

Разработка высоконагруженных проектов

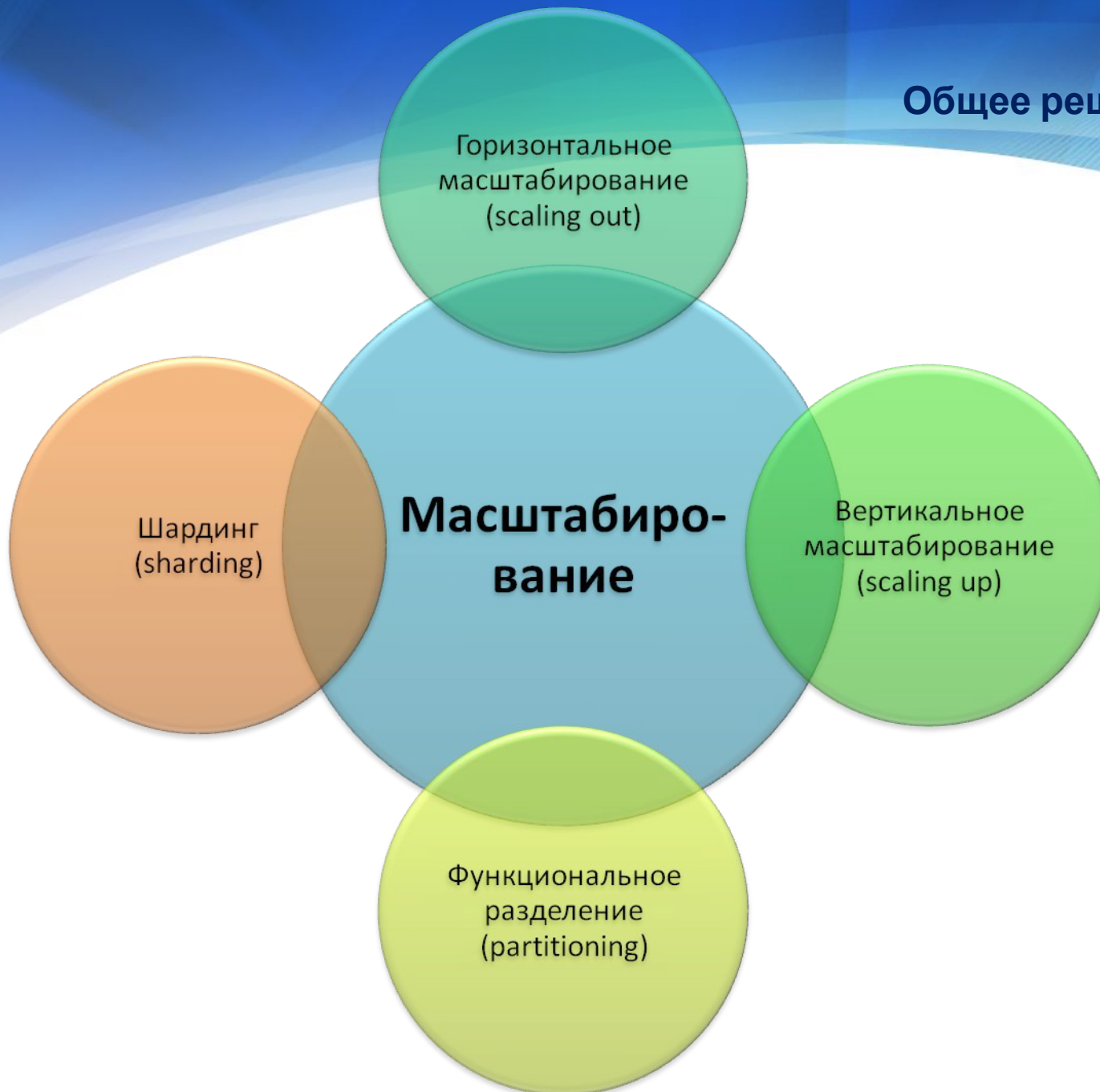
(например – сайтов для сообществ)

Олег Бунин

Что такое “большой” проект?

- Сотни тысяч, миллионы, десятки миллионов **ХИТОВ**;
- Бесперебойная работа;
- Сложная структура: серверный парк, количество
- Большое количество



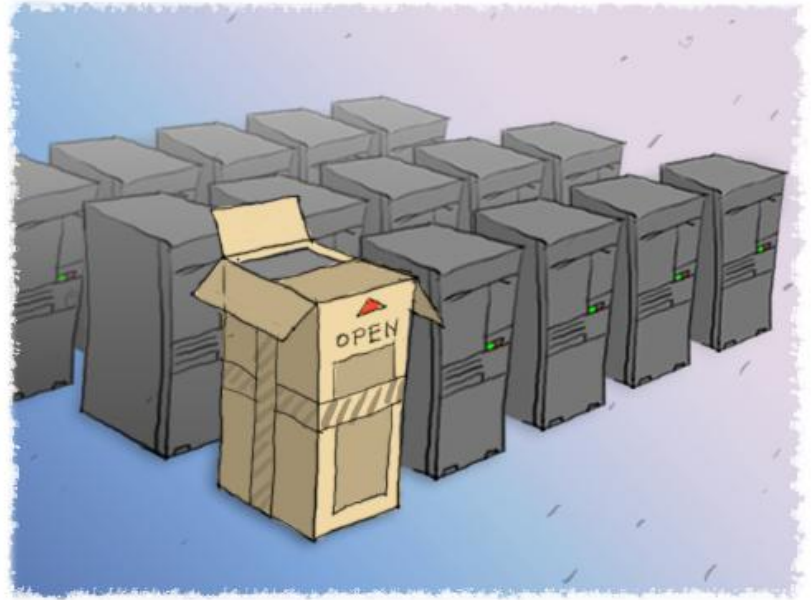


Горизонтальное масштабирование

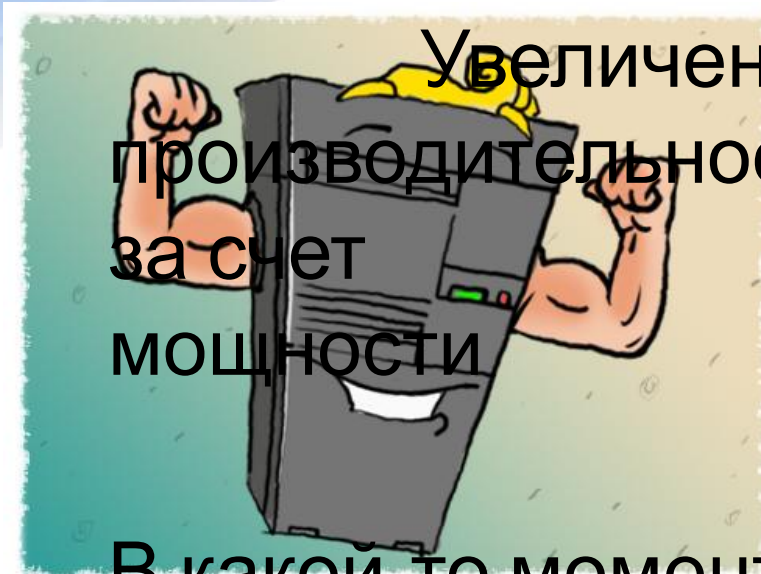
Увеличение системы за счёт подключения серверов.

Отлично работает для серверов, а как быть с базс

Что делать со связанными общими для нескольких серверов данными?



Вертикальное масштабирование



Увеличение
производительности
за счет
мощности

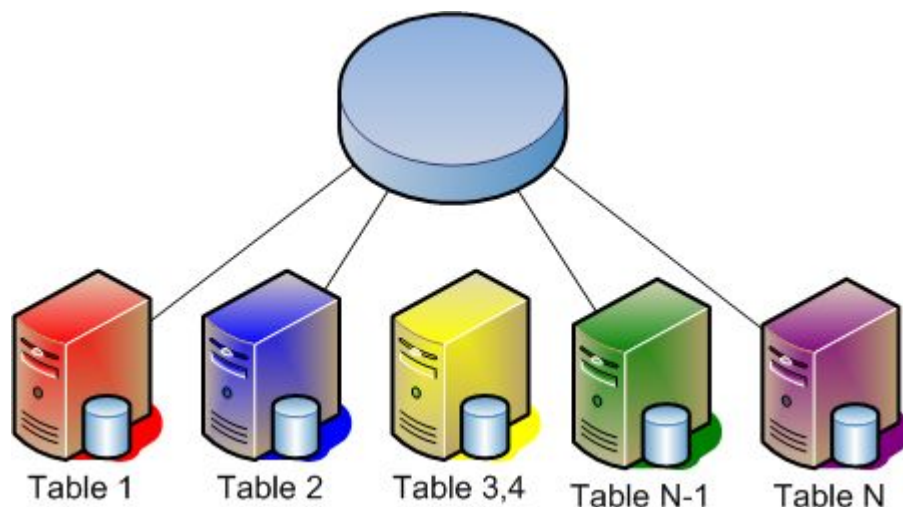
СИСТЕМЫ
увеличения
сервера.

В какой-то момент мы все равно достигнем предела по процессору, памяти или жесткому диску.

Функциональное разбиение

Разные функциональные части работают и хранятся на разных серверах системы.

В какой то момент мы все равно упрямся в физические возможности сервера.



Шардинг

Разбиение данных на
которые раскладываются по
серверам-шардам.

Как правильно разбить
шардинга? Как правильно
идентифицировать данные?

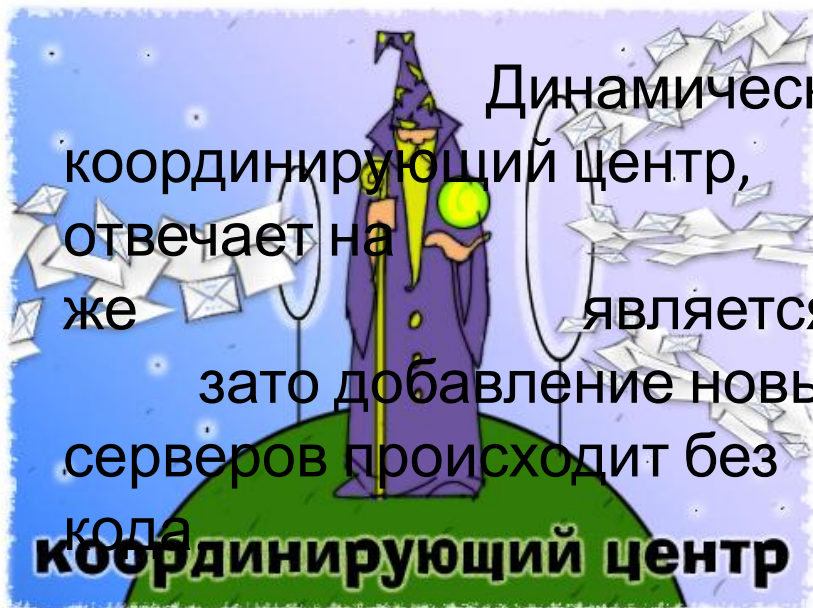


У них просто нет выбора:

Google™ Яндекс

Разбиение данных для шардинга

Статическое: по первой букве логина, хэширование идентификаторов или логинов. Единого центра нет, соответственно нет узкого места, зато есть сложности с разрешением заранее непредусмотренных ситуаций.



Динамическое: есть

координирующий центр,
отвечает на
же

зато добавление новых
серверов происходит без
кода

координирующий центр

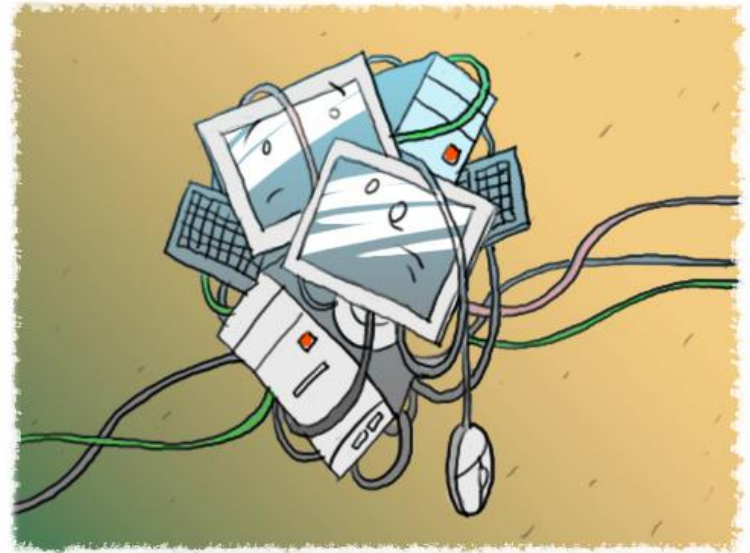
который
вопрос “где лежит”? Он

является узким местом,

изменения

Как облегчить масштабирование?

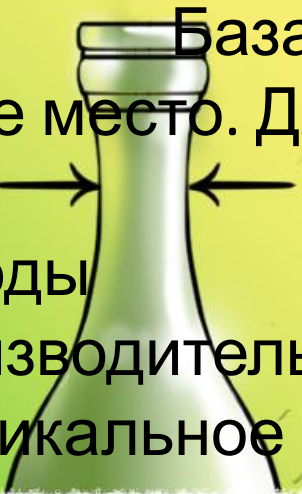
- Низкая степень связности данных и кода;
- Разделение кода на слои минимум слой связи с и слой кэширование;
- Рефакторинг, высокое качество минимизация workaround'ов;
- Контроль над системой, мониторинг;
- Минимизация академических решений (построение таблиц “на лету”, ORM).



Серебряная пуля



Отдельно о базах данных



База данных – типичное узкое место. Для базы данных все методы актуальны вышеперечисленные увеличения производительности: горизонтальное и вертикальное масштабирование, функциональное разбиение, шардинг.

Горизонтальное масштабирование в случае с БД достигается с помощью **репликации**.

Репликация

Синхронизация
объекта.

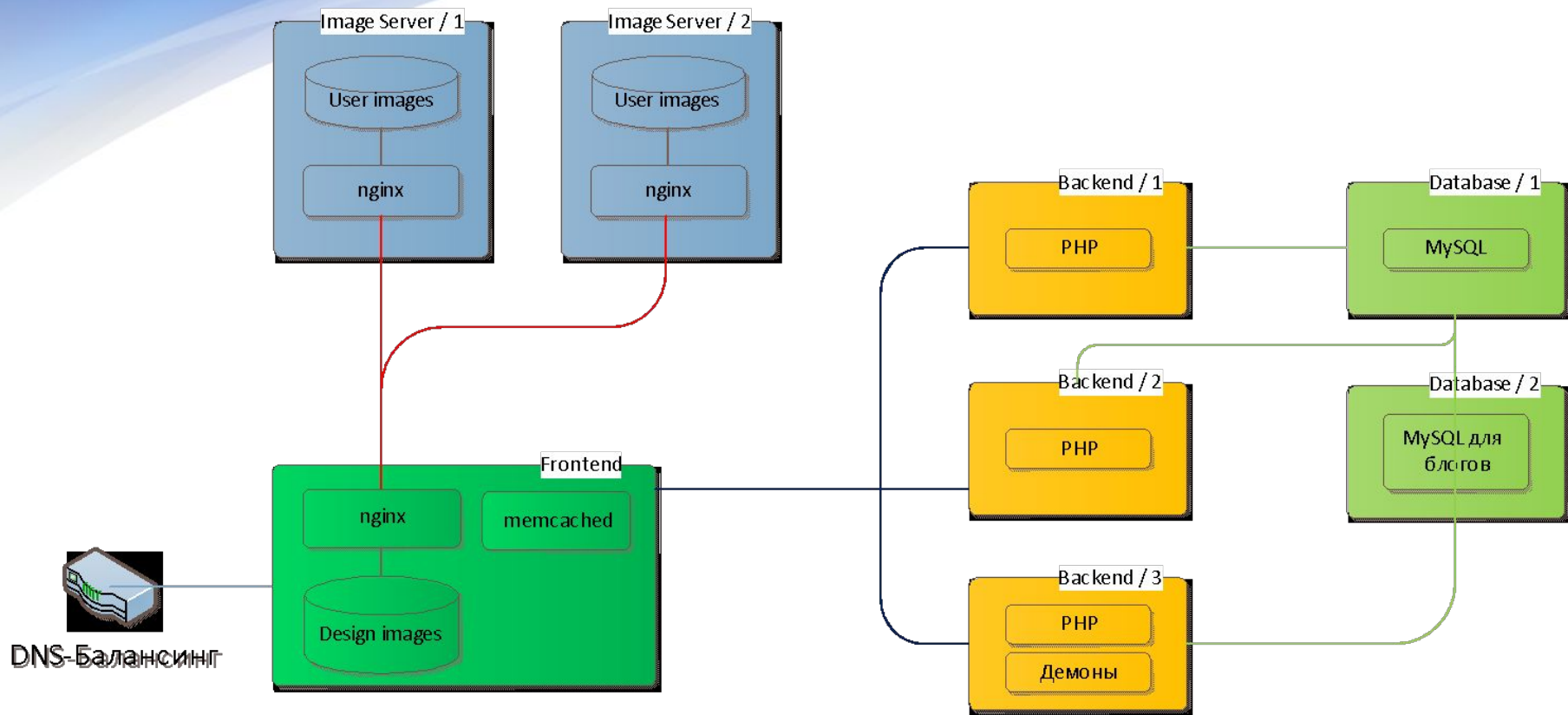
Наиболее эффективна при
количестве
усложняется

распространения изменений, которое, в
дальнейшем, становится узким местом.

Усложнение программной архитектуры – например,
чтение данных с слейва, до которого не докатились
изменения.



Типичная архитектура: обычный сайт



Специфика сообществ

Большое количество связей между объектами □
сложная программная архитектура.

Высокий hit ratio □
серверов.

Наличие сложных
реализуемых
алгоритмами (поиск,
переписка,





oleg.bunin@ontico.ru