

# Реализация динамической балансировки нагрузки во фрагментированных программах методом WorkStealing

Выполнили: Тренин С.А, Чмиль А.В.  
Руководитель: Перепёлкин В.А.

ФИТ НГУ 2019  
Кафедра параллельных вычислений.

# Содержание

- 1) Введение
- 2) Метод WorkStealing
  - 1) Основы
  - 2) Проблемы
- 3) Выбор среды разработки
- 4) Предлагаемое решение
- 5) Пример
- 6) Тестирование
  - 1) Параметры
  - 2) Результаты
- 7) Заключение

# Введение

Проблема — дисбаланс вычислительной нагрузки при исполнении фрагментированных программ в силу неоднородности конфигурации, динамики явления и т. д.

Решение — применение динамической балансировки вычислительной нагрузки.

# Метод WorkStealing (основы)

Суть метода — недогруженные (или простаивающие) узлы запрашивают работу у остальных узлов. Те, в свою очередь, принимают решение о выделении работы, основываясь на своей загруженности.

Работа — фрагмент вычислений без побочных эффектов и с заданными параметрами.

# Метод WorkStealing (проблемы)

Проблемы метода:

- 1) Избыточные коммуникации могут значительно замедлить исполнение.
- 2) Недостаточные коммуникации также замедляют исполнение.
- 3) При бесконтрольной передаче запросов на получение работы, узлы будут передавать данные по кругу.

# Выбор среды разработки

В качестве среды разработки была выбран язык автоматизации генерации фрагментированных программ LuNA, который подходит для нашей задачи по ряду причин:

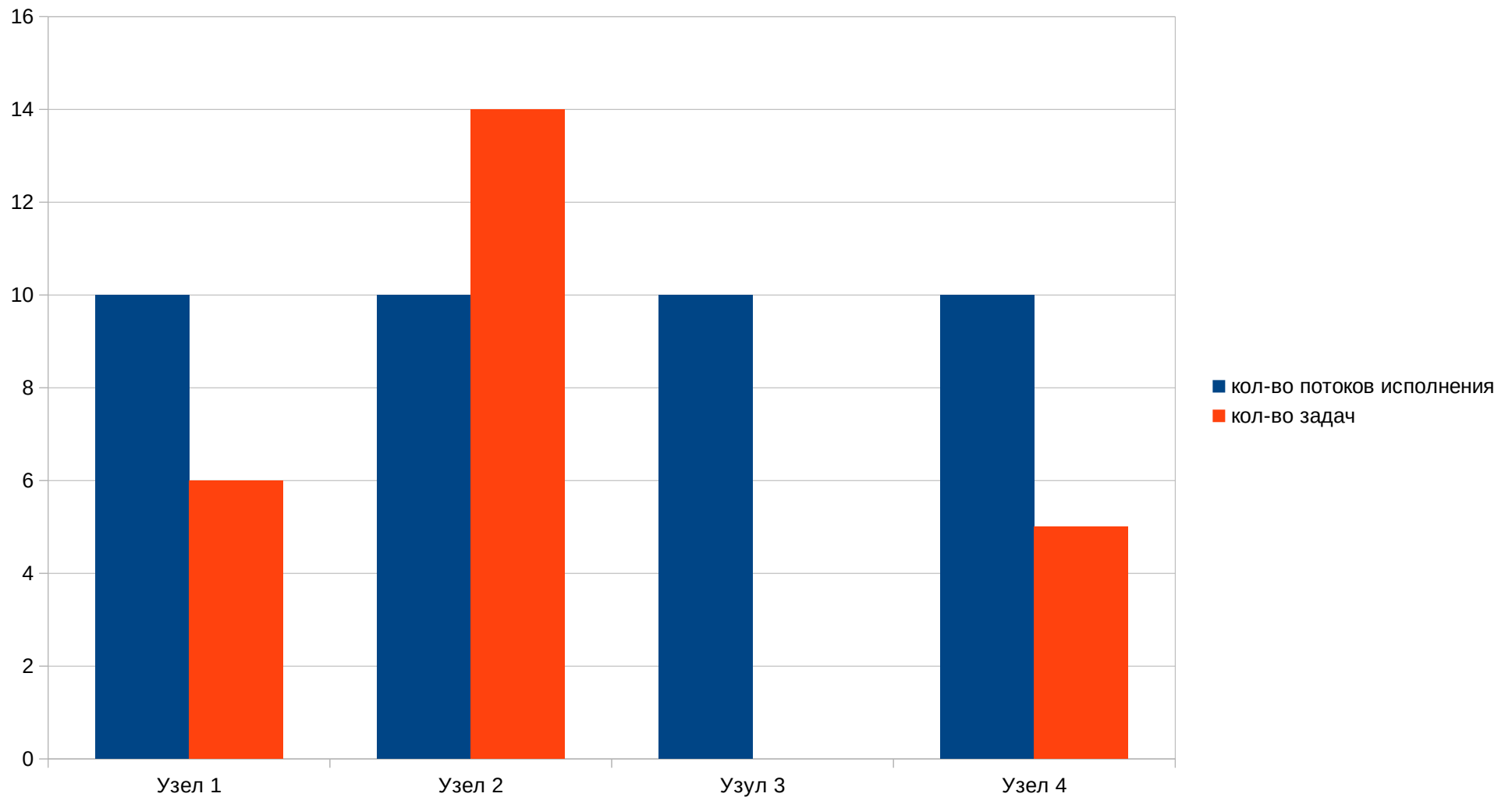
- 1) Вынесенное отдельно поведение, позволяет подключать балансировку, не затаргивая код программы.
- 2) Дописав балансировку мы добавим её для целого класса задач, а не для одной конкретной.
- 3) Удобство тестирования и дальнейших исследований на целом классе задач.

# Предлагаемое решение

Детали реализации:

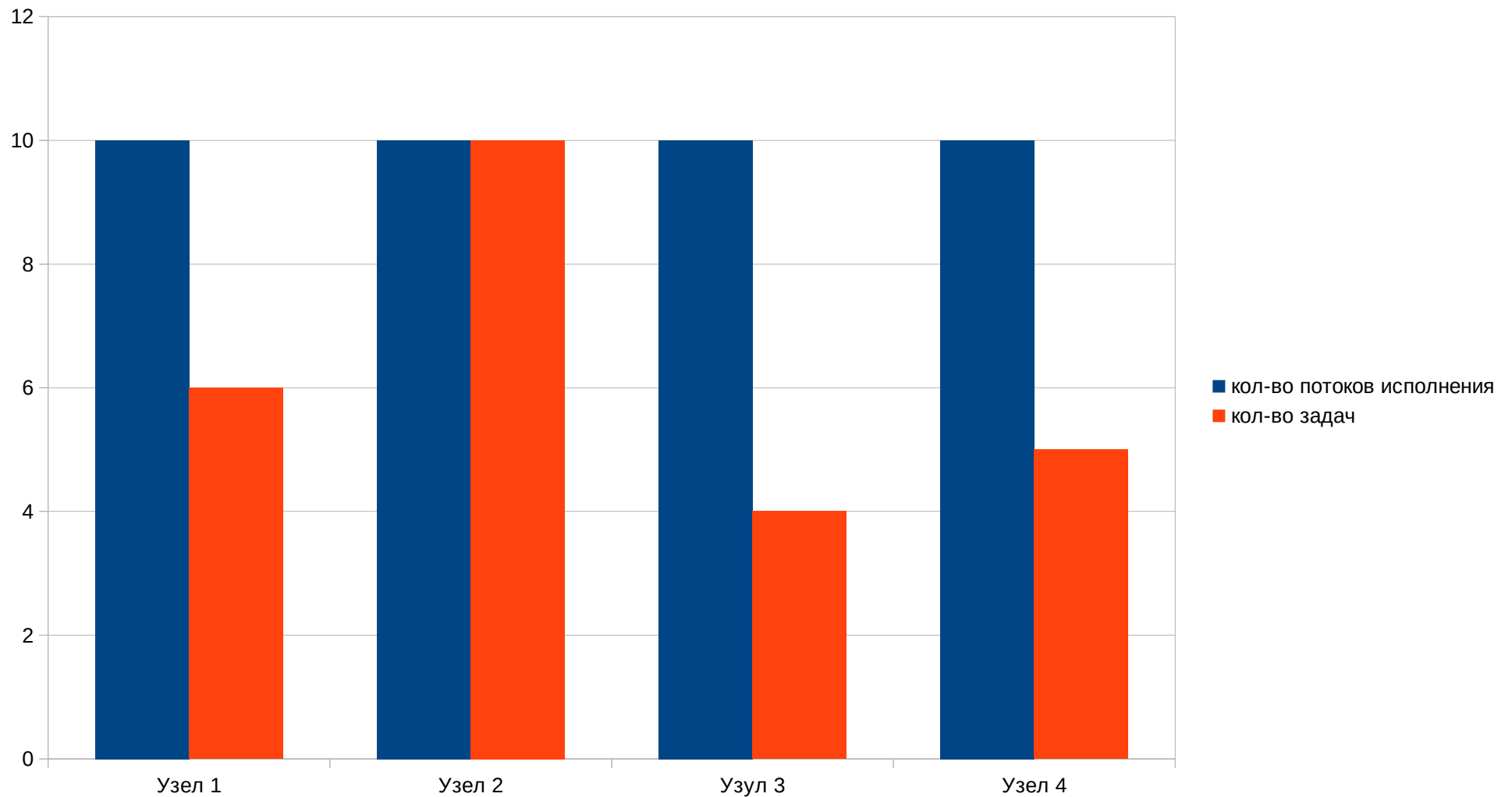
- 1) Узел отправляет запрос на получение работы, только если он простаивает (вообще не загружен работой).
- 2) Запрос отправляется только на соседние узлы.
- 3) При получении запроса узел передаёт работу, только если число задач на нём больше, чем количество потоков исполнения (он перегружен).

# Пример (без балансировки)





# Пример (с балансировкой)



# Тестирование (параметры)

Для тестирования была взята программа на языке LuNA, решающая трёхмерную задачу Пуассона с декомпозицией на «линейке».

Запуск производился на кластере МВС-10П ОП Межведомственного Суперкомпьютерного Центра.

Задача запускалась на 4 вычислительных узлах с 2 потоками исполнения на каждом. При этом, чтобы наглядно продемонстрировать эффект динамической балансировки, один из узлов изначально не был загружен работой.

# Тестирование (результаты)

Использование WorkStealing	Циклов исполнения задачи	Размер одного фрагмента вычислений	Общее число фрагментов	Время работы / Идеальное время работы
Нет	1	8x64x960	32	333.081413 sec./ 249.75 sec.
Да	1	8x64x960	32	558.757576 sec./ 249.75 sec.
Нет	250	8x64x960	32	104.544873 sec./ 78.00 sec.
Да	250	8x64x960	32	85.281108 sec./ 78.00 sec.
Нет	1000	16x64x960	32	310.256728 sec./ 232.50
Да	1000	16x64x960	32	228.624841 sec./ 232.50

# Заключение

Удалось добавить к run-time системе LuNA возможность включать при исполнении задачи динамическую балансировку WorkStealing. На данный момент ускорение на кластере происходит только при большом числе циклов исполнения задачи. В силу больших задержек коммуникации.

В дальнейшем планируется улучшать метод балансировки, опираясь на полученные результаты профилирования. В первую очередь, следует продолжить избавляться от излишних запросов и прочих коммуникаций.