

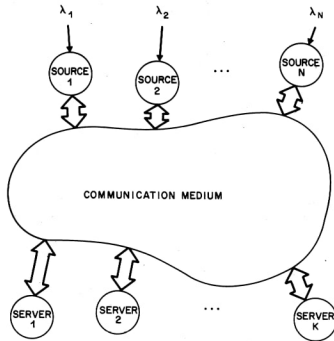
Модификация балансировщика нагрузки для распределенной вычислительной системы с высокой степенью гетерогенности

Дадашев Р.Р., МФТИ
Курочкин И.И., ИППИ РАН

Москва, 2025

Что такое распределенная вычислительная система?

Гетерогенные распределенные системы – это распределенные вычислительные системы, которые характеризуются разнородностью составляющих их элементов. Разнородность может проявляться в различных аспектах: аппаратное обеспечение, программное обеспечение и др.



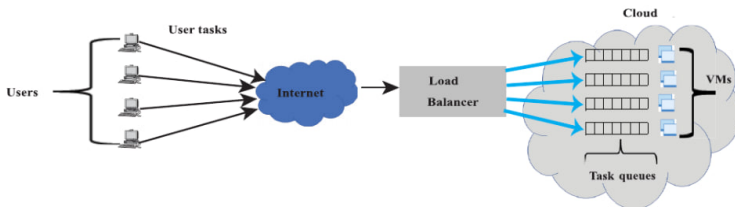
* David Anderson. *"Emulating Volunteer Computing Scheduling Policies"*, 2011, IEEE International Symposium on Parallel Distributed Processing

Базовые задачи балансировщика

- В конечном итоге завершить все задания.
- Избегать простоя ресурсов обработки.
- Завершать задания к их крайнему сроку, тем самым избегая создания дополнительных экземпляров задания.
- Использовать наиболее производительную версию приложения для данного задания и сервера.
- Обеспечить передачу заданий исполнителю с учетом вычислительных предпочтений.

Зачем нужны балансировщики нагрузки?

- Предотвращение перегрузки отдельных вычислительных узлов.
- Улучшение производительности (распределение нагрузки).
- Реплицирование задач (отказоустойчивость).

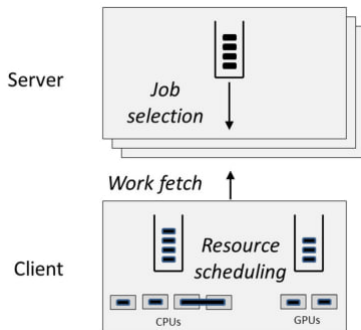


Цель: повысить эффективность использования вычислительных ресурсов, сократив время простоя.

* Stavros Souravlas, Sofia Anastasiadou, Nicoleta Tantalaki, Stefanos Katsavounis. "A Fair, Dynamic Load Balanced Task Distribution Strategy for Heterogeneous Cloud Platforms Based on Markov Process Modeling", 2022, IEEE Access

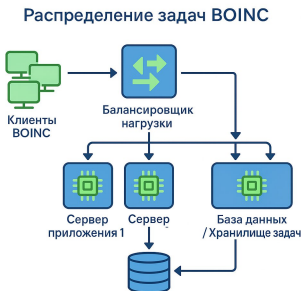
Ограничения балансировщика

- Клиент BOINC поддерживает очередь заданий и выбирает когда запрашивать новые задания.
- Оценки времени выполнения по своей сути неопределенны. Потому что вычислительный ресурс может отказать в любой момент.
- Выбор задания - это выбор сервером того, какие задания отправлять в ответ на запрос клиента о выполнении работы.



* David Anderson. *"BOINC: A Platform for Volunteer Computing"*, 2019, J Grid Computing

Краткое описание архитектуры BOINC



- Основные компоненты: Сервер приложений, База данных, Планировщик, Клиенты.
- Неоднородные, непостоянно доступные ресурсы.
- Планировщик = одно из узких мест → роль балансировщика нагрузки.

* Stavros Souravlas, Sofia Anastasiadou, Nicoleta Tantalaki, Stefanos Katsavounis. "A Fair, Dynamic Load Balanced Task Distribution Strategy for Heterogeneous Cloud Platforms Based on Markov Process Modeling", 2022, IEEE Access

Обзор подхода

- Рассматривать планировщик как динамический балансировщик нагрузки.
- Использовать симулятор **ComBoS** (Complete BOINC Simulator) для прототипирования и тестирования политик в автономном режиме.
- Сосредоточиться на минимизации времени оборота задачи и времени простоя хоста.

Среда моделирования BOINC

ComBoS (Complete BOINC Simulator) – симулятор, разработанный для моделирования реалистичных сценариев с учетом всей инфраструктуры BOINC.

- **SimBOINC** не моделирует серверную часть и сеть, только клиентскую часть.
- **SimBA** (Simulator of BOINC Applications) не может моделировать проекты с сотнями тысяч клиентов или моделировать разные проекты одновременно (в нем не реализован клиентский планировщик).
- **EmBOINC** не учитывает важные аспекты проектов, такие как продолжительность выполнения задач или размер файлов, в дополнение к этому, является эмулятором, а не симулятором.

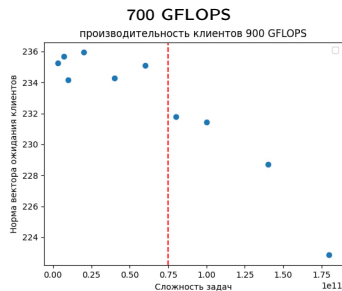
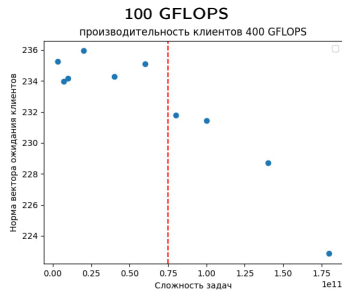
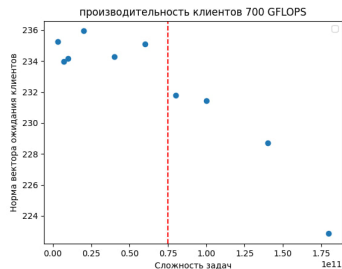
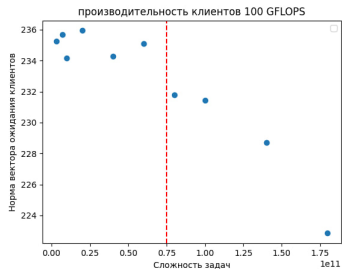
Процесс моделирования

- Исторические данные проекта **RakeSearch**, 8 месяцев, 12 тыс. узлов.
- Варьируемые параметры: вычислительная мощность клиента (1 – 1000 GFLOPS), сложность задачи (5 уровней: 3 – 100 GFLOP).

Отслеживаемые метрики

- Процент простоя хоста.
- Количество успешно выполненных заданий.
- Накладные расходы CPU планировщика.
- Учитывать стабильность очереди (станд. отклонение ожидающих задач).

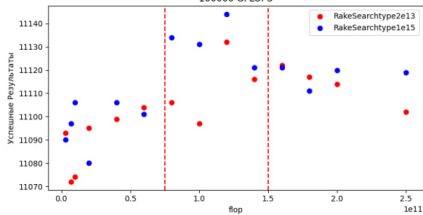
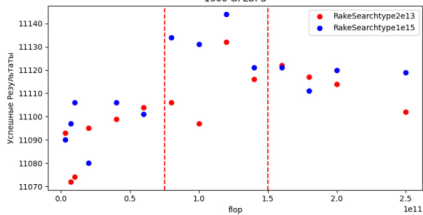
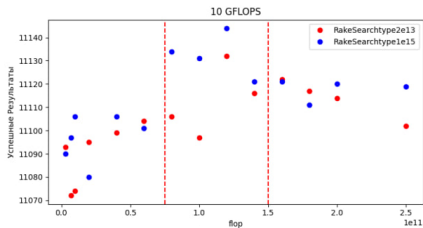
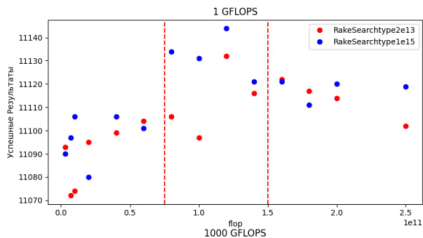
Основные результаты



400 GFLOPS

900 GFLOPS

Основные результаты



Выводы и дальнейшие шаги

Выводы

- Получены референсные параметры для последующей доработки балансировщика.
75 GFLOP – сложность задач
100 GFLOPS – производительность клиентов
- **ComBoS** ускоряет исследования и разработки планировщика, не подвергая риску доступность работающих систем.

Дальнейшие шаги

- Доработать адаптивный балансировщик что бы динамически подстраиваться под имеющиеся ресурсы клиентов.
- Возможность проведения А/В-тестирования на работающих BOINC проектах.
- Адаптивная политика существенно повышает эффективность использования ресурсов.

Спасибо за внимание

Авторы выражают благодарность за предоставление данных
по проекту RakeSearch
Максиму Олеговичу Манзюку и Эдуарду Игоревичу Ватутину

Доп. материал

