Урок 1 > Где/зачем нужен системный дизайн

> Оглавление

> Сбор требований к системе и расчет нагрузки на систему

Часто задают вопрос, зачем вообще нужно считать нагрузки и собирать требования?

На собеседовании можно растеряться. В таком случае наличие стандартного плана действий поможет справиться со стрессом и начать действовать. Тем не менее, во время сбора требований вы можете либо загнать себя в ловушку, записав что-то, что вообще невозможно выполнить, либо сделаете все верно, обозначив ограниченный скоуп для интервью.

Совет для собеседования по System design: вы должны лидировать, то есть вести партнера (собеседующего).

В идеале интервьюер сначала задает свой вопрос, а вы, в виде монолога, говорите, что собираетесь делать на несколько шагов вперед.

В реальной жизни этап сбора требований и расчета нагрузки самый нервный. Именно на нем инженеру необходимо понять, сколько ресурсов требуется для выполнения, а менеджеру — какой нужен бюджет.

> Высокоуровневый дизайн

После сбора требований и расчета нагрузки приходит время высокоуровнево изобразить систему. Часто люди начинают вдаваться в подробности уже на этом этапе, но так лучше не делать. Указав основные компоненты, у собеседующего можно уточнить, согласен ли он с дизайном или есть вопросы.

В реальной жизни выделение основных компонент позволяет понять:

- есть ли вообще люди в команде, которые способны выполнить проект;
- есть ли все необходимое для компонент;
- или же надо это закладывать в headcount, в бюджет и так далее.

> Выбор подходящих баз данных

Описав общие концепции и компоненты, пора углубиться в частности.

На интервью данному шагу уделяют меньше внимания, так как можно просто указать, какую базу вы бы хотели видеть.

В реальной жизни нельзя просто так взять и выбрать любую из существующих БД. Если в вашей компании всю жизнь работали с PostgreSQL, вряд ли можно просто выбрать Cassandra — нет людей, которые могут ее поддерживать и использовать. Поэтому необходимо четко понимать альтернативы. Возможно, идеальная БД для проекта существует, но вам выгоднее взять чуть менее идеальную, с которой вы уже можете работать и не потратите много денег и времени на внедрение и обучение персонала.

> Модульный подход к дизайну

Теперь каждую из компонент рассматриваем как отдельный модуль.

Различные сценарии взаимодействия выделяются в подсервисы, появляются брокеры сообщений для асинхронного общения. Тут же можно упомянуть права доступа, но на интервью, возможно, не будет на это много времени.

В реальной жизни все очень похоже. Можно уточнить, например, нужен ли вообще брокер сообщений или другой частный компонент и, опять же, есть ли люди, которые с ним могут работать. Менеджерам данный курс поможет понять, есть ли что-то лишнее или, наоборот, чего-то не хватает. Инженерам курс поможет не получать таких вопросов от своего менеджера.

> Масштабирование системы

Когда все модули уже есть, пора поговорить про масштабирование системы — устранение избыточности и способы репликации БД и сервисов.

Тут возникают такие вещи, как:

- консистентное хэширование;
- балансировщики нагрузок;
- масштабирование БД;
- и так далее.

Иногда на интервью этому этапу уделяют большое внимание, в то время как в жизни везде load balancer'ы не ставятся. Когда они действительно нужны, возникает вопрос, что они должны из себя представлять: физические машины или nginx, настроенный как load balancer.

> Повышение отзывчивости

В разделе нефункциональных требований обычно есть latency.

На этом этапе разбирается:

- как и что кэшировать;
- как индексировать БД;
- действительно ли все хэндшейки нужны.

В реальной жизни кэш — дело дорогое, поэтому нужно понять, нужен ли он: возможно, latency > 200 мс для вас подходит. Если же нужен, он должен быть заложен в расчет нагрузки на систему.

> Подсистема для поиска

Данная тема часто встречается на интервью, поэтому было решено добавить ее в курс. В данной лекции будут рассмотрены полезные алгоритмы и структуры данных.

> Мониторинг и точки выхода

Успешно завершив предыдущие этапы собеседования, нужно показать, что спроектированный сервис хорошо работает. Очевидно, для этого необходимо мониторить какие-то метрики. Появляются сразу несколько вопросов:

- какие?
- почему именно они?

Метрики стоит выбирать:

- которые о чем-то говорят;
- на которые можно повлиять.

Во время собеседования можно обозначить сами метрики и сказать, как они будут мониториться.

В реальной жизни важно, чтобы сервис работал! :) На случай, если метрики ведут себя не так, нужно понимать, можем ли быстро понять причины. Может помочь

иерархия метрик. Необходимо продумать, какие системы аналитики будут использоваться и как будет вестись сбор данных.

> Козыри в рукаве

Вероятно, вы — специалист некоей области. По окончании собеседования можно показать свои глубокие знания и уточнить какие-то подробности дизайна.

Если после собеседования останется время и если вы ранее указали требования, но условились обговорить только основную часть, можно попробовать описать решение для одного из отложенных пунктов.

В реальной жизни после написания дизайн дока нужно ждать отзывов от читающих его коллег. Если критика адекватная, то это повод внести изменения. Однако если коллеги топят за технологии, не аргументируя, то к такому лучше не прислушиваться.