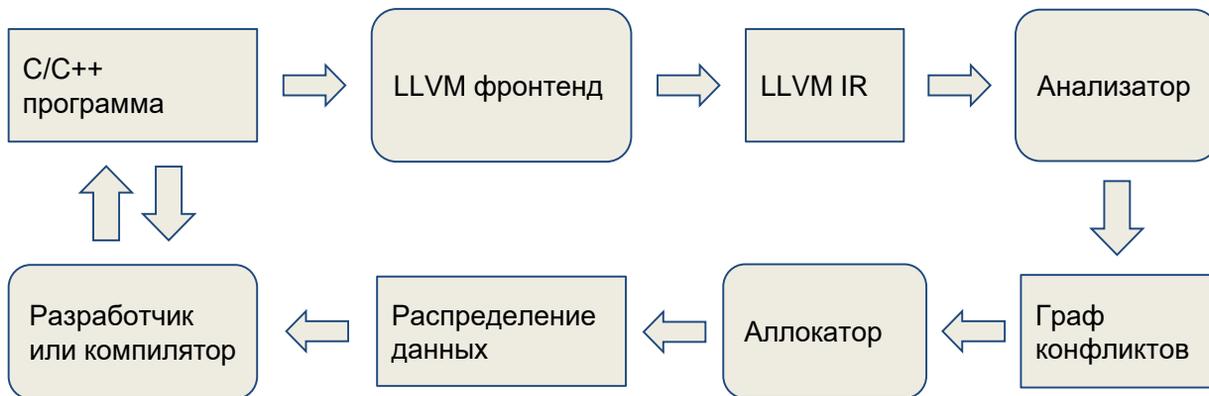


Архитектура Аллокатора



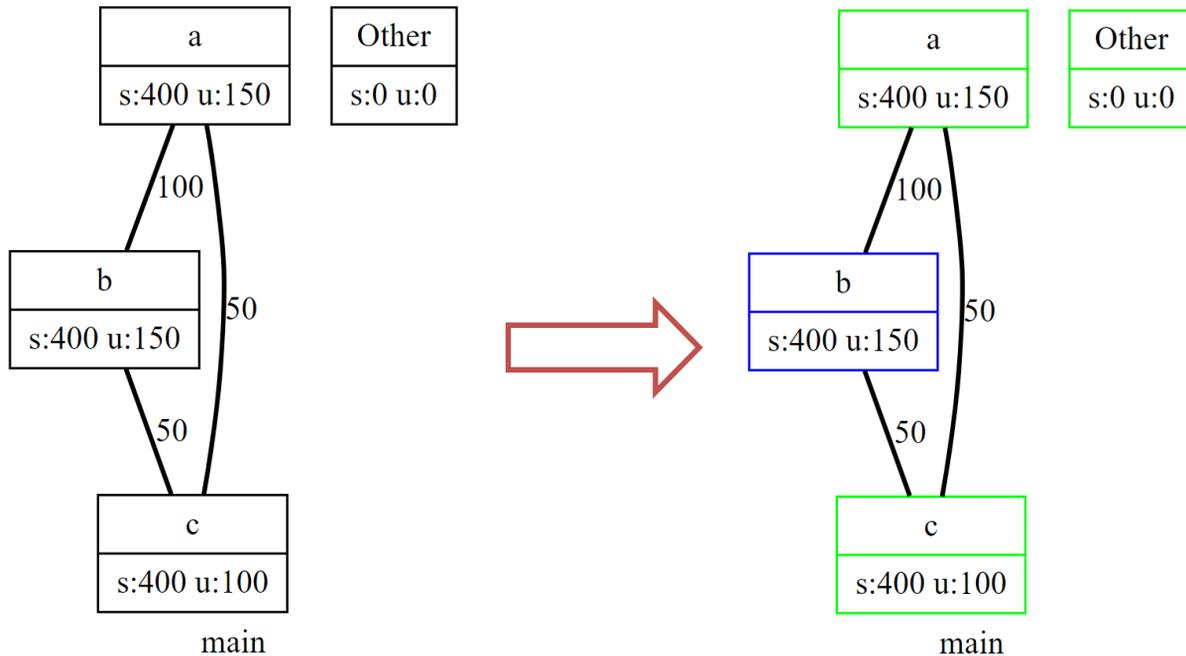


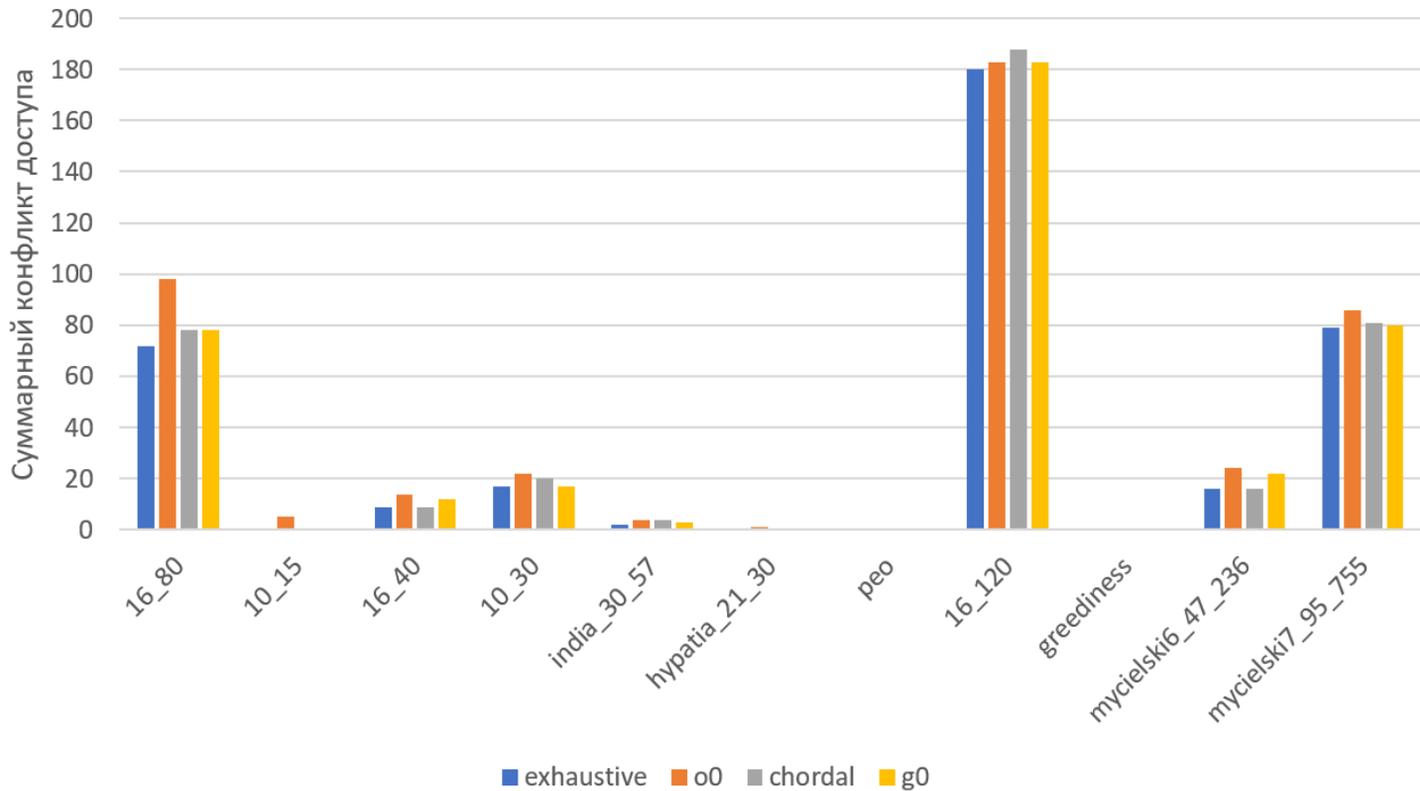
Сценарий использования

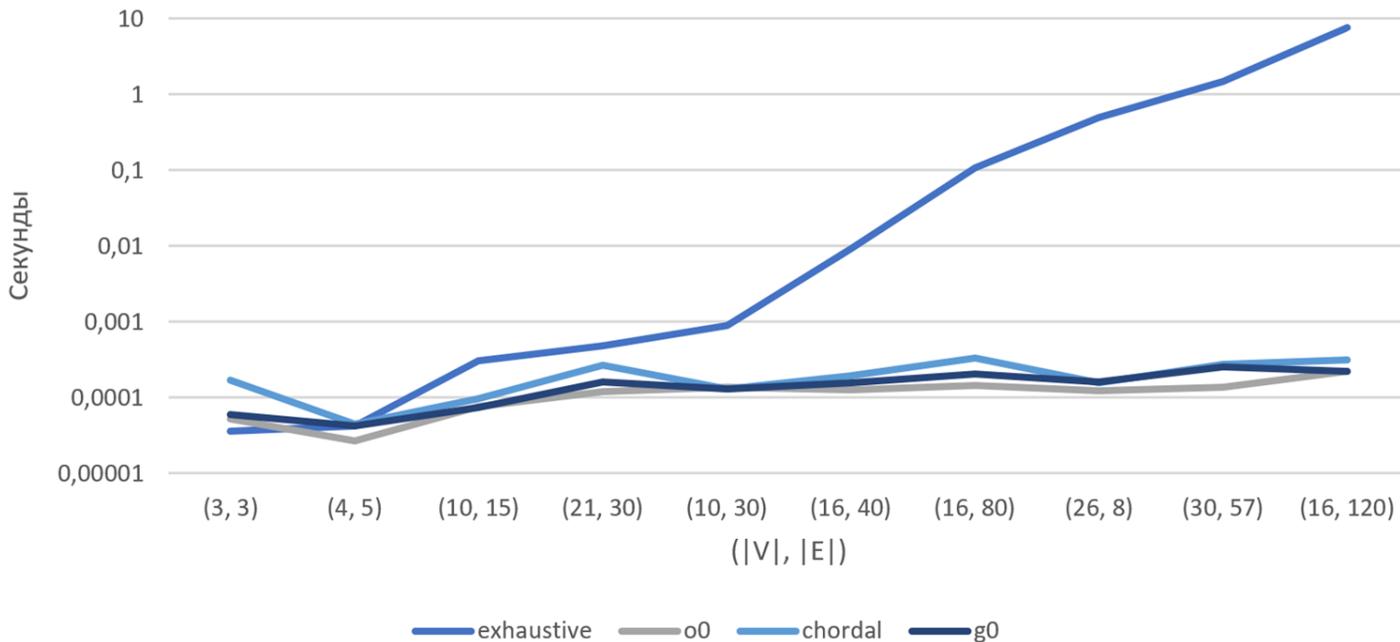




```
1. int a[100], b[100], c[100];
2.
3. int dot_product(const int *x, const int *y, int n) {
4.     int acc = 0;
5.
6.     for(int i = 0; i < n; i++) {
7.         acc += x[i]*y[i];
8.     }
9.
10. return acc;
11.}
12.
13.int main() {
14. return dot_product(a, b, 100) + dot_product(a, c, 50) + dot_product(b, c, 50);
15.}
```









- Создан инструмент для анализа использования данных программой и построения её графа конфликтов
- Создан инструмент, распределяющий данные по банкам памяти на основе графа конфликтов и конфигурации памяти
- Реализован точный алгоритм раскраски графа, использующий перебор с отсечениями, а также несколько приближенных жадных алгоритмов с использованием эвристик
- Произведена апробация на тестовых программах, в том числе реализующих алгоритм обнаружения объектов на изображении с использованием нейронной сети, обученной на наборе CIFAR-10
- Промежуточные результаты работы были представлены на конкурсе-конференции СТТПП-2019, где заняли первое место