

Разработка распределенной вычислительной системы на мобильных устройствах

Зимняя школа, 2019

Полушин Кирилл, ФИТ НГУ, 3 курс

Руководитель: Городничев Максим Александрович

План доклада

- Постановка задачи
- Обзор возможных решений
- Идея решения
- Архитектура решения
- Применяемые инструменты
- Заключение

Постановка задачи

Имеем: множество разнородных
простаивающих без дела вычислителей

Хотим: повысить эффективность вычисления
параллельных задач

Обзор имеющихся решений

Существует множество средств решающих данную проблему.

Например:

- <https://cyberleninka.ru/article/v/primeneniye-parallelnykh-vychisleniy-v-sisteme-upravleniya-gruppy-mobilnykh-robotov-sapyorov>
- <https://www.sciencedirect.com/science/article/pii/S0167739X1732767X>
- <https://www.sciencedirect.com/science/article/pii/S1574119216304114>
- http://ece.nmsu.edu/~pdeleon/Research/Publications/ICASSP_2014_2.pdf
- <https://arxiv.org/pdf/1707.07452.pdf>
- <https://www.computer.org/csdl/proceedings/ficloud/2018/7503/00/750300a244-abs.html>

И один из самых известных: BOINC

Идея решения

Создадим унифицированный интерфейс вычислителей для параллельных задач, которые могут раздаваться единым центром на множество различных устройств для параллельного исполнения.

Функциональные требования (с. 1)

- Пользовательский интерфейс для авторизации и выставления задач на исполнение
- Сбор технической информации о вычислителях для возможности уточнения минимальных требуемых ресурсов исполнителей задачи
- Клиентское приложение на вычислителе, позволяющее управлять включенностью в вычисления
- Высокопроизводительные инструменты для достижения эффективного использования ресурсов и низкого энергопотребления
- Возможность вычислителей эффективно коммуницировать с другими вычислителями

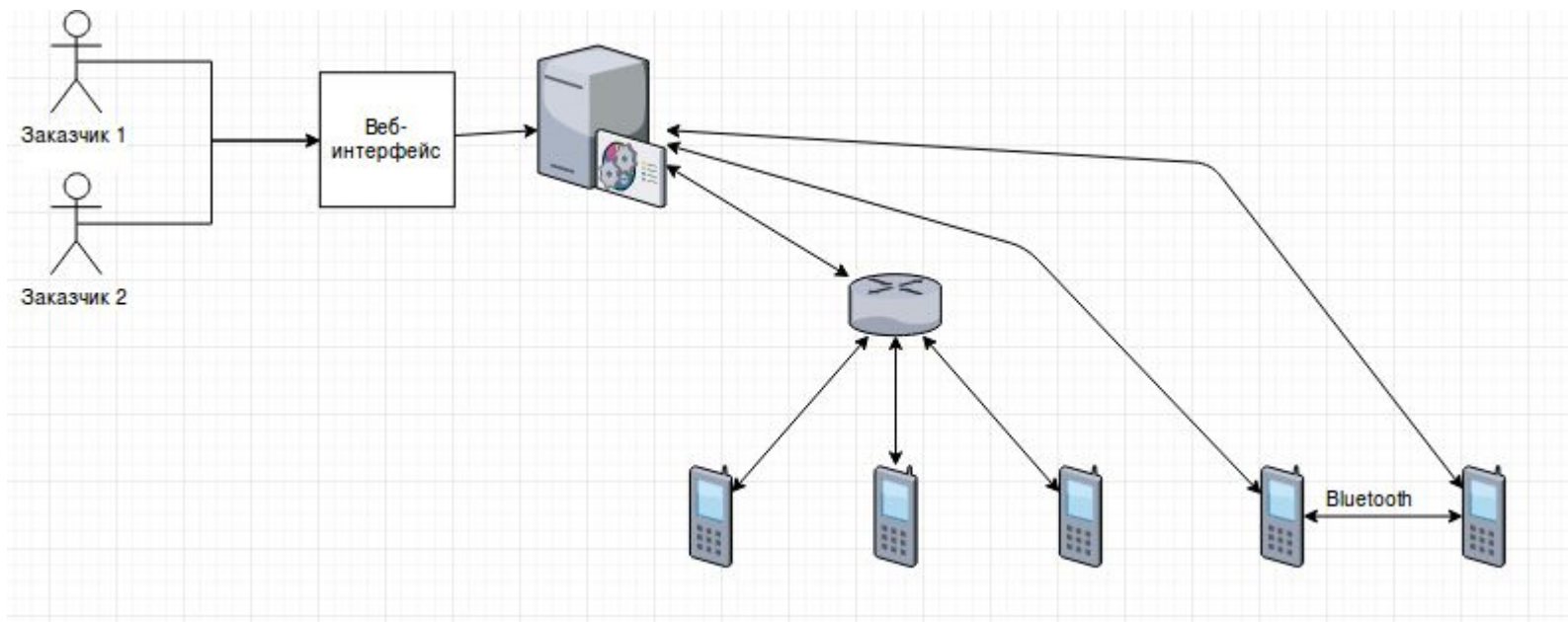
Функциональные требования (с. 2)

- Унифицированный интерфейс разнообразных вычислителей
- Возможность верификации результатов вычисления
- Возможность анонимизации данных для вычислений

Описание архитектуры

- Центральный сервер
- Пользовательский интерфейс (веб-сайт/приложение с авторизацией) для добавления задач в очередь на исполнение
- Вычислительные устройства с предустановленным приложением

Архитектура



Функциональность сервера

- Пользовательский интерфейс для авторизации пользователей распределенной системы
- Возможность пользователей добавлять новые задачи с указанием требований к исполнителям задачи
- Прием, учет и сортировка исполнителей по техническим характеристикам
- Выдача задач исполнителям
- Проверка статуса исполнения
- Обработка результатов от каждого исполнителя (в т.ч. перераспределение подзадач в случае потери связи)

Функциональность клиента-вычислителя (с. 1)

- Сбор и отправка собственных технических характеристик
- Сбор и отправка данных об окружении вычислителя (соседние устройства в LAN, Bluetooth)
- Получение от сервера информации о других вычислителях, которые находятся поблизости, для быстрого обмена данными между ними.
- Прием, компиляция и исполнение кода для вычислительных задач
- Обмен данными с другими вычислителями
- Обработка результатов исполнения (успешный, ошибка исполнения, ошибка компиляции)

Функциональность клиента-вычислителя (с. 2)

- Приостановка и сохранение состояния исполнения задачи по требованию пользователя
- Отслеживание состояния устройства на предмет чрезмерного использования вычислительных ресурсов (нехватки памяти, перегрев, разряд аккумулятора) и прерывание/приостановка исполнения
- Передача сохраненного состояния задачи другим вычислительным устройствам для перераспределения нагрузки

API для параллельных задач

- Получение информации о технических характеристиках и доступности ресурсов данного вычислителя и других вычислителей данной задачи
- Получение информации о ближайших вычислителях с которыми возможен наиболее быстрый обмен данными
- Обмен данными с другими вычислителями
- Обработка текущего статуса задачи и ее результатов
- Возможность сохранения/восстановления состояния исполнения

Инструментарий

- Язык реализации API: C++
- Система сборки для целевых устройств: CMake
- Компилятор: clang
- Протокол для обмена данными: HTTP/2 over TLS

Результаты

- Спроектирована общая архитектура распределенной системы
- Описан функционал частей архитектуры и выбраны подходящие инструменты
- Спроектирован API для вычислительных задач
- Выполнена сборка clang для Android
- Спроектирован протокол общения между клиентами-вычислителями и центральным сервером

Ближайшие задачи

- Реализация центрального сервера
- Создание протокола обмена данными между локальными клиентами
- Реализация приложения клиентов-вычислителей
- Реализация веб-сайта с авторизацией и выставлением задач на исполнение

Планы на будущее

- Проверка достоверности результатов, которые возвращают вычислители
- Балансировка нагрузки между вычислителями
- Интеграция с системой LuNA