

# 1. Глоссарий

В настоящем документе используются следующие понятия:

- пользователь – человек, формирующий задание комплексу на проведение некоего расчёта;
- задача – программа научно-прикладного характера, предоставленная в виде исполняемого файла;
- расчёт – процесс выполнения задачи, результатом которого являются некие файлы (зависящие от задачи), содержащие результаты его работы. Подразумевается, что расчёт занимает значительное (от нескольких часов и до нескольких дней) время;
- пользовательский интерфейс, ПИ – интерфейс, используемый для постановки задач комплексу и управления ходом их расчётов;
- база данных, БД – выделенный сервер или программный компонент, отвечающий за хранение и доступ к данным;
- персональный компьютер, ПК – электронно-вычислительная машина архитектуры IBM PC;
- вычисляющий компьютер, ВК – ПК с установленным ПО, обеспечивающим взаимодействие данного ПК с комплексом и проведение расчётов на данном ПК;
- СОА – “сервис-ориентированная архитектура(SOA)”, подход к разработке программного обеспечения на основе слабосвязанных компонентов, взаимодействующих посредством стандартизованных интерфейсов;
- бекенд – сервер, элемент декомпозиции СОА, отвечающий за выполнение определенной подзадачи(работы с определенным типом данных, балансировку и т.д.);
- фронтенд – сервер, элемент декомпозиции СОА, отвечающий за перенаправление запросов бекендам и предоставление ПИ и/или интерфейса приложения.

## 2. Введение

Настоящее техническое задание распространяется на разработку программного комплекса, выполняющего задачи распределения вычислительных мощностей персональных компьютеров в рамках исследовательских коллективов. Разрабатываемый комплекс позволит задействовать простаивающие вычислительные ресурсы под расчёты научных задач. Техническое задание выполняется в соответствии со стандартом ГОСТ 34.602–89 “Техническое задание на создание автоматизированной системы”.

В ходе исследовательских работ в разных областях у членов научно-исследовательских и технических коллективов часто возникают задачи расчёта небольших программ, призванных проверить какую-либо гипотезу. Время работы подобных программ, несмотря на их простоту, может достигать нескольких часов, и в рамках коллектива часта ситуация, когда в любой момент времени кто-либо проводит какие-либо расчёты. В то же время, для каждого конкретного исследователя время расчёта подобных программ не занимает всё доступное. Част режим работы, в котором конкретный сотрудник несколько дней планирует вычислительный эксперимент, после чего ему необходимо поставить его на выполнение на несколько часов. В эти несколько часов его компьютер находится под высокой нагрузкой; однако во время нескольких дней планирования он по большей части простаивает.

В связи с этим возникла необходимость реализации программного комплекса, позволяющего исследователям в рамках коллектива загружать вычислительными задачами компьютеры друг друга. Комплекс должен быть прост в обращении и не требовать особой доработки программного обеспечения вычислительных экспериментов для их расчёта.

### 3. Существующие аналоги

Подобные системы разрабатываются с 1994 года, и в общем случае их называют системами “добровольных вычислений”. Среди программного обеспечения, используемого для организации таких вычислений, наиболее распространены системы XtremWeb, Xgrid, Grid MP и BOINC. Все подобные программы работают по одному и тому же принципу – пользователь в заданном формате передаёт системе свою программу; система отправляет эту задачу на выполнение какому-либо из вычислительных узлов, получает ответ и отдаёт его пользователю.

Xgrid – технология, разработанная компанией Apple, позволяющая объединять группу компьютеров в виртуальный суперкомпьютер для проведения распределённых вычислений. Из преимуществ данной системы можно выделить наличие предустановленных клиентов на компьютерах под управлением MAC OS X; однако её недостатки весьма существенны – во-первых, существует только реализация для MAC OS X; во-вторых, для доступа к каким-либо функциям комплекса, кроме просто однопоточного запуска программы, исполняемая программа должна быть специально спроектирована с учётом особенностей системы и только на языке Objective-C.

Grid MP – технология, разработанная компанией Univa. Символы MP в названии не имеют официальной расшифровки. Предоставляет web API для манипулирования объектами системы, что позволяет вести разработку для комплекса на практически любом языке программирования, но в то же время требует разработки программ специально под комплекс.

BOINC – открытая программная платформа университета Беркли для GRID вычислений. Обеспечивает валидацию вычислений за счёт избыточности, отслеживание конкретного вклада пользователей в расчёты, управление участием в различных экспериментах; однако рассчитан на огромные по масштабам проекты (тысячи и сотни тысяч вычислителей; наиболее крупные проекты насчитывают до 15 миллионов участников). В связи с этим, процесс настройки проекта занимает значительное время.

Таким образом, из известных систем подобного рода ни одна не занимает целевую нишу ввиду следующих особенностей:

- Xgrid – поддерживает только MAC OS системы, исполняемая программа должна быть написана специально для работы с данной системой;
- Grid MP – коммерческий продукт, данных о ценах нет в наличии; исполняемая программа должна быть написана специально для работы с данной системой;

- BOINC – избыточная для поставленной задачи функциональность, исполняемые прикладные программы должны сильно дорабатываться для совместимости с проектом;

## **4. Основание для разработки**

Программа разрабатывается на основе учебного плана кафедры “Программное обеспечение ЭВМ и информационные технологии”.

## **5. Назначение разработки**

Основным назначением комплекса является утилизация простаивающих вычислительных мощностей в рамках исследовательских коллективов.

## **6. Требования к программному изделию**

### **6.1. Функциональные требования**

Комплекс должен обеспечивать возможность выполнения следующих функций:

#### **С точки зрения пользователя, устанавливающего задачу**

- регистрация пользователя в сети;
- постановка произвольной задачи с указанием минимальных параметров технических и программных средств, необходимых задаче;
- отслеживание состояния выполнения задачи;
- досрочная остановка выполнения задачи;
- просмотр отчёта о завершении задачи или всех таких отчётов в случае, если выполнение задачи дублировалось на разных компьютерах;

#### **С точки зрения пользователя, предоставляющего вычислительные мощности**

- регистрация компьютера в сети;
- начало предоставления вычислительных мощностей;
- установка лимита использования процессорного времени;

- выключение компьютера по завершению текущего расчёта;
- остановка предоставления вычислительных мощностей;

### **С точки зрения администратора**

- динамическое добавление вычисляющих компьютеров в сеть и удаление их из сети должно происходить прозрачно для пользователей;
- наличие системы мониторинга для отслеживания состояния сети;

## **6.2. Входные параметры комплекса**

В качестве входных параметров со стороны пользователя комплексу предоставляются:

- исполняемый файл задачи;
- текстовый документ формата, выработанного в ходе разработки, содержащий:
  1. требования к аппаратному и программному обеспечению ВК;
  2. список выходных файлов задачи;
- сведения о необходимости дублирования расчёта на нескольких ВК.

Со стороны ВК в автоматическом либо ручном режиме работы комплексу предоставляются сведения о аппаратном и программном обеспечении, позволяющие балансировщику принимать решения о назначении задачи конкретному ВК.

## **6.3. Выходные параметры комплекса**

Комплекс должен предоставлять пользователю следующую информацию:

- данные о текущем статусе поставленных задач – выполнена, выполняется, ожидает свободного ВК, приостановлена, отменена;
- для завершённых или частично завершённых задач – список поступивших отчётов о завершении расчёта;
- данные о текущем состоянии сети – список в данный момент подключенных ВК, данные по их активности.

## 7. Сценарии функционирования системы

- Постановка задачи на исполнение

1. Пользователь осуществляет вход в ПИ со своими учётными данными
2. Авторизованный пользователь добавляет новую задачу через ПИ, предоставляя все данные, указанные в пункте “Входные данные”
3. Задача ставится в очередь на выполнение
4. Система ожидает запроса от свободного ВК на получение задачи
5. Система передаёт ВК задачу и необходимые ей данные и ожидает ответа
6. Далее возможны варианты:
  - Расчёт успешно завершается и ВК передаёт результаты расчёта системе
  - Расчёт завершается ошибкой и ВК передаёт отчёт об ошибке системе
  - В ходе расчёта ВК перестаёт сообщать балансировщику о ходе работы. Балансировщик ждёт сообщений от ВК в течение указанного максимального времени ожидания, и считает ВК нештатно завершившим работу. Балансировщик повторяет помещение задачи в очередь на выполнение с повышенным приоритетом. Если ВК вновь окажется в сети, переданные им результаты не будут учтены.

- Просмотр статуса выполнения задачи

1. Пользователь осуществляет вход в ПИ со своими учётными данными
2. Авторизованный пользователь выбирает одну из собственных задач через ПИ
3. Система генерирует отчёт, содержащий информацию о статусе выполнения задачи и все полученные на данный момент отчёты о завершении расчётов, полученные от ВК

- Отмена исполнения задачи

1. Пользователь осуществляет вход в ПИ со своими учётными данными

2. Авторизованный пользователь отменяет выполнение одной из собственных задач через ПИ
3. Система помечает задачу как отменённую, и при получении следующего отчёта от ВК о статусе выполнения задачи в ответ передаёт команду остановки расчётов

## 7.1. Требования к структуре

Комплекс должен удовлетворять следующим требованиям:

- структура системы должна следовать принципам СОА;
- количество качественно различных сервисов, из которых должна состоять система, должно быть не меньше 4-х;
- не менее 3-х сервисов(бекендов) должны быть горизонтально масштабируемыми;
- взаимодействие сервисов должно осуществляться по HTTP протоколу с учётом рекомендаций REST, если не доказана необходимость отказа от такого решения;
- отслеживание авторизационных ключей должно осуществляться отдельно выделенным сервисом;
- база данных комплекса должна поддерживать репликацию;

## 7.2. Требования к надёжности

Комплекс должен быть спроектирован с учётом контролируемой деградации, то есть в случае ошибки продолжать обеспечивать хотя бы часть функционала. Примеры таких ситуаций:

- внезапное отключение ВК не должно приводить к “зависанию” задачи на стадии выполнения, балансировщик должен определять такие ситуации и передавать задачу другому исполнителю;
- внезапное завершение работы задачи не должно приводить к остановке работы ВК, он должен отправить отчёт о работе задачи балансировщику и запросить новую задачу;
- внезапное отключение балансировщика не должно влиять на ход расчёта задач и должно быть отражено в интерфейсе администратора;

- внезапное отключение сервера, осуществляющего взаимодействие с пользователем, не должно влиять на ход расчёта уже поставленных комплексу задач.

### **7.3. Требования к информационной и программной совместимости**

На программное обеспечение серверных частей комплекса ограничения не накладываются. Комплекс должен поддерживать как вычисляющие компьютеры под управлением ОС Windows (начиная с 7), так и Linux. Желательно использование нескольких языков программирования для реализации разных сервисов.

## **8. Требования к программной документации**

Программная документация должна удовлетворять следующим требованиям:

- разрабатываемые программные модули должны быть самодокументированны, т.е. тексты программ должны содержать все необходимые комментарии;
- документация должна включать в себя отчёты о модульном, интеграционном и системном тестированиях программы
- документация должна включать в себя краткое руководство пользователя