**Афанасьев Илья Викторович Исследование и разработка методов эффективной реализации графовых алгоритмов для современных векторных архитектур**

ОГЛАВЛЕНИЕ ДИССЕРТАЦИИ

кандидат наук Афанасьев Илья Викторович

Введение

Глава 1. Обзор подходов к реализации графовых алгоритмов

для векторных систем с быстрой памятью

1.1 Три основных свойства графовых алгоритмов

1.2 Взаимосвязь свойств современных вычислительных архитектур

и графовых алгоритмов

1.3 Исследуемые в работе графовые задачи

1.4 Реализации графовых алгоритмов для современных архитектур: multicore CPU, NVIDIA GPU, векторные процессоры

1.5 Существующие подходы к оптимизации графовых алгоритмов

1.6 Подходы к оценке производительности, эффективности и локальности реализаций графовых алгоритмов

1.7 Типы и свойства входных графов, используемых в работе

1.8 Выводы главы

Глава 2. Целевые архитектуры: основные свойства,

характеристики, взаимосвязь

2.1 NEC SX-Aurora TSUBASA

2.2 Графические ускорители NVIDIA

2.2.1 NVIDIA Pascal

2.2.2 NVIDIA Volta

2.3 Intel Knight Landing

2.4 Векторные процессоры и NVIDIA GPU как представители SIMD-архитектур

2.5 Примеры приложений различных классов, подтверждающие взаимосвязь векторных архитектур и графических ускорителей NVIDIA

2.6 Выводы главы

Глава 3. Типовые алгоритмические структуры графовых

алгоритмов и подходы к их эффективной реализации на векторных системах с быстрой памятью

3.1 Почему важно исследовать типовые алгоритмические

структуры графовых алгоритмов?

3.2 Исследование информационных графов фундаментальных графовых алгоритмов

3.3 Типовые абстракции данных

3.4 Реализация абстракций данных для векторных систем с

быстрой памятью

3.4.1 Векторно-ориентированный формат хранения графов

3.4.2 Подмножество вершин и его векторно-ориентированная реализация

3.5 Реализация алгоритмических абстракций для векторных систем

с быстрой памятью

3.5.1 Алгоритмическая абстракция: генерация подмножества вершин

3.5.2 Алгоритмическая абстракция Advance

3.5.3 Алгоритмическая абстракция Compute

3.5.4 Алгоритмическая абстракция Reduce

3.6 Анализ эффективности реализованных абстракций

3.6.1 Использование Roofline-модели для анализа эффективности разработанных реализаций типовых алгоритмических абстракций

3.6.2 Анализ эффективности абстракции «генерация подмножества вершин»

3.6.3 Анализ эффективности абстракции advance

3.6.4 Анализ эффективности абстракции compute

3.6.5 Анализ эффективности абстракции reduce

3.6.6 Исследование динамических характеристик реализованных абстракций

3.7 Метод создания эффективных реализаций графовых

алгоритмов для векторных систем

3.8 Выводы главы

Глава 4. Программный комплекс для создания эффективных архитектурно-независимых реализаций графовых

алгоритмов

4.1 Актуальность разработки архитектурно-независимого фреймворка для векторных систем с быстрой памятью

4.2 Основные абстракций VGL: описание, характеристики, реализация

4.2.1 Абстракции данных: граф

4.2.2 Абстракции данных: подмножество вершин

4.2.3 Вычислительные абстракции: advance

4.2.4 Обертки вычислительной абстракции advance: gather и scatter

4.2.5 Вычислительные абстракции: generate\_new\_frontier

4.2.6 Вычислительные абстракции: compute

4.2.7 Вычислительные абстракции: reduce

4.3 Программная структура фреймворка VGL

4.4 Типовые схемы использования фреймворка VGL для

реализации графовых алгоритмов

4.5 Пример использования фреймворка VGL для реализации графовых алгоритмов на архитектуре NEC SX-Aurora TSUBASA

4.6 Сравнительная производительность реализаций на основе фреймворка с оптимизированными вручную реализациями

4.7 Выводы главы

Глава 5. Анализ производительности, эффективности и

энергоэффективности разработанных реализаций

5.1 Анализ производительности и сравнение с существующими библиотечными реализациями

5.2 Анализ эффективности

5.3 Анализ энергоэффективности

5.4 Выводы главы

Заключение

Список сокращений и условных обозначений

Словарь терминов

Список литературы

Список рисунков

Список таблиц