НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
Факультет информационных технологий
Кафедра параллельных вычислений

РАЗРАБОТКА И РЕАЛИЗАЦИЯ АЛГОРИТМОВ УПРАВЛЕНИЯ ПАМЯТЬЮ В СИСТЕМЕ АКТИВНЫХ ЗНАНИЙ LUNA

Выполнил: Олимпиев Юрий Юрьевич, студент группы 21210 ФИТ НГУ

Руководитель: Перепёлкин Владислав Александрович, к.т.н., доц. каф. ПВ ФИТ НГУ

Соруководитель: Матвеев Алексей Сергеевич, ст. преп. каф. ПВ ФИТ

20.06.2025

Новосибирск, 2025

Актуальность проблемы

Задача автоматического управления памятью возникает во многих информационных и вычислительных системах.

Специфика управления памятью зависит от требований к системе и класса решаемых системой задач.

Унификация подходов затруднена как разнообразием вычислительных архитектур, так и различиями в требованиях к прикладным системам.

Актуальность проблемы

Система активных знаний LuNA представляет собой пригодную для исследования механизмов автоматического управления памятью среду.

Основная задача системы активных знаний - упрощение процесса разработки прикладных программ, генерация высокопроизводительного кода.

Цель. Задачи

Целью работы является разработка механизмов управления памятью вычислителя в системе активных знаний LuNA.

Для достижения этой цели в работе были решены следующие **задачи**:

- Анализ требований к управлению памятью в вычислительных системах и выявление особенностей, влияющих на архитектуру решения;
- разработка модели управления памятью для снижения избыточного выделения памяти в типичных вычислительных задачах системы активных знаний LuNA;
- реализация модуля управления памятью вычислителя на основе разработанных алгоритмов;
- оценка эффективности предложенной реализации, сравнительный анализ.

Существующие подходы

Критерии обзора существующих решений:

- Возможность сокрытия механизмов управления ресурсами вычислителя;
- обеспечение пригодных для практического использования показателей производительности выходных программ.

Существующие подходы

C/C++

Rust

Java™

NVIDIA CUDA

julia Programming

GOLANG

Существующие подходы

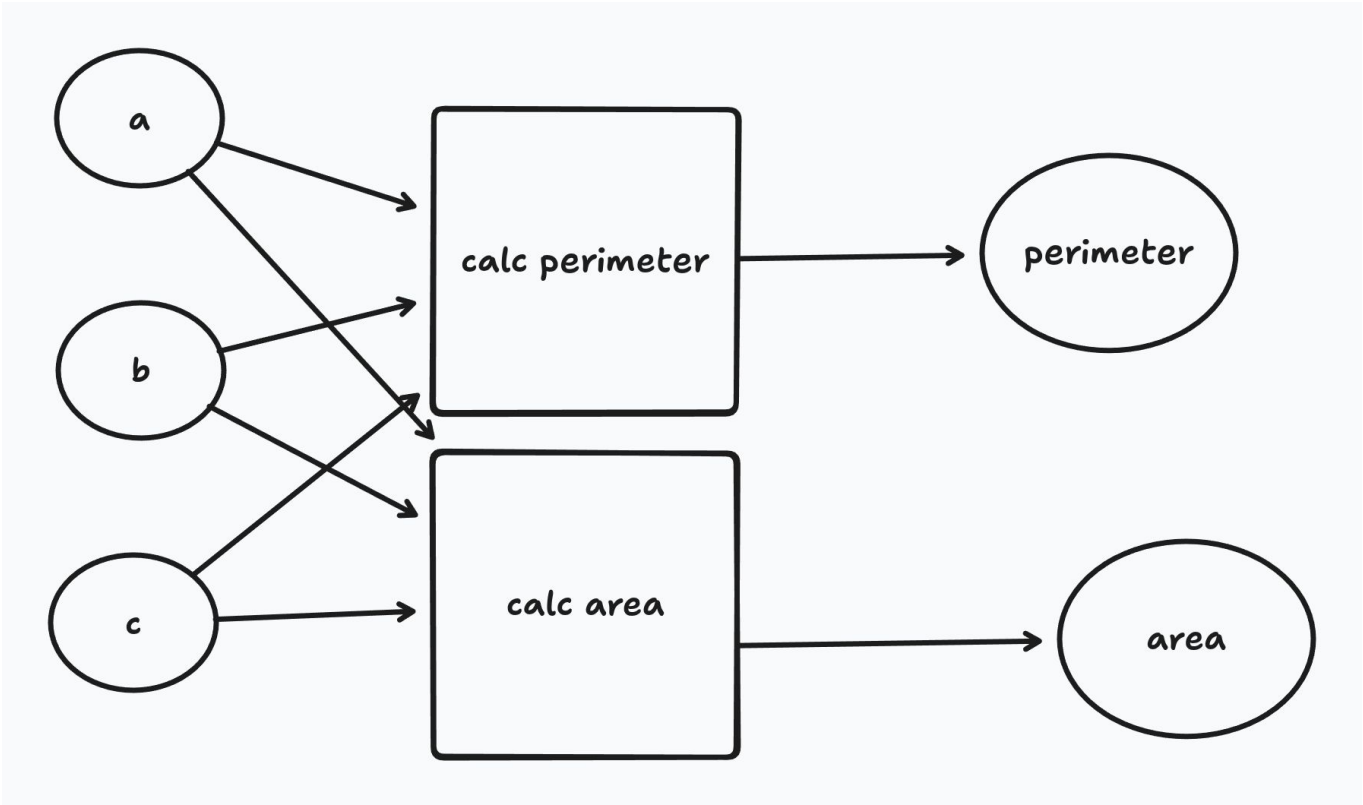
- Высокую производительность дают системы с ручным управлением памятью, такие как C/C++, CUDA.
- Системы с встроенным сборщиком мусора позволяют скрывать детали управления памятью от разработчика, но влекут за собой дополнительные накладные расходы (Java, Golang).
- гибридные решения, соединяющие в себе оба подхода (Rust, Julia), хорошо показывают себя на практике, но не могут быть в явном виде применены к системам активных знаний.

Вывод: рассмотренные подходы к управлению памятью обладают преимуществами и недостатками, которые определяют их применимость к системам автоматической генерации программ, подобным LuNA. Для их интеграции необходимы дальнейшие исследования и разработка специализированных инструментов, способных работать в условиях генерации кода с учетом параллелизма и разнообразия архитектур.

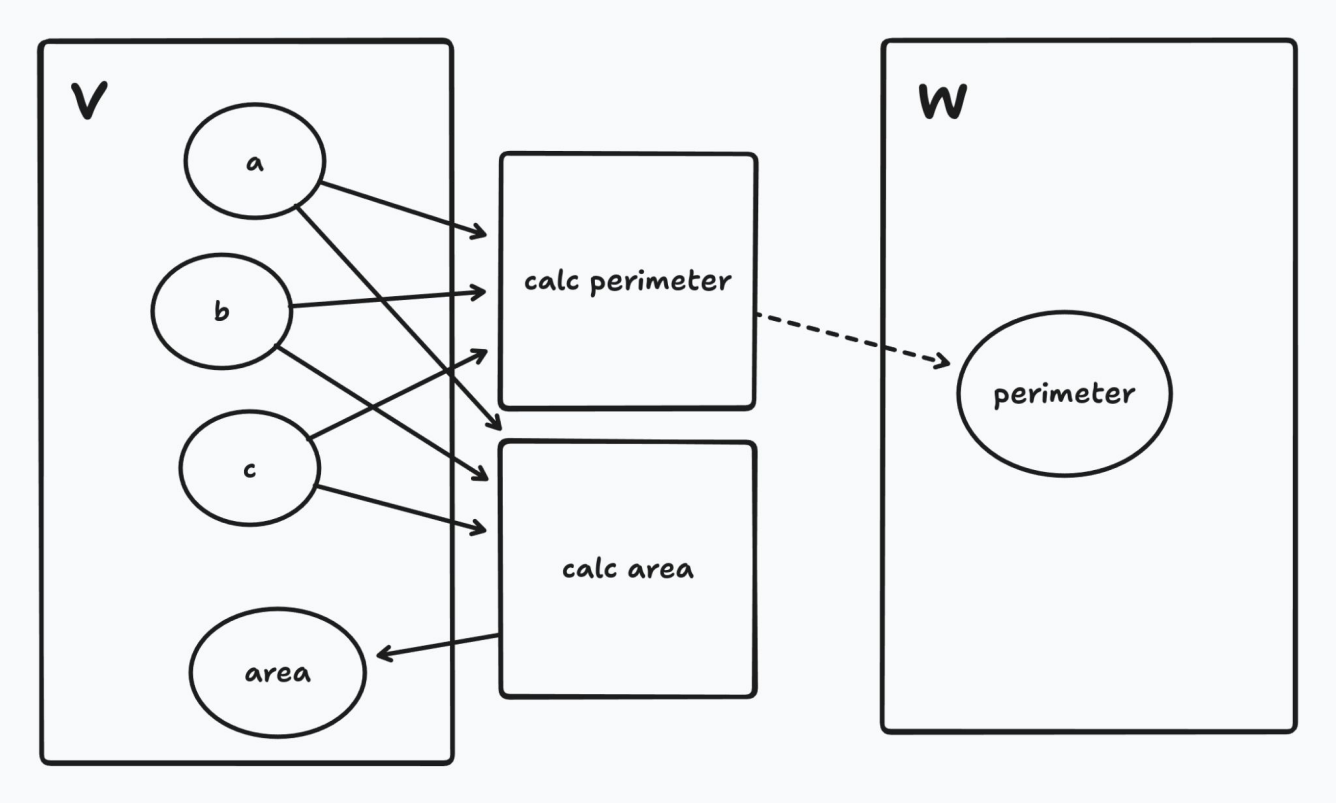
Система активных знаний LuNA

LuNA — это система активных знаний, основная цель которой — автоматическая генерация высокопроизводительных программ на основе формальной постановки задачи.

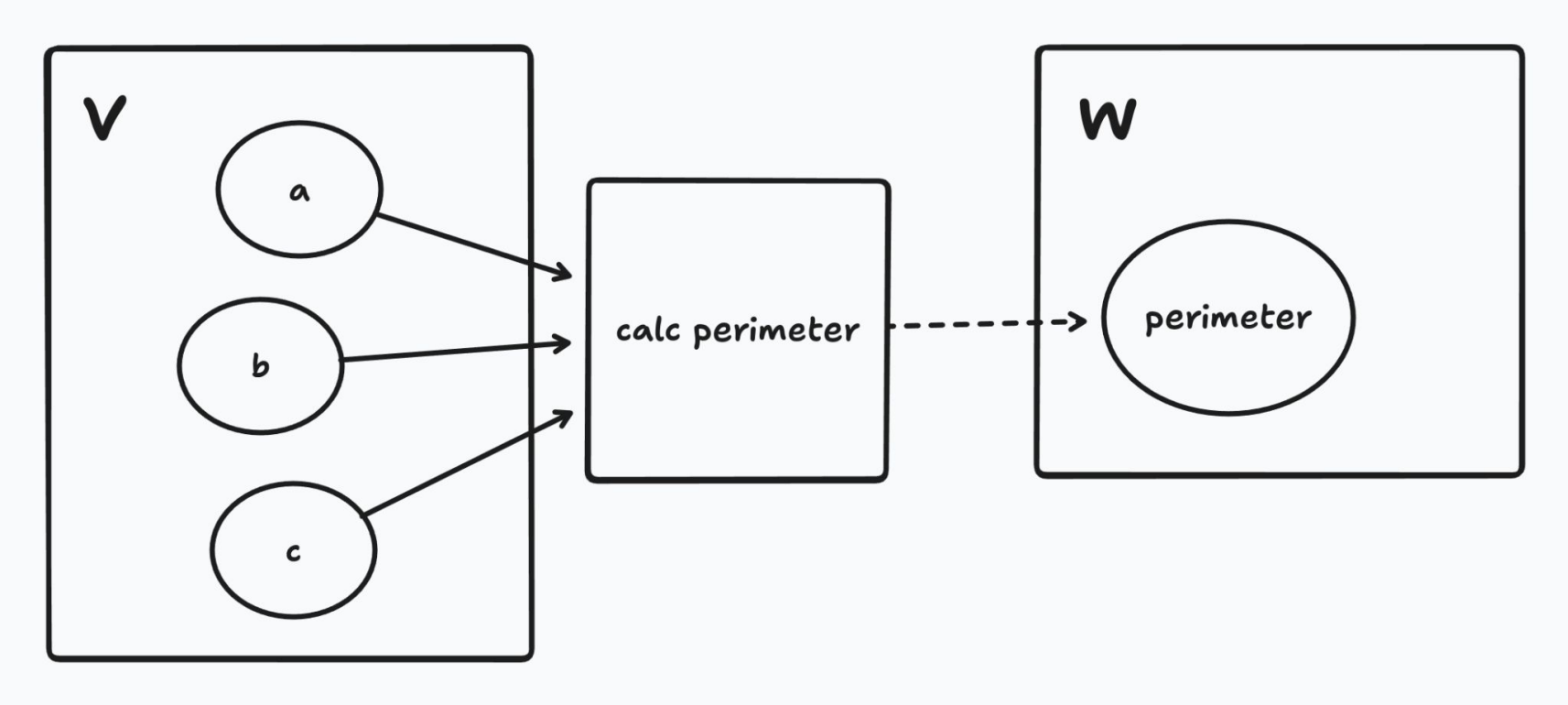
Вычислительная модель



VW-задача



VW-план



Система активных знаний LuNA

Решение задачи в LuNA может быть получено:

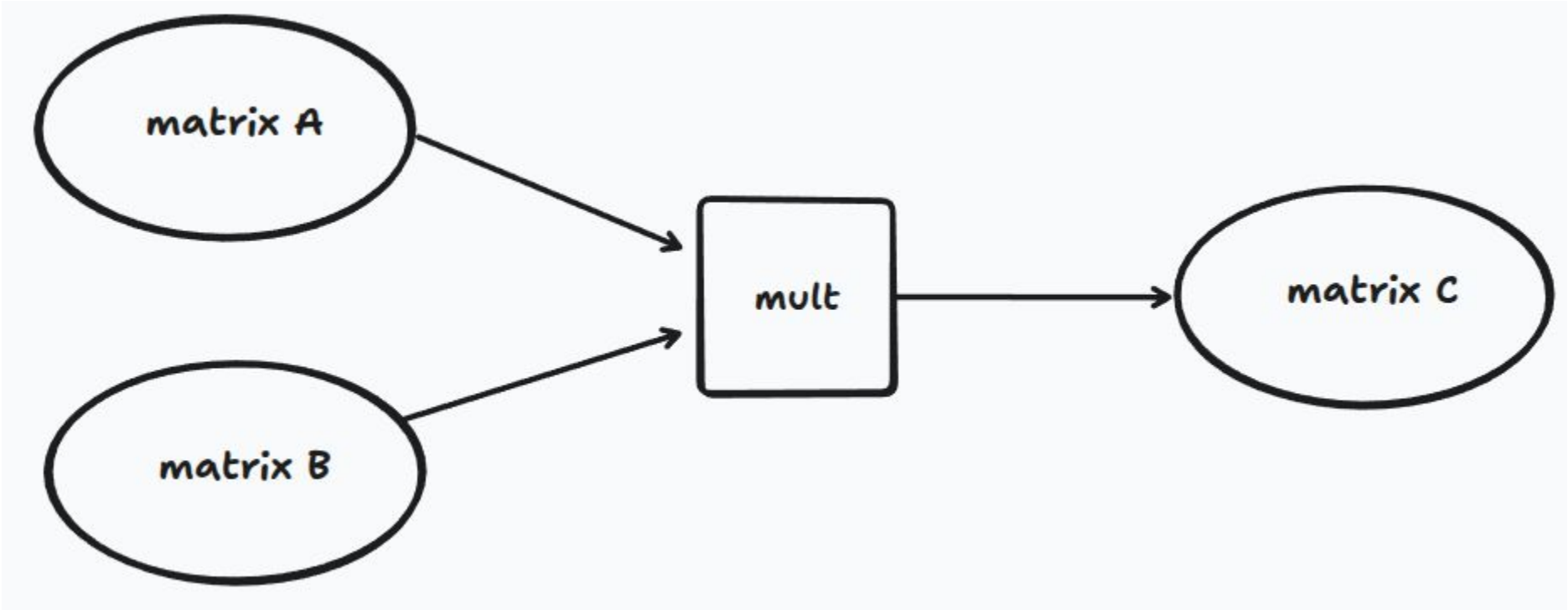
- либо предварительным построением плана,
- либо динамическим выбором операций во время исполнения (runtime-интерпретатор).

Идея решения

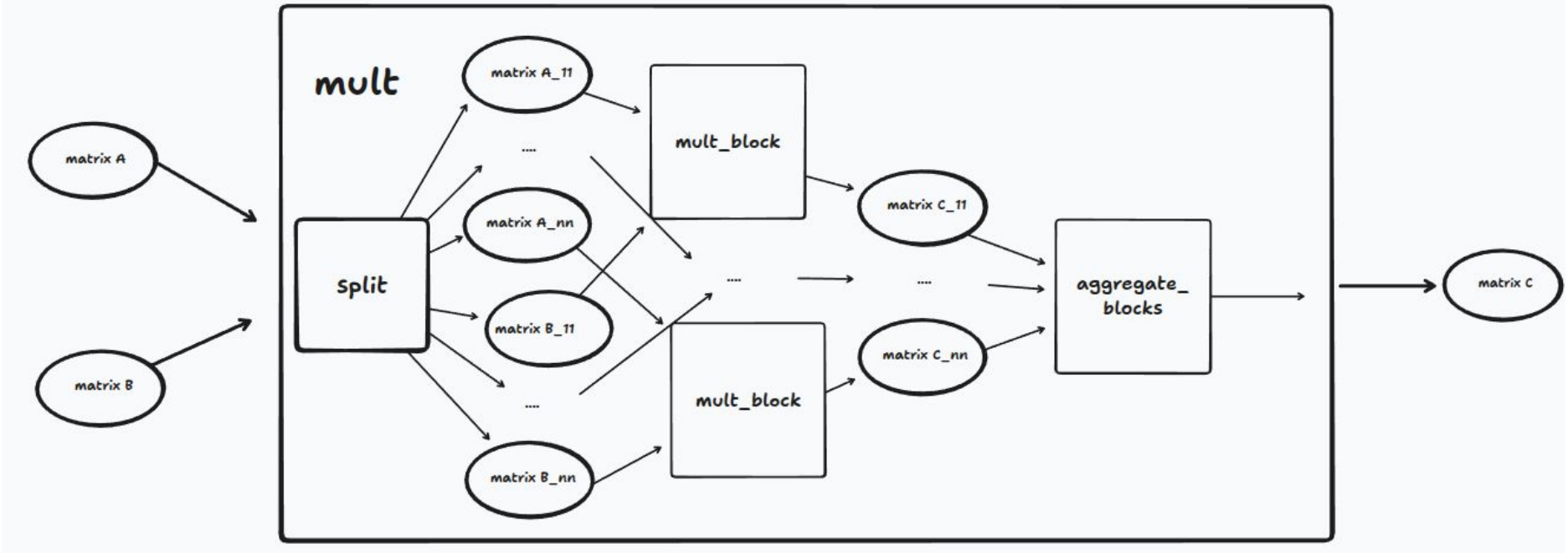
Предполагаемый подход — интеграция управления переменными как части вычислительной логики, но:

- с сокрытием этой логики от пользователя;
- с открытостью этой логики для системы планирования и исполнения.

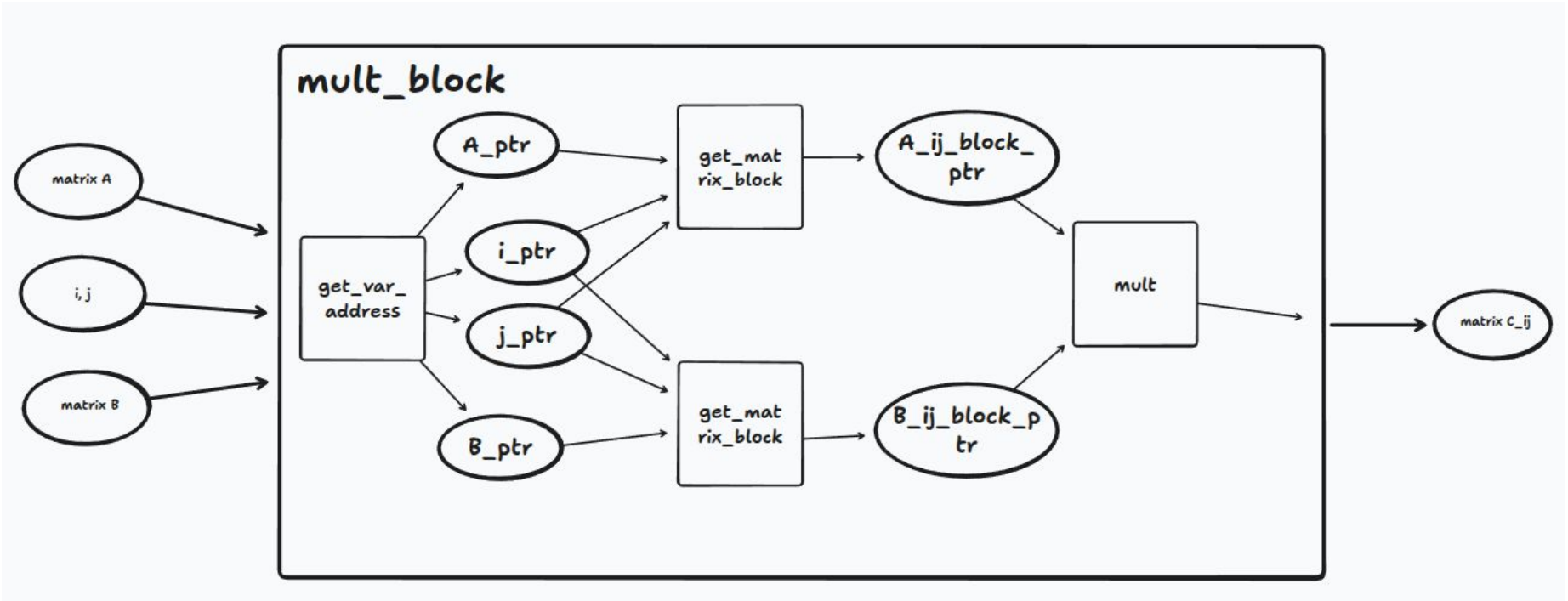
Идея решения. Пример



Идея решения. Пример



Идея решения. Пример



Формальная модель

$V = \{v_1, \ldots, v_n\}$ — множество переменных,

$T = \{t_0, t_1, \ldots, t_p\}$ — дискретное множество шагов исполнения программы,

$s: V \times T \rightarrow S$, S - множество кортежей вида (name, type, shape, state), которое задает состояние переменной v в момент времени t.

shape in {0, (n), (n,m)} - скаляры, массивы и матрицы.

type in {int, double, float}.

state in {ready, undefined}.

Таким образом, состояние переменной может изменяться от шага к шагу. Например, на шаге t_0 переменная может быть создана, на шаге t_1— записано значение, а на t_2 — произведено чтение.

Формальная модель. Операции

Введем набор допустимых операций доступа с учетом времени. Каждая операция теперь определяется на моменте времени и может порождать изменение состояния:

create(v, t): s(v, t):= (name, type, shape, undefined).

write(v, d, t) → Unit, s(v, t).state = ready.

read(v, t) → **d** блокирует исполнение пока s(v, t`).state != ready.

wait(v, t) → Unit: блокирует выполнение до тех пор, пока s(v, t`).state != ready.

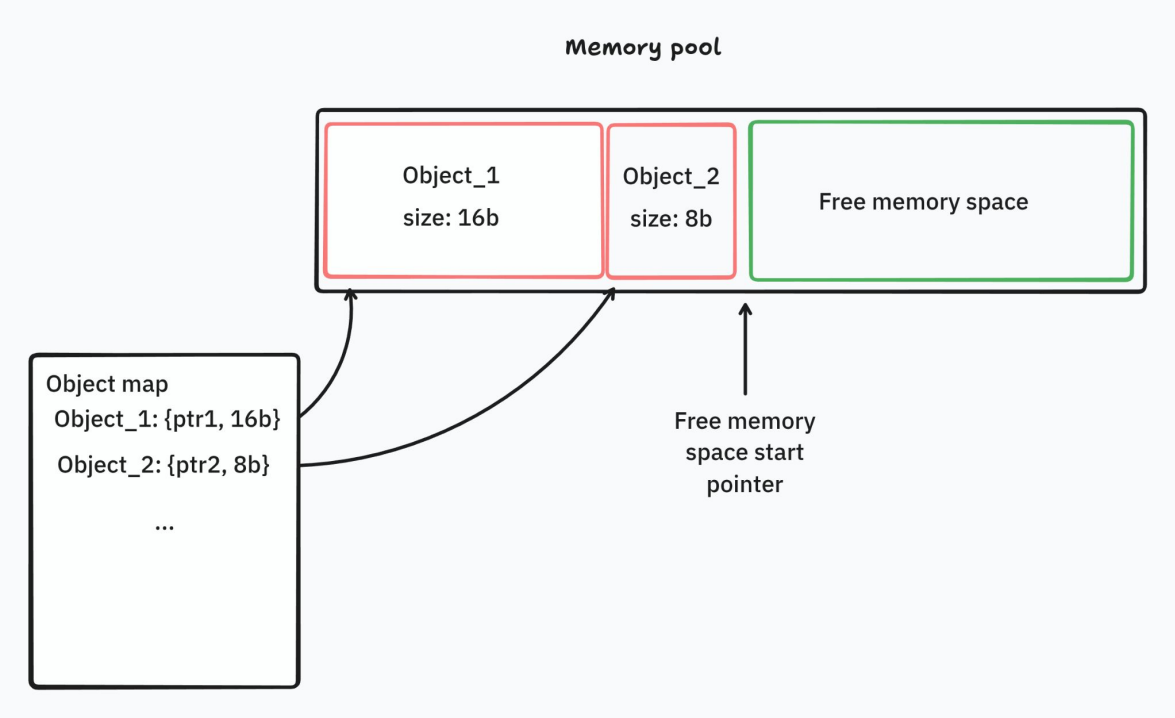
Формальная модель. Операции

consume([v_1, …, v_k], f, t): при выполнении условия s(v, t_i).state = ready выполняется функция f(v_1, …, v_k), которая создает изменения в состояниях. Выполняет её со значениями v1, …, vn, когда они все готовы.

slice(v, i_1, i_2, t): shape(v) = (n), при s(v, t`).state = ready возвращает подмассив v` = v[i_1: i_2].

submatrix(v, i_1, i_2, j_1, j_2): если shape(v) = (n, m), при s(v, t`).state = ready возвращает подматрицу v` = v[i_1:i_2, j_1:j_2].

Формальная модель. Аренная модель



Формальная модель. Аренная модель

V → Offset in A, где Offset содержит информацию о позиции и длине блока в арене.

Дополнительное отображение

map: Offset → RAM | VRAM | SharedMem

отвечает за физическое связывание логического блока в арене с конкретной памятью вычислителя: основной ОЗУ, видеопамятью или общей памятью при распределенном хранении.

Реализация

Язык реализации — C. Это обусловлено:

- совместимостью с уже реализованными модулями системы LuNA,
- распространенностью C в целевых высокопроизводительных предметных областях.

Структура хранения служебной информации о переменных — односвязный список.

В будущем возможна замена на хеш-таблицу для ускорения поиска по имени переменной.

Реализация. Расширение арены

Рассматривались два основных варианта расширения арены:

realloc-модель:

- требует полного копирования,
- инвалидирует все указатели,

Чанковая модель:

- Арена разбивается на последовательность чанков;
- Новый чанк создается по мере необходимости;
- Ранее выделенная память не перемещается — указатели остаются валидными;
- Уменьшается нагрузка при расширении области используемой памяти.

Реализация. Альтернативные идеи реализации расширения арены

Альтернативные идеи

- Индексная адресация: ссылка на переменные через таблицу указателей.
- Разделение данных и метаинформации на области памяти: эффективный доступ, сложность синхронизации.

Обе идеи рассматриваются как перспективные направления развития и исследования.

Реализация. Потокобезопасность

Для синхронизации доступа используется глобальный мьютекс, что обеспечивает защиту операций создания и поиска переменных.
Планируется опциональное отключение синхронизации (для однопоточных сценариев) и
возможность использования других примитивов.

Экспериментальное тестирование

Размер матрицы	Ручная реализация	Менеджер памяти	Реализация на LuNA v6
512	6293568	6819710	11088771
1024	25167936	27266942	29963139
2048	100665408	102912642	105460322
4096	402655296	405854370	407450146

Заключение

В рамках выполнения выпускной квалификационной работы был разработан, реализован и протестирован модуль управления памятью, ориентированный на использование в системах автоматической генерации параллельных программ.

Защищаемые положения:

1. Разработана формальная модель управления переменными в памяти вычислителя;
2. Реализован модуль управления переменными в памяти вычислителя, проведено экспериментальное исследование работоспособности алгоритма управления переменными.

Планы

Планируется продолжение работы по следующим направлениям:

- Интеграция разработанного модуля в систему LuNA;
- Расширение модели для поддержки распределенных систем и гетерогенных вычислений;
- Разработка механизмов статического анализа для автоматического синтеза правил управления памятью.

Апробация работы

Данная работа была опубликована на 63-й Международной научной студенческой конференции, г. Новосибирск, 2025 г.

Спасибо за внимание

НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ
Факультет информационных технологий
Кафедра параллельных вычислений

РАЗРАБОТКА И РЕАЛИЗАЦИЯ АЛГОРИТМОВ УПРАВЛЕНИЯ ПАМЯТЬЮ В СИСТЕМЕ АКТИВНЫХ ЗНАНИЙ LUNA

Выполнил: Олимпиев Юрий Юрьевич, студент группы 21210 ФИТ НГУ

Руководитель: Перепёлкин Владислав Александрович, к.т.н., доц. каф. ПВ ФИТ НГУ

Соруководитель: Матвеев Алексей Сергеевич, ст. преп. каф. ПВ ФИТ

20.06.2025

Новосибирск, 2025