Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра суперкомпьютеров и квантовой информатики

Разработка подхода к реализации альтернативного планировщика для менеджера ресурсов "SLURM"

Кислов Евгений Витальевич
423 группа

Научный руководитель
к.ф.-м.н., в.н.с. НИВЦ МГУ
Жуматий С.А.

Москва, 2022

Введение

- На данный момент на суперкомпьютере “Ломоносов-2” используется менеджер ресурсов SLURM.
- Преимущества SLURM: масштабируемость и использование по умолчанию алгоритма планирования Backfill.
- Недостатки SLURM:
  - Ограниченность в параметрах алгоритмов.
  - Отсутствует возможность выставлять лимиты или приоритеты по произвольным признакам.
  - Отсутствуют расширенные обработчики событий.

Постановка задачи

Разработать подход к реализации альтернативного планировщика для менеджера ресурсов SLURM, позволяющего:

- Добавлять алгоритмы планирования или последовательность разных алгоритмов планирования и выбирать, какие алгоритмы использовать для определенной очереди.
- Задавать в алгоритмах планирования приоритеты и лимиты по любым параметрам.
- Предоставлять выделенной группе пользователей возможность назначать лимиты и приоритеты на постановку задач, а также просматривать статистику очереди за любой промежуток времени.
- Добавлять обработчики на произвольные события планировщика.
- Добавлять способы сбора статистики.
- Использовать внутренние механизмы SLURM для исполнения задач.

Описание SLURM

- SLURM - это отказоустойчивая система управления кластерами и планирования заданий для больших и малых кластеров Linux.
- Используется многими суперкомпьютерами из рейтинга Top500, включая половину суперкомпьютеров в top 10.
- Предоставляет пользователям доступ к ресурсам на определенный период времени.
- Обеспечивает основу для запуска, выполнения и мониторинга работы на наборе выделенных узлов.
- Использует протокол wiki2 для взаимодействия с внешними планировщиками.

Обзор аналогов

- **SLURM spank plugins** - не имеет достаточного функционала.
- **MAUI** - стал коммерческим, не подходит из-за высокой стоимости.
- **lpSched** - перестал поддерживаться, не поддерживает нужного функционала.

Таким образом, **аналогичных решений найдено не было**.

Сравнение подходов - внешний или внутренний планировщик

- **Подход 1** - писать плагин в существующий код внутри SLURM.
- **Подход 2** - писать независимое приложение, взаимодействующее с SLURM по сетевому протоколу.

Недостатки первого подхода:

- Любые изменения в ядре SLURM приведут к тяжело поддерживаемому коду.
- Многие внутренние механизмы SLURM будут недоступны.
- Разработка и поддержка кода на C сложнее для разработчиков, чем более высокоуровневый язык.

Таким образом, был **выбран подход 2**.

Архитектура системы

- Разработана модульная архитектура, состоящая из 3 основных компонентов:
  - Компонент логики внешнего планировщика.
  - Компонент взаимодействия с SLURM.
  - Компонент взаимодействия с СУБД.

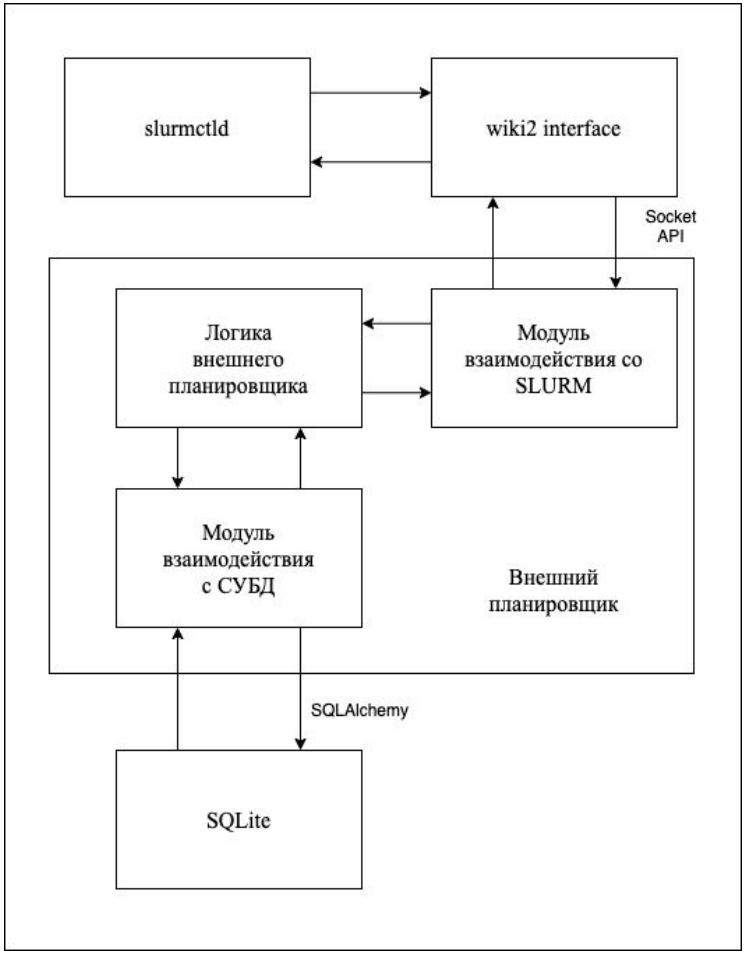

Рис. 1. Архитектура разработанного планировщика

Компонент взаимодействия с SLURM

- Отвечает за взаимодействие с менеджером ресурсов SLURM.
- Предоставляет интерфейс остальным компонентам системы для подписки на события и постановки задачи в очередь.
- Содержит цикл с опросом событий от SLURM через неблокирующий socket-сервер.

Компонент логики внешнего планировщика

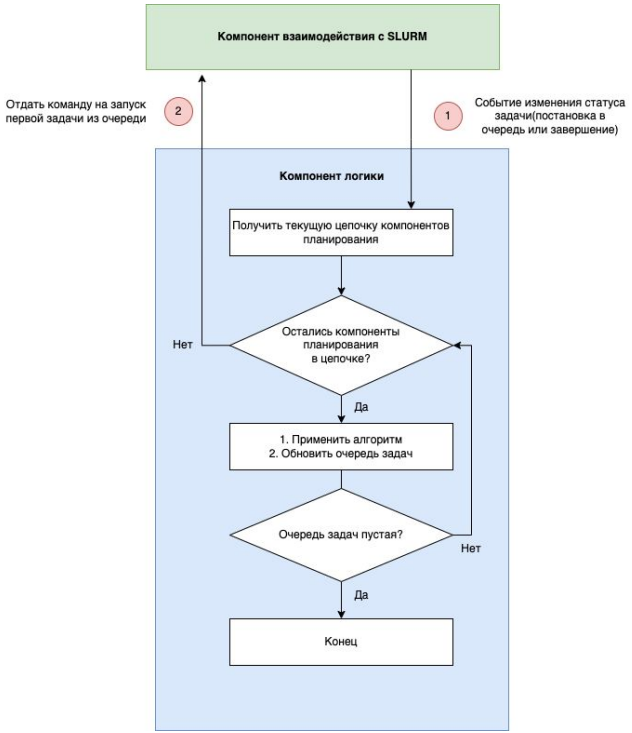

Рис. 2. Блок-схема, описывающая работу цепочки алгоритмов

- Позволяет задать цепочку алгоритмов планирования, которые будут выполняться в момент, когда планировщик начнет обрабатывать новую задачу.
- Позволяет подписаться на события планировщика - передав функцию, которая принимает на вход тип события и содержание.

Условия апробации

- Была использована архивная выборка задач с суперкомпьютера Ломоносов-2 за февраль 2022 года, состоящая из 7500 задач.

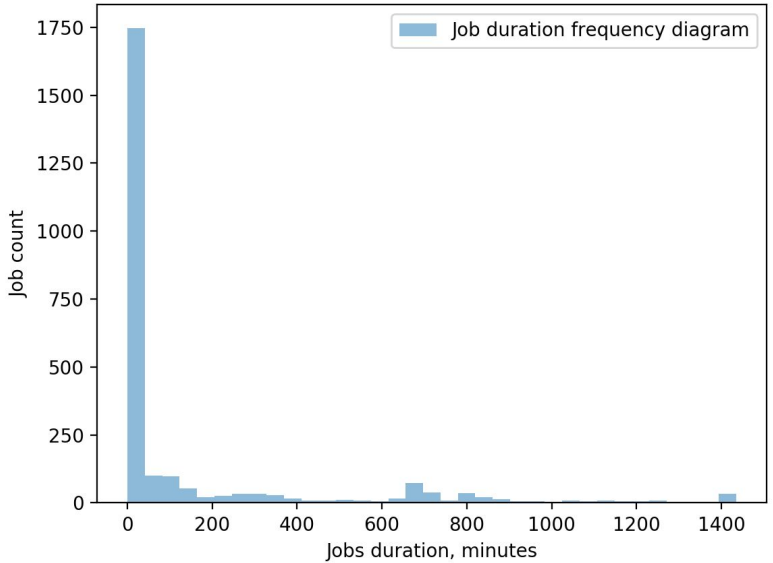

Рис 4. Распределение времени выполнения задачи.

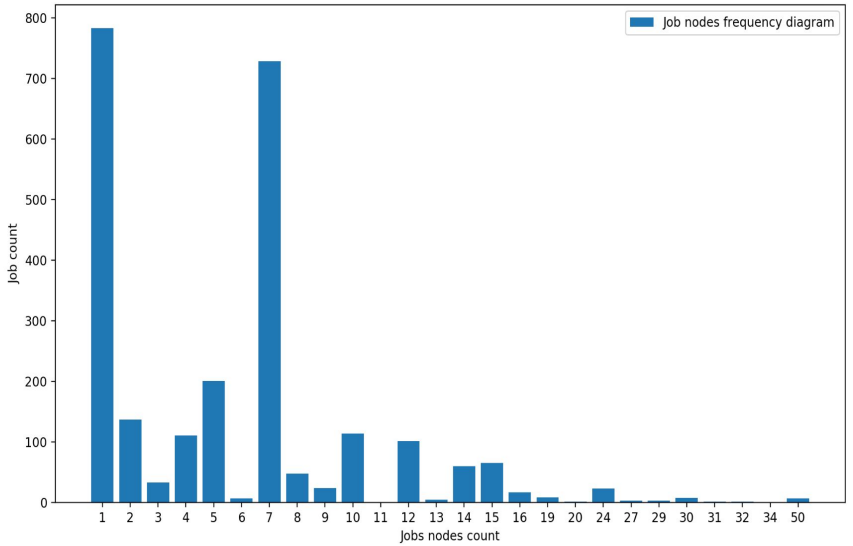

Рис 5. Распределение количества занимаемых задачей узлов.

Условия апробации

- Параметры задач: количество занимаемых задачей узлов - от 1 до 50, время запуска от 1 секунды до 400 минут.
- Для того, чтоб эмулировать различное время запуска, использовался bash-скрипт, вызывающий функцию sleep.
- Было запущено 280 задач в течение 1.5 часа с равным интервалом в 20 секунд.
- С целью более точного сравнения работы встроенного в SLURM планировщика backfill и разработанного внешнего планировщика были сгенерированы единые параметры для обоих случаев.

Результаты апробации

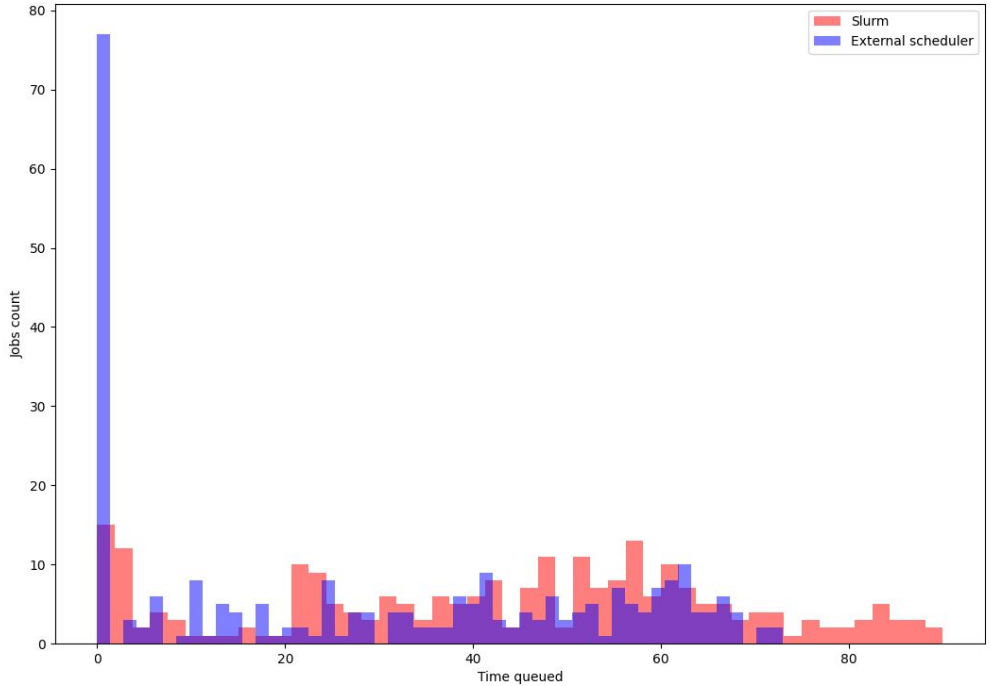

Рис 6. Сравнительная диаграмма частот по времени ожидания задачи в очереди.

Анализ **времени ожидания задачи в очереди**:

Встроенный алгоритм backfill

- Среднее - 42 секунды
- Медиана - 25 секунд
- 95-й перцентиль - 83 секунды

Разработанный планировщик:

- Среднее - 29 секунд
- Медиана - 29 секунд
- 95-й перцентиль - 66 секунд

Результаты

- Разработан подход к реализации планировщика, использующий внутренние механизмы менеджера ресурсов SLURM, позволяющий добавлять произвольные алгоритмы планирования и обработчики событий.
- На базе реализованного подхода разработан модифицированный алгоритм backfill.
- Прототип апробирован на исторических данных, взятых с суперкомпьютера Ломоносов-2.

Предлагаемый алгоритм планирования

```
nodes_coeff = 1
queued_time_coeff = 2.5
max_time_coeff = 1

nodes_fit = 100.0 * task_nodes / nodes_available
queued_time = current_time - task_queue_time

priority =
  nodes_coeff * nodes_fit +
  queued_time_coeff * nodes_fit +
  max_time_coeff + max_time
```