

The background of the slide is a large, stylized eye. The iris is a light blue circle containing a white database diagram with three nodes and connecting lines. The pupil is a bright white starburst. The eyelids are grey, and the eyelashes are long, thin, and grey.

Apache Cassandra

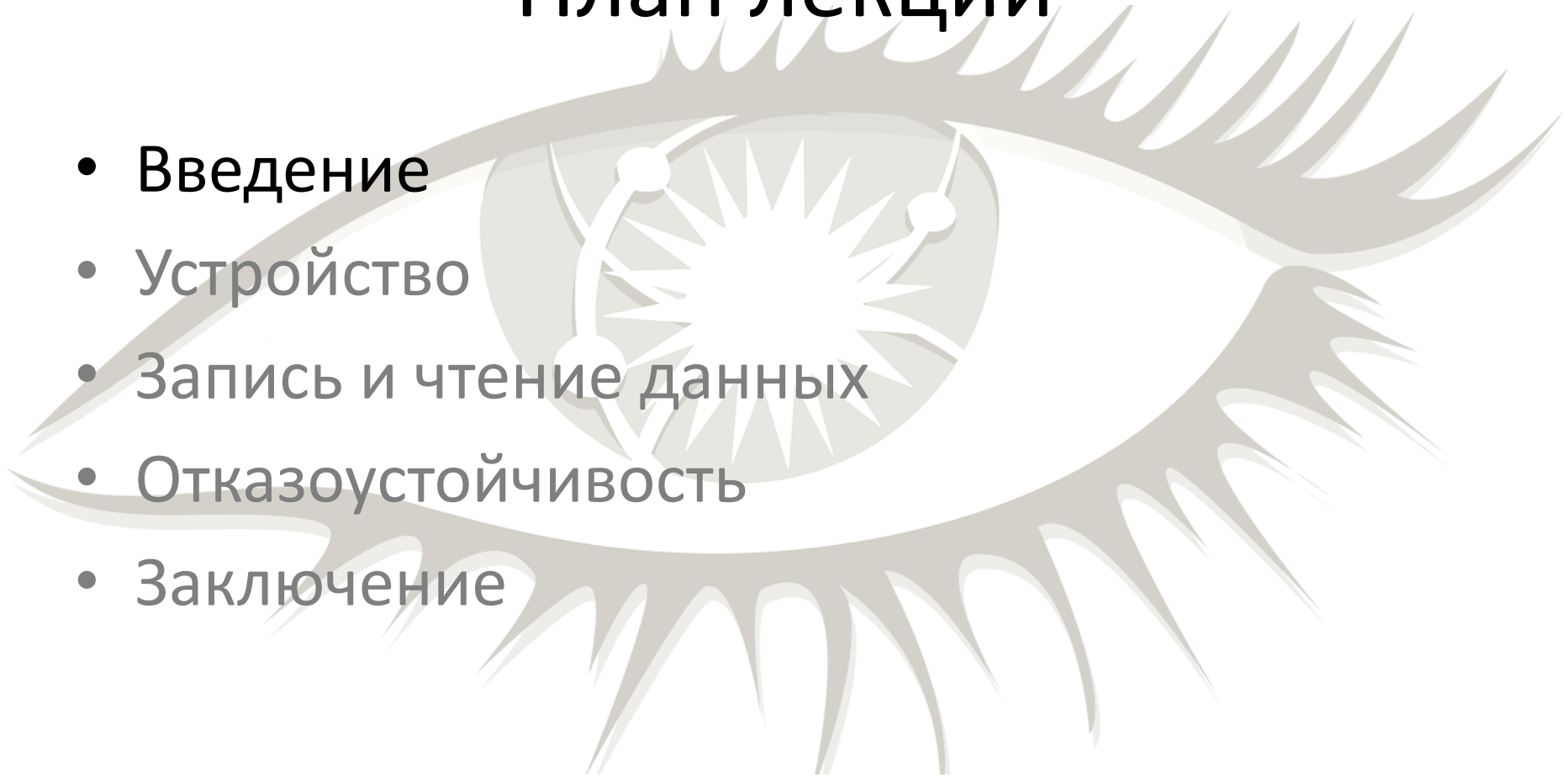
Иван Пузыревский
Разработчик Яндекс

Презентация: Ивченко Олег, 094а

Москва, 2015 (Апрель)

План лекции

- Введение
- Устройство
- Запись и чтение данных
- Отказоустойчивость
- Заключение

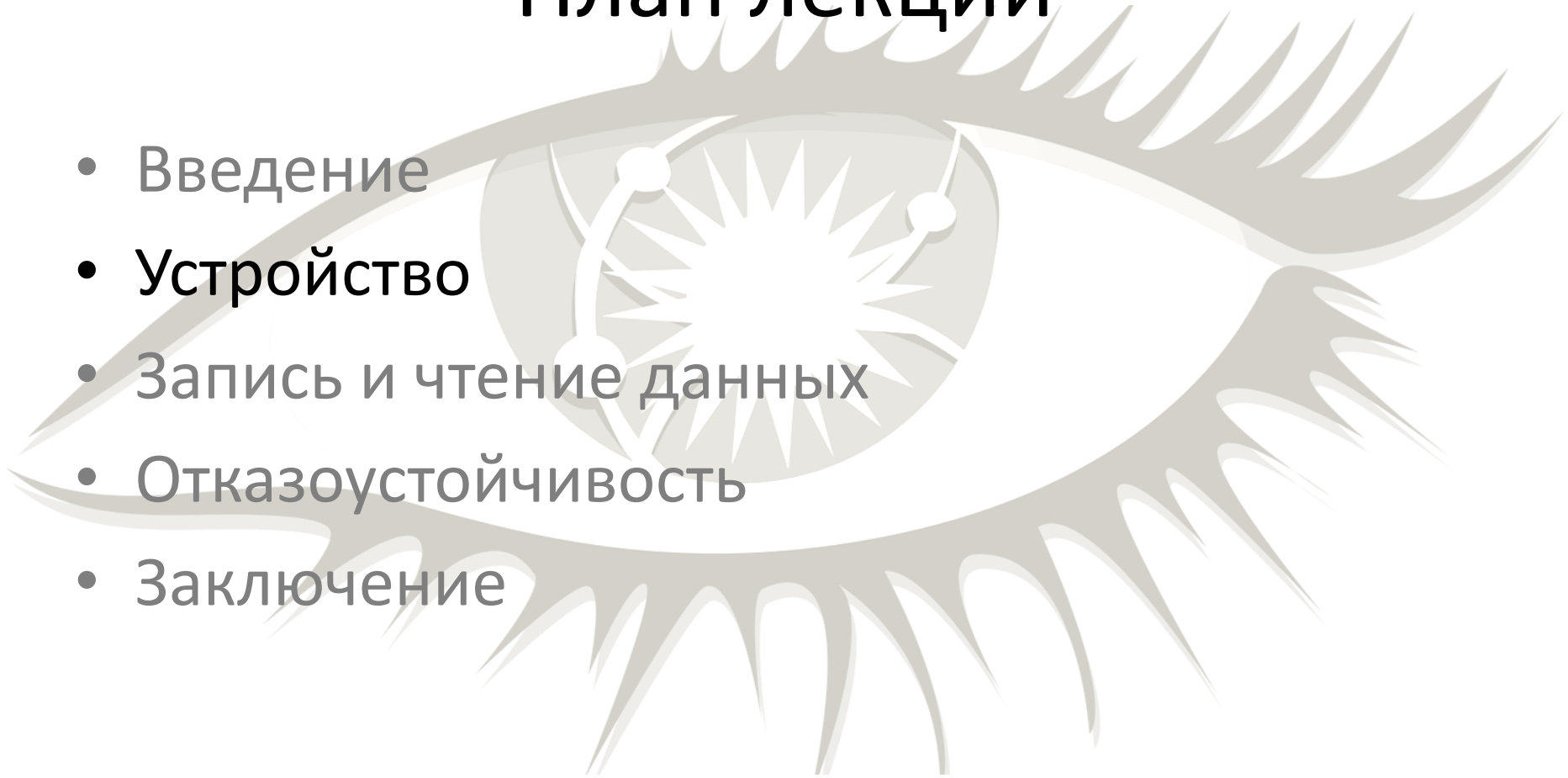


Предпосылки

- Amazon – интернет-магазин:
 - Нужно обрабатывать корзины
 - т.е. данные, которые имеют версии
 - Хранить изображения товаров
 - т.е. редко изменяемые данные

План лекции

- Введение
- Устройство
- Запись и чтение данных
- Отказоустойчивость
- Заключение



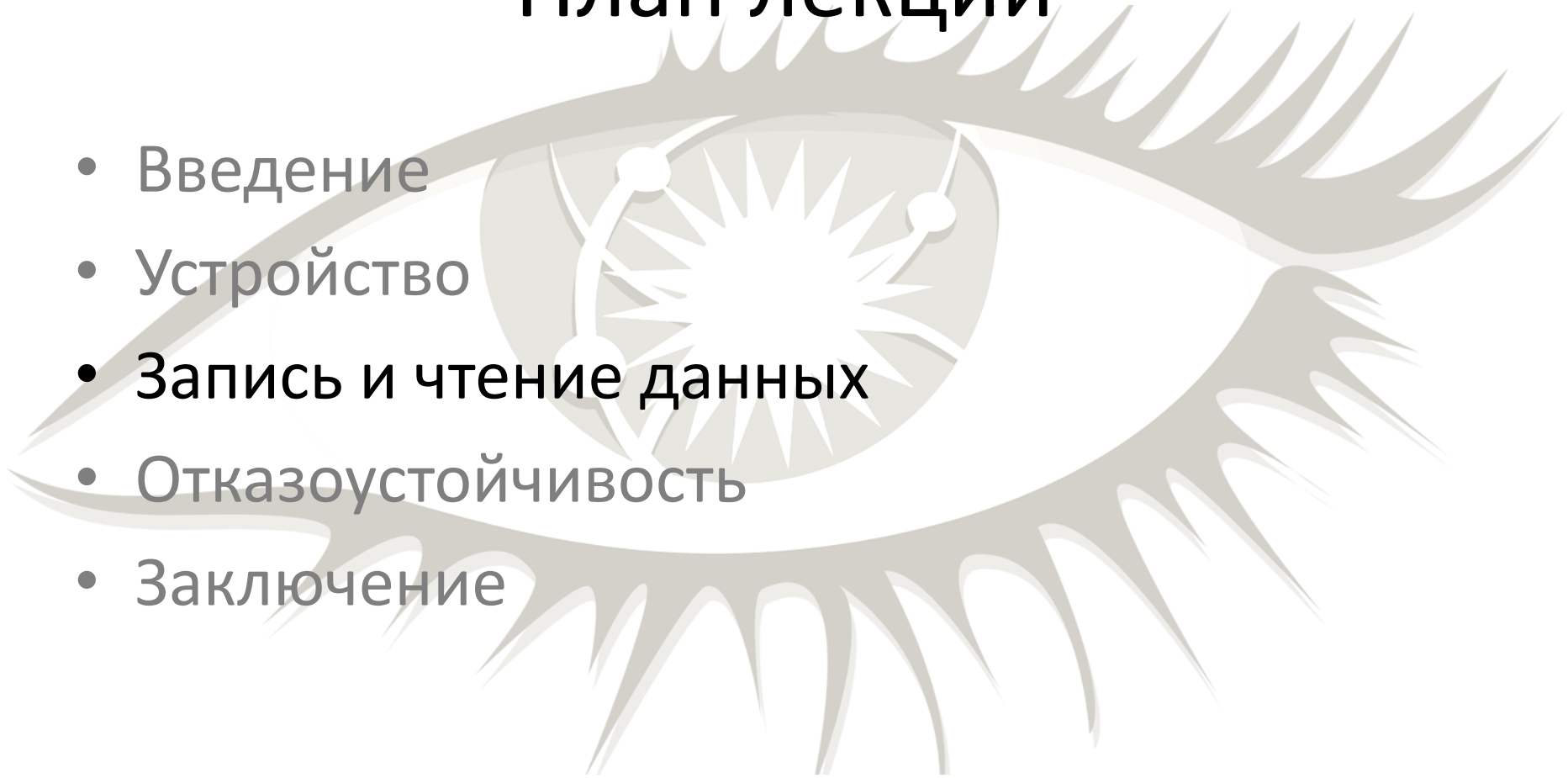
Устройство

- Хранилище вида key-value
- Нет единого центра
 - Часть функций Master server передана клиенту
- Консистентность:
 - Запись высокодоступна
 - Запись будет успешной если есть **хотя бы 1** живая машина
 - Высокодоступность чтения снижена (могут возвращаться несколько устаревшие данные)

Модель **eventual consistency**

План лекции

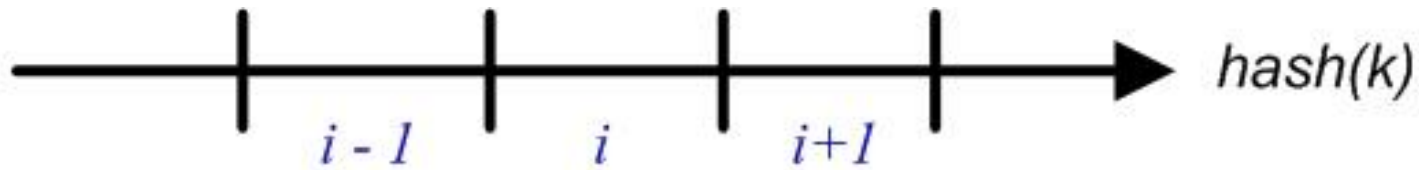
- Введение
- Устройство
- **Запись и чтение данных**
- Отказоустойчивость
- Заключение



Партиционирование

- Алгоритм

1. Для каждого ключа k рассчитывается $hash(k)$
2. Пространство хешей разбивается на участки

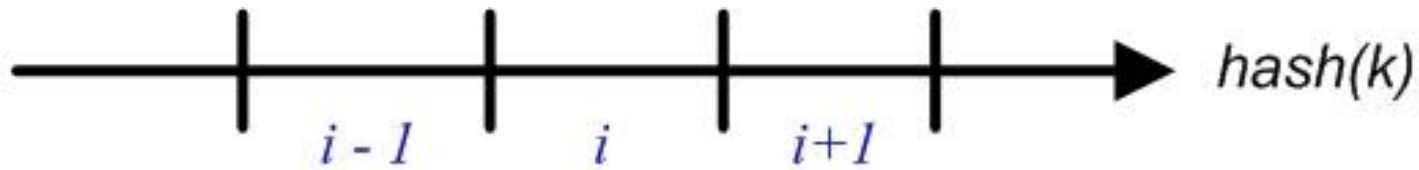


- Плюсы / минусы (по сравнению с HBase)?

Партиционирование

- Алгоритм

1. Для каждого ключа k рассчитывается $hash(k)$
2. Пространство хешей разбивается на участки



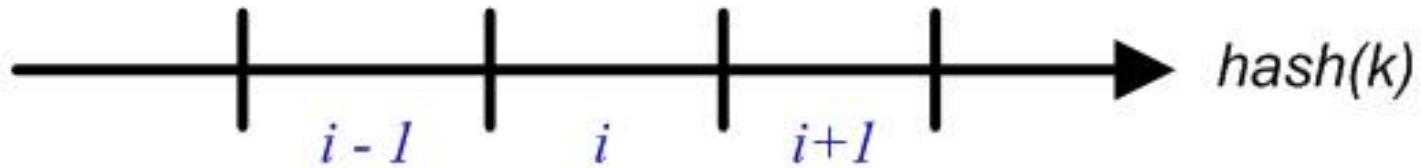
- Плюсы / минусы (по сравнению с Hbase)

- + данные распределяются равномерно
- нелинейное чтение (2 соседних ключа могут быть на разных машинах)

Реплицирование

- Наивное решение

- Каждой машине выдаём диапазон хешей

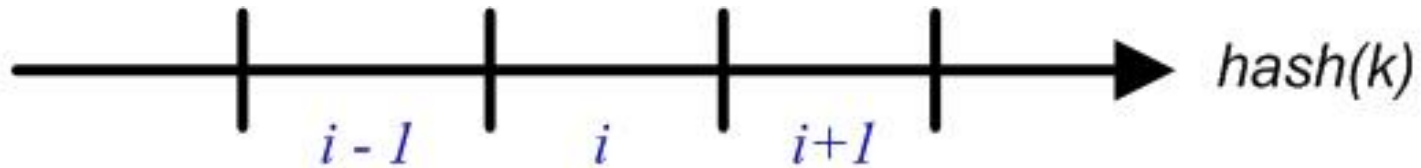


- Недостатки?

Реплицирование

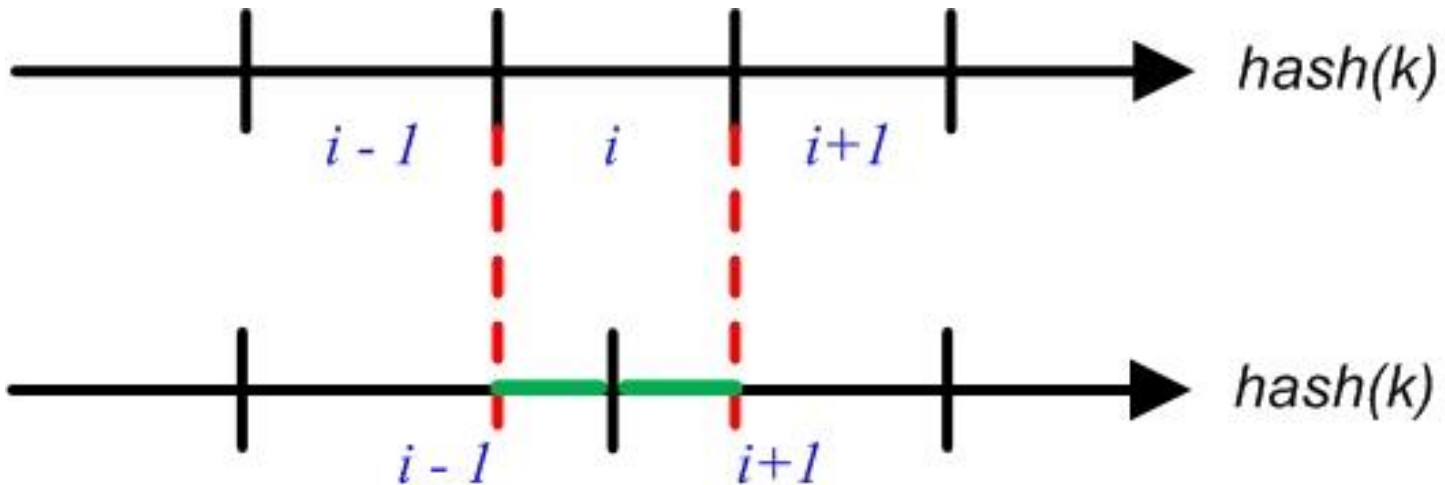
- Наивное решение

- Каждой машине выдаём диапазон хешей



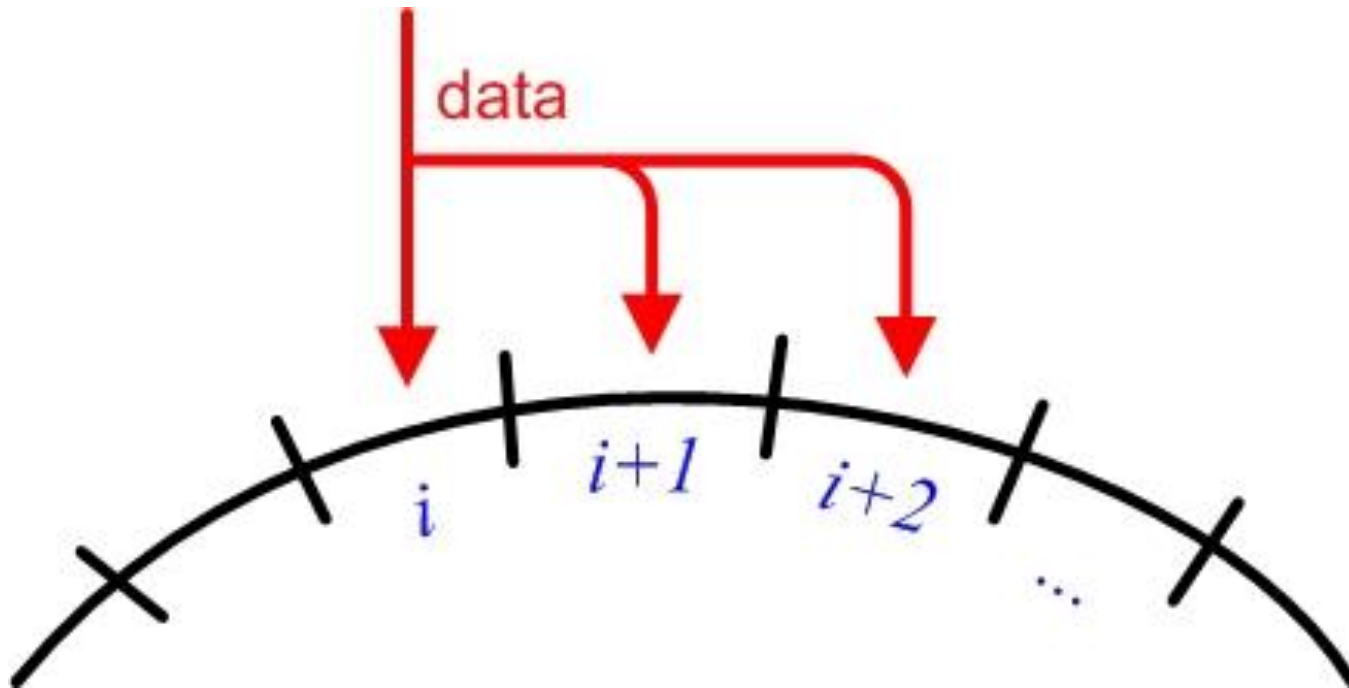
- Недостатки

- при добавлении / удалении машин нужно перемещать много данных



Реплицирование

- Консистентное хеширование
 - Изменения **не** приводят к большому перемещению данных (затронуты только ближайшие машины)



- Диапазоны имеют разные размеры

Реплицирование

- Консистентное хеширование

- Регулируется двумя параметрами:

- R – со сколько реплик можем читать копии
 - W – на сколько реплик копируются данные

$R + W > N \Rightarrow$ консистентное чтение и запись (N – целевое кол-во реплик)

Реплицирование

- Проблемы

- Конфликты копий

- Можно ввести версию Каждому записываемому значению автоматически присписывается № версии
 - В зависимости от структуры данных возможны разные решения (выбираются на стороне клиента)
 - Множество – объединение конфликтных копий
 - Экстремальное значение
 - Значение самой последней версии
 - ...

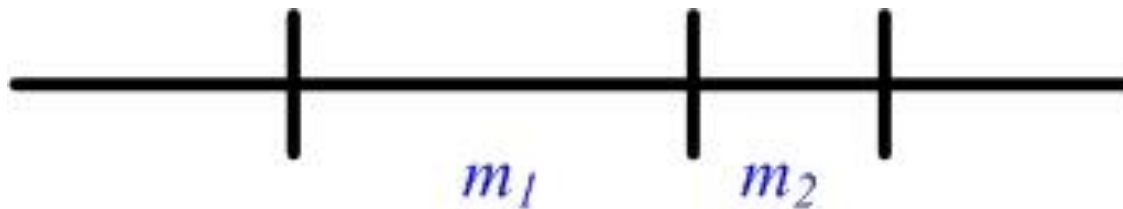
Реплицирование

- Проблемы

- Конфликты копий

- Можно ввести версию Каждому записываемому значению автоматически приписывается № версии
 - В зависимости от структуры данных возможны разные решения
 - Множество – объединение конфликтных копий
 - Экстремальное значение
 - Значение самой последней версии
 - ...

- Переполнение на машине



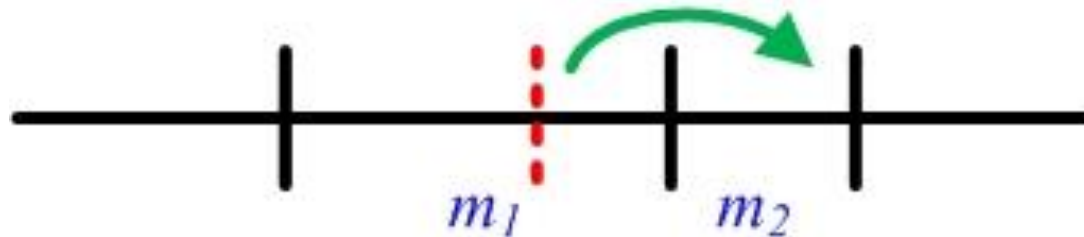
Реплицирование

- Проблемы

- Конфликты копий

- Можно ввести версионность Каждому записываемому значению автоматически приписывается № версии
 - В зависимости от структуры данных возможны разные решения
 - Множество – объединение конфликтных копий
 - Экстремальное значение
 - Значение самой последней версии
 - ...

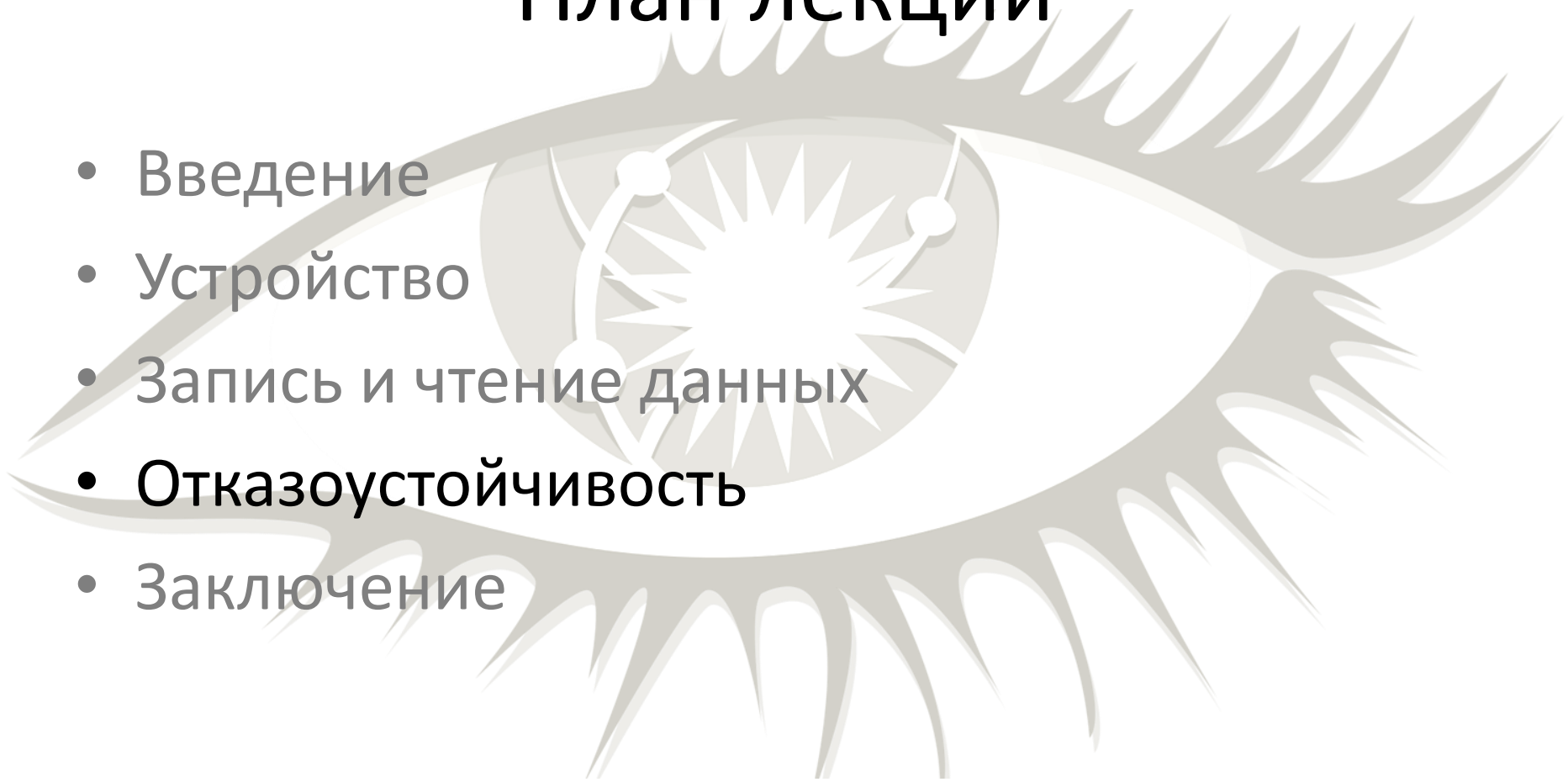
- Переполнение на машине



- Добавляется нов. точка разбиения
 - Следующая машина забирает данные из новой области

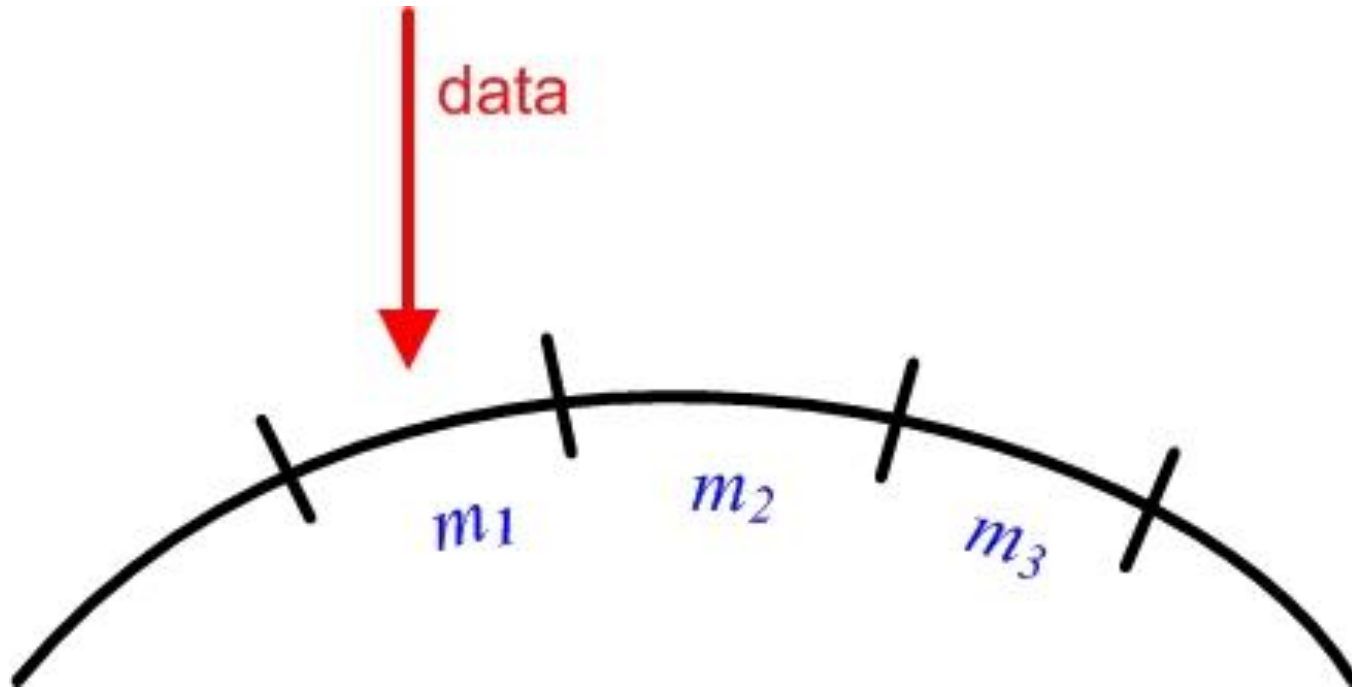
План лекции

- Введение
- Устройство
- Запись и чтение данных
- **Отказоустойчивость**
- Заключение



Восстановление данных

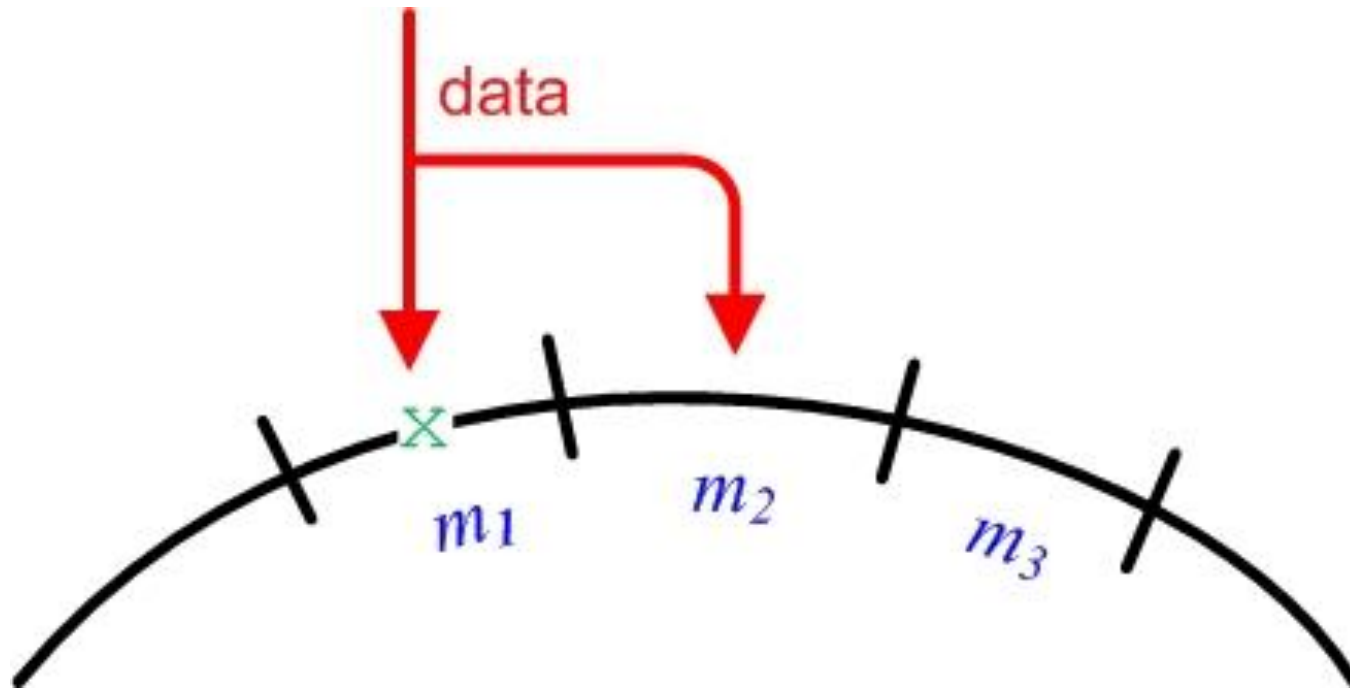
- Временные (temporary) выпадения



Данные приходят на машину m_1

Восстановление данных

- Временные (temporary) выпадения



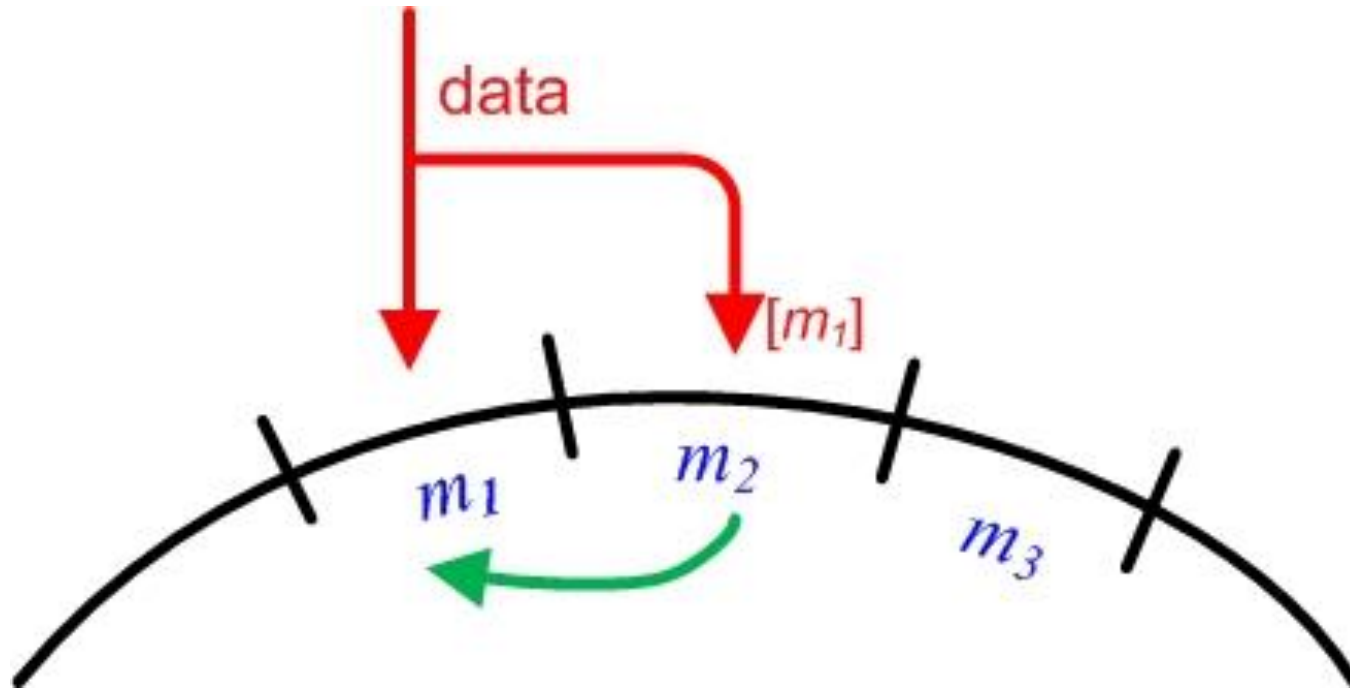
Машина m_1 падает, данные передаются на m_2

- Когда m_1 оживёт, на ней будет старая версия
- Если $R = 1$, то при чтении получим устаревшие данные

Решение?

Восстановление данных

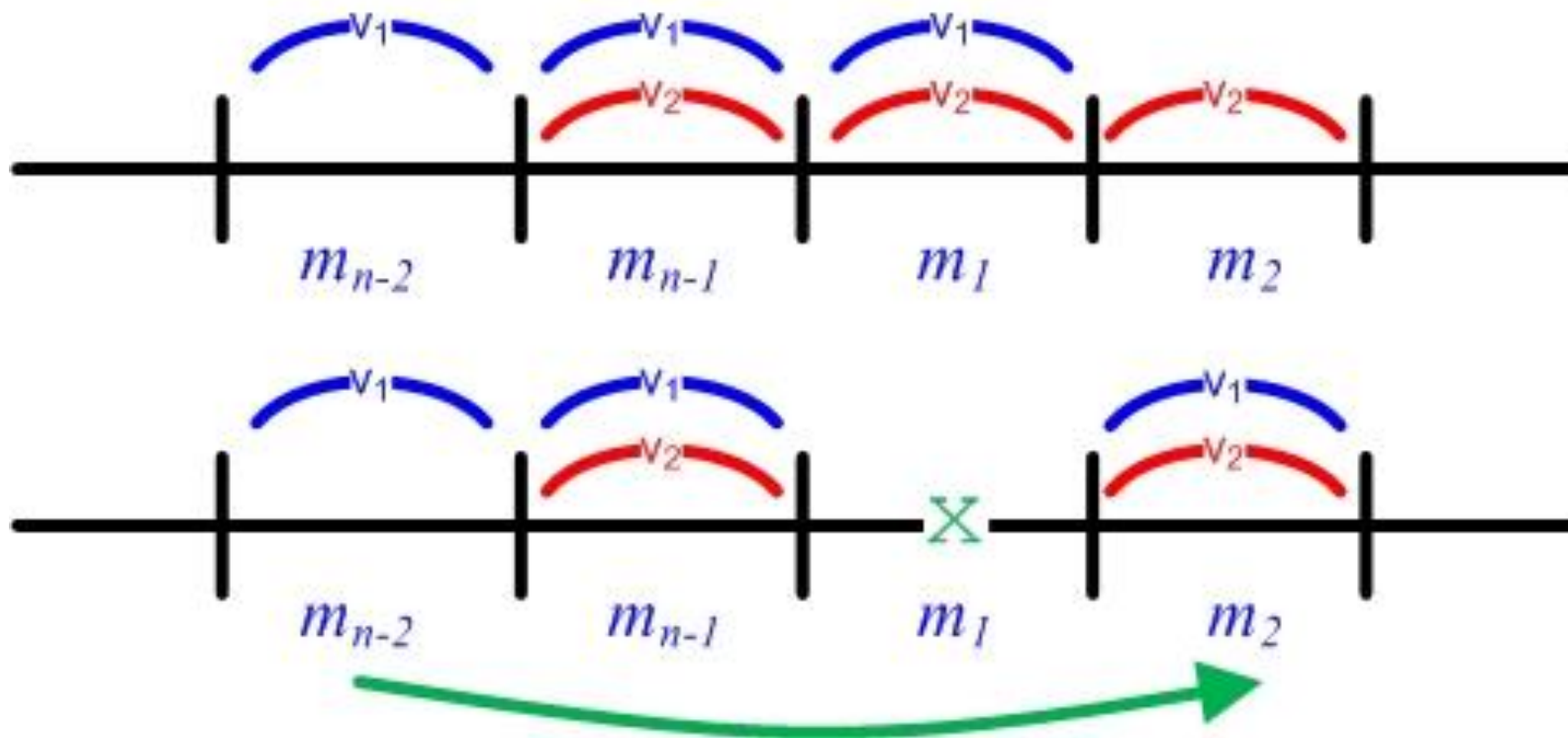
- Hinted handoff



- Данные аннотируются подсказкой m_1
- Каждая машина хранит список подсказок и проверяет не поднялись ли машины-члены списка
- По поднятии машины, на неё копируются данные

Восстановление данных

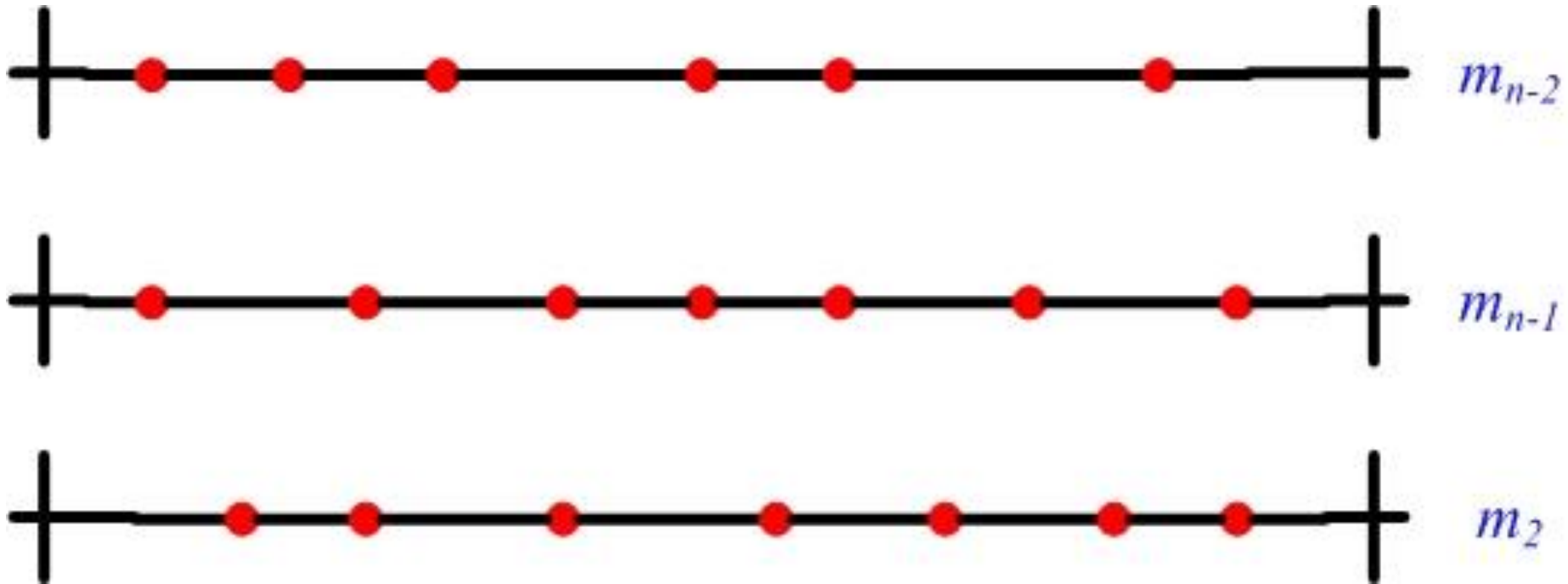
- Перманентные (permanent) выпадения



Как синхронизировать реплики на m_{n-1} , m_{n-2} и m_2 ?

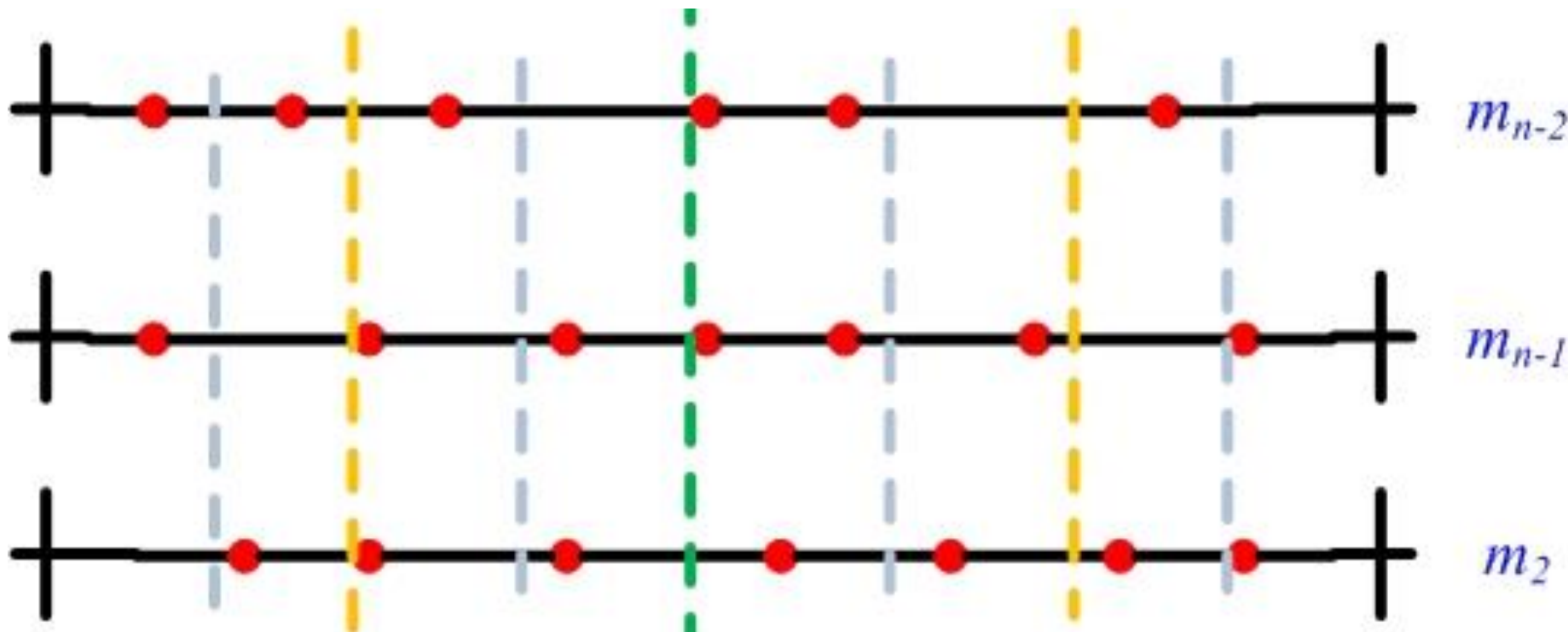
Восстановление данных

- Синхронизация реплик



Восстановление данных

- Синхронизация реплик



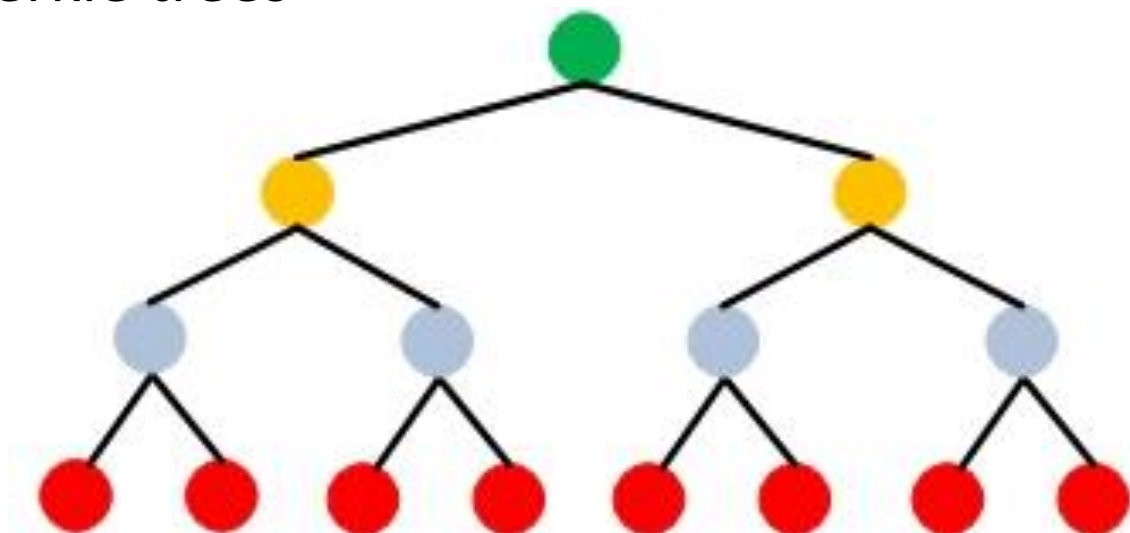
— Разбиваем диапазоны одним из способов:

- Заданное кол-во раз
- Пока в каждом поддиапазоне не останется по 1 значению

Восстановление данных

- Синхронизация реплик

- Merkle trees



- Для каждой машины считаем дерево **контрольных сумм** (поэтапно, начиная с наименьших поддиапазонов):
- Выделяем поддиапазоны, в которых суммы расходятся и синхронизируем только их
- [Используется в BitTorrent](#)

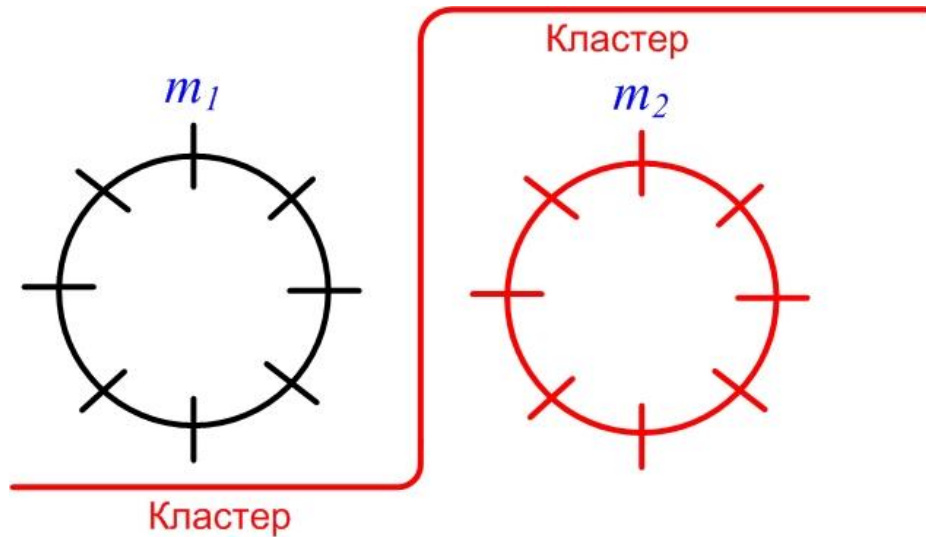
Добавление и удаление машин

- **Протокол сплетничества**

— в конфиге указано 2-3

узла для каждой

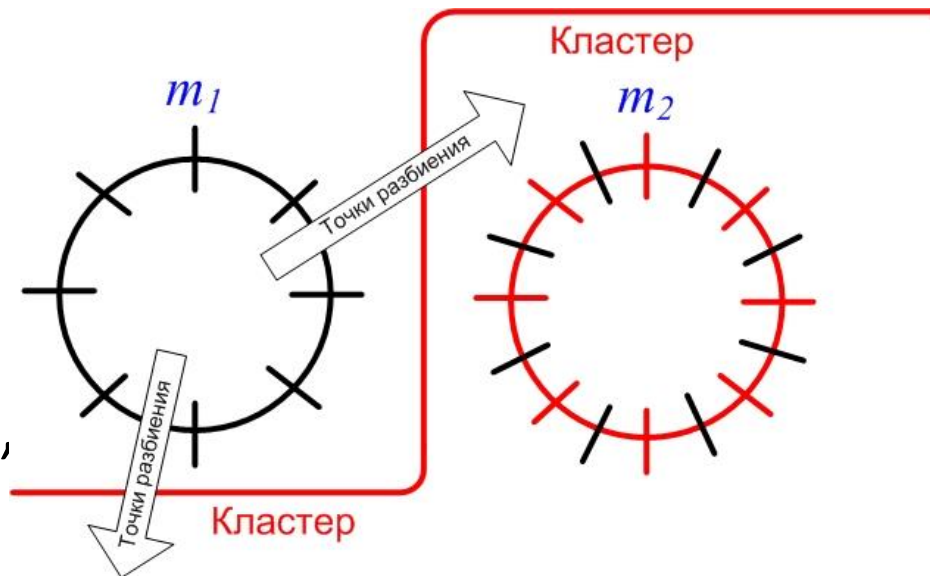
машины



Добавление и удаление машин

- **Протокол сплетничества**

- в конфиге указано 2-3 узла для каждой машины
- когда процесс сходится, машина добавляется в кластер



Добавление и удаление машин

- **Протокол сплетничества**

- в конфиге указано 2-3

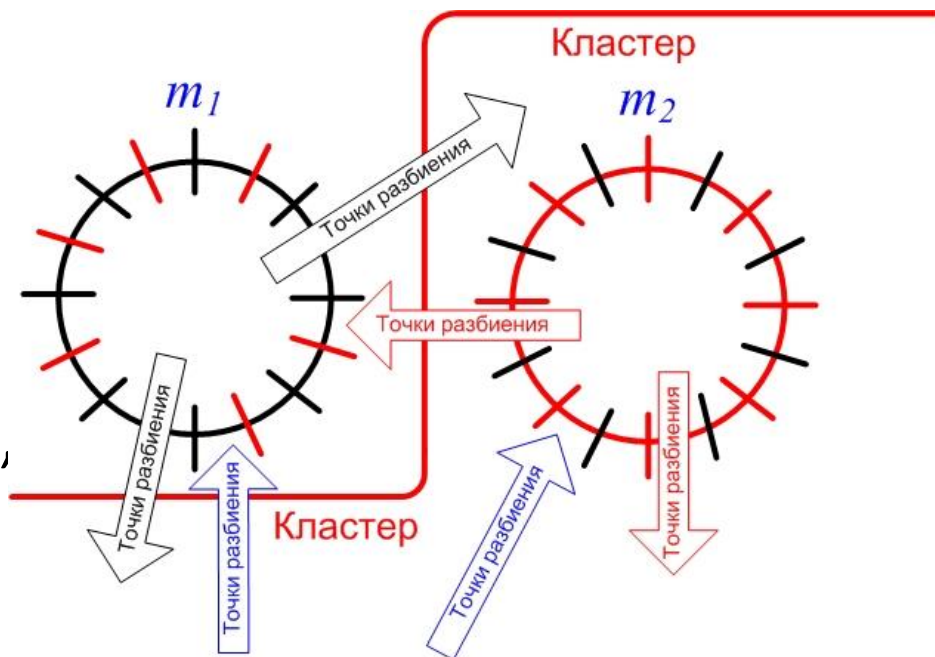
- узла для каждой

- машины

- когда процесс сходится,

- машина добавляется в

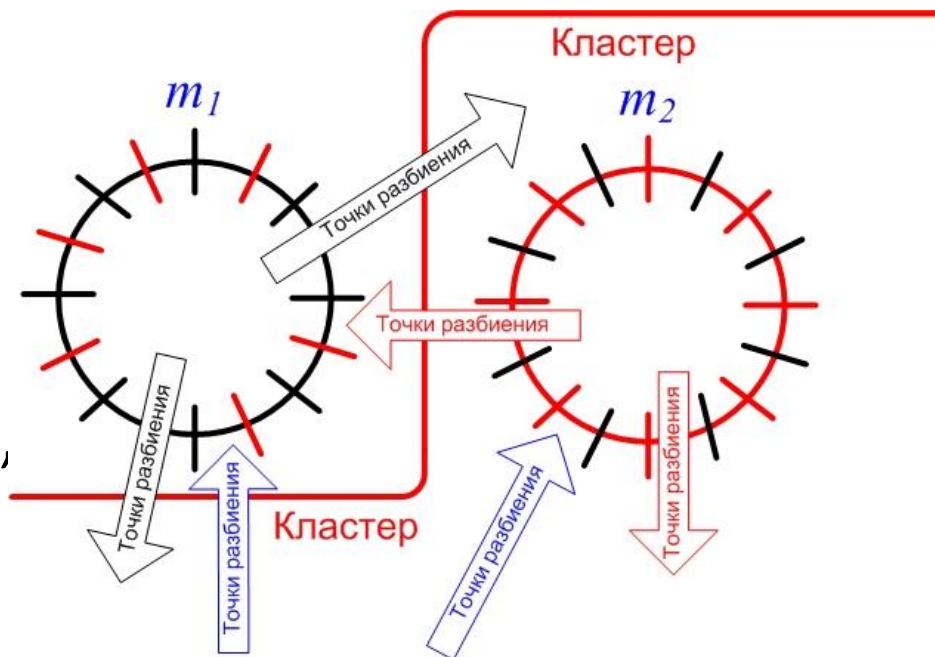
- кластер



Добавление и удаление машин

- **Протокол сплетничества**

- в конфиге указано 2-3 узла для каждой машины
- когда процесс сходится, машина добавляется в кластер

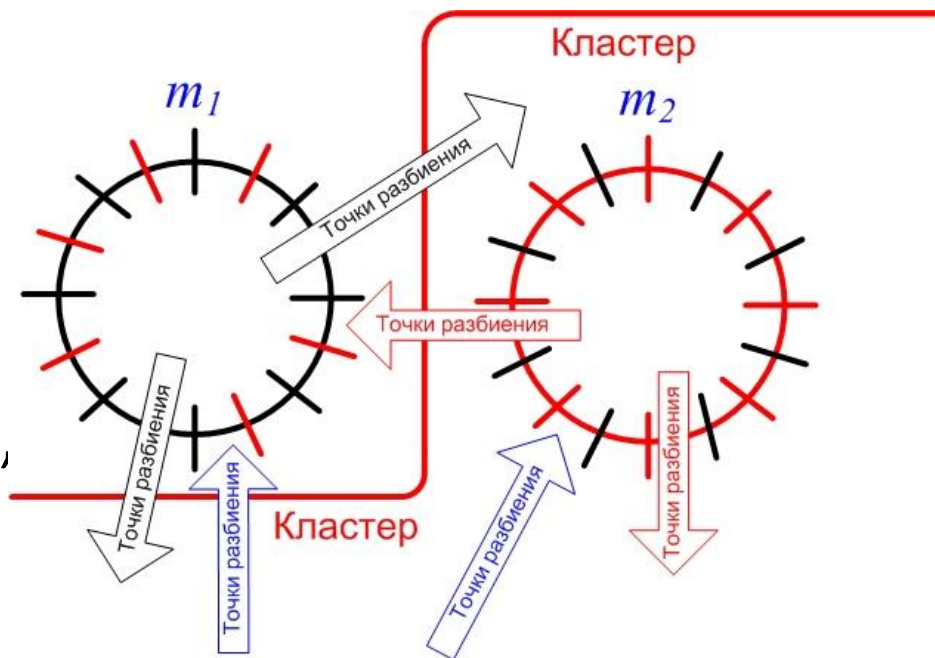


- **Как удалять лишние точки разбиения?**

Добавление и удаление машин

- **Протокол сплетничества**

- в конфиге указано 2-3 узла для каждой машины
- когда процесс сходится, машина добавляется в кластер

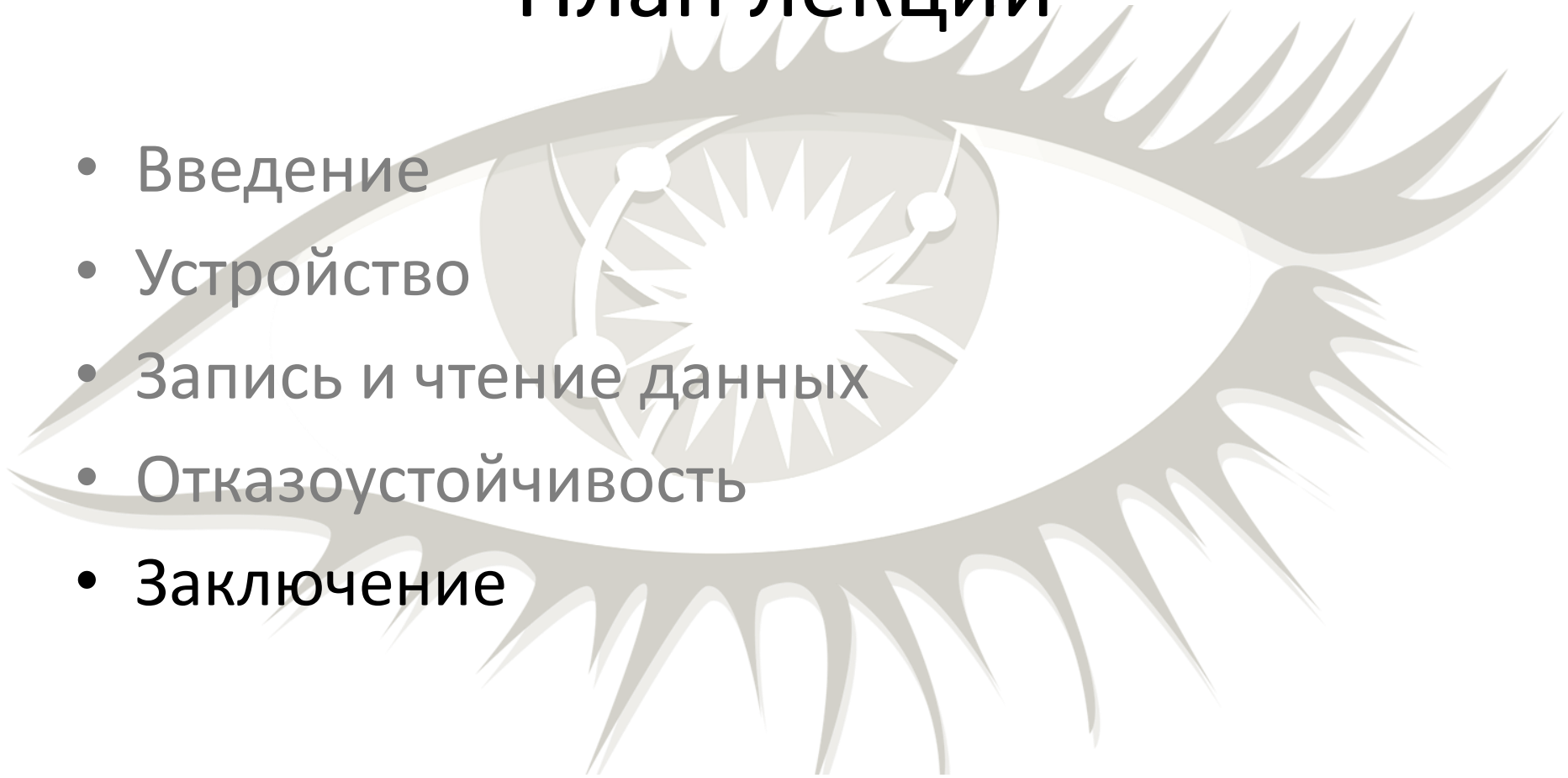


- **Как удалять лишние точки разбиения?**

- точкам разбиения присваивается **timestamp**
- старые точки удаляются
 - соответственно, если точки какой-то машины долго время не обновляются, она удаляется

План лекции

- Введение
- Устройство
- Запись и чтение данных
- Отказоустойчивость
- Заключение



Пользователи Cassandra

<http://planetcassandra.org/companies/>

