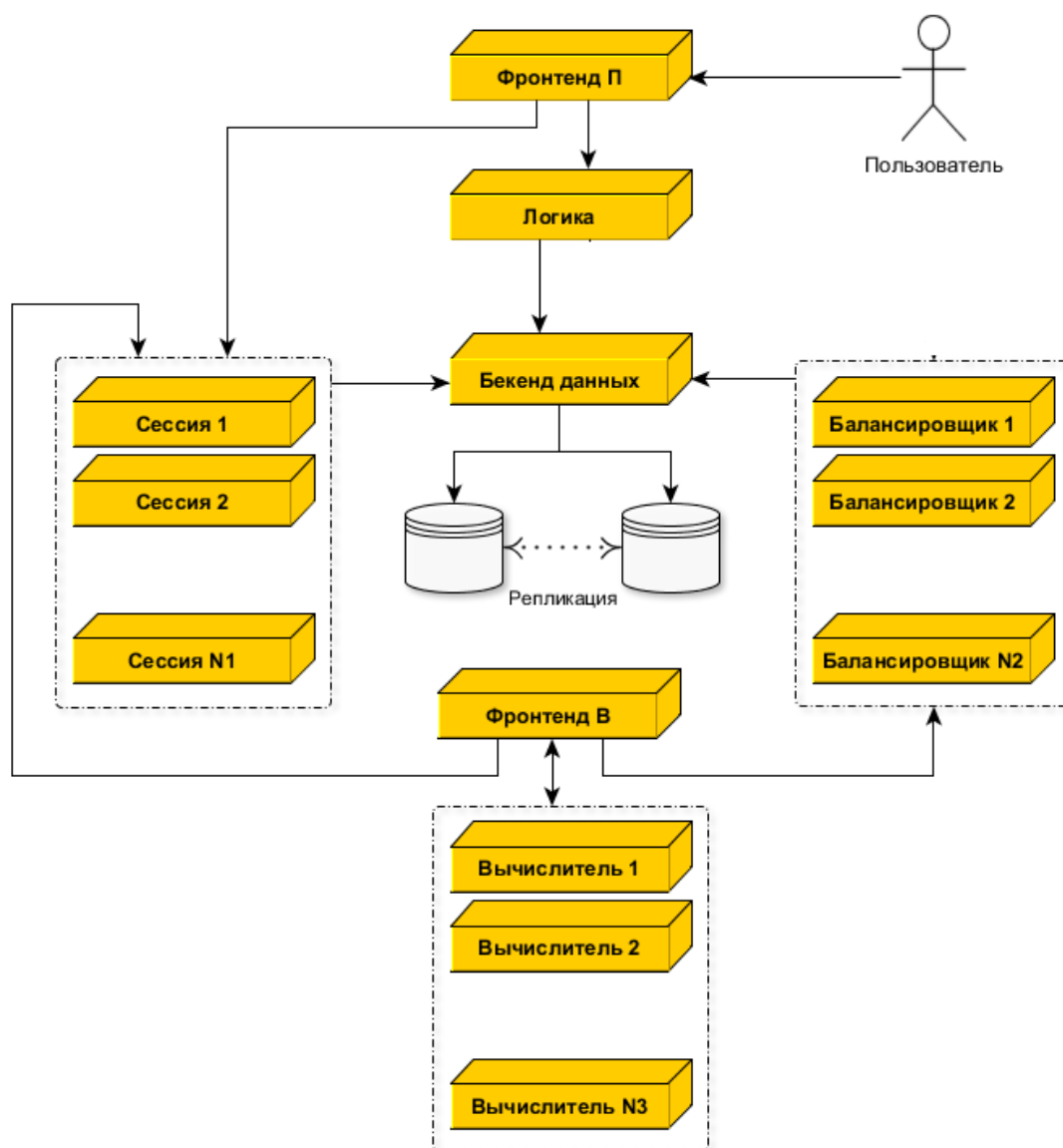


Введение

1. Аналитическая часть

1.1. Проектирование общей структуры комплекса



1.2. Проектирование базы данных

Определение требований к структуре БД

БД должна осуществлять функцию коммуникации между отдельными узлами сети. Это накладывает следующие требования на её структуру:

- Чтобы по возможности уменьшить степень дублирования похожих описаний задач, в отдельную сущность должны быть вынесены “черты” задач (traits). Чертой, к примеру, является требование задачи к вычислителю иметь окружение “Cuda v4.0” или “.net 3.5”.
- Чтобы позволить отслеживать состояние всех подзадач задачи, запущенной с дублированием вычислений, в отдельную сущность должны быть вынесены “подзадачи”, наследующие все атрибуты родительской задачи и хранящие данные, относящиеся непосредственно к ходу вычислений (на каком вычислителе производятся вычисления, результат вычислений и т д).
- Чтобы комплекс имел возможность выбора подходящего вычислительного узла для задачи, узлам (Agents) должны соответствовать такие же наборы “черт”, как и диспетчеризуемой в данный момент задаче.

ER-диаграмма

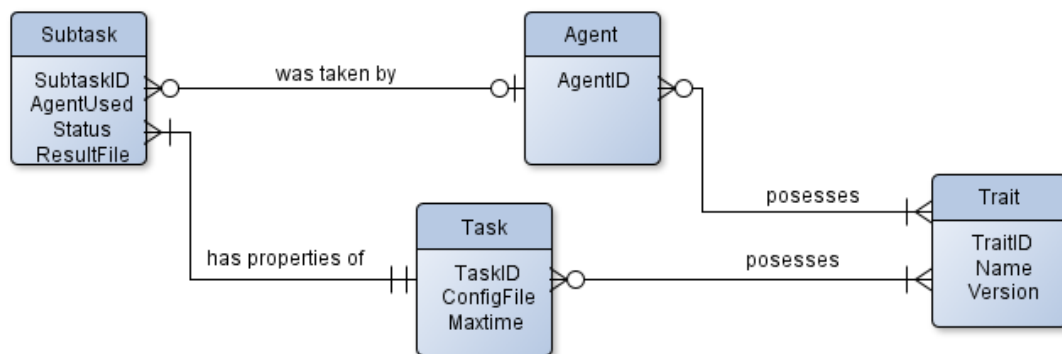


Таблица полей и типов данных

“тип?” соответствует Nullable<Тип>

Сущность	Поле	Тип	Описание
Subtask	AgentUsed	UID?	Использованный узел
	Status	Scheduled In process Terminated Node unavailable Completed	Статус задачи
	ResultFile	string?	Имя архива с результатами
Task
Agent
Trait

1.3. Проектирование API отдельных сервисов

Сервер БД

REST API, полное отражение структуры БД на набор эндпойнтов-объектов (/agents/id, /traits/id, tasks/id и т д)

Балансировщик

Функционал	Endpoint	Хедеры-параметры
Получение новой задачи	GET /newtask	AgentID
Выдача результатов выполнения	POST /result	AgentID, SubtaskID, Status, Body
Оповещение о статусе работы	POST /heartbeat	AgentID

Фронтенд вычлительных узлов

Дублирует эндпойнты балансировщика; добавлены функции:

- Регистрация нового узла в сети
- Подключение зарегистрированного узла к сети

Запросы на эндпойнты балансировщика проходят проверку безопасности и уходят к балансировщику(ам), запросы на регистрацию/подключение идут сразу к серверу сессии.

Сервер сессии

Отвечает за проверки ключей на обоих фронтендах. В случае несоответствия ключей ожидаемым запрос не пропускается “внутрь” комплекса. Внутри комплекса - “доверенная” область, проверок безопасности нет.

Сервер логики

Должен обеспечивать функционал:

- Регистрация нового пользователя
- Вход пользователя в свой аккаунт
- Постановка новой задачи на выполнение
- Просмотр статусов поставленных задач
- Для выполненных задач - получение результатов в виде архива

Пользовательский фронтенд

Дублирует эндпойнты сервера логики, проверяя ключи перед перенаправлением запроса на сервер логики.