

# Асинхронные операции над письмами в Я.Почте

Денис Кутуков, разработчик

Я.Субботник в Минске, 30.08.2014

# Операции над письмами

- Пометить прочитанным
- Удалить
- Переместить в другую папку
- Пометить как спам/хам

# Единичные операции

Письма

Диск

Контакты

Подписки

Календарь



Написать



Проверить



Ответить



Переслать



Удалить



Это спам!



Не прочитано



Добавить кнопку

Живите вне очереди



Яндекс.Деньги <inform@money.yandex.ru>



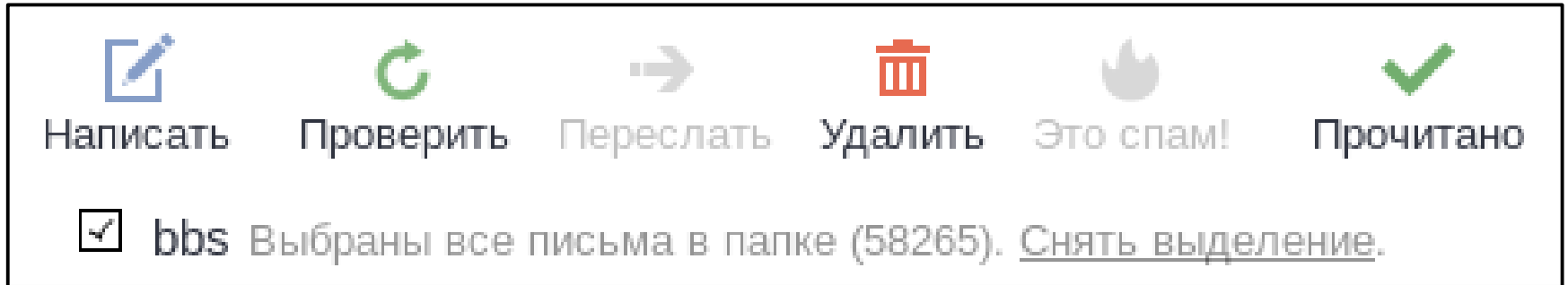
4 и

Кому [mabinist@yandex.ru](mailto:mabinist@yandex.ru)

[Перевести](#) [Создать правило](#) [Свойства письма](#)

Яндекс  
ДЕНЬГИ

# Массовые операции



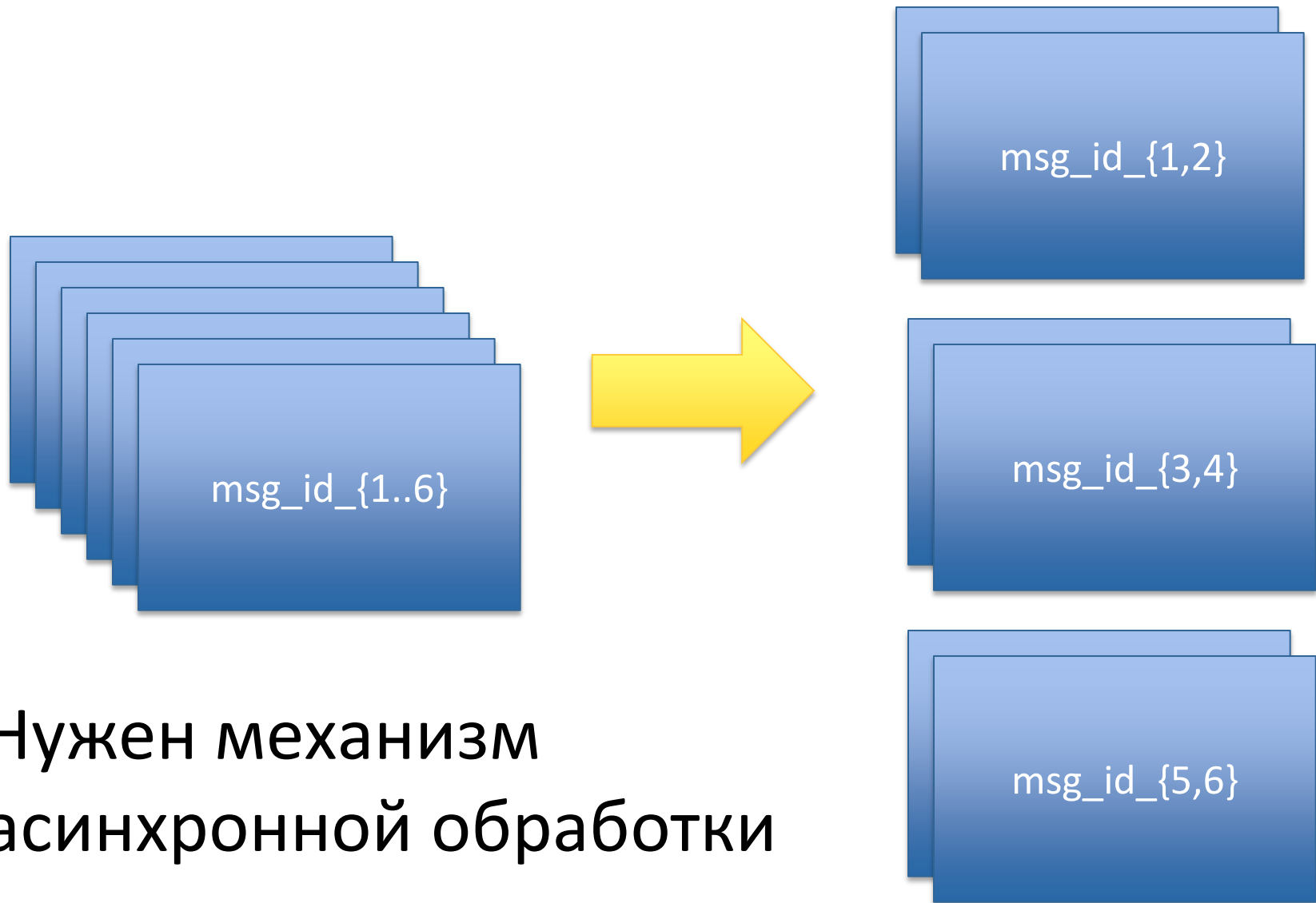
Типичные операции:

- Удалить весь INBOX
- Пометить всю папку прочитанной

# Массовые операции: проблемы

- долгие запросы забивают БД
- отваливаются по таймауту
- user-experience ухудшается
- пользователи жалуются

# Массовые операции: решение



Нужен механизм  
асинхронной обработки

# Oracle Scheduler

<http://bit.ly/yasb-orasch>

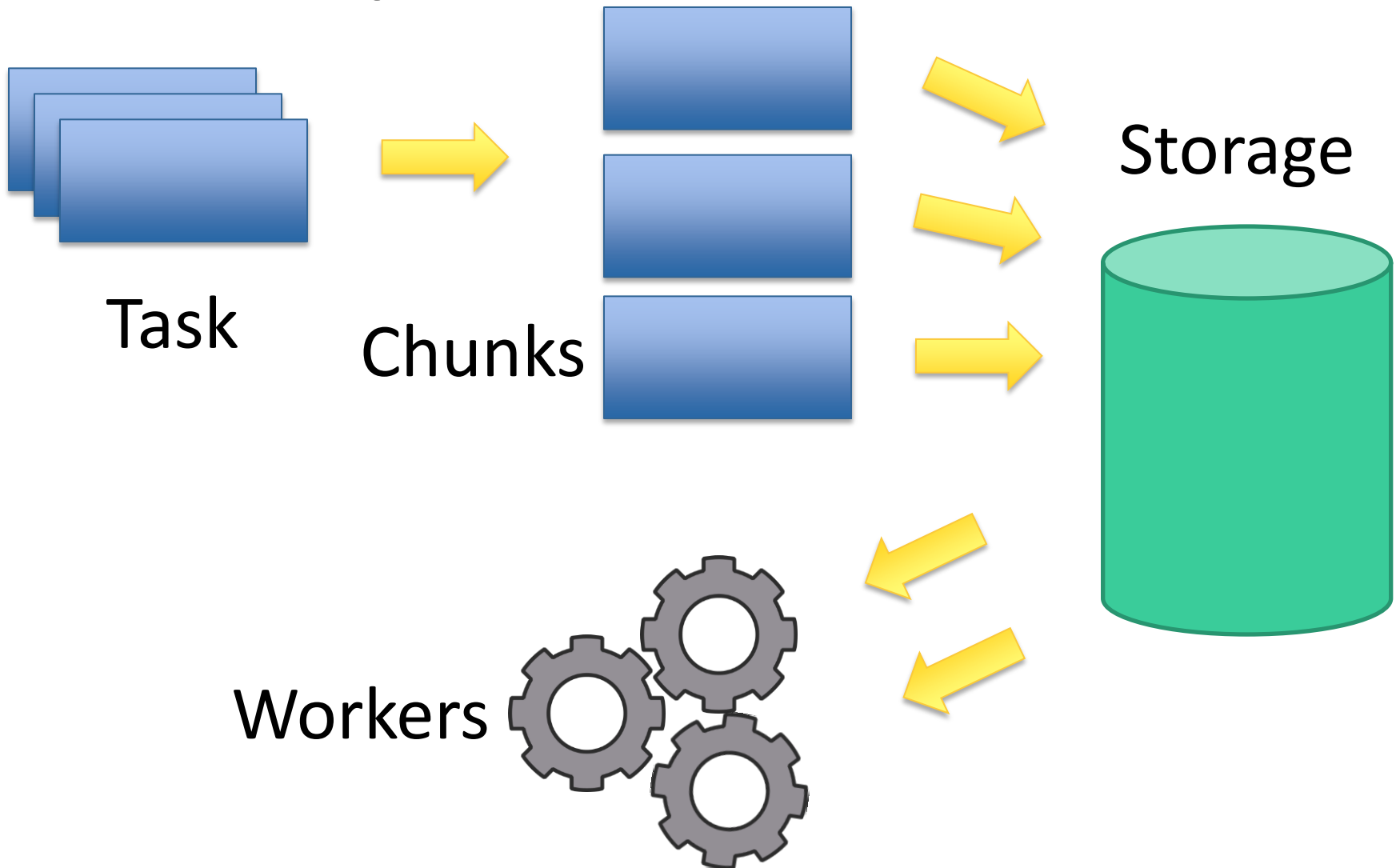
Достоинства:

- Работает

Недостатки:

- Сложная логика в БД
- Неприятно для DBA
- Трудно выкатывать изменения
- Oracle = \$\$\$

# Операции в middle-tier





# Что надо реализовать

- Хранение данных о тасках
- Синхронизация воркеров

# Apache Zookeeper

<http://bit.ly/yasb-zk>

- Файловая система
- Локи
- Координация

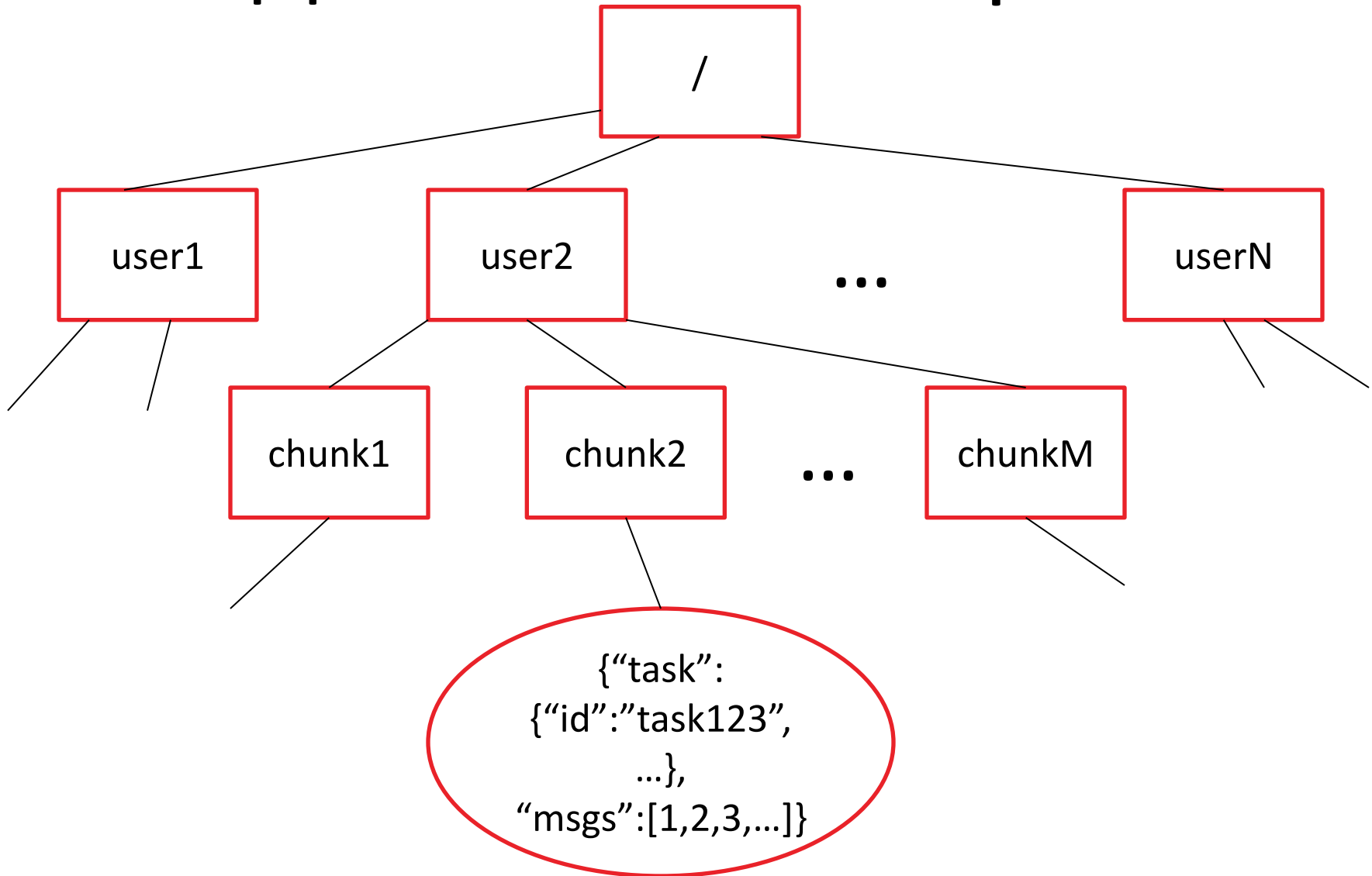
Все это - распределенное



# Почему Zookeeper?

- Переживает потерю ДЦ
- Хорошо задокументирован, активное сообщество
- Есть примеры успешного использования в Яндексе

# Данные в Zookeeper-е



# Конфликты

Пользователь:

- сначала удаляет большую папку
- затем перемещает одно письмо

Система:

- удаляет письмо из запланированного задания

Требование:

- быстро искать конфликты

# Добавление задания

```
void push(Task task) {  
    lock(task.user);  
    resolve(task.user, task.mids);  
    chunks = divide(task);  
    storage.push(chunks);  
    storage.push(task.params);  
    unlock(task.user);  
}
```

# Обработка чанка

```
void processChunk(Handler handler) {  
    while (true) {  
        chunk = storage.nextChunk();  
        if (tryLock(chunk)) {  
            handler(chunk);  
            storage.remove(chunk);  
            unlock(chunk);  
            break;  
        }  
    }  
}
```

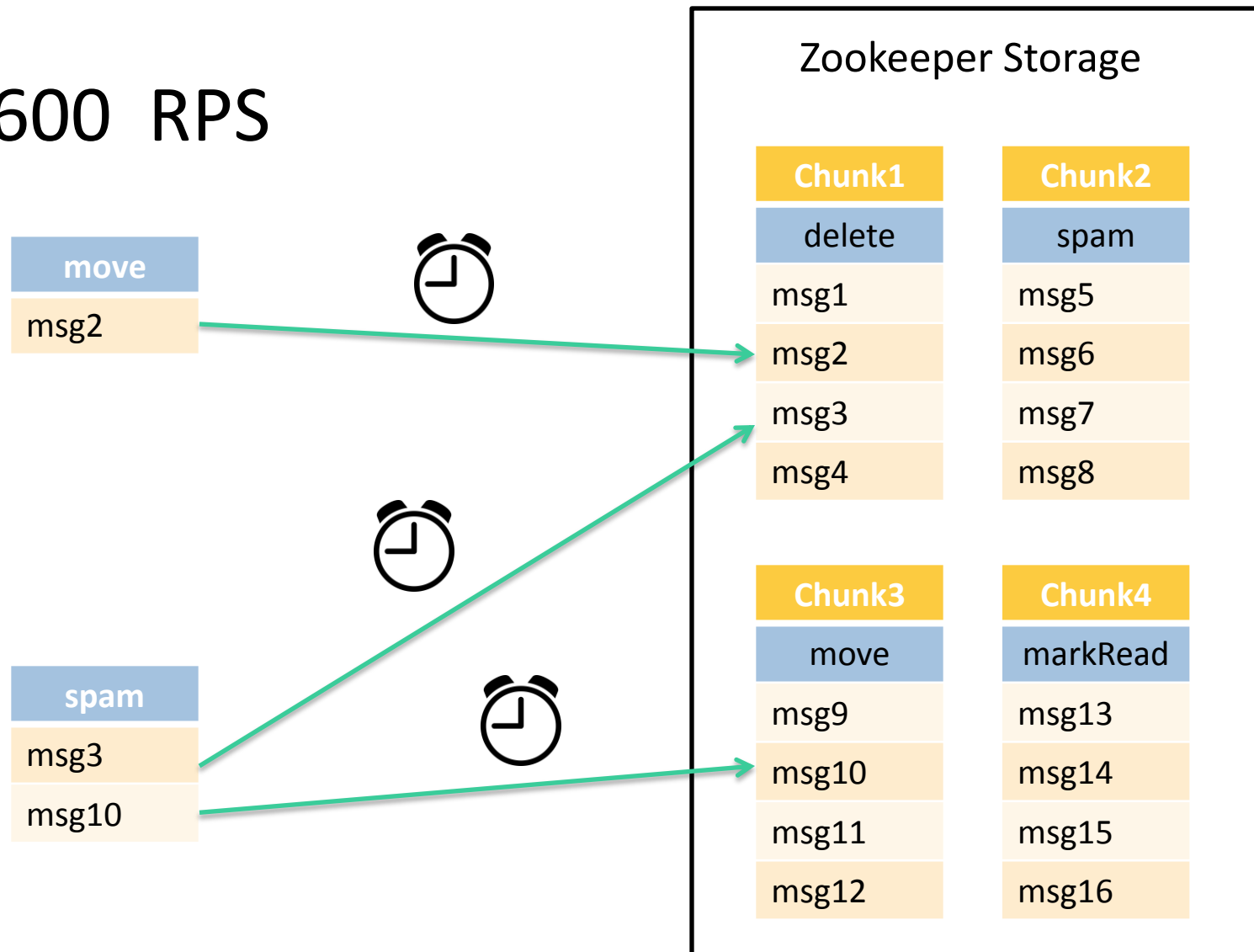
# Разрешение конфликтов

```
void resolve(Request req) {  
    lock(req.user);  
    conf = storage.intersect(req.mids);  
    for (chunk: conf.chunks) {  
        msgs = conflictMsgs(conf, req);  
        chunk.remove(msgs);  
    }  
    unlock(req.user);  
}
```



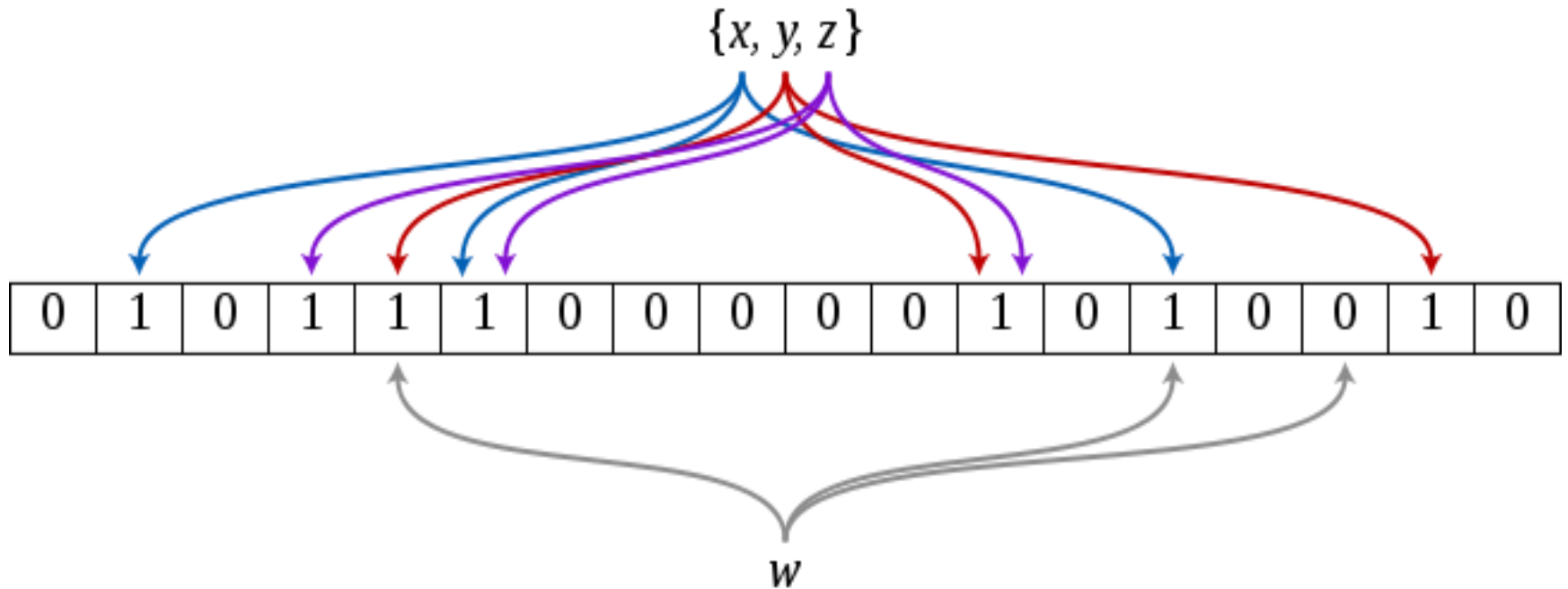
# Проблема с конфликтами

~ 600 RPS



# Блум-фильтр

<http://bit.ly/yasb-bloom>

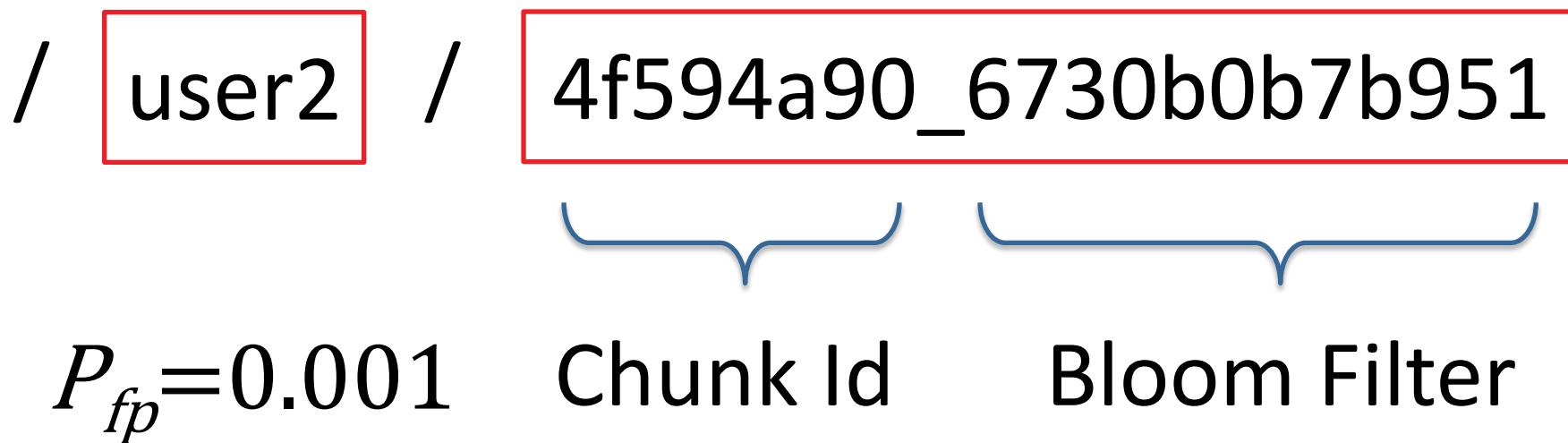


$$P_{fp} > 0$$

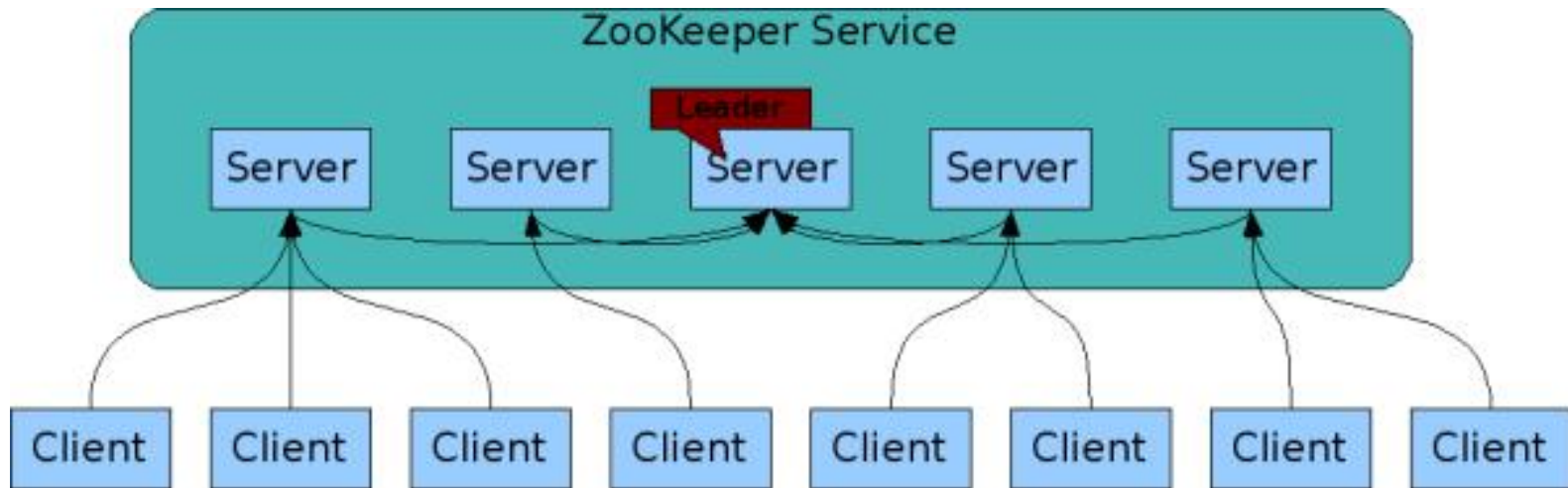
# Фильтр чанка

Предфильтруем поиск письма в чанке.

Название Z-ноды с чанком:

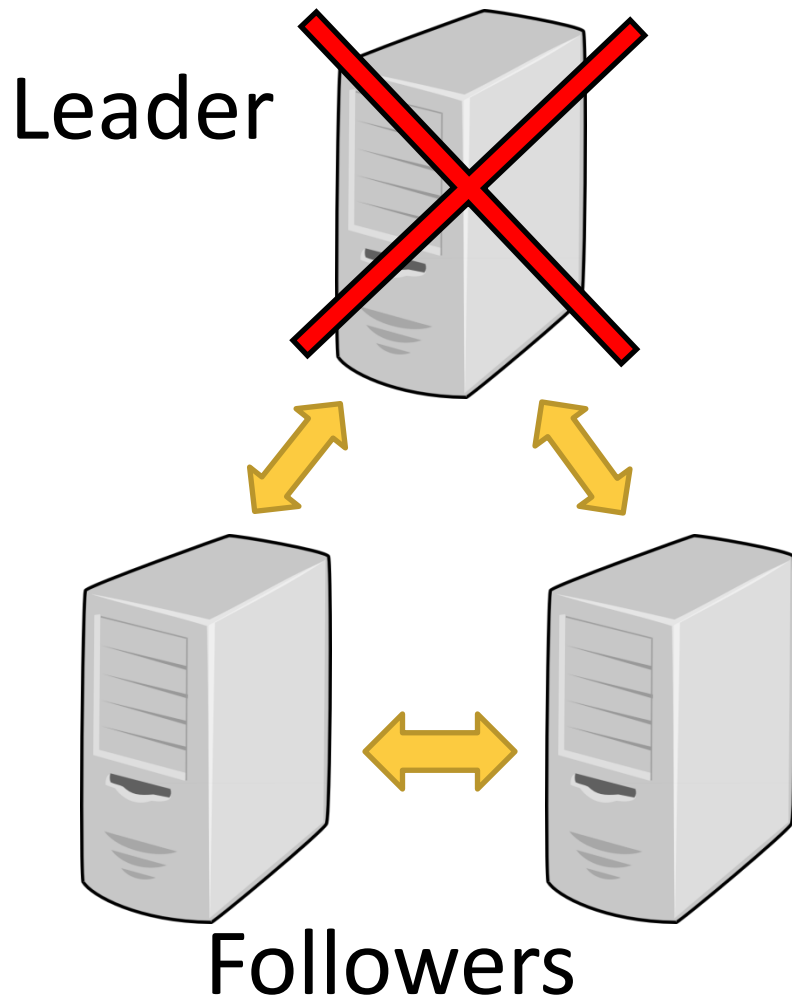


# Zookeeper и несколько машин



- Leader-follower
- Синхронные записи в кворум
- Чтение с любой ноды

# Наши эксперименты



Объем 10Gb

Перевыборы 7 мин  
(кластер недоступен)

Долгая синхронизация  
данных

# Что делать?

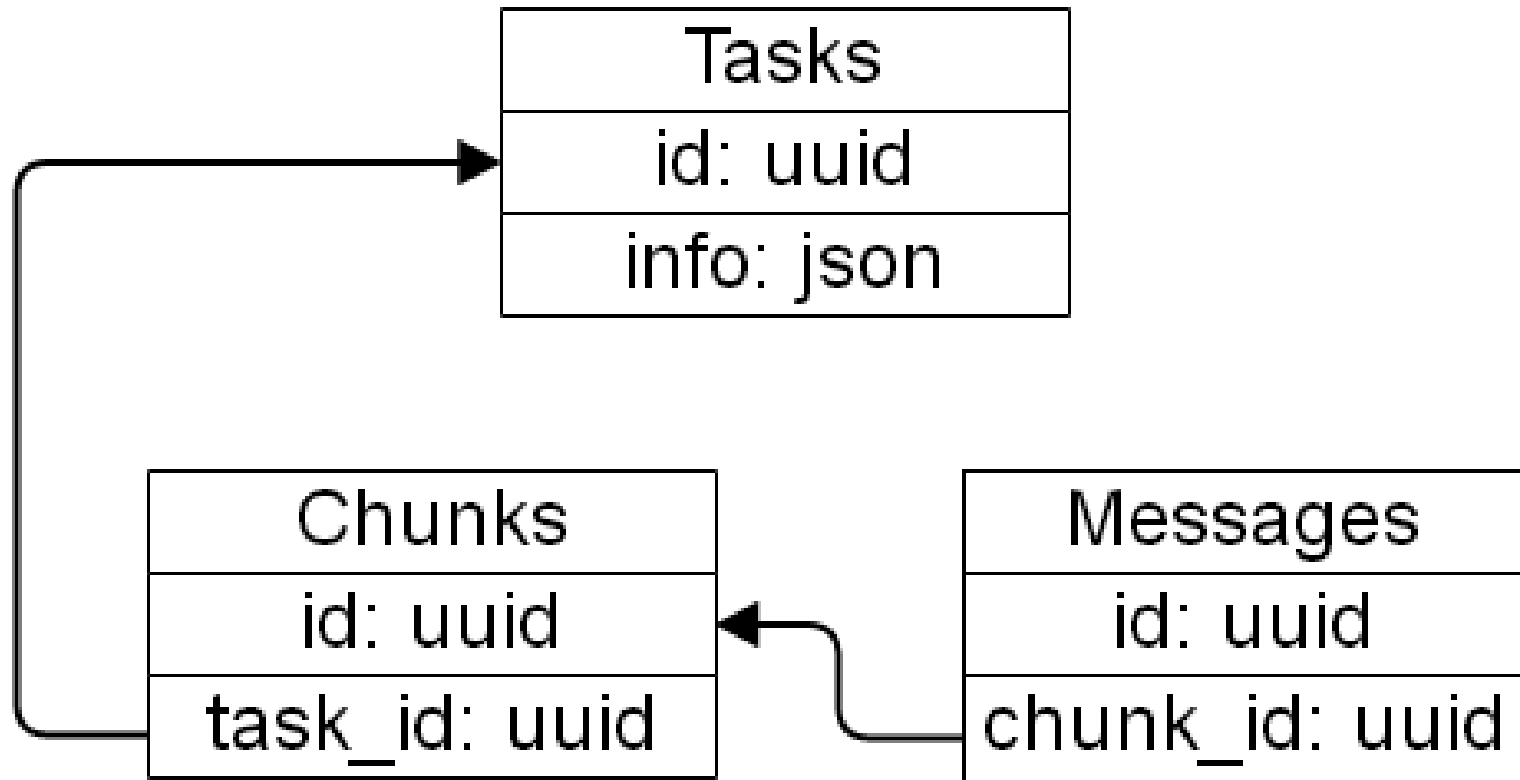
Облегчаем Zookeeper, выносим данные.

Хранилище – Postgresql (<http://bit.ly/yasb-pg>)

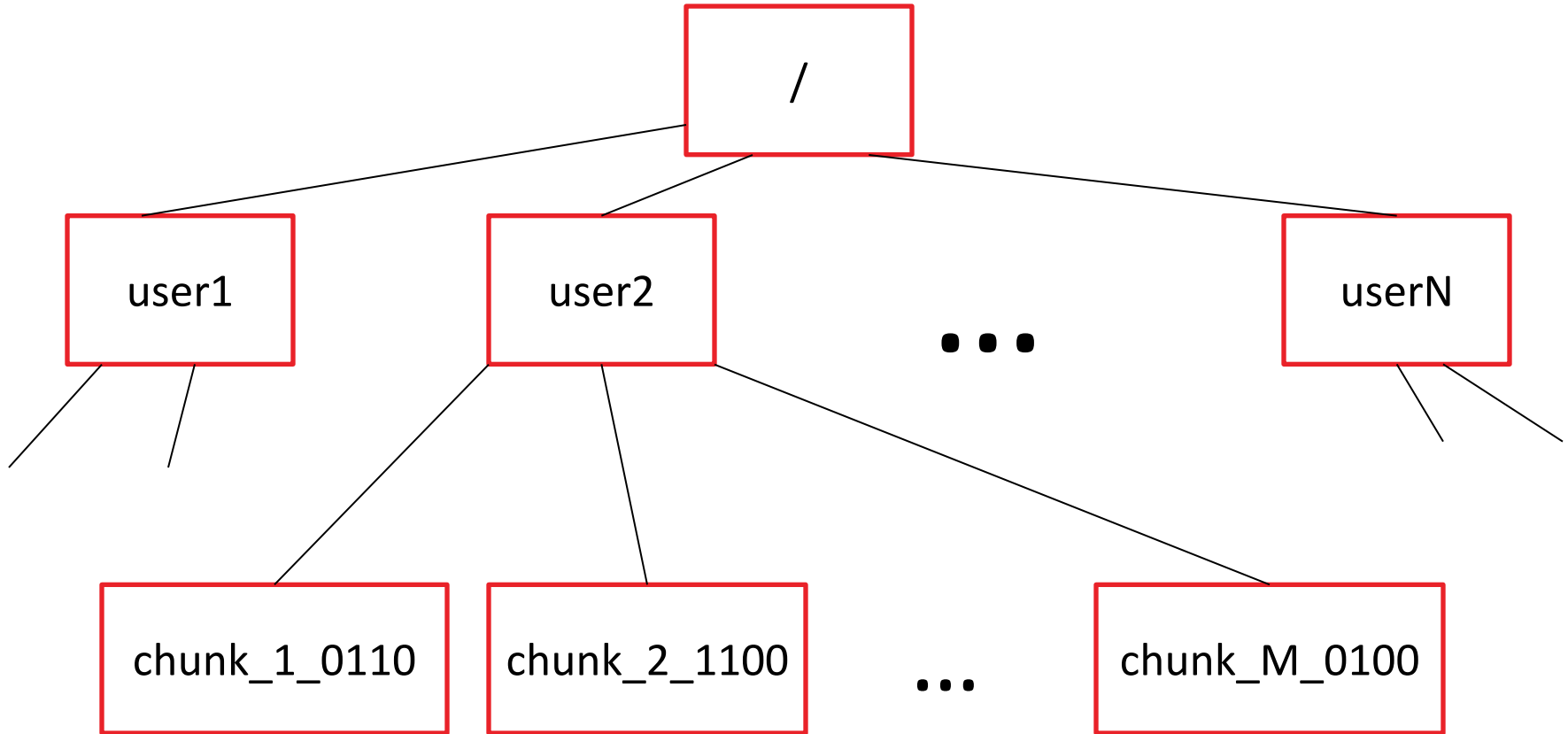
- Реляционная структура
- OpenSource
- У админов и разработчиков  
есть библиотеки и инструменты
- Нормально переживает потерю  
мастера



# Схема данных



# Данные в Zookeeper-е





# В итоге

- Zookeeper подходит для блокировки, но не очень – для данных
- Запускаем массовые операции над письмами в Я.Почте

# ССЫЛКИ

- Oracle Scheduler

<http://bit.ly/yasb-orasch>

- Zookeeper

<http://bit.ly/yasb-zk>

- Bloom Filter

<http://bit.ly/yasb-bloom>

- Postgresql

<http://bit.ly/yasb-pg>

Спасибо за внимание!

Вопросы?

[dskut@yandex-team.ru](mailto:dskut@yandex-team.ru)