Здравствуйте! Меня зовут Габидуллин Ибрагим, студент группы 11-308. Темой моей дипломной работы является разработка клиент-серверного фреймворка для высоконагруженных вычислений на базе Linux и MPI.

Исходя из темы можно сказать что целью моей работы является разработать готовый к развёртыванию и использованию клиент-серверный фреймворка для управлений параллельными вычислениями. Почему фреймворк? Потому что подразумевается, что проект может быть модифицирован для определенных систем. К примеру, в базовом варианте система управляет MPI вычислениями, но при желании можно легко добавить управление другими технологиями параллельных вычислений.

Для достижения цели, я поставил перед собой задачи:

* Для того, чтобы знать, чем нужно управлять, мне нужно изучить парадигмы и технологии параллельных вычислений;
* Также нужно исследовать существующие системы управления параллельными вычислениями, для того чтобы знать какие системы уже существуют и на что ориентироваться при разработке;
* Очевидно, что нужно спроектировать архитектуру фреймворка, клиентской и серверной части. В данном случае в архитектуре важна гибкость и ясность;
* После этого приняться за разработать серверную и клиентскую части;
* В конце работы нужно исследовать работу фреймворка на разных операционных системах.

Существует так называемая классификация Флинна, созданная в 60-х годах американским ученым Майклом Флинном.

Рассмотрим её.

ОКОД – классический ПК

ОКМД – OpenMP

МКОД – буран шаттл

МКМД – MPI

Системы управления

Разниц между системами практически. Все обладают примерно одинаковым функционалом, таким как запуск задачи, остановка, запуск её по времени. Подстройка программы под определенную систему.

PBS (Portable Batch System) — система управления распределенными вычислениями. Основная функция PBS — запуск вычислительных задач в вычислительной среде по расписанию.

TORQUE (Terascale Open-Source Resource and QUEue Manager) — менеджер [распределенных ресурсов](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D1%8B%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) для вычислительных [кластеров](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%82%D0%B5%D1%80_(%D0%B3%D1%80%D1%83%D0%BF%D0%BF%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BE%D0%B2)) из [машин](https://ru.wikipedia.org/wiki/%D0%A5%D0%BE%D1%81%D1%82) под управлением [Linux](https://ru.wikipedia.org/wiki/Linux) и других [Unix-подобных](https://ru.wikipedia.org/wiki/UNIX-%D0%BF%D0%BE%D0%B4%D0%BE%D0%B1%D0%BD%D0%B0%D1%8F_%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) [операционных систем](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0). Является современной версией PBS. Одна из самых развитых систем управления распределенными вычислениями, поддерживается большим организаций и научных лабораторий.

В вычислительном кластере на базе Казанского (Приволжского) Федерального университета для управления вычислениями используется такая система как IBM Platform LSF. Данная система является мощной платформой управления нагрузок для распределенных сред высокопроизводительных вычислений. Но имеет несколько минусов, являясь коммерческой и зарубежной разработкой.

Проект разделяется на три большие части: клиент, сервер управления и сервис параллельных задач, что вы можете увидеть на картинке. Архитектура довольно классическая, а также простая для модернизации

С другой стороны, в проекте используется модель слоёв Мартина Фаулера. Это добавляет читаемости для программиста. Такая структура ограничивает разделяет функции отдельных частей. Данный паттерн часто используемый и большинство программистов с ходу в нём разберутся.

В разработке использовались такие технологии как C#, .NET Core, Web API. C# один из самых используемых языков в мире, по индексу TIOBE на май 2017 года занимал 5 место. От этого зависит выбор программной платформы для разработки, в данном случае был выбран .NET Core, фреймворк является молодым, но быстроразвивающимся и поддерживает все функции нужные для проекта, таки как Web API для общения между клиентом и сервером.

Можно посмотреть на код, здесь менеджеры и энджины.

Таким образом, перейдем к последней стадии разработки, благодаря .NET Core, система работает как на системах Windows, так и на системах Linux. Фреймворк позволяет запускать MPI задачи, а также управлять выполнением запущенной программы. Таким образом, все поставленные задачи были выполненные, а заданная цель достигнута.

В дальнейшем планируется развитие продукта, создание веб клиента для управления. Также планируется добавление новой функциональности, аткой как получение статистики работы параллельной программы, работа через протокол FTP, запуск вычислений по определенному расписанию.

СПАСИБО ЗА ВНИМАНИЕ