

ТЕХНОЛОГИИ В ОБРАЗОВАНИИ УНИВЕРСИТЕТ

МИКРОЭЛЕКТРОНИКА
ИННОВАЦИИ
КАТАЛИТИЧЕСКИЕ
МАТЕРИАЛЫ

ДИЗАЙН
ЛЕКАРСТВ
ТОЧКА
СБОРКИ

НАУЧНАЯ
ЛАБОРАТОРИЯ
ГЕОХИМИЯ
ИНЖИНИРИНГ
ГЕОФИЗИКА

ГИБРИДНЫЕ
МАТЕРИАЛЫ
ЭНЕРГОСБЕРЕЖЕНИЕ

ВЫСОКИЕ
ЭНЕРГИИ

БИОТЕХНОЛОГИИ
МОДЕЛИРОВАНИЕ
НАНОТЕХНОЛОГИИ
СЕМИОТИКА

НАУКА
МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

ЭЛЕМЕНТАРНЫЕ
ЧАСТИЦЫ
ГЕОЛОГИЯ

КВАНТОВЫЕ
ТЕХНОЛОГИИ
БИОЛОГИЯ

ТЕМНАЯ
МАТЕРИЯ
ФОТОНИКА

БИОМЕДИЦИНА
ПРИКЛАДНЫЕ
ИССЛЕДОВАНИЯ
РАЗВИТИЕ

АСТРОНОМИЯ
ГЛОБАЛЬНЫЕ ПРИОРИТЕТЫ

АСТРОФИЗИКА
БИОИНФОРМАТИКА

ЛАЗЕРНАЯ
ФИЗИКА

АРХЕОЛОГИЯ

ЭКОНОМИКА
ЗНАНИЙ

СОТРУДНИЧЕСТВО

IT
DEEP
LEARNING
ИЗУЧЕНИЕ

МОЗГ
АРКТИКА
КОГНИТИВНЫЕ ТЕХНОЛОГИИ

N* Новосибирский
государственный
университет
*НАСТОЯЩАЯ НАУКА

Новосибирский государственный университет, НГУ



Разработка конфигурируемой подсистемы профилирования для системы LuNA

09.04.01 Информатика и вычислительная техника.
Технология разработки программных систем;

Студент: Симонов Владислав Игоревич

Руководитель: Малышкин Виктор Эммануилович,
д.т.н., профессор, зав. каф. ПВ ФИТ НГУ

Соруководитель: Киреев Сергей Евгеньевич, ст.
преп. каф. ПВ ФИТ НГУ

Рецензент: Вяткин Юрий Викторович, директор по
инновациям, ООО "Новые Программные Системы"

Дата защиты: 26.06.2025

* АКТУАЛЬНОСТЬ ПРОБЛЕМЫ

- * Рост масштабов высокопроизводительных вычислений
- * Необходимость гибких инструментов профилирования в системе LuNA
- * Существующие решения были частными, не было общего подхода, были под конкретную задачу, не совместимы между собой

* ЦЕЛЬ РАБОТЫ И ЗАДАЧИ

Цель: разработка конфигурируемого компонента системы LuNA, который позволит собирать и анализировать сведения о выполнении фрагментированных программ. Были поставлены следующие задачи:

- * спроектировать и реализовать событийную модель профилирования;
- * спроектировать и реализовать механизм расширения через модули;
- * спроектировать и реализовать средства конфигурирования.

* ОБЗОР ОБЩЕПРИНЯТЫХ ИНСТРУМЕНТОВ ПРОФИЛИРОВАНИЯ

- * **Intel Trace Analyzer and Collector** – инструмент для сбора и визуализации трасс MPI-программ, помогающий анализировать производительность параллельных приложений.
- * **HPC Toolkit** – это набор инструментов для сбора, анализа и визуализации производительности параллельных программ с привязкой к исходному коду, позволяющий выявлять узкие места и дисбаланс нагрузки в HPC-приложениях.
- * **TAU** – это инструмент для профилирования и трассировки параллельных программ, поддерживающий различные модели (MPI, OpenMP, CUDA и др.). Он собирает данные о времени выполнения, ресурсах и коммуникациях, интегрируется с компиляторами и поддерживает визуализацию через ParaProf и PerfExplorer.

* ОБЗОР ПРОФИЛИРОВОЧНЫХ МОДУЛЕЙ LuNA

- * **WorkflowTracer** – это инструмент для трассировки жизненного цикла вычислительных фрагментов в распределённых приложениях, помогающий выявлять зависшие фрагменты и семантические ошибки с помощью специализированных методов и последующего анализа утилитой `luna_trace`.
- * **Evlog** – это модуль для автоматического логирования ключевых событий исполнения параллельных программ, позволяющий анализировать память, сеть и параллелизм с помощью `LuNA Evlog Analyzer`.
- * **Evlog Profiler** – это инструмент для визуализации исполнения LuNA-программ, разработанный в работе Голикова М. О. Он собирает данные о событиях во время выполнения через встроенное логирование, отображает ход выполнения и статистику программы, а также позволяет анализировать производительность на уровне фрагментов.

* ОБЗОР ЛИТЕРАТУРЫ

- * **Абрамушкина Е. С.** Разработка и реализация подсистемы профилирования для системы фрагментированного программирования LuNA // Выпускная работа бакалавра, НГУ Факультет информационных технологий. – 2023.
- * **Лямин А. С.** Разработка и реализация алгоритма распределения ресурсов фрагментированных программ на основе профилирования // Выпускная работа бакалавра, НГУ, Факультет информационных технологий. – 2021.
- * **Саяпин М. П.** Разработка и реализация алгоритмов динамической балансировки вычислительной нагрузки для подсистемы воспроизведения трасс в системе LuNA. Выпускная работа бакалавра, НГТУ, Факультет прикладной математики и информатики. – 2022.
- * **Мустафин Д. Э.** Реализация централизованного подхода к динамической балансировке вычислительной нагрузки в системе фрагментированного программирования LuNA и его сравнение с децентрализованным подходом. Выпускная работа бакалавра, НГУ, Факультет информационных технологий. – 2022.

* КОНТЕКСТ

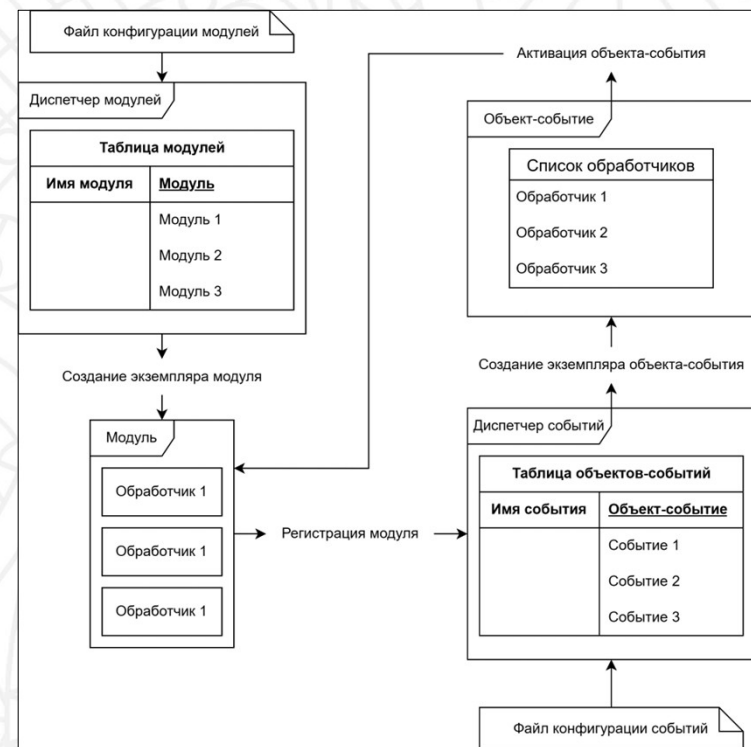
- * Система LuNA – инструмент автоматического построения параллельных программ численного моделирования
- * Фрагмент кода – участок кода C++ с заданным интерфейсом без побочных эффектов
- * Фрагменты данных – переменные единственного присваивания, служащие входными и выходными параметрами для фрагментов вычислений
- * Фрагменты вычислений – это применение некоторых фрагментов кода к конкретным фрагментам данных

* ТРЕБОВАНИЯ

- * Унифицированный подход к сбору данных
- * Модульная архитектура
- * Простота расширения
- * Гибкая настройка профилирования

* ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

- * Архитектура: централизованная событийная система с множеством независимых модулей
- * Событийная модель: фиксируются и обрабатываются события
- * Модули: каждый модуль отвечает за свой вид анализа
- * Конфигурирование: пользователь задаёт, какие события и модули включены



Архитектура подсистемы

* СОБЫТИЙНАЯ МОДЕЛЬ

- * За каждое событие отвечает отдельный объект-событие, который хранит ссылки на подписанные обработчики. Обработчики – это функции, которые могут использовать данные, переданные объектом-событием.

```
template<typename... Args>
class Event
{
public:
    typedef std::function<void(Args...)> Slot;

    void connect(const Slot& slot)
    {
        slots_.push_front(slot);
    }

    void emit(Args... args)
    {
        for (auto i = slots_.begin(); i != slots_.end(); ++i)
            (*i)(args...);
    }

private:
    std::list<Slot> slots_;
};
```

Класс объекта-события

* МОДУЛИ И ОБРАБОТЧИКИ

- * Модуль – это логическая единица профилировщика LuNA, состоящая из набора обработчиков с общим внутренним состоянием. Модули подключаются к системе через специальный интерфейс и регистрируют обработчики для реакции на события.

```
class DFSizerEventModule : public IEventModule {
public:
    void registerEvents() override {
        CONNECT_EVENT(DFEvents, onCreateSize, [&]
            (const char* message, size_t size) {
                std::lock_guard<std::mutex> lock(mutex_);
                totalSizeCreated_ += size;
            });
        CONNECT_EVENT(DFEvents, onDestroySize, [&]
            (const char* message, size_t size) {
                std::lock_guard<std::mutex> lock(mutex_);
                totalSizeDestroyed_ += size;
            });
    }
};
```

Пример реализации модуля

* КОНФИГУРИРУЕМОСТЬ СОБЫТИЙ

- * Для минимизации накладных расходов и гибкости сценариев предусмотрена настройка событий через JSON-файл. Пользователь может включать или отключать отдельные события, а также объединять их в группы для удобного управления.

```
{
  "eventsSettings": {
    "CFEvents": {
      "onStarted": true,
      "onFinished": true
    },
    "DFEvents": {
      "onCreate": true,
      "onDestroy": true
    }
  },
  "groups": {
    "LifeCycle": {
      "enabled": true,
      "events": [ "CFEvents::onStarted", "CFEvents::onFinished" ]
    },
    "Memory": {
      "enabled": false,
      "events": [ "DFEvents::onCreate", "DFEvents::onDestroy" ]
    }
  }
}
```

Пример конфигурации событий

* КОНФИГУРИРУЕМОСТЬ СОБЫТИЙ

- * Для каждого события можно задать правила вставки объекта-события и перечень аргументов, которые будут переданы обработчикам. Компилятор LuNA автоматически добавит вызов объекта-события с нужными параметрами в соответствующее место программы.

```
"onStarted": {  
  "enabled": true,  
  "rule": "after BlockRetStatus",  
  "args": [ "__func__", "std::chrono::high_resolution_clock::now()" ]  
},  
"onExited": {  
  "enabled": true,  
  "rule": "before return EXIT",  
  "args": [ "__func__", "std::chrono::high_resolution_clock::now()" ]  
},  
"onMigrated": {  
  "enabled": true,  
  "rule": "before return MIGRATE",  
  "args": [ "__func__", "std::chrono::high_resolution_clock::now()" ]  
},
```

Пример написания правила

* КОНФИГУРИРУЕМОСТЬ МОДУЛЕЙ

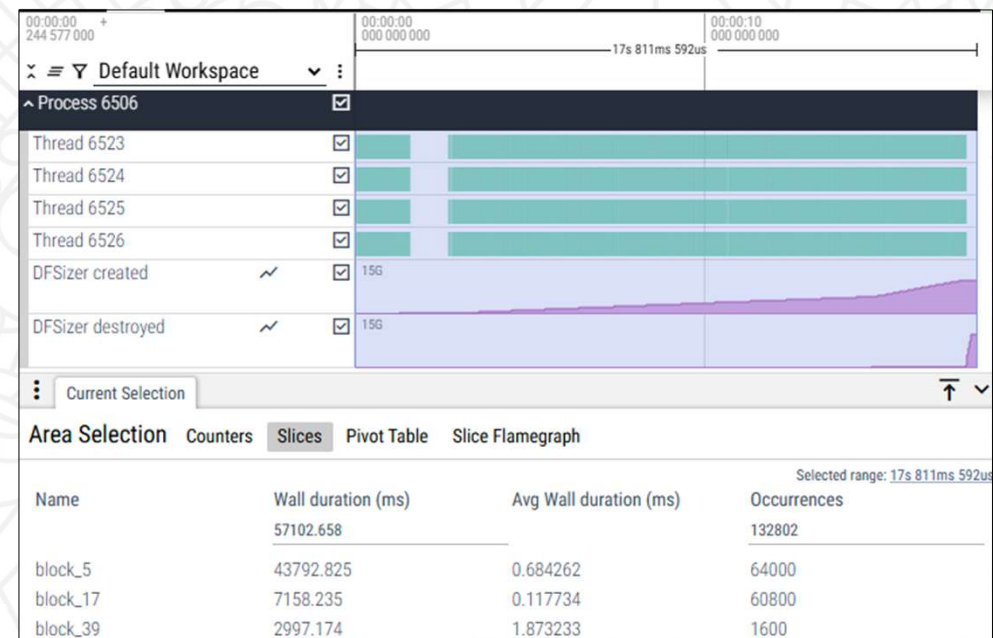
- * Модули, как и события, настраиваются через JSON-файл: их можно включать и отключать как отдельно, так и глобально. Параметр `overrideEnabled` позволяет задать индивидуальное поведение, а `priority` – управлять порядком вызова обработчиков.

```
{  
  "globalSettings": {  
    "enabled": false  
  },  
  "libdf_sizer_events_module": {  
    "overrideEnabled": true  
  },  
  "libwrite_file_module": {  
    "enabled": false,  
    "priority": 1  
  },  
  "libread_file_module": {  
    "enabled": false,  
    "priority": 2  
  }  
}
```

Пример конфигурации модулей

* ДЕМОНСТРАЦИЯ РАБОТЫ

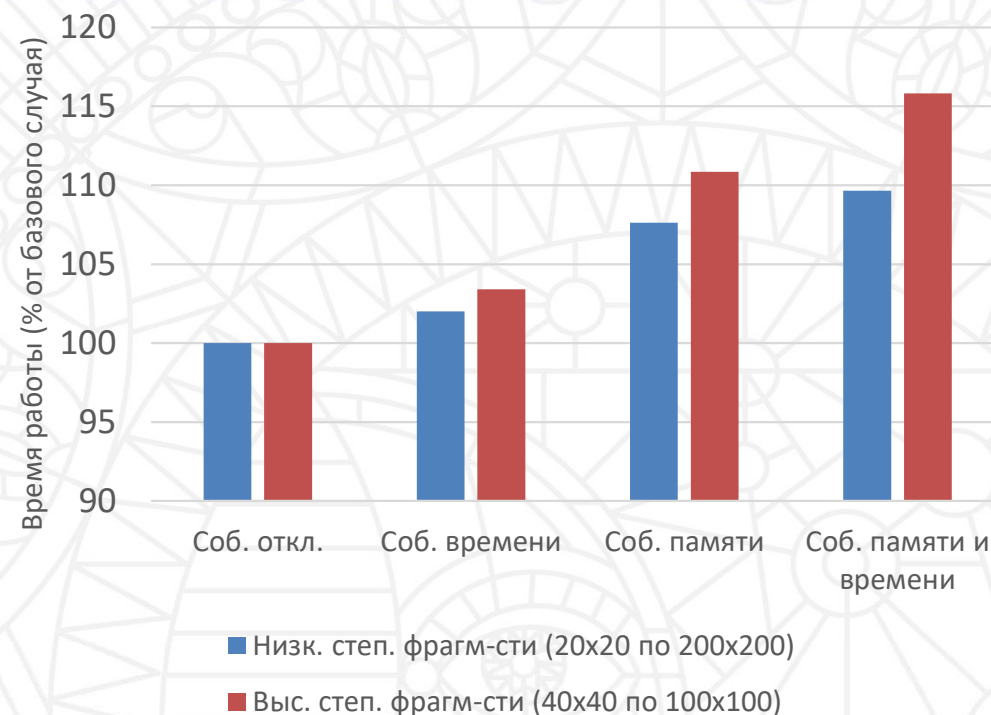
- * Для демонстрации работы был создан модуль, который собирает информацию о времени выполнения блоков параллельных вычислений, а также фиксирует объём выделенной и освобождённой памяти. По завершении работы модуль агрегирует все данные и сохраняет их в формате, совместимом с сервисом Perfetto.



Просмотр результатов в интерфейсе Perfetto

* ВЛИЯНИЕ НА ПРОИЗВОДИТЕЛЬНОСТЬ

- * Для анализа производительности была протестирована программа умножения матриц в двух режимах: с низкой фрагментированностью (разделение на 20×20 блоков по 200×200 элементов) и с высокой фрагментированностью (разделение на 40×40 блоков по 100×100 элементов). Эксперименты проводились с различными комбинациями настроек профилировочной системы.



* ВЫВОДЫ

- * Показан пример модуля для отслеживания событий памяти и времени выполнения, который автоматически формирует совместимый с визуализатором файл.
- * Анализ производительности показал, что профилирование вносит лишь умеренные накладные расходы

* АПРОБАЦИЯ

- * Работа была представлена в рамках 63-й международной научной студенческой конференции МНСК-2025 и получила диплом третьей степени
- * Работа была продемонстрирована на студенческом рабочем научно-организационном семинаре «Активные знания и система LuNA» в Лаборатории синтеза параллельных программ ИВМиМГ СО РАН 6 мая 2025 г.

* ЗАКЛЮЧЕНИЕ

- * Реализован единый механизм сбора данных через события и обработчики, обеспечивающий гибкость и расширяемость
- * Разработан интерфейс и система подключения внешних модулей, позволяющая добавлять новые функции без изменения основного кода
- * Внедрена поддержка конфигурационных файлов для настройки событий и модулей
- * Проведены экспериментальные исследования, подтвердившие корректность работы подсистемы и приемлемый уровень нагрузки на ресурсы системы при её использовании

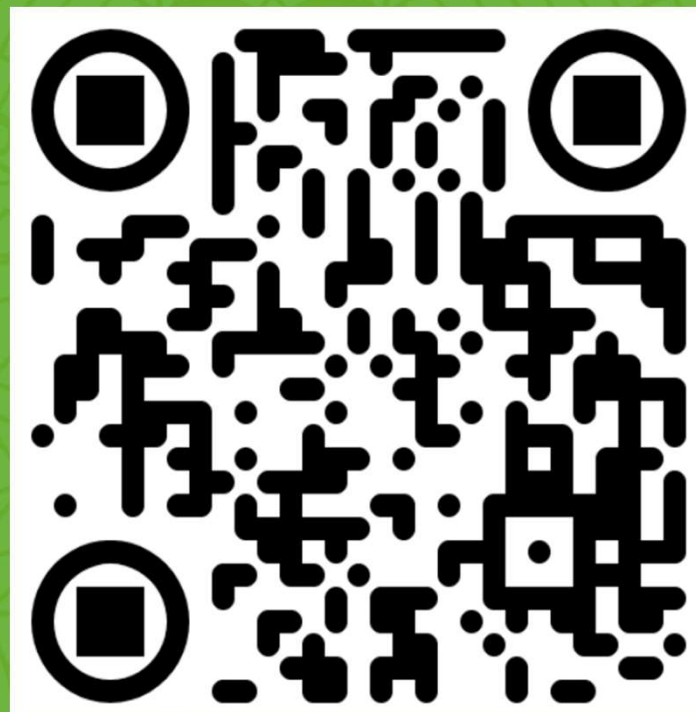
* НАПРАВЛЕНИЯ ДАЛЬНЕЙШЕЙ РАБОТЫ

- * Возможность задавать параметры модулей через конфигурационные файлы для большей гибкости
- * Интеграция профилировщика с балансировщиком нагрузки для динамического распределения задач
- * Разработка инструментов автоматического анализа и визуализации данных профилирования

Спасибо за внимание!



Ветка с кодом



Мой контакт

N* Новосибирский
государственный
университет
***НАСТОЯЩАЯ НАУКА**