8. Масштабирование

Сухорослов Олег Викторович

23.10.2023

План лекции

- Масштабируемость и масштабирование
- Техники масштабирования и связанные задачи
  - Репликация stateless-сервиса и балансировка нагрузки
  - Кэширование
  - Разбиение stateful-сервиса по данным (шардинг)

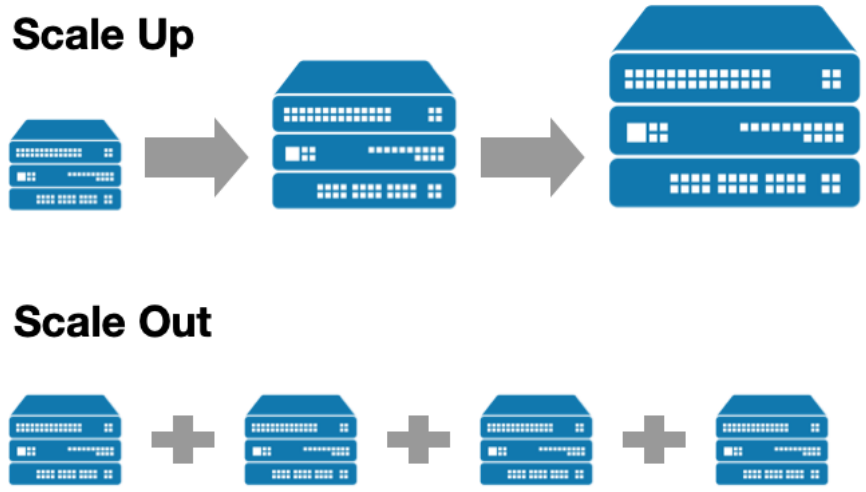
Проблема

Масштабируемость

- Способность системы "расти" в некотором измерении без потери производительности и других характеристик, а также без необходимости изменять реализацию
- **Нагрузочная масштабируемость** – способность системы увеличивать свою производительность при увеличении нагрузки путем замены или добавления аппаратных средств
- Параметры нагрузки
  - Число обрабатываемых запросов
  - Объем хранимых данных
  - Объем вычислений/данных на запрос

Масштабирование

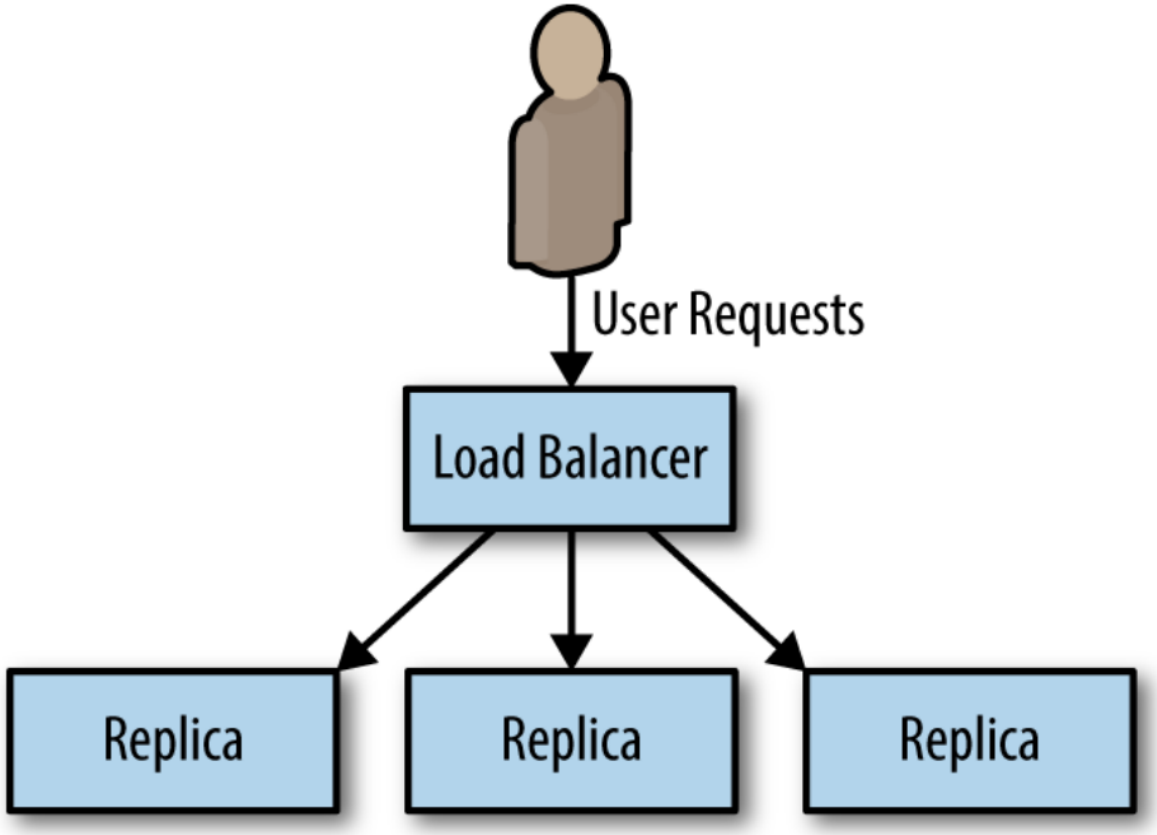
- **Вертикальное масштабирование** – увеличение производительности каждого компонента системы с целью повышения общей производительности.
- **Горизонтальное масштабирование** – разбиение системы на более мелкие структурные компоненты и разнесение их по отдельным физическим машинам, и (или) увеличение количества серверов, выполняющих одну и ту же функцию.



Техники горизонтального масштабирования

- Репликация stateless-сервиса + балансировка нагрузки
- Кэширование
- Разбиение stateful-сервиса по данным (шардинг)
- Параллельная обработка (далее в курсе)

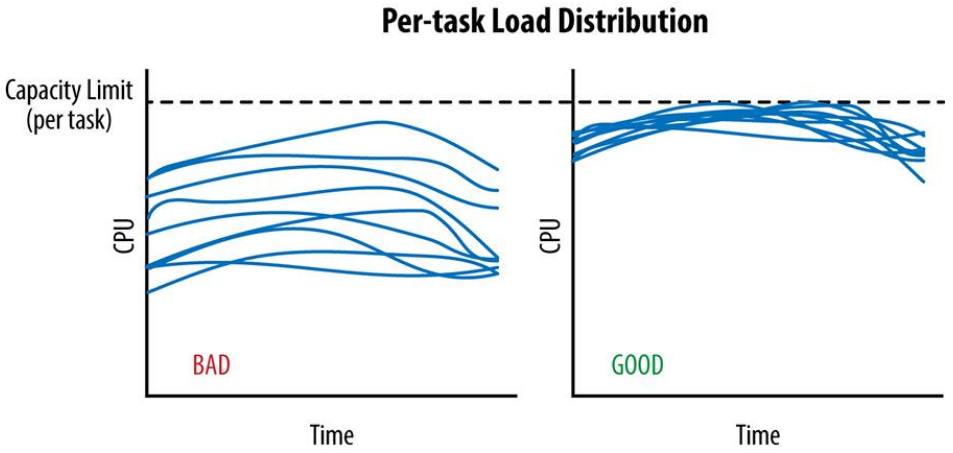
Репликация (stateless сервис)



Load Balancer: варианты реализации

- Layer 4 (connection/session)
  - Оперирует данными (пакетами) на транспортном (TCP, UDP) уровне
- Layer 7 (application)
  - Распределяет запросы на прикладном уровне (HTTP) на основе их содержимого
- Балансировка с помощью DNS
- Hardware vs Software
- Load Balancer vs Proxy

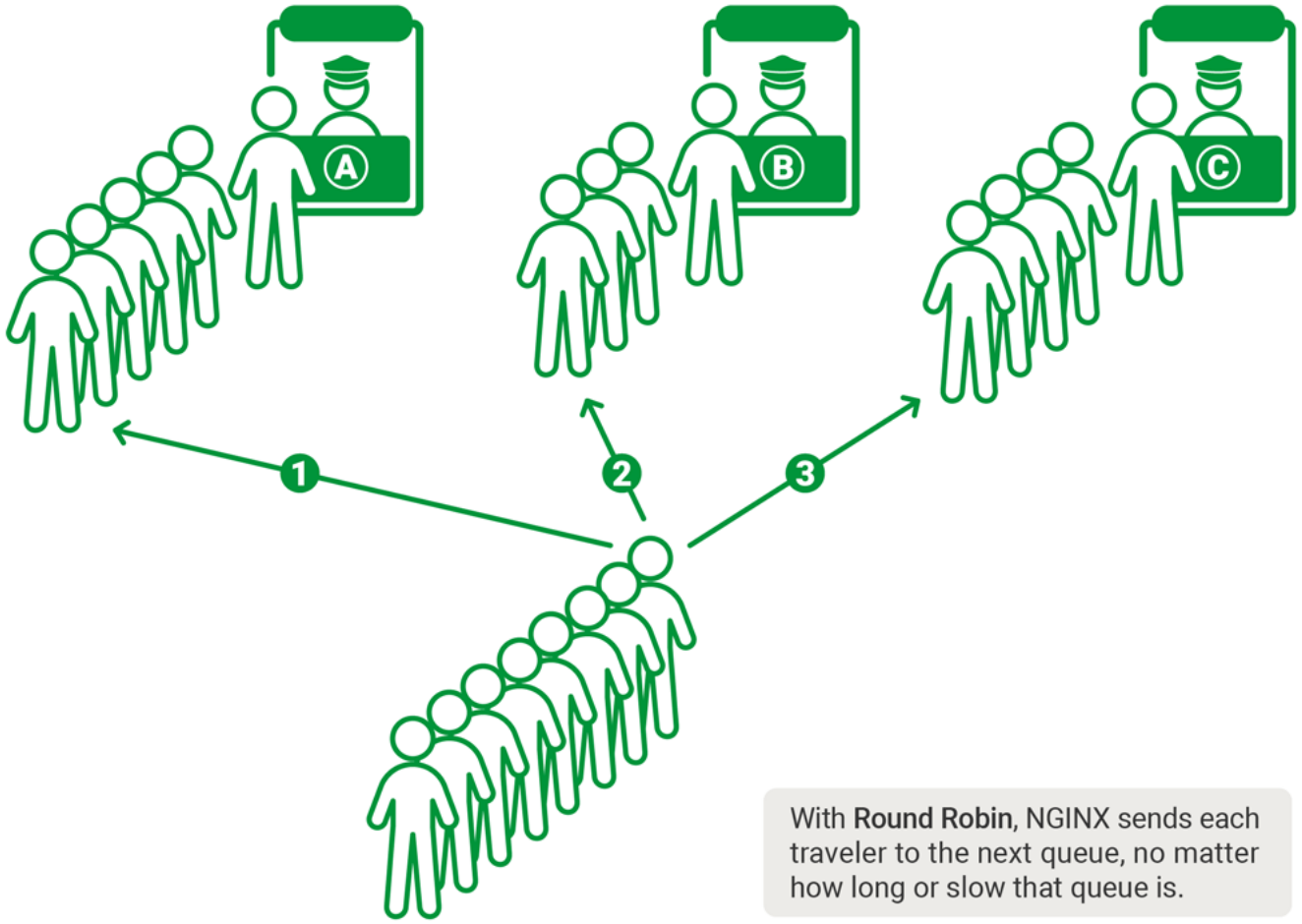
Балансировка нагрузки



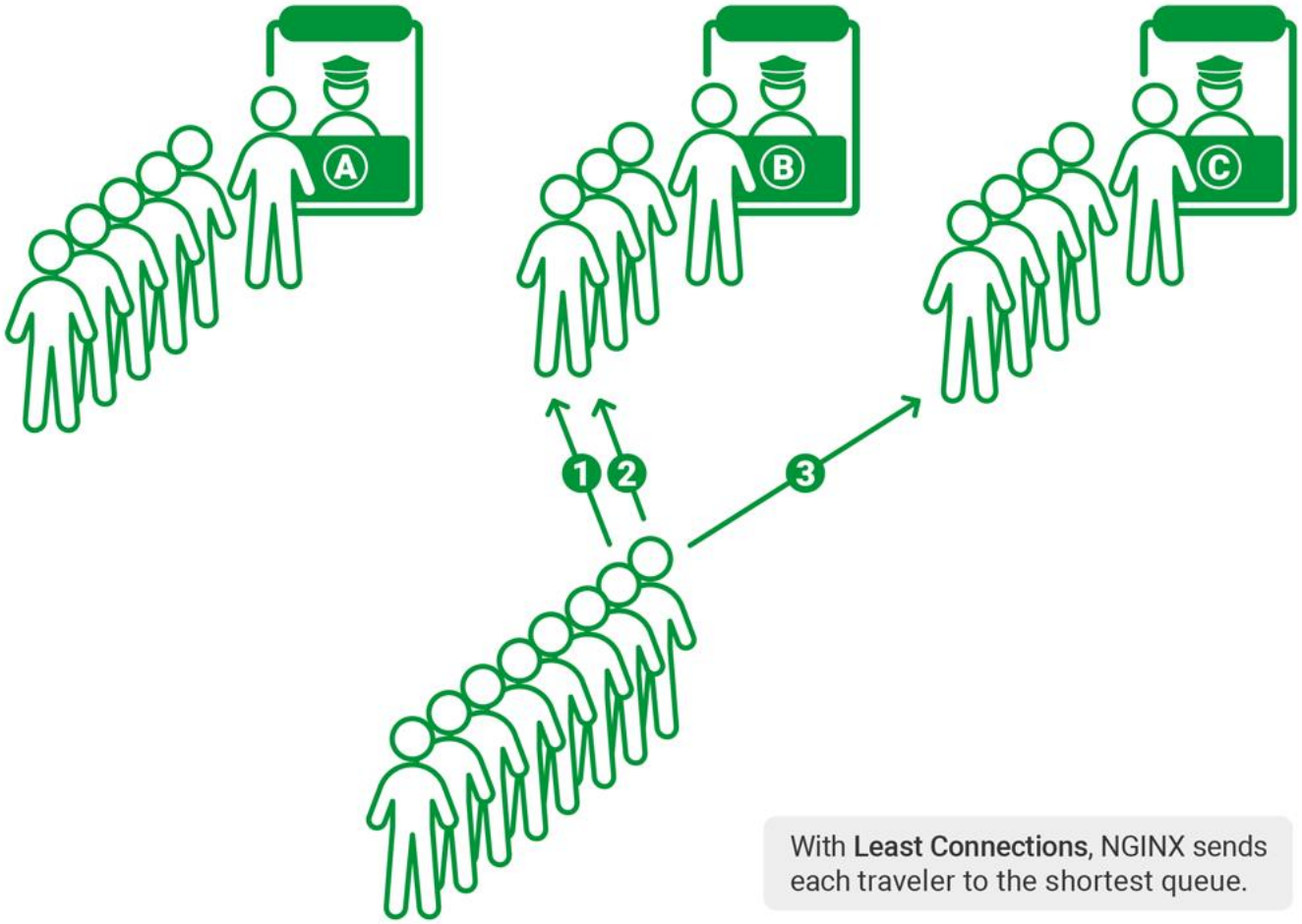
Алгоритмы

- Random
- Round Robin
- Least Connections
- Least Time
- Power of Two Choices

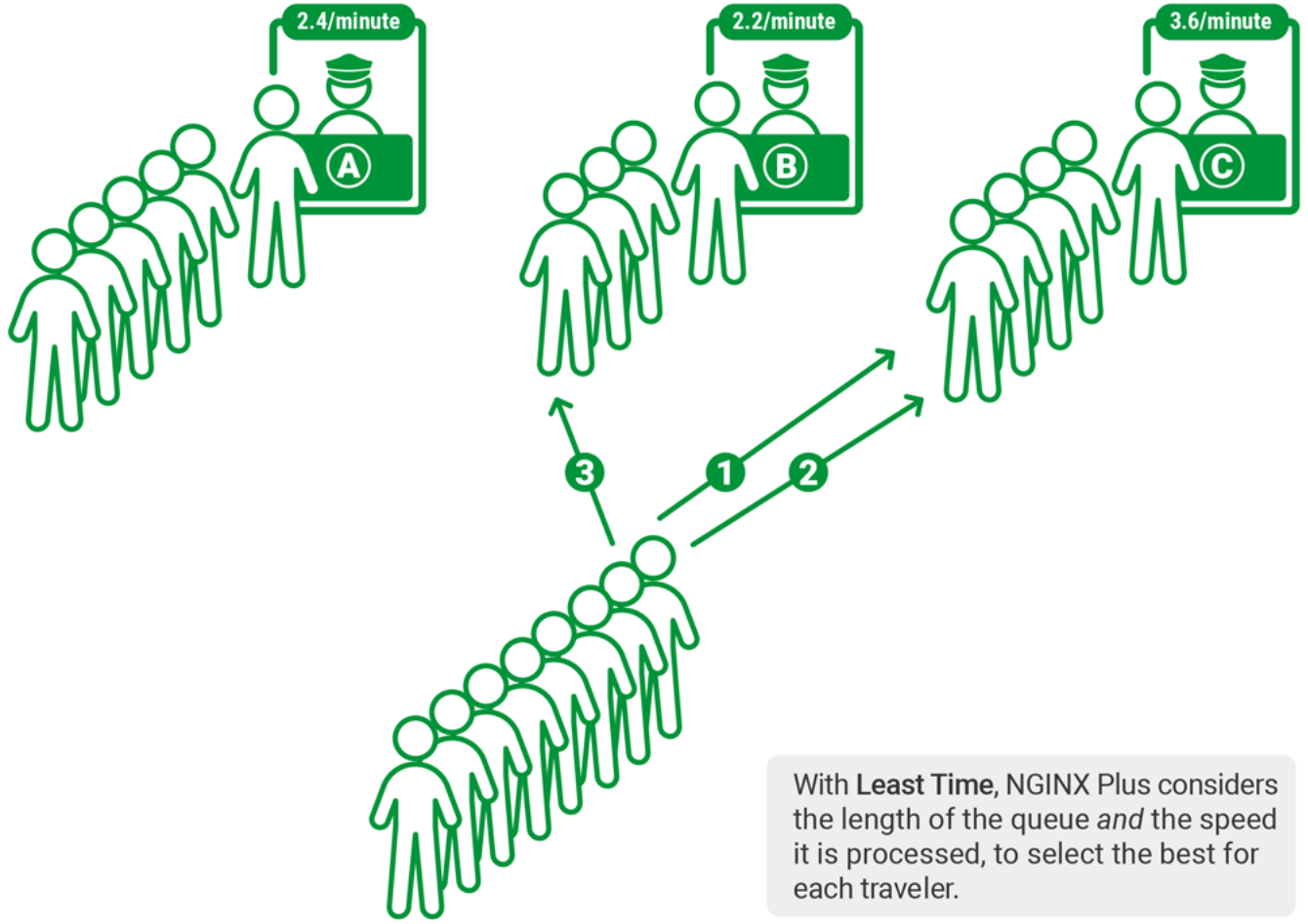
Round Robin


A
B
C
1
2
3
With **Round Robin**, NGINX sends each traveler to the next queue, no matter how long or slow that queue is.

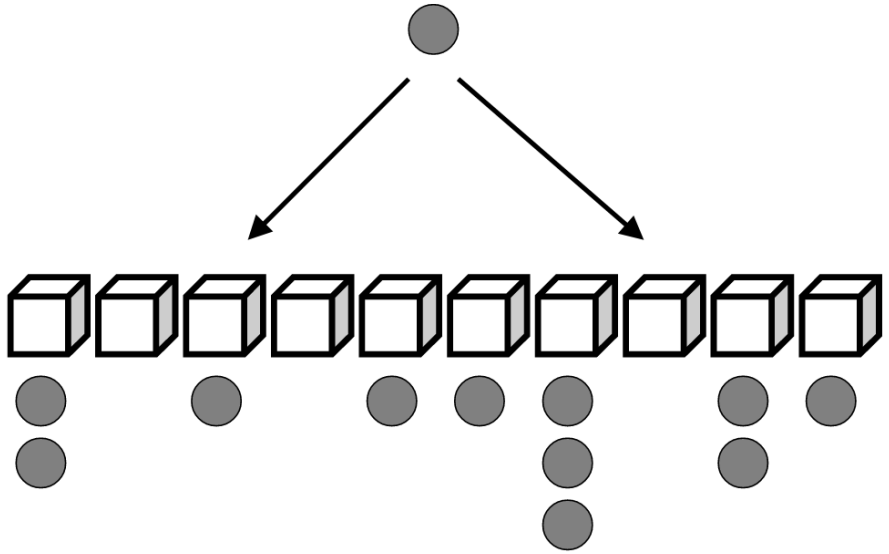
Least Connections / Shortest Queue First



Least Time

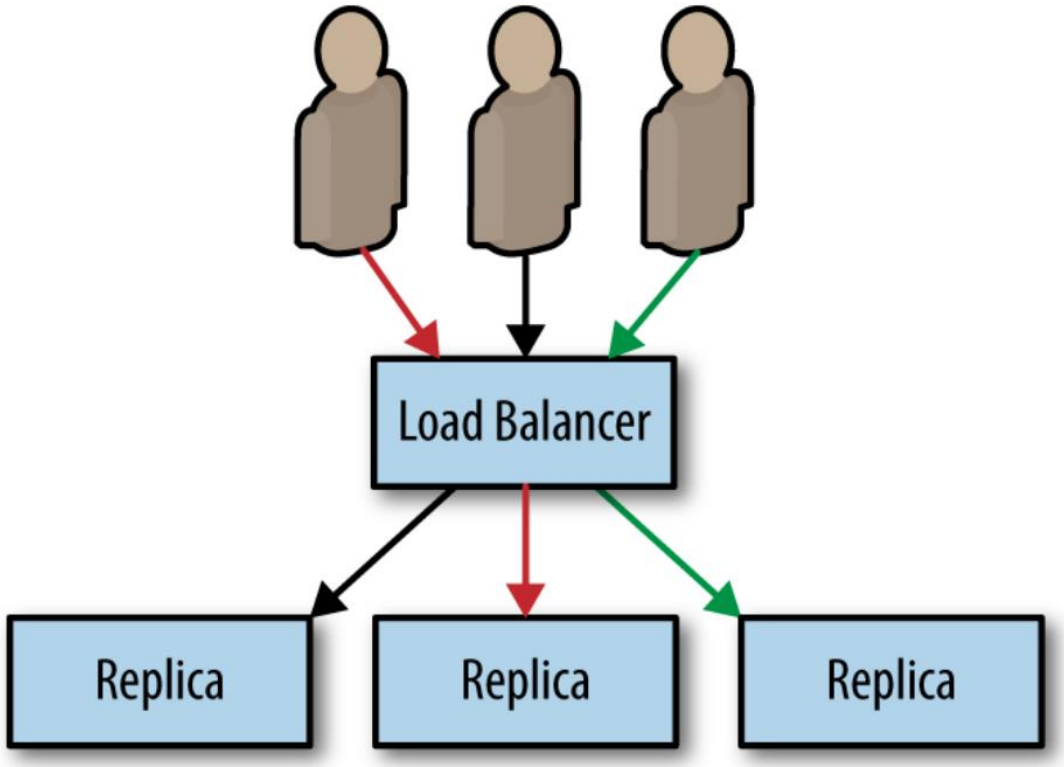

2.4/minute
A
2.2/minute
B
3.6/minute
C
3
1
2
With **Least Time**, NGINX Plus considers the length of the queue *and* the speed it is processed, to select the best for each traveler.

Power of Two Choices

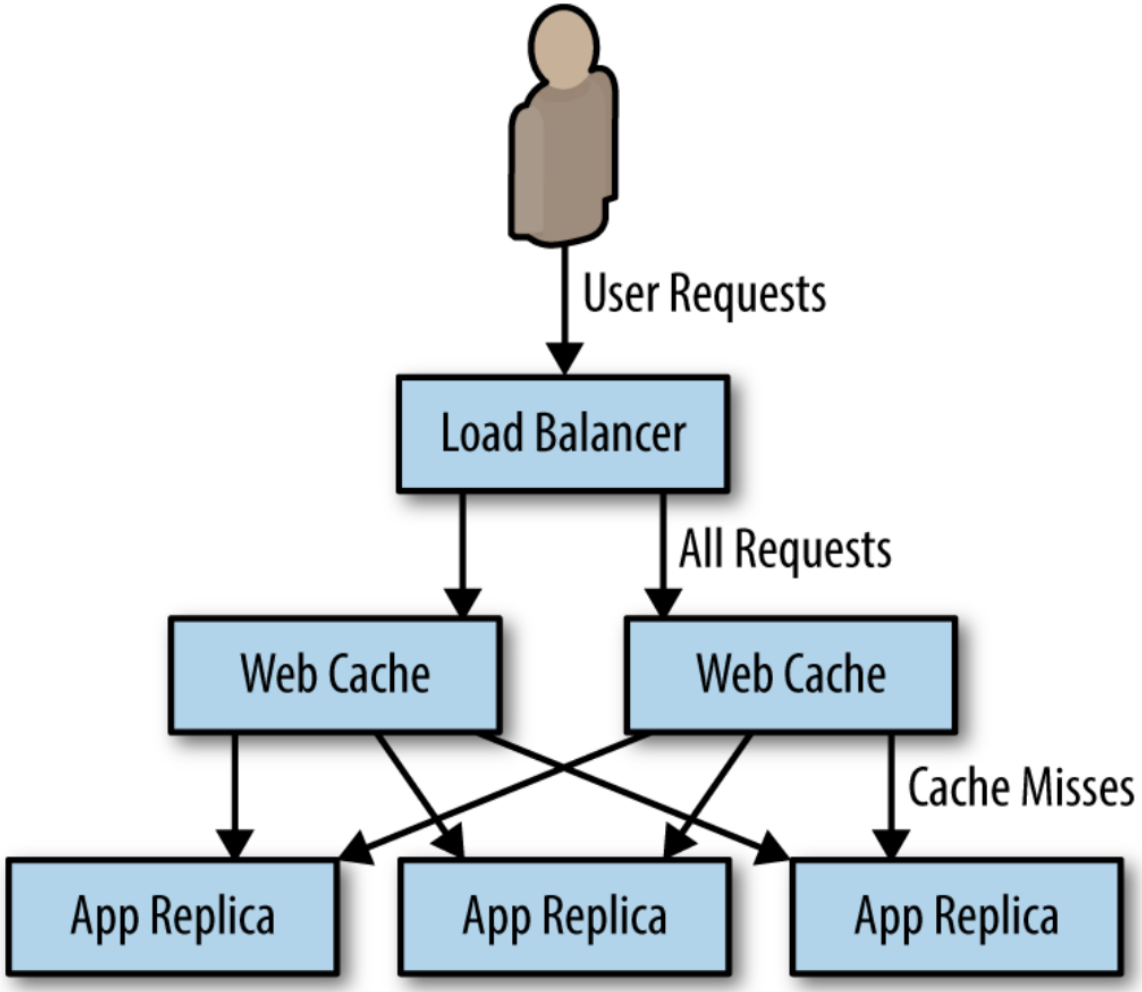


- Максимальная нагрузка:
  - Random: $O(\log n \,/\, \log \log n)$
  - Two random choices: $O(\log \log n)$
- The Power of Two Choices in Randomized Load Balancing (1996, 2001)

Отслеживание клиентских сессий



Кэширование



Варианты реализации кэша

- Кэш на стороне клиента
- Промежуточные кэширующие прокси-сервера
- Content Delivery Network (CDN)
- Caching HTTP reverse proxy (Varnish, Nginx)
- Специализированные хранилища (memcached, Redis)

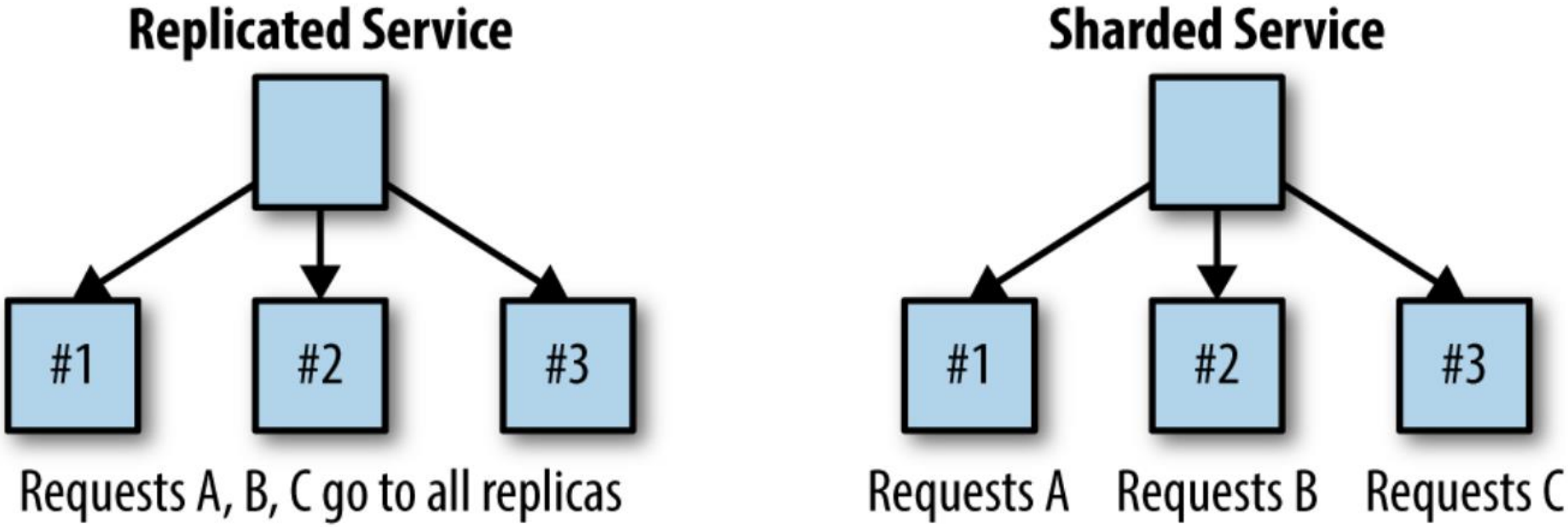
Политики вытеснения

- Least Recently Used
- Least Frequently Used
- Least Frequently Recently Used
- First In First Out

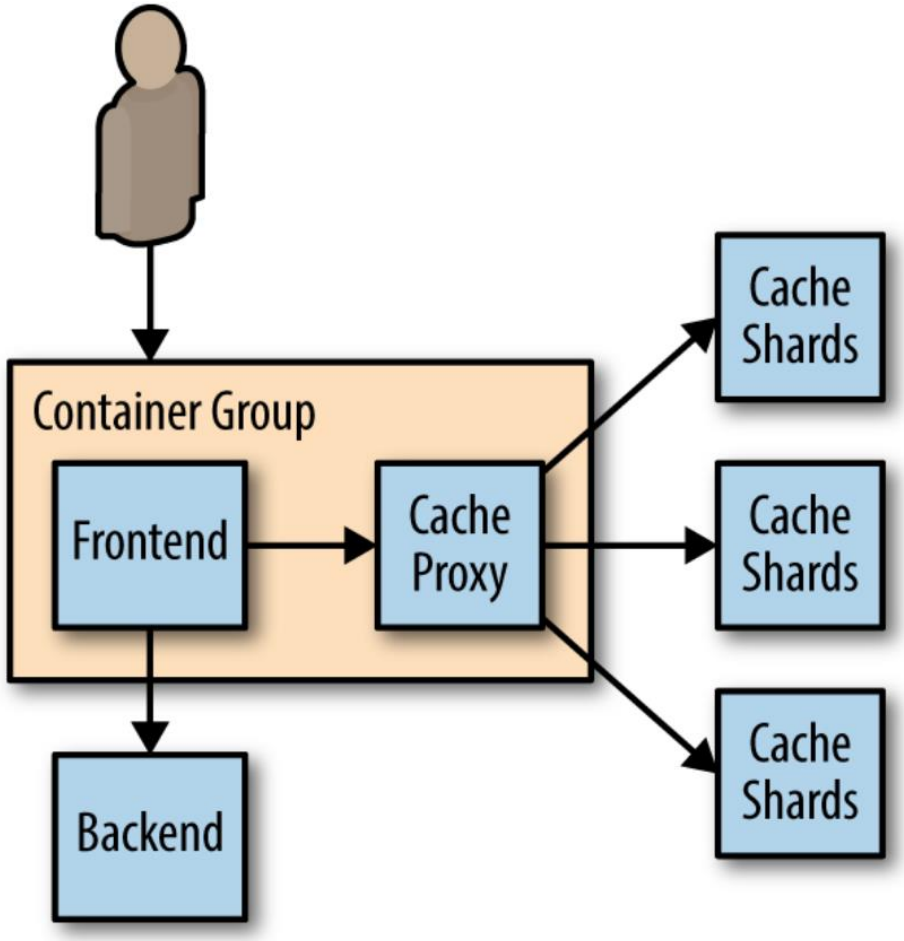
For a significant number of workloads, FIFO has similar or lower miss ratio performance as LRU for in-memory caching workloads.

A large scale analysis of hundreds of in-memory cache clusters at Twitter (2020)

Шардинг (stateful сервис)



Шардинг кэша

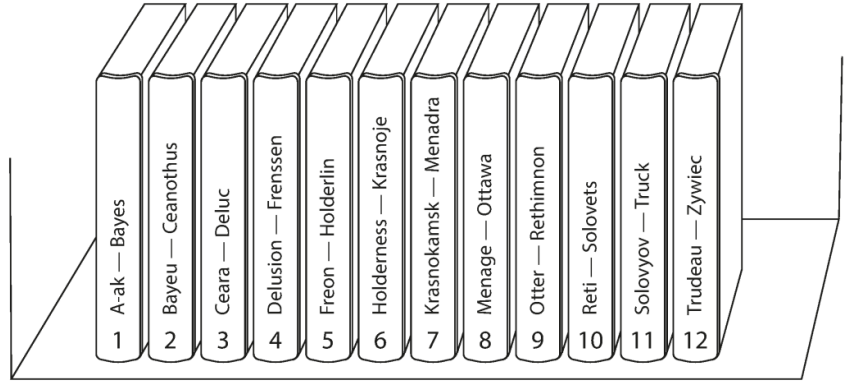


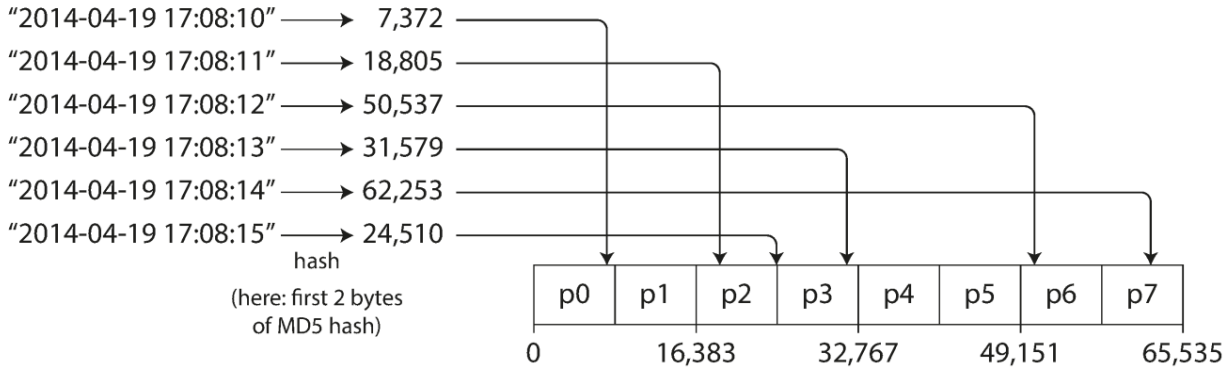
Принцип разбиения на шарды

- $Shard = ShardingFunction(ObjectKey)$
- Требования к функции
  - Детерминированность
  - Равномерность
  - Устойчивость к изменениям состава узлов (rebalancing)

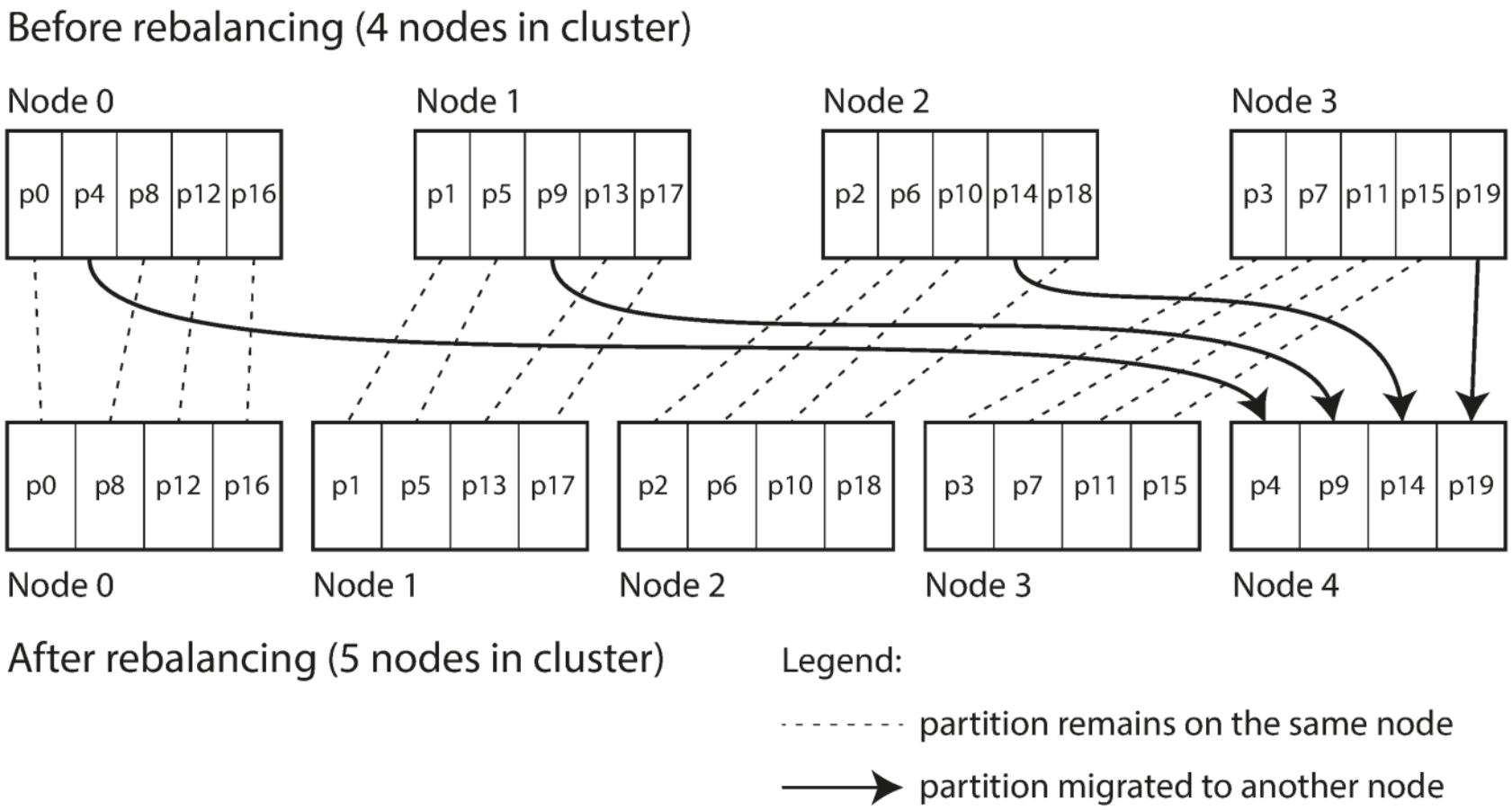
Подходы

- Вертикальный шардинг
- Горизонтальный шардинг
  - (Выбор ключа)
  - Интервалы ключей
  - Хэширование ключей
  - $hash(k)\ mod\ N$
  - Согласованное хеширование
- Число шардов
  - Фиксированное
  - Пропорционально числу узлов
  - Пропорционально размеру данных



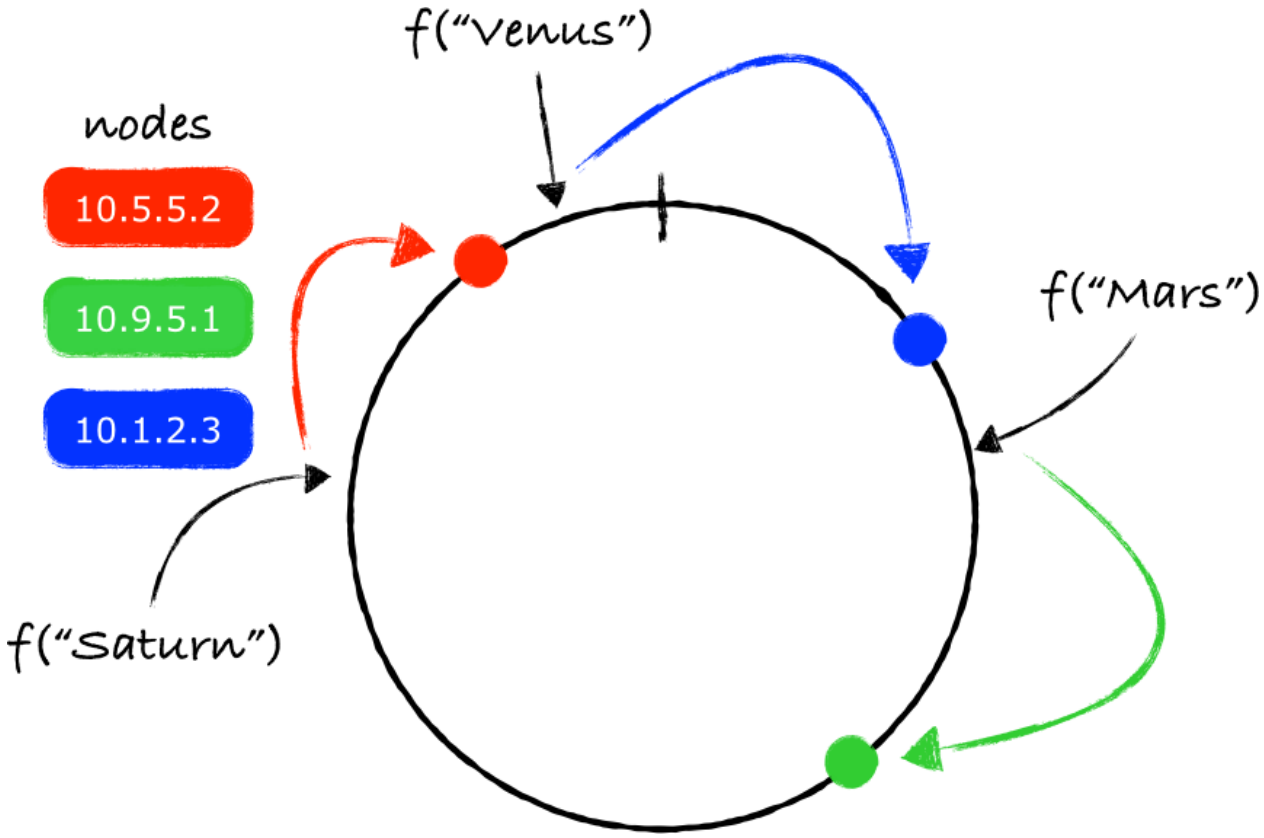


Фиксированное число шардов



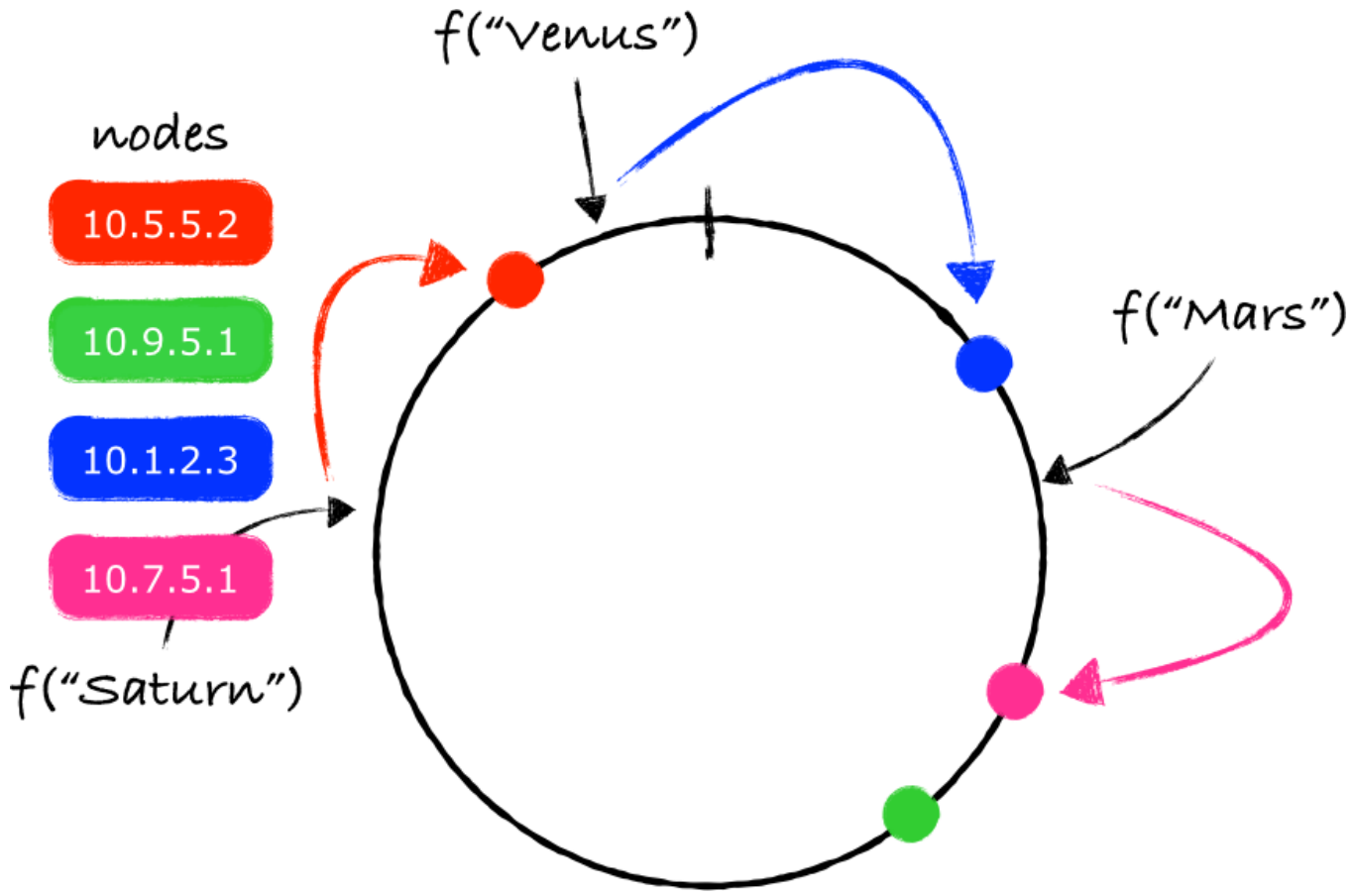
Согласованное хэширование

Consistent hashing and random trees: Distributed caching protocols... (1997)

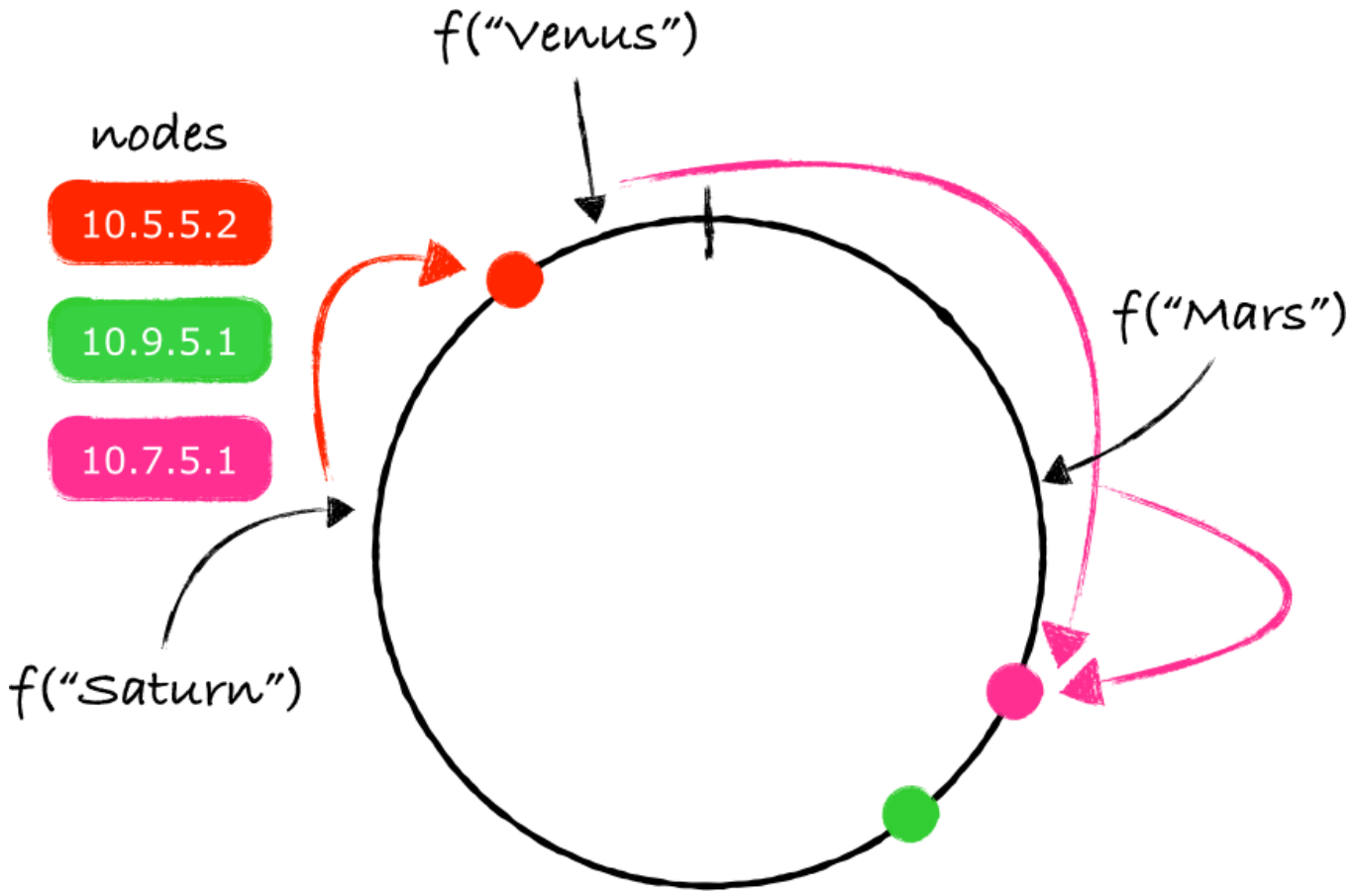


https://blog.carlosgaldino.com/consistent-hashing.html

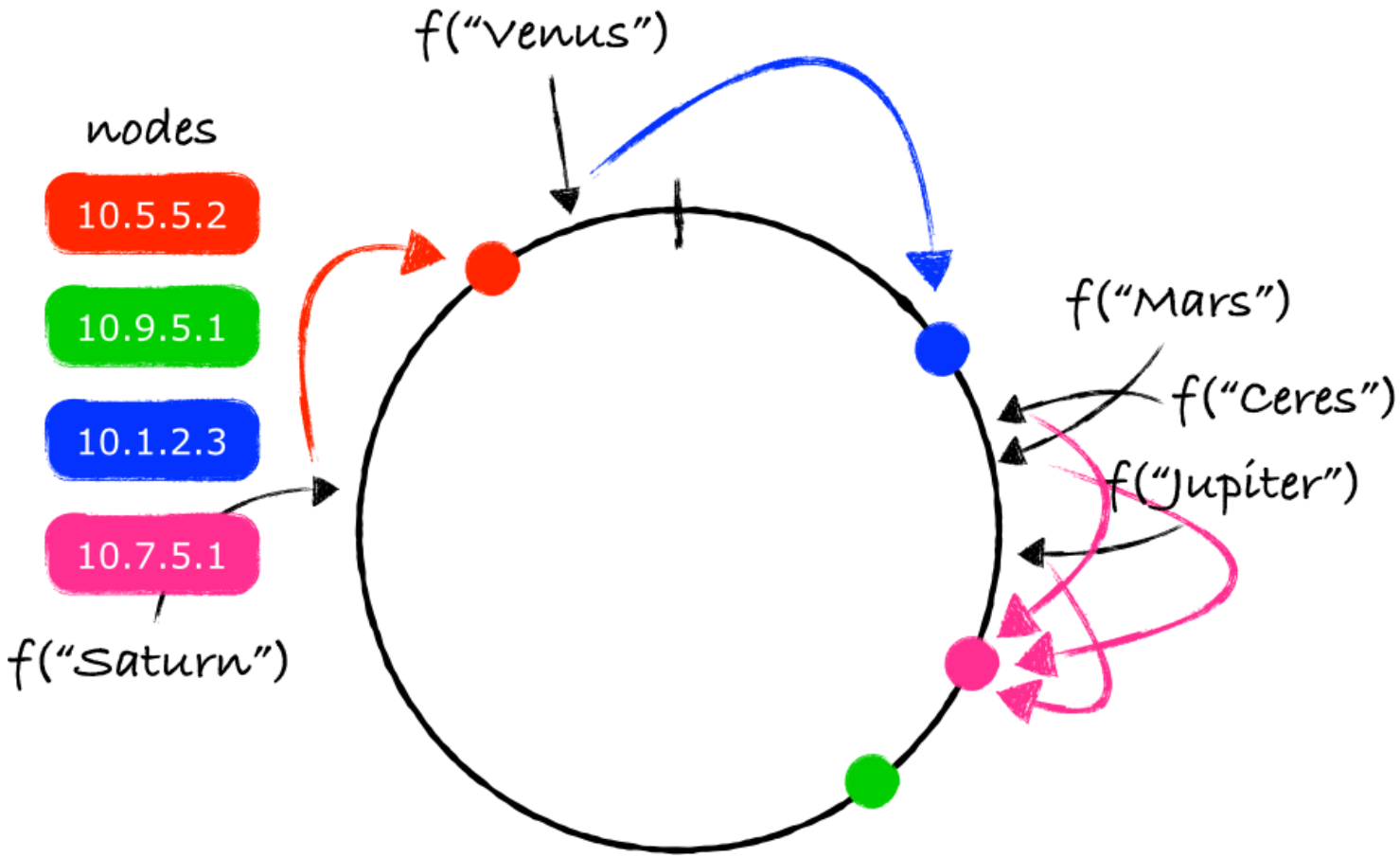
Добавление узла



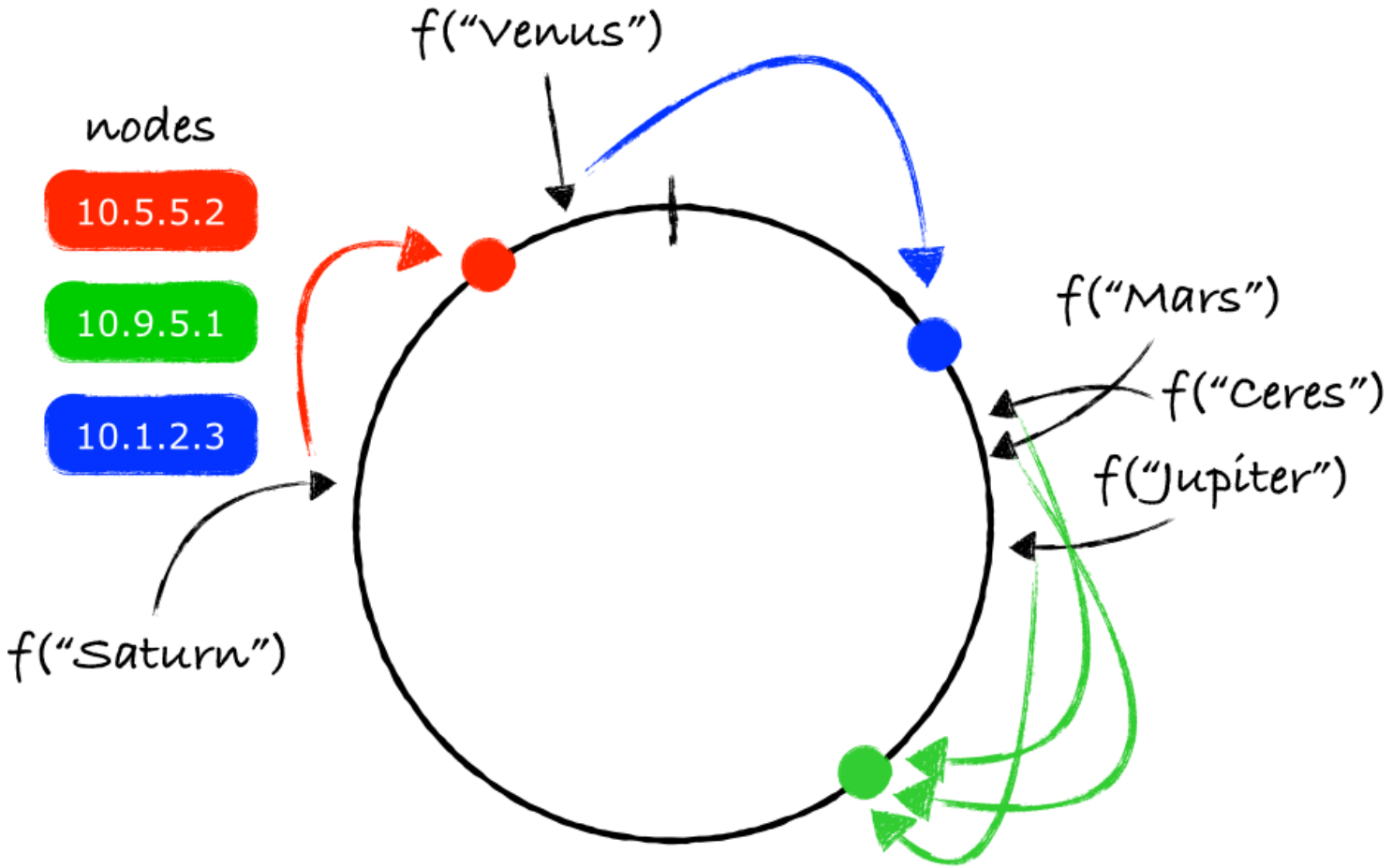
Удаление узла



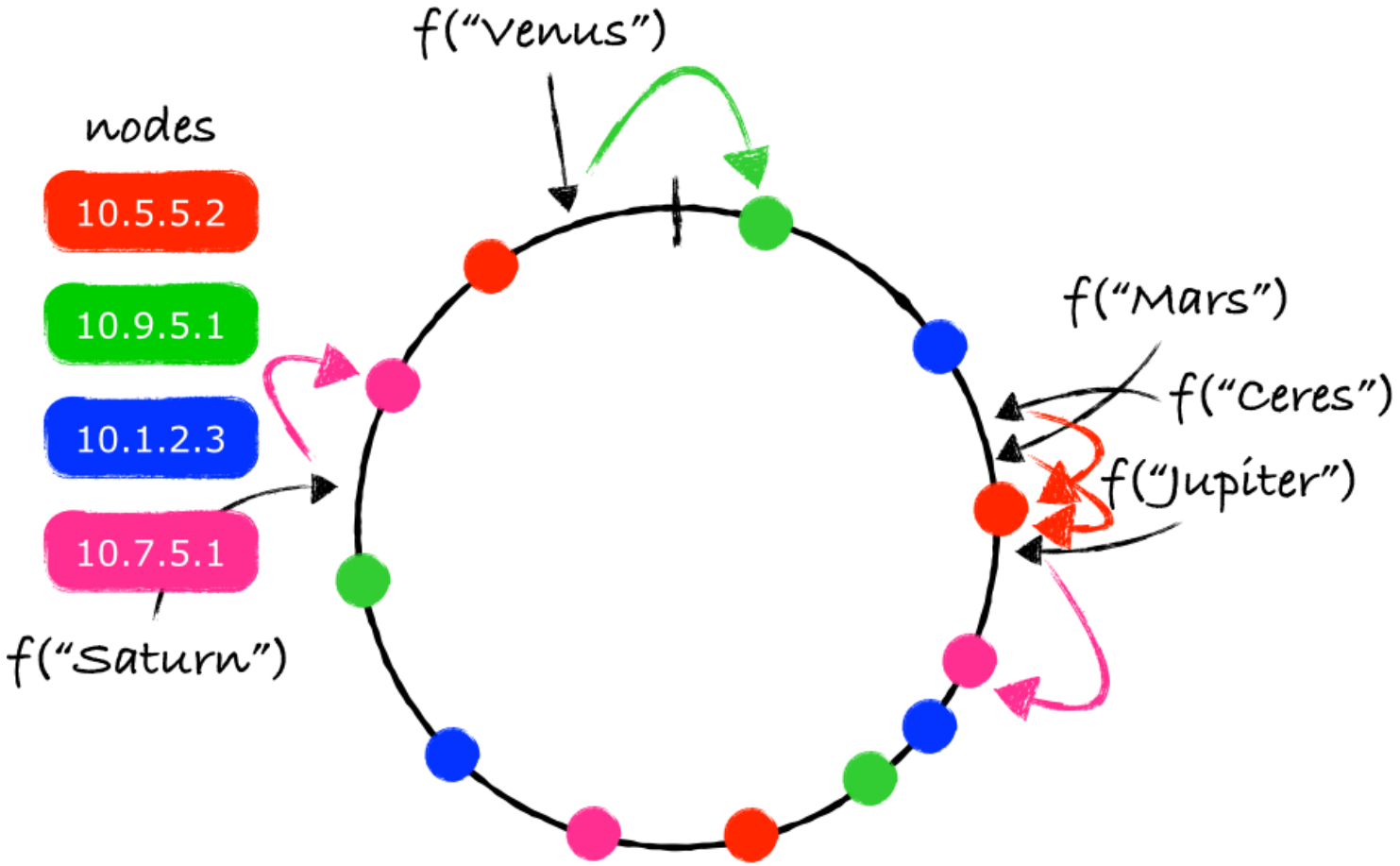
Проблема 1


f("Venus")
nodes
10.5.5.2
10.9.5.1
10.1.2.3
10.7.5.1
f("Saturn")
f("Mars")
f("Ceres")
f("Jupiter")

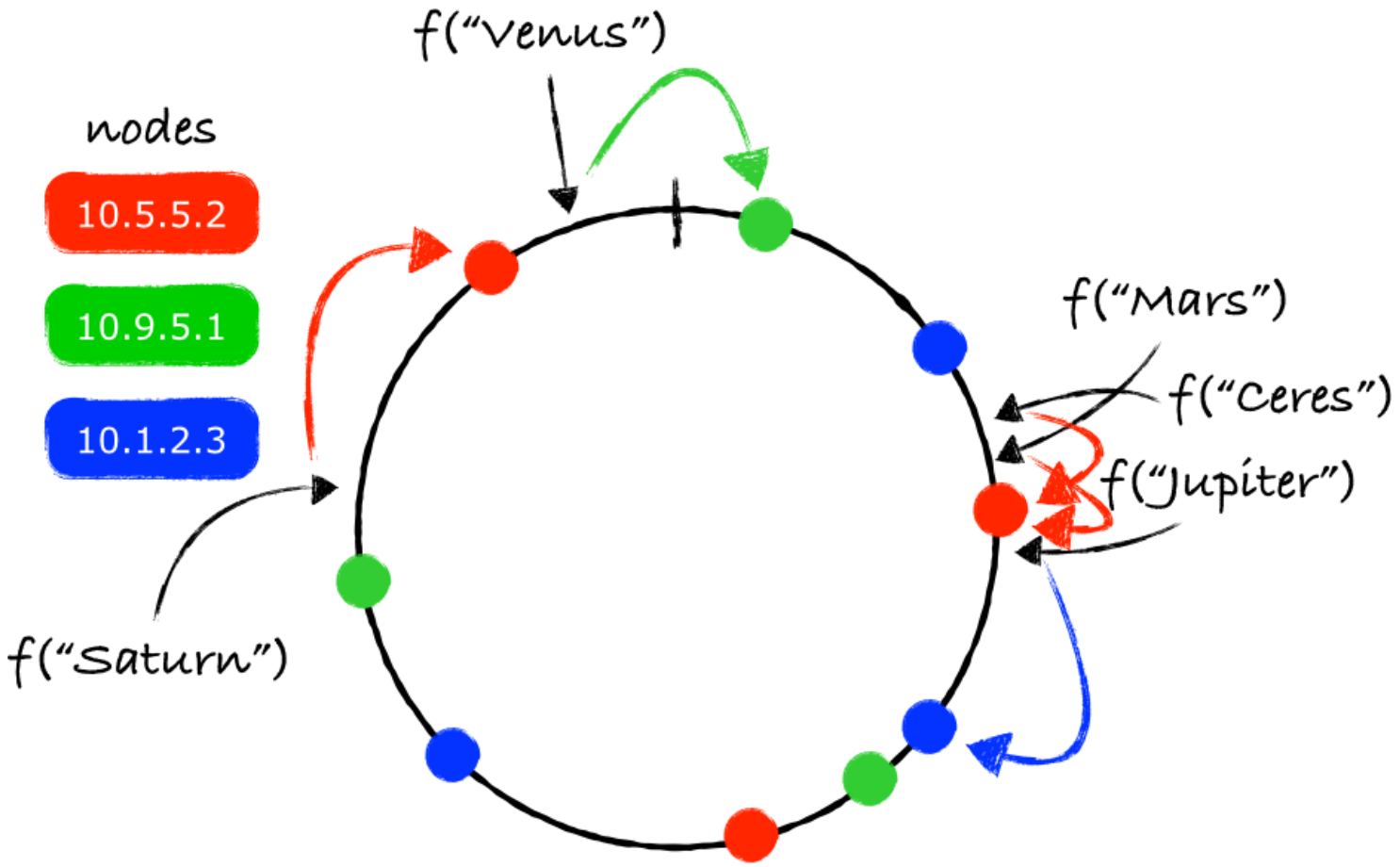
Проблема 2


f("Venus")
nodes
10.5.5.2
10.9.5.1
10.1.2.3
f("Mars")
f("Ceres")
f("Jupiter")
f("Saturn")

Виртуальные узлы



Удаление узла



Rendezvous Hashing

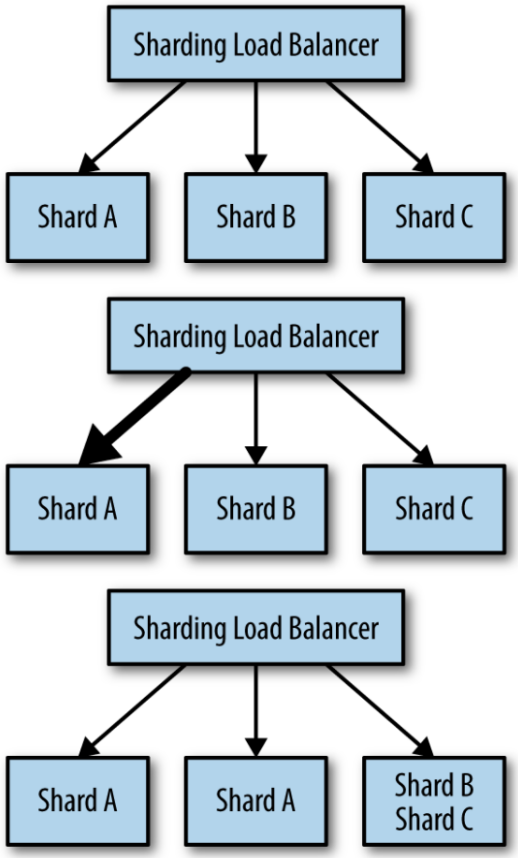
- Using name-based mappings to increase hit rates (1998)
- Другое название - Highest Random Weight (HRW)
- Вес сервера для ключа вычисляется как $hash(key, server)$
- Выбирается сервер с максимальным весом

Другие подходы

- A fast, minimal memory, consistent hash algorithm (2014)
- Multi-probe consistent hashing (2015)
- Maglev: A fast and reliable software network load balancer (2016)
- Consistent Hashing with Bounded Loads (2016)

Балансировка нагрузки при шардинге

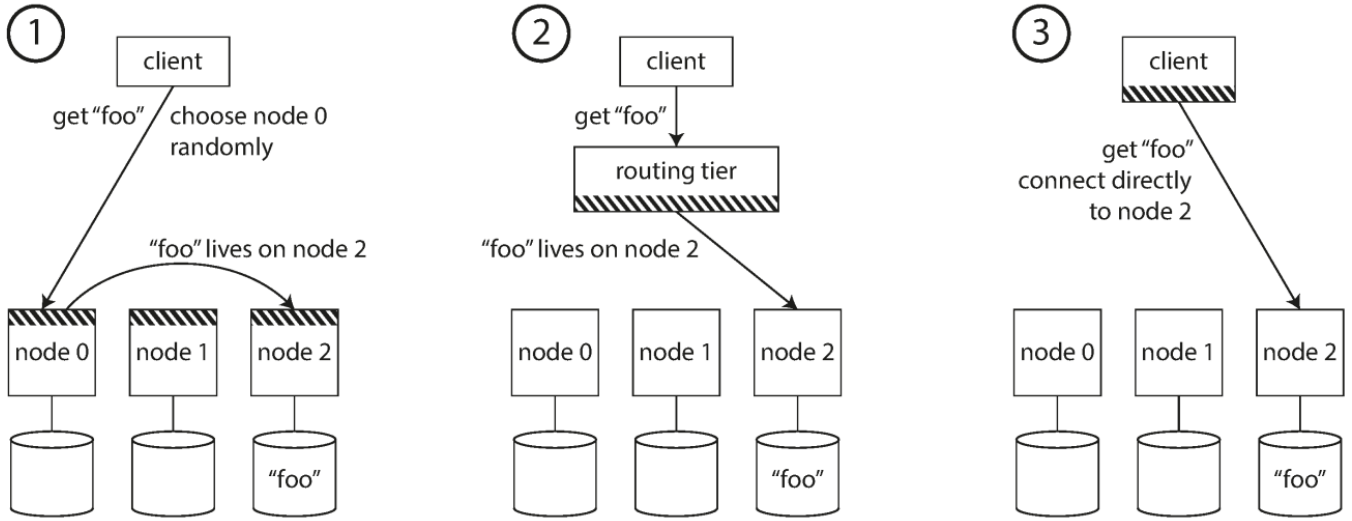
- Выбор "хорошего" принципа разбиения на шарды
  - Не гарантирует отсутствие "горячих" шардов
- Микро-шарды
  - Число шардов >> машин
- Реконфигурация шардов
  - Разбиение и слияние шардов
- Выборочная репликация шардов
  - Изменение числа реплик в зависимости от нагрузки



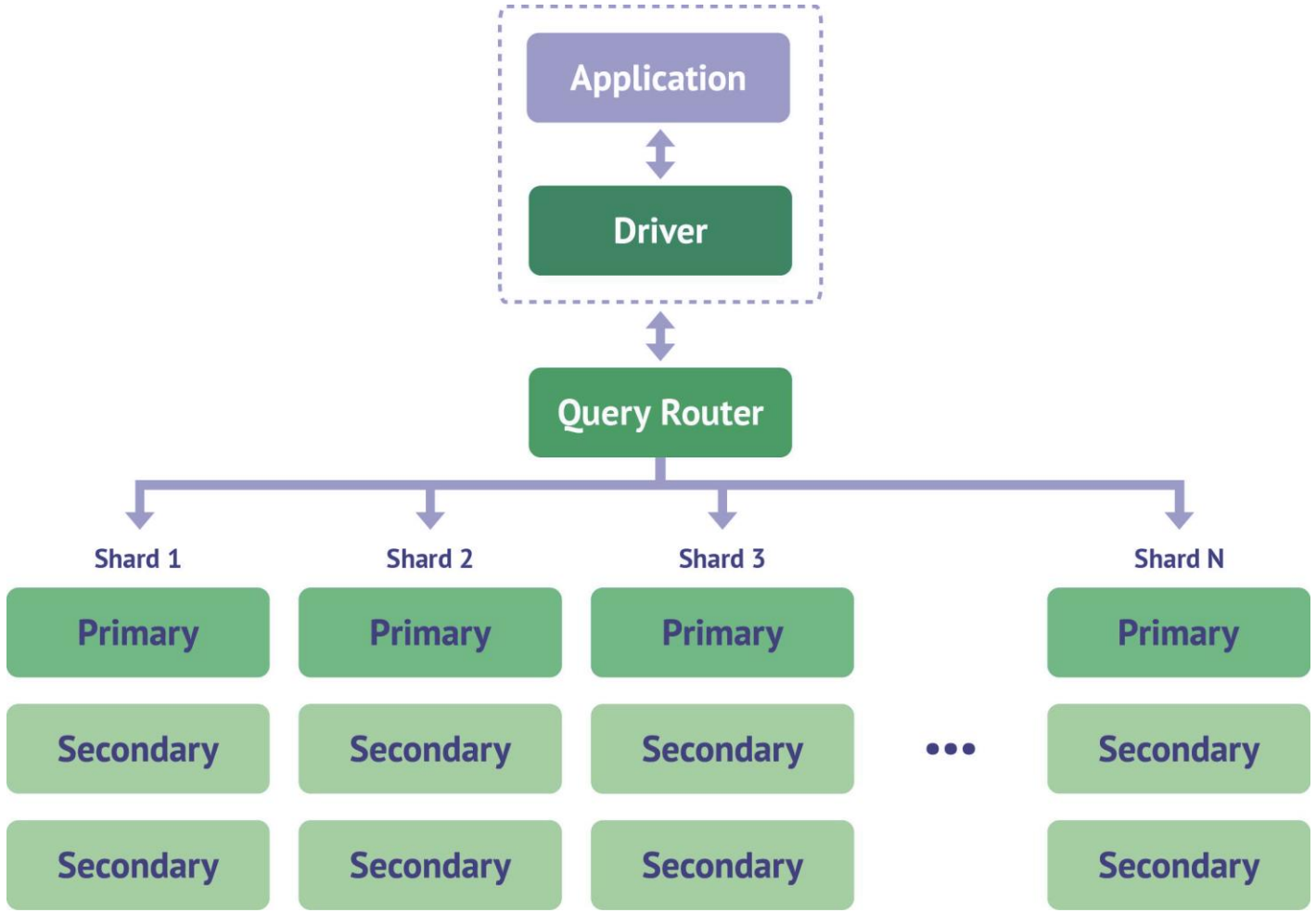
Маршрутизация запросов

Как клиент определит, на какой узел надо отправить запрос?

1. Обратиться к любому узлу, а тот перенаправит запрос при необходимости
2. Использовать отдельный промежуточный слой, знающий о шардах
3. Хранить информацию (шард, узел) на клиенте, возможно частично (DHT)



Шардинг + репликация данных



Материалы

- Designing Distributed Systems: Patterns and Paradigms for Scalable, Reliable Services (главы 5-6)
- Site Reliability Engineering (главы 19-20)
- Designing Data-Intensive Applications (глава 6)
- Introduction to modern network load balancing and proxying
- NGINX and the “Power of Two Choices” Load-Balancing Algorithm
- CS265: Balls in Bins and Power-of-Two-Choices
- CS168: Consistent Hashing
- Consistent Hashing: Algorithmic Tradeoffs

Материалы 2

- Test Driving “Power of Two Random Choices” Load Balancing
- Predictive Load Balancing: Unfair but Faster & more Robust
- Load Balancing is Impossible
- How Etsy caches: hashing, Ketama, and cache smearing
- Building a Consistent Hashing Ring
- Теория шардирования
- Упомянутые статьи