Принципы построения высоконагруженных систем
Институт прикладных компьютерных наук ИТМО

Домашнее задание 4. Проектирование высоконагруженной системы

Георгий Семенов
georgii.v.semenov@mail.ru
Мягкий дедлайн: Вт, 23.12.2025, 23:59 МСК
Жесткий дедлайн: Вт, 30.12.2025, 23:59 МСК

Имя Фамилия
DDIA25-HW4-NameSurname.pdf

December 12, 2025

Это домашнее задание – финальное в курсе и представляет собой курсовую работу, которую необходимо очно защитить. В ней необходимо спроектировать высоконагруженную систему, рассматривая те аспекты, которые были освещены в рамках занятий и домашних заданий. Давайте их структурированно перечислим:

- **SLA** (uptime, latency) – то, что система предоставляет пользователям;
- **Пропускная способность** системы ограничивается её bottlenecks – аппаратными (CPU, RAM, Disk), сетевыми (bandwidth), слоями хранения данных (СУБД, кэш и т.д.) и пр.;
- Мы хотим оставить себе возможность увеличивать пропускную способность системы с помощью **масштабирования** (вертикального и горизонтального);
- Свойство возможности наблюдения за ходом работы системы – **observability** – в real-time обеспечивается мониторингом SLO/SLI, проверками здоровья;
- **Устойчивость** системы к отказам (resilience) обеспечивается избеганием **SPoF** и стратегиями устойчивости (retry, failover, bulkhead, hedging, circuit breaker и пр.);
- В случае отказов (внешней перегрузки из-за DDoS, и из-за внутренних) хочется аккуратно сдеградировать – **graceful degradation**, **failover**;

- **Балансировка нагрузки** между компонентами системы **в различных зонах** позволяет равномерно распределять нагрузку (RR, LC, IP Hash и т.д.) и избегать перегрузок отдельных компонентов;
- Стратегия управления инцидентами (**incident management**) и их ретроспективный анализ (**post mortem**) помогают повышать **надежность** системы со временем;
- **Слой хранения данных (stateful)** реализуется на основе хранилищ (ACID/BASE), кэшей (стратегии кэширования); в паттернах **федерации** и **шардирования**;
- На практике мы вынуждены балансировать между **latency** и **consistency** системы (PACELC-теорема);
- Важный паттерн проектирования высоконагруженных систем - **очереди** (event sourcing, microservices, CQRS)

Возьмите любой свой проект (в т.ч. можно прошлый учебный) и спроектируйте его высоконагруженную архитектуру, освещая **все аспекты, упомянутые выше,** и следуя схеме, предложенной ниже.

1 Введение

1.1 Предметная область

что за предметную область вы рассматриваете, проектируя вашу систему? какие у нее особенности? какая проблема решается?

1.2 Требования к системе

1.2.1 Функциональные требования

что должна делать система? какие пользовательские истории она должна поддерживать?

1.2.2 Нефункциональные требования

SLA (uptime, latency), SLO/SLI (error rate и т.д.), throughput (RPS), scalability (какие бизнес-риски и насколько заставляют масштабировать систему в будущем?). Обоснуйте выбор метрик

2 Архитектура системы

2.1 Потоки данных

deployment-диаграмма системы; DFD-диаграмма системы; слой хранения данных (хранилища, кэши), федерация и шардирование; как реализуется микросервисная архитектура, какие очереди (event sourcing) используются? какая модель репликации, consistency?

2.2 Отказоустойчивость

балансировка, как обеспечивается graceful degradation, zone redundancy, failover, resilience strategies?

рефлексия: почему нет единственных точек отказа (или все же есть)? какой фактор вмешательства человека? MTTR? какая целевая архитектура может быть предложена, если приведенная выше - текущая?

3 Эксплуатация и управление надежностью

3.1 Ресурсы и их масштабирование

Предложите количество ресурсов CPU, RAM, LAN, NVMe для каждого пода микросервиса в вашей архитектуре; количество подов; пропускную способность пода; как обеспечивается автоматическое горизонтальное масштабирование под нагрузкой?

3.2 Мониторинг и дежурная смена

Отслеживаемые SLO/SLI, как реализуются столпы observability, организационное устройство дежурной смены

3.3 Реагирование на инциденты

Алерты, runbooks, порядок эскалации инцидентов, коммуникация в команде и с пользователями ; disaster recovery (backup-стратегия, recovery процедуры, RTO/RPO)

3.4 Порядок расследования инцидентов и проведение постмортемов

Методы расследования инцидентов, протокол проведения встречи с разборами, артефакты разбора, отслеживание выполнения артефактов

4 Заключение

Какая работа была проделана, рефлексия, какие направления дальнейшего проектирования SRE-процессов для системы?