

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)
Факультет информационных технологий
Кафедра параллельных вычислений

09.03.01 Информатика и вычислительная техника. Программная инженерия и
компьютерные науки

Разработка и реализация алгоритма распределения ресурсов фрагментированных программ на основе профилирования

Выполнил: Лямин Артём Сергеевич, студент группы 17202 ФИТ НГУ

Научный руководитель: Маркова Валентина Петровна, д.т.н., доцент каф. ПВ ФИТ НГУ

Соруководитель: Перепёлкин Владислав Александрович, ст. преп. каф. ПВ ФИТ НГУ.

Рецензент: Григорьева Ирина Владимировна, к.ф.-м.н., н.с. ИВТ СО РАН

23.06.2021

Новосибирск 2021

Введение в предметную область

Система автоматического конструирования программ - система, которая строит программу автоматически по её описанию на относительно высоком уровне абстракции (Charm++, PaRSEC, Legion, LuNA).

Системы автоматического конструирования программ:

- Упрощают работу программиста
- Позволяют избегать ошибок

Эффективность таких систем, в частности, зависит от эффективности алгоритмов балансировки нагрузки по узлам вычислительной системы.

Алгоритмы балансировки нагрузки

Алгоритмы балансировки можно разделить на:

- **Статические** - конструирующие распределение на этапе подготовки к исполнению программы.
- **Динамические** - корректирующие распределение во время исполнения программы.
- **На основе профилирования** - корректирующие распределение с учётом профиля на этапе подготовки к исполнению программы.

Актуальность

С точки зрения оптимизации профилирование является ценным источником информации. Поэтому актуальна разработка алгоритмов, которые этот потенциал будут использовать.

Цель работы

Цель работы – разработка и реализация алгоритмов автоматической балансировки нагрузки на основе профилирования.

Задачи для достижения цели:

- Анализ систем автоматического конструирования параллельных программ и их средств профилирования, возможности применения алгоритма;
- Разработка модели профиля программы;
- Программная реализация;
- Экспериментальное исследование алгоритма.

Системы автоматического конструирования параллельных программ

Чтобы в систему было возможно встроить балансировщик нагрузки на основе профилирования, необходимо:

- Наличие средств профилирования или возможность их добавления;
- Возможность распределять задачи, не меняя логику программы, используя при этом результат работы алгоритма автоматической балансировки нагрузки на основе профилирования;
- Возможность анализировать производительность программы.

Системы автоматического конструирования параллельных программ

Система	Преимущества	Недостатки
Charm++	Программа представлена множеством структурных единиц, называемых чарами (chare); RunTime-система; Средства сбора профилировочной информации.	Ограниченное управление исполнением
PaRSEC	Runtime-система; Функционал по получению профиля и его визуализации.	Ограниченность применимости классом задач типа линейной алгебры
Legion	Программа состоит из регионов (region) - совокупность данных, которые являются локальными и независимыми от других регионов; Средства сбора профиля.	Необходимость писать управляющий код самостоятельно
LuNA	Программа имеет фрагментированную структуру; RunTime-система; Средства профилирования; Управляющий код может генерироваться системой.	Плохо проработана оптимизация на основе профилирования

LuNA

- LuNA (Language for Numerical Algorithms) – система фрагментированного программирования, разрабатываемая в ИВМиМГ СО РАН;
- Task based система с крупноблочным параллелизмом;
- Задачи (task) – **фрагменты вычислений(ФВ)**, независимые единицы программы, которые содержат описание входных/выходных фрагментов данных и кода фрагмента;
- **Фрагмент данных** – агрегат из переменных.
- RunTime-система LuNA управляет исполнением программы и позволяет логировать информацию, связанную с исполнением фрагментов вычислений.

Постановка задачи

Задача автоматического распределения фрагментов вычислений сводится к считыванию логированной RunTime-системой информации, получению профиля программы и нахождению корректирующего распределения фрагментов вычислений на узлы мультимпьютера, при котором эффективность программы повышается.

Разработка

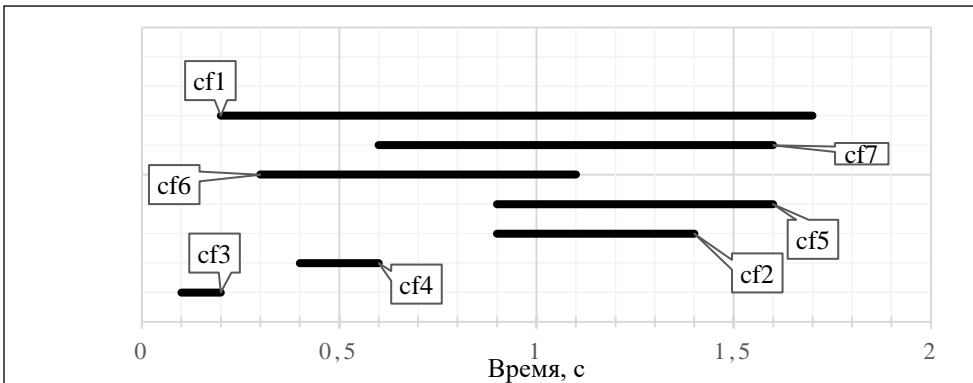
Разработка. Сбор профиля

Для профилирования необходима информация, связанная фрагментами вычислений:

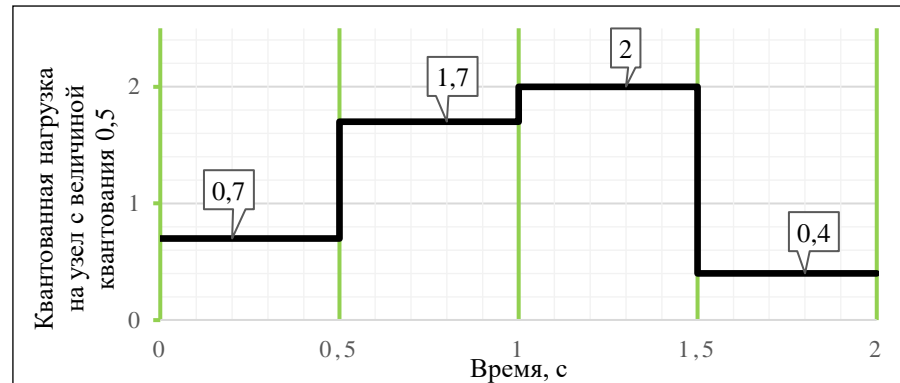
- Время и узел, на котором ФВ был создан;
- Информация о миграциях ФВ между узлами;
- Время и узел, на котором ФВ исполнялся.

Используя эти данные, можно восстановить ход исполнения каждого отдельного фрагмента вычислений.

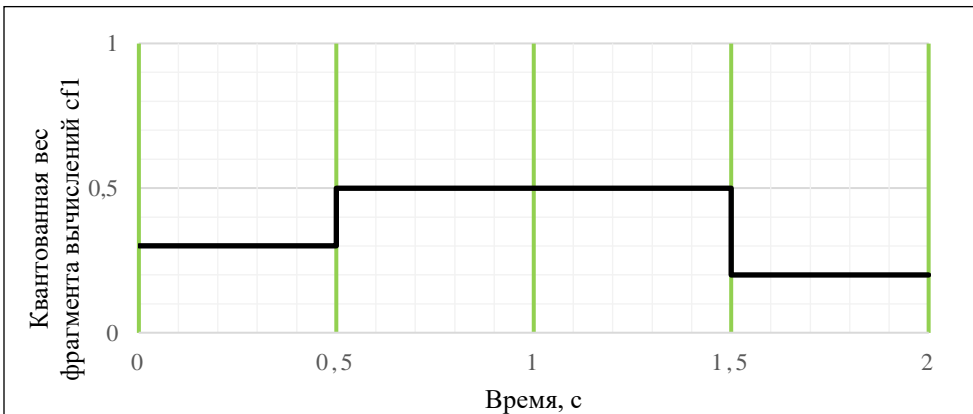
Разработка. Определение нагрузки на узлы



Отрезки исполнения ФВ



Определение квантованной нагрузки на узел



Определение квантованного веса ФВ cf1

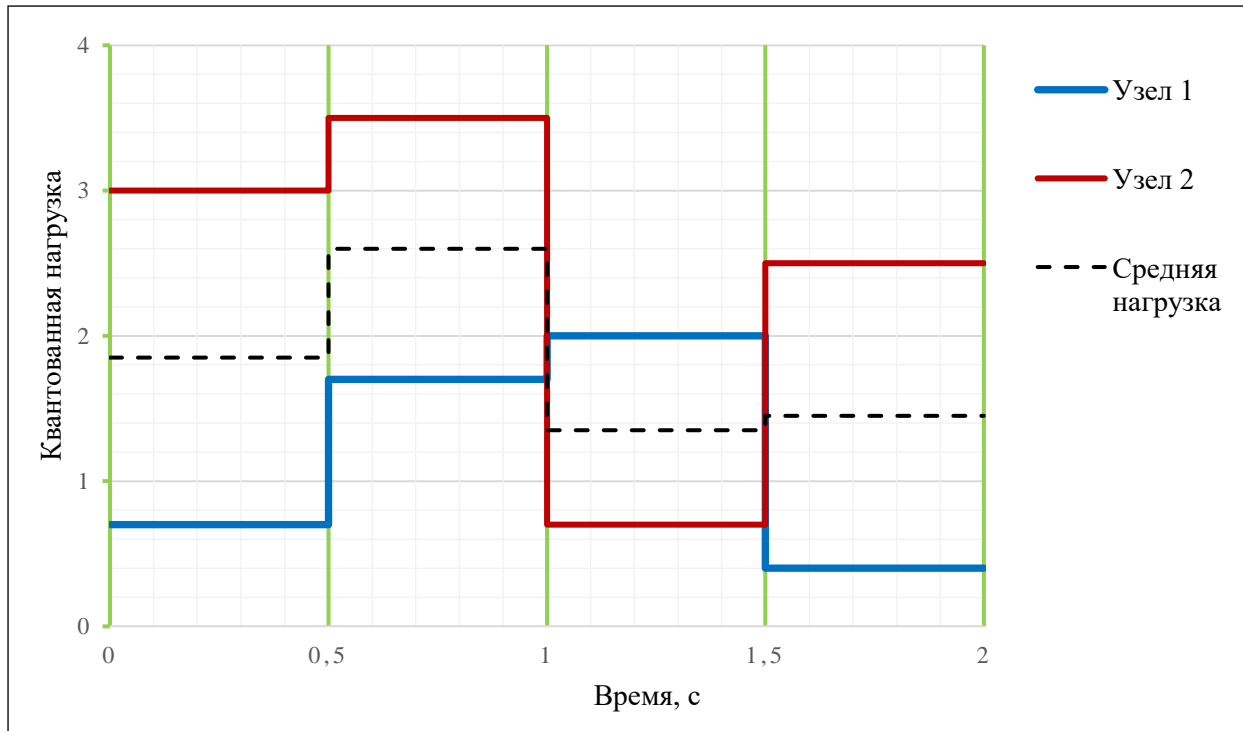
q – величина квантования;
 $N_q = \left\lceil \frac{t_{last}}{q} \right\rceil$ – число квантованных промежутков;
 Квантованные промежутки:
 $x_i = [q * i ; q * (i + 1))$, где $0 \leq i < N_q$;
 Квантованный вес ФВ: $w_{cf,j} = \|x_j \cap [t_b ; t_e]\|$;
 Квантованная нагрузка на узел: $W_{i,j} = \sum_{cf \in CF_i} w_{cf,j}$

Разработка. Определение дисбаланса нагрузки

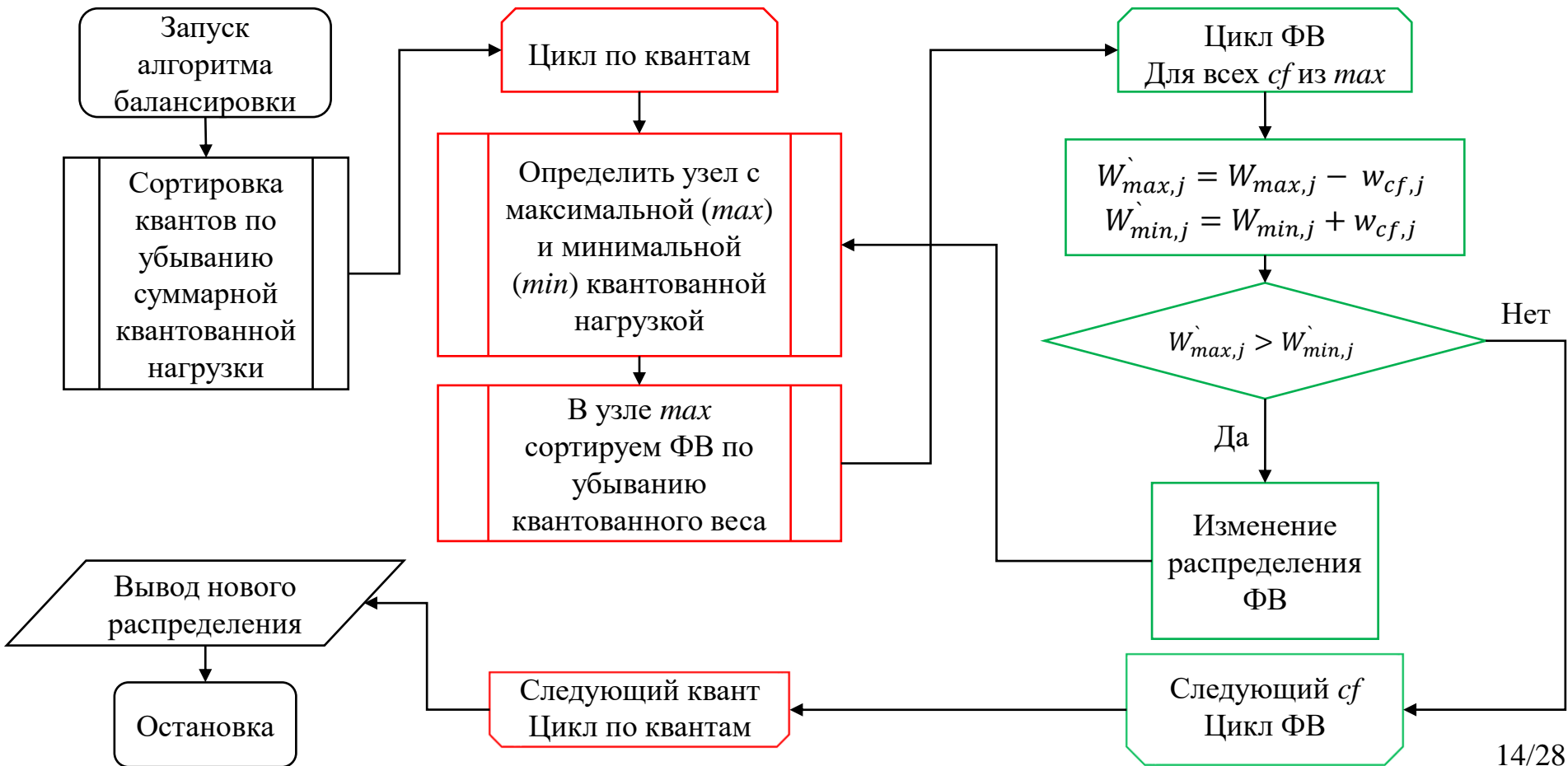
Для определения дисбаланса считается средняя квантованная нагрузка в каждый квант:

$$W_{avg,j} = \frac{\sum_{i \in N} W_{i,j}}{\|N\|}$$

$\|N\|$ – мощность
множества
вычислительных узлов
системы.



Разработка. Определение перемещаемых фрагментов вычислений



Реализация

Реализация. Средства профилирования LuNA

```
0 | 0.851619 | Invoke_start CF 139815815236640 (9_2_8_8_6) : 13 | ./src/rts/
0 | 0.869083 | Invoke_finish CF 139815815236640 (9_2_8_8_6) : 13 0.016414 | .
```

Номер узла

Момент времени

Тип события

Глобальный идентификатор
фрагмента вычислений

Вес фрагмента
вычислений

В результате логи содержат всю необходимую информацию для работы алгоритма балансировки:

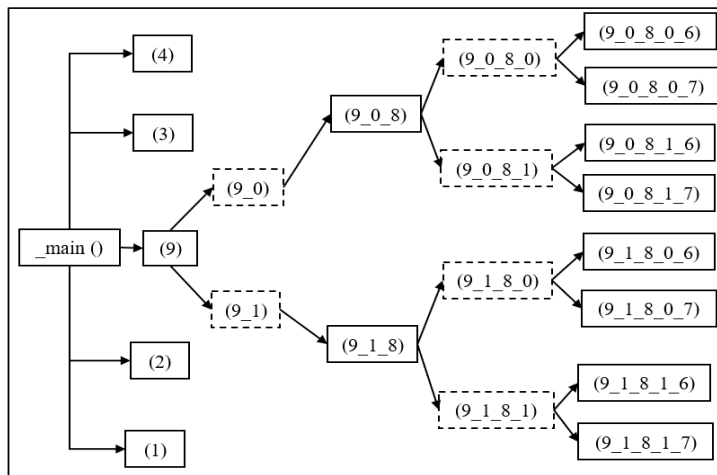
- **Номере узла**, на котором произошло событие;
- **Момент времени** от начала исполнения программы;
- **Тип события**: Invoke_start – начало исполнения, Invoke_finish – конец исполнения;
- **Глобальный идентификатор** – уникальный идентификатор ФВ;
- Для события Invoke_finish – вес фрагмента вычислений.

Реализация. Средства профилирования LuNA. Глобальный идентификатор

Ранее существовал только локальный идентификатор. Проблема локального идентификатора – не идемпотентность.

Глобальный идентификатор – вектор целых чисел, соответствующий стеку вызовов (порождений).

При использовании операторов `for` и `while` к глобальному идентификатору добавляется номер итерации.



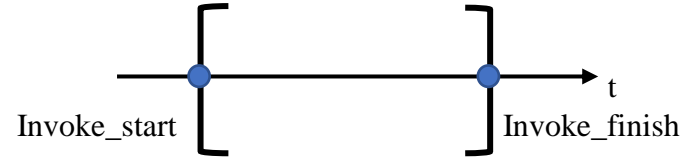
Пример дерева порождений фрагментов вычислений

Реализация. Средства профилирования LuNA.

Изменения в системе логирования

Ранее промежуток исполнения ФВ вычислялся как:

$[Invoke_start.time; Invoke_Finish.time]$.



Для более точного определения промежутка исполнения ФВ, во время исполнения производится замер времени, которое выводится при событии `Invoke_finish` в лог-файл. Тогда отрезок исполнения определяется:

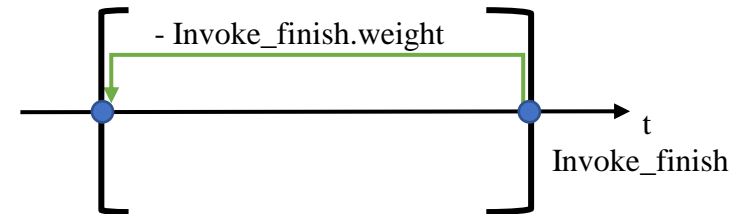
$[Invoke_Finish.time - Invoke_Finish.weight; Invoke_Finish.time]$

```
struct timespec t0, t1;

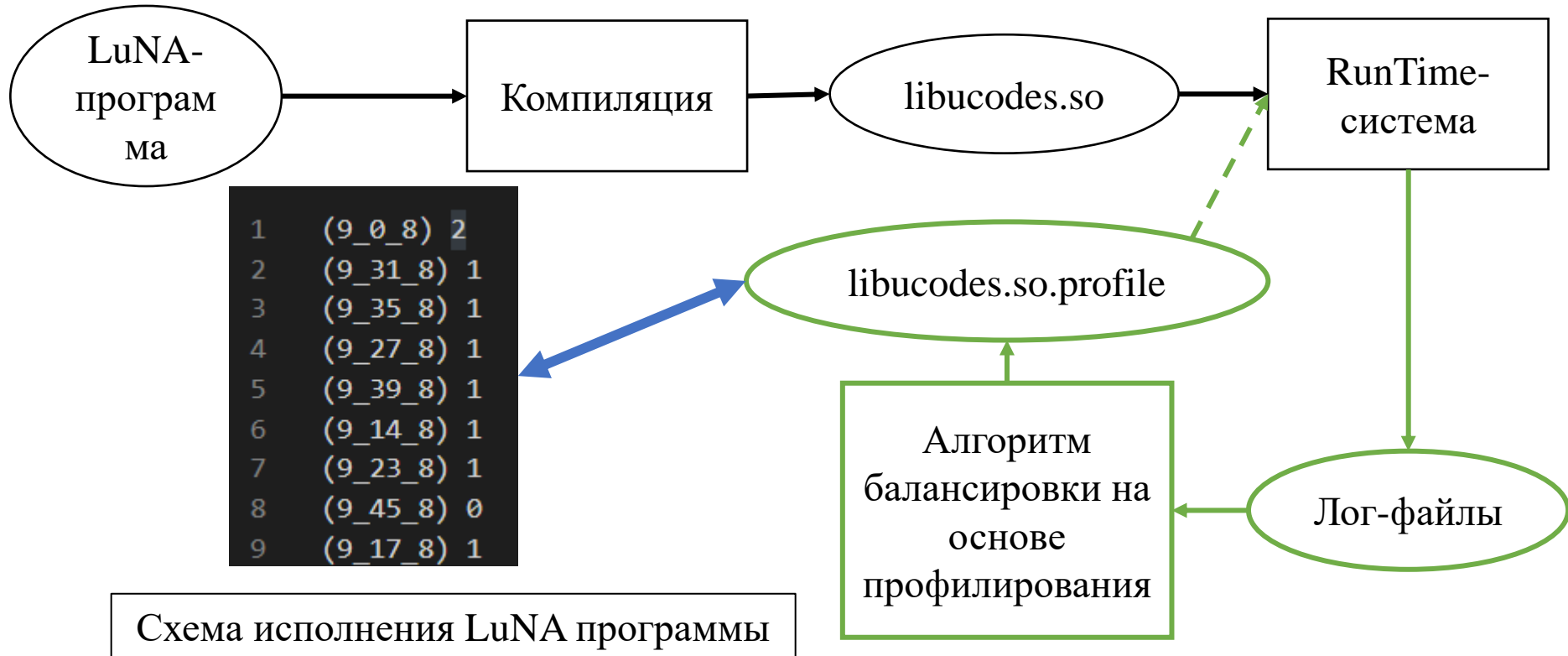
auto name = cf->to_string();
LOG("Invoke_start " + name)
clock_gettime(CLOCK_MONOTONIC, &t0);
```

```
clock_gettime(CLOCK_MONOTONIC, &t1);
double elapsed_time = t1.tv_sec - t0.tv_sec;
elapsed_time += (t1.tv_nsec - t0.tv_nsec) * 1e-9;
```

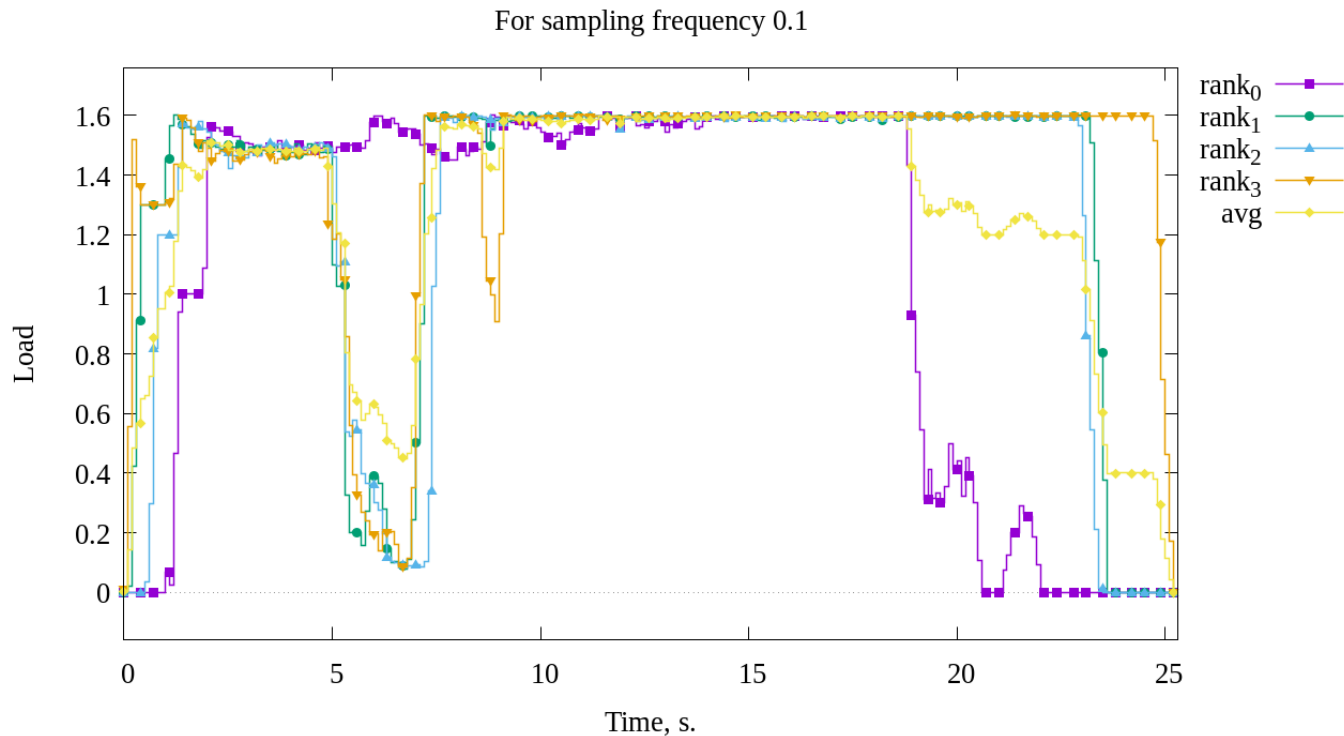
```
LOG("Invoke_finish " + name + " " + std::to_string(elapsed_time))
```



Реализация. Поддержка созданного распределения в RunTime-системе



Реализация. Построение графиков квантованной нагрузки



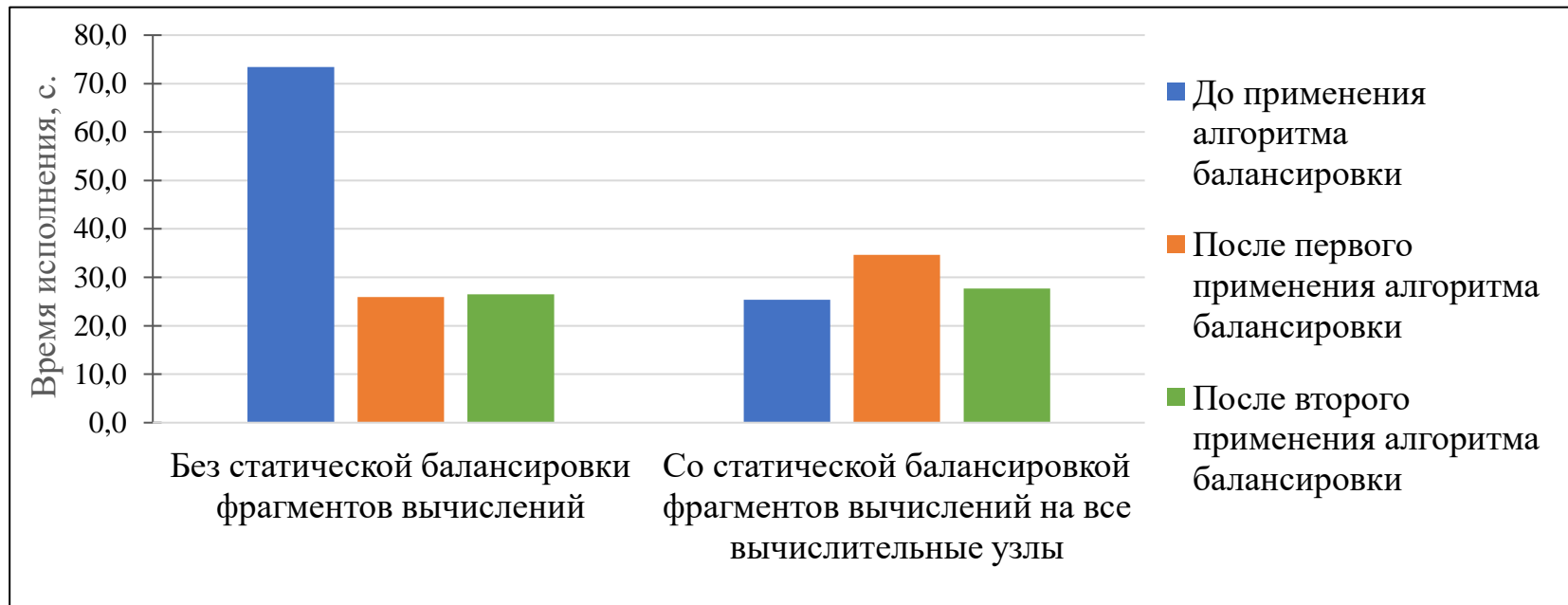
Тестирование

- Тестирование проводилось на кластере MVS-10p МСЦ РАН.
- Для проверки эффективности работы алгоритма балансировки были проведены тесты на 2-х программах на языке LuNA:
 - Прибавление единицы ко всем элементам матрицы;
 - Решение 3-х мерного уравнения теплопроводности методом конвейерной прогонки.
- Цель тестирования – оценить влияние алгоритма балансировки на скорость работы программы.

Результаты тестирования

В данном тесте использовалась программа, осуществляющая добавление единицы к каждому элементу матрицы.

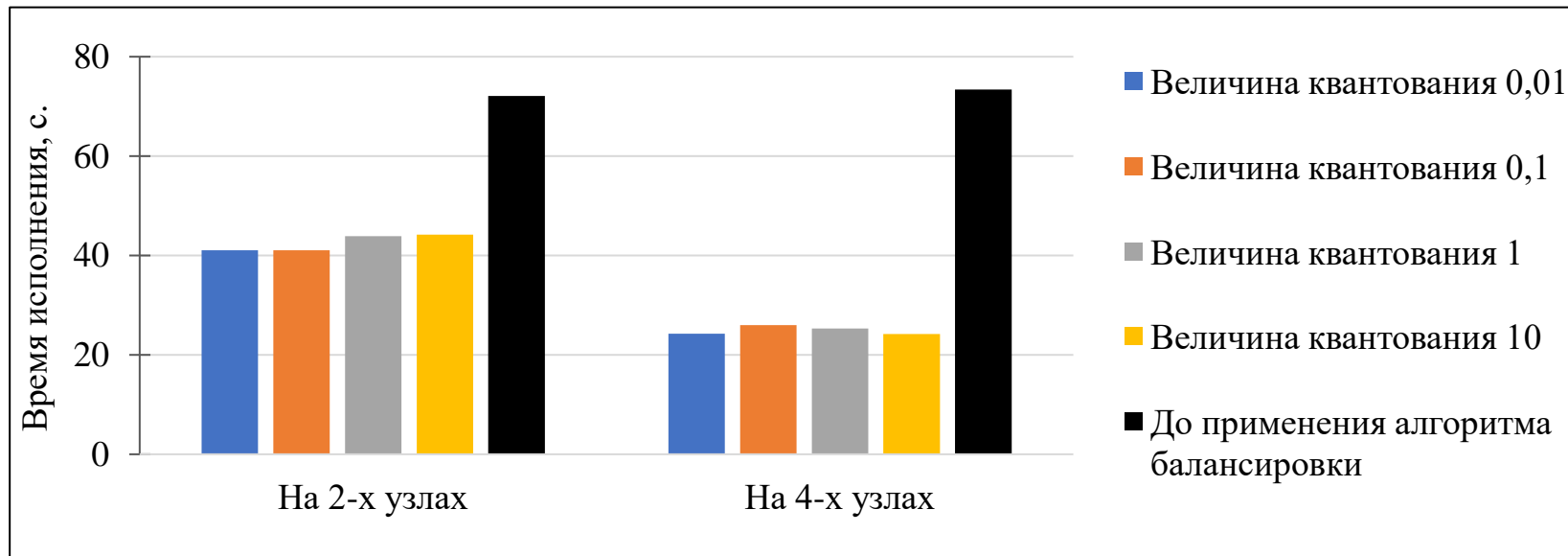
Цель теста: проверить эффективность алгоритма на программах с различным статическим распределением



Результаты тестирования

В данном тесте использовалась программа, осуществляющая добавление единицы к каждому элементу матрицы.

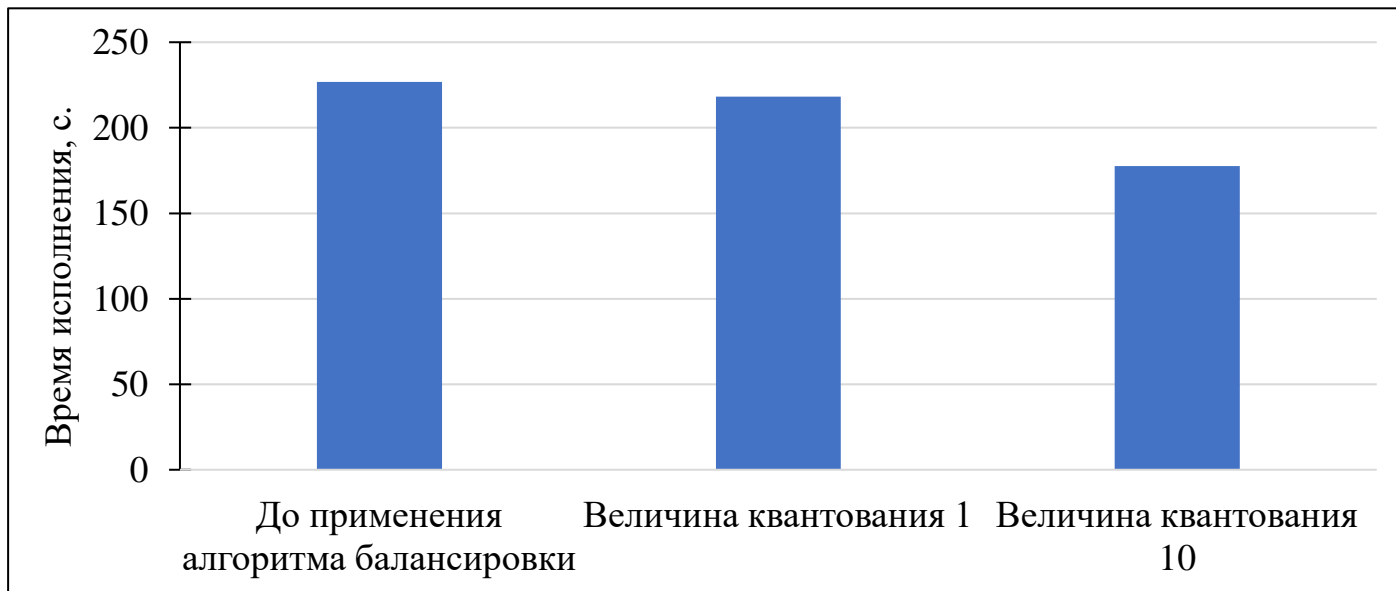
Цель теста: проверить, как влияет величина квантования на эффективность распределения.



Результаты тестирования

В данном тесте использовалась программа решения трёхмерного уравнения теплопроводности.

Цель теста: проверить эффективность алгоритма на реальной задаче.



Заключение

- В ходе выполнения работы был разработан и реализован алгоритм автоматического распределения ресурсов фрагментированных программ на основе профилирования в системе LuNA. Применение алгоритма улучшило производительность LuNA-программ.
- **Защищаемые положения:**
 - Разработан и реализован алгоритм балансировки нагрузки на основе профилирования в системе LuNA;
 - Проведено экспериментальное исследование работоспособности алгоритма.

Заключение

- Дальнейшие планы:
 - Разработка новых алгоритмов устранения дисбаланса;
 - Разработка рекомендаций по автоматической балансировке на основе профилирования с возможностью указывать величину квантования;
 - Разработка эвристик на основе профиля для оптимального выбора величины квантования.

Апробация работы

Данная работа была опубликована на 59-й Международной научно-студенческой конференции, г. Новосибирск, 2021 г. и была награждена дипломом второй степени.

Публикация по теме:

Лямин, А. С. Разработка и реализация алгоритма распределения ресурсов фрагментированных программ на основе профилирования / А. С. Лямин // Информационные технологии : Материалы 59-й Междунар. науч. студ. конф. 12–23 апреля 2021 г. / Новосиб. гос. ун-т. — Новосибирск : ИПЦ НГУ, 2021. — С. 114.

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет информационных технологий

Кафедра параллельных вычислений

09.03.01 Информатика и вычислительная техника. Программная инженерия и
компьютерные науки

Разработка и реализация алгоритма распределения ресурсов фрагментированных программ на основе профилирования

Выполнил: Лямин Артём Сергеевич, студент группы 17202 ФИТ НГУ

Научный руководитель: Маркова Валентина Петровна, д.т.н., доцент каф. ПВ ФИТ НГУ

Соруководитель: Перепёлкин Владислав Александрович, ст. преп. каф. ПВ ФИТ НГУ.

Рецензент: Григорьева Ирина Владимировна, к.ф.-м.н., н.с. ИВТ СО РАН

23.06.2021

Новосибирск 2021