

Обзор проекта LuNA

В.А. Перепелкин

Область: эффективная параллельная
реализация больших численных
моделей для суперкомпьютеров

Проблематика

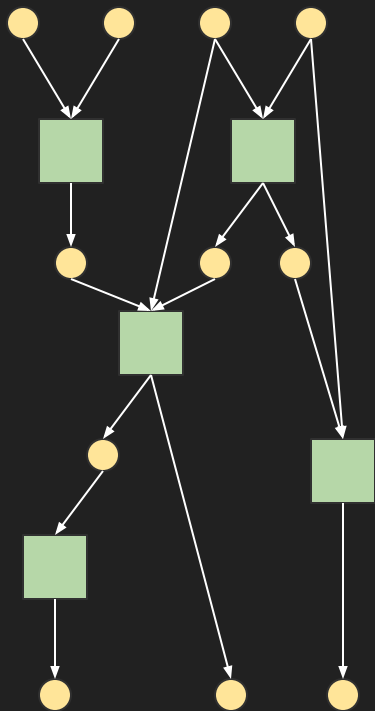
Необходимо сложное системное программирование для обеспечения эффективности

- Масштабируемость
- Учёт неоднородности
- Организация вычислений
- Распределение и перераспределение данных и вычислений
- Малые накладные расходы

Основа подхода

- Задачу конструирования параллельной программы можно разбить на подзадачи
 - Легко автоматизируемые
 - Трудно автоматизируемые
 - Практически неавтоматизируемые
- Язык программирования может быть удобен или неудобен для автоматизации
- Выберем удобную модель алгоритма и поддержим автоматизацию системно

Базовая модель: фрагментированный алгоритм



Фрагменты данных (ФД) — агрегированные переменные единственного присваивания, реализуемые как сериализуемые пассивные объекты

Фрагменты вычислений (ФВ) — операции над ФД, реализуемые как вызовы процедур без побочных эффектов

Фрагментированный алгоритм (ФА) — двудольный орграф ФВ и ФВ

Особенности модели

- Явный параллелизм
- Агрегация данных
- Возможность манипуляций с фрагментами (перемещение, репликация, ...)
- Возможность отдельно решать задачу отображения ФА на вычислитель

Качество исполнения

- распределение фрагментов по узлам: равномерность и полнота нагрузки, объём коммуникаций
- порядок выполнения фрагментов вычислений: доступность вычислений и текущий объём промежуточных данных
- управление распределёнными данными: поиск, миграция, сохранение соседства, сборка мусора

Генерация параллельной программы



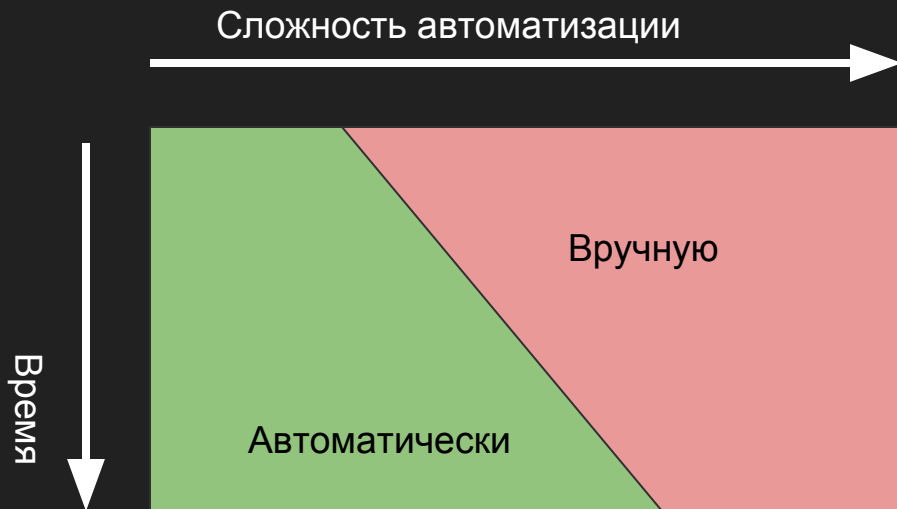
Рекомендации

Рекомендации — средство управления исполнением ФА на высоком уровне абстракции, без программирования

Виды рекомендации:

- Информационные
- Специфицирующие требуемое поведение
- Императивно определяющие исполнение

Автоматизация конструирования рекомендаций

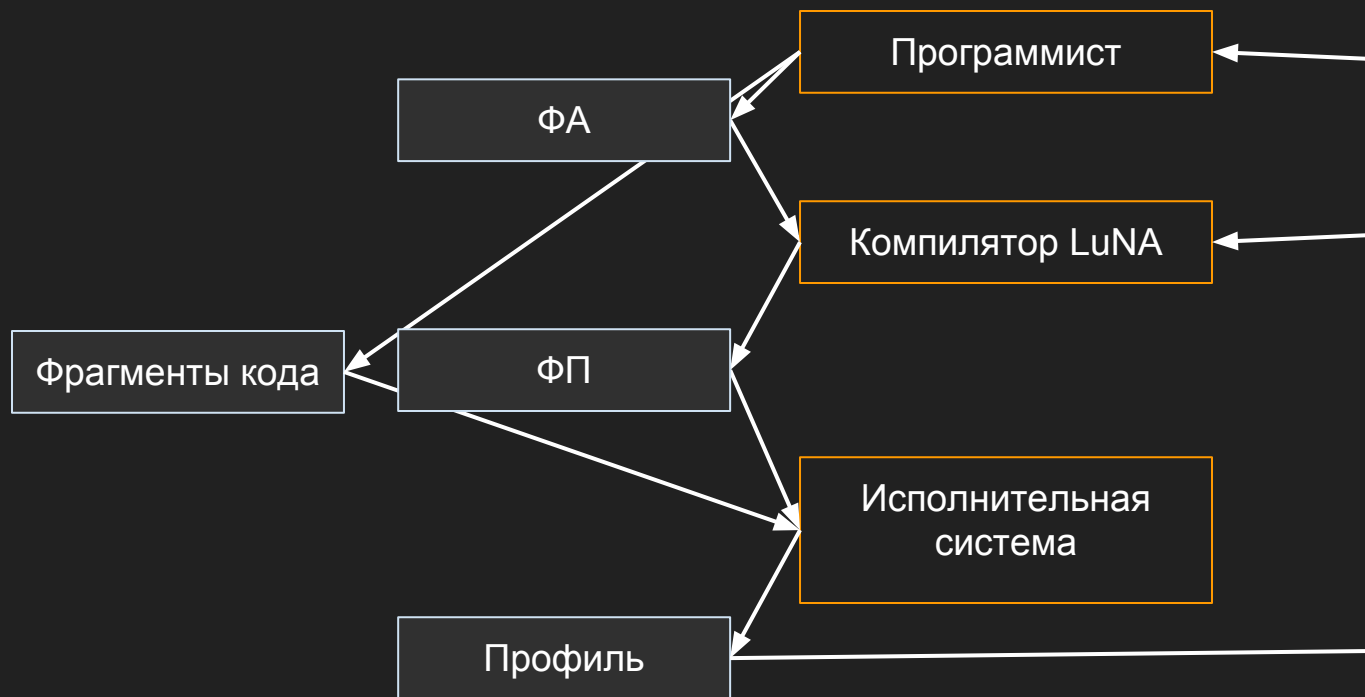


Цель: устранить параллельное программирование

Остаётся:

- Последовательное программирование
- Фрагментация алгоритма
- Спецификация требуемого поведения

Структура системы LuNA



Основные направления работы

- Методика фрагментации алгоритмов
- Язык LuNA
 - Рекомендации
 - Повышение уровня абстракции
- Компилятор LuNA
 - Инфраструктура
 - Алгоритмы проверки на ошибки
 - Алгоритмы статического анализа и анализа профиля
 - Алгоритмы конструирования рекомендаций
 - Генерация кода (прямое управление)
- Исполнительная система LuNA
 - Менеджер фрагментов вычислений
 - Менеджер фрагментов данных
 - Динамический балансировщик нагрузки

Основные направления работы (продолжение)

- Система RuSh: реализация ФА на базе прямого управления
 - Средства описания прямого управления
 - Алгоритмы конструирования прямого управления
 - Системные алгоритмы исполнения ФА
- Профилировщик
 - Язык спецификации профилирования
 - Алгоритмы сбора и сохранения профильной информации
 - Алгоритмы анализа профиля
- Визуализатор
 - Пост-мортем
 - Онлайн
 - Онлайн с функциями управления исполнением
- Использование спецвычислителей (CUDA, Xeon Phi)
- Интеграция с HPC Community Cloud

Основные направления работы (продолжение)

- Реализация прикладных задач
 - Метод частиц в ячейках
 - Метод IADE
 - Краевая задача фильтрации трёхфазной жидкости
- Библиотека численных подпрограмм
- Поддержка спецвычислителей
- Отладчик
 - Поведенческий (автоматический)
 - Традиционный (в терминах ТФП)
- Обеспечение отказоустойчивости
 - Разработка коммуникационной подсистемы
 - Системная поддержка отказоустойчивости
- Интеграция системы LuNA с NumGRID

Timeline

2008 Система RFES (Runtime Fragmented Execution System) — общая память, нет фрагментов данных, есть рекомендации (приоритеты, группы)

2010 Система LuNA-4 — распределённая память, неуниверсальный язык, рекомендации по распределению ресурсов

2012 Универсальный язык описания ФА

2013 Система LuNA-5 — распределённая память, универсальный язык

Timeline (продолжение)

2014-2015:

- Динамическая балансировка нагрузки
- Система RuSh: прямое управление на базе сетей Петри
- Масштабируемые системные алгоритмы с локальными взаимодействиями

Дальнейшие планы

- Выведение системы на публичный уровень
 - Доработка эффективности
 - Подготовка пользовательской документации
 - Подготовка tutorиала
 - Подготовка сайта с демо-системой (в НРС2С)
- Развитие системных алгоритмов по приоритетным направлениям
- Переход к задаче синтеза фрагментированных алгоритмов