Aioriak
Зачем?

Белоусов Максим

RAMBLER&Co

Вы в хорошей компании

RAMBLER&Co

40

млн человек

суммарная аудитория группы

50+

количество изданий,

сервисов и проектов

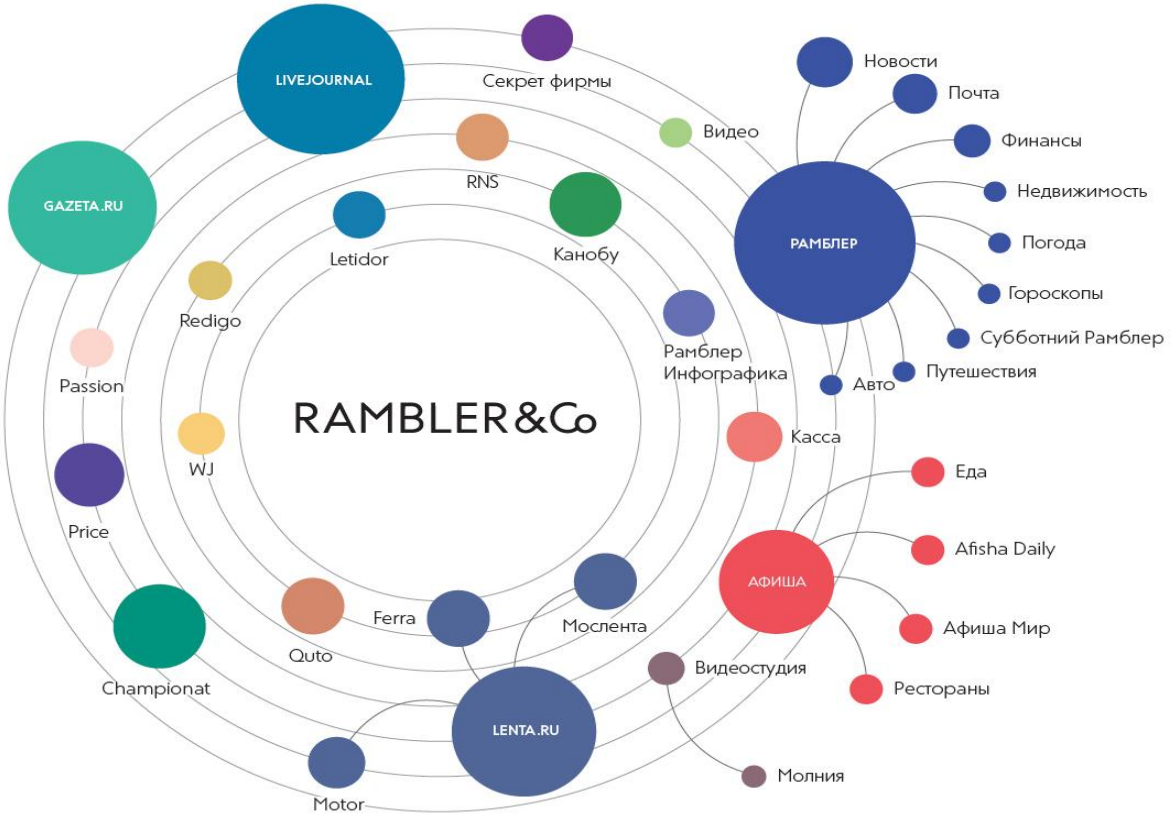
700

разработчиков

1700+

человек в хорошей

компании


LIVEJOURNAL
Секрет фирмы
Видео
RNS
GAZETA.RU
Letidor
Канобу
Redigo
Passion
WJ
Price
Championat
Quto
Ferra
Мослента
LENTA.RU
Motor
Видеостудия
Молния
RAMBLER&Co
Рамблер Инфографика
РАМБЛЕР
Новости
Почта
Финансы
Недвижимость
Погода
Гороскопы
Субботний Рамблер
Путешествия
Авто
Касса
АФИША
Еда
Afisha Daily
Афиша Мир
Рестораны

Языки и технологии

RAMBLER&Co

Контакты

В группе компаний Rambler&Co всегда есть открытые вакансии для тех, кто хочет профессионально расти и развиваться, занимаясь тем, что по-настоящему нравится

hr@rambler-co.ru

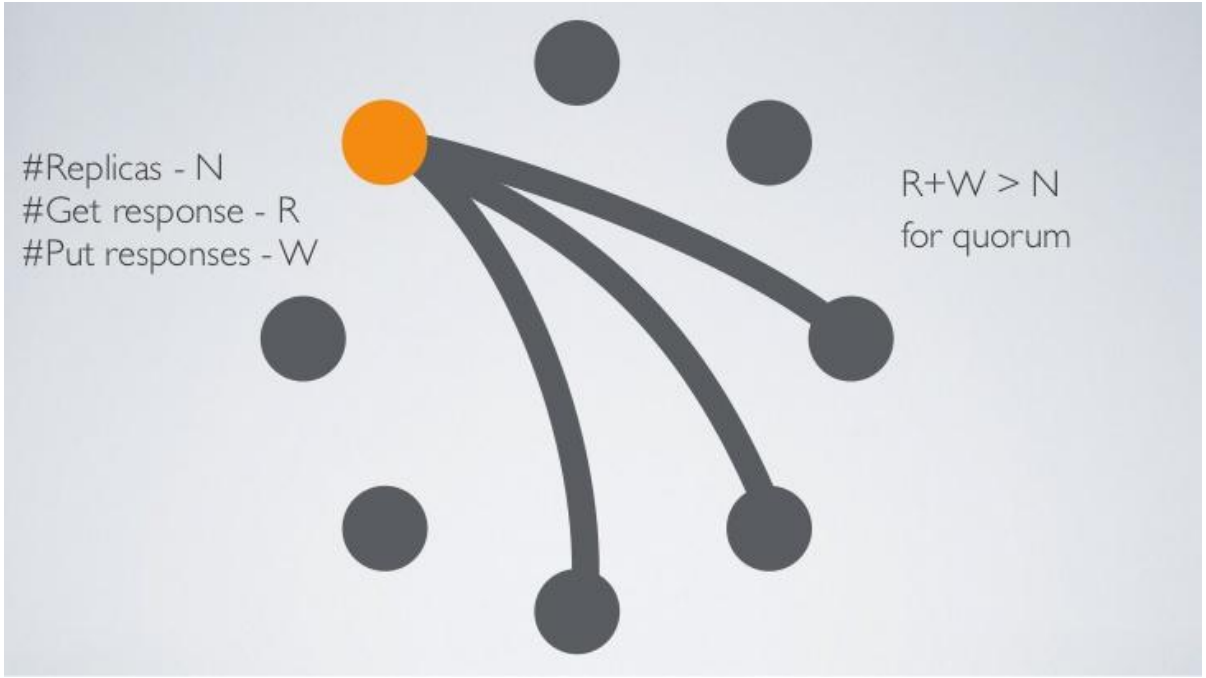
www.rambler-co.ru/jobs

RAMBLER&Co

- Key-value storage;
- Distributed hash table;
- Наследие Amazon DynamoDB;
- Eventual consistency;
- MapReduce;
- Riak search (Apache solr)

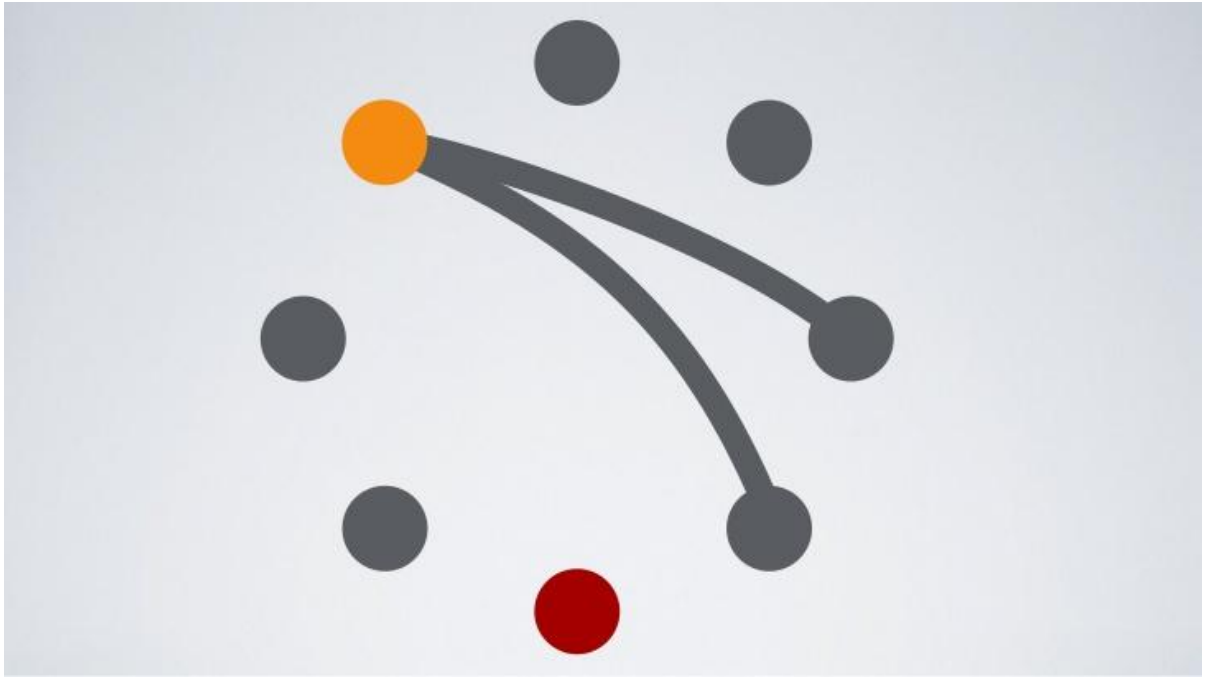
riak

Запись данных



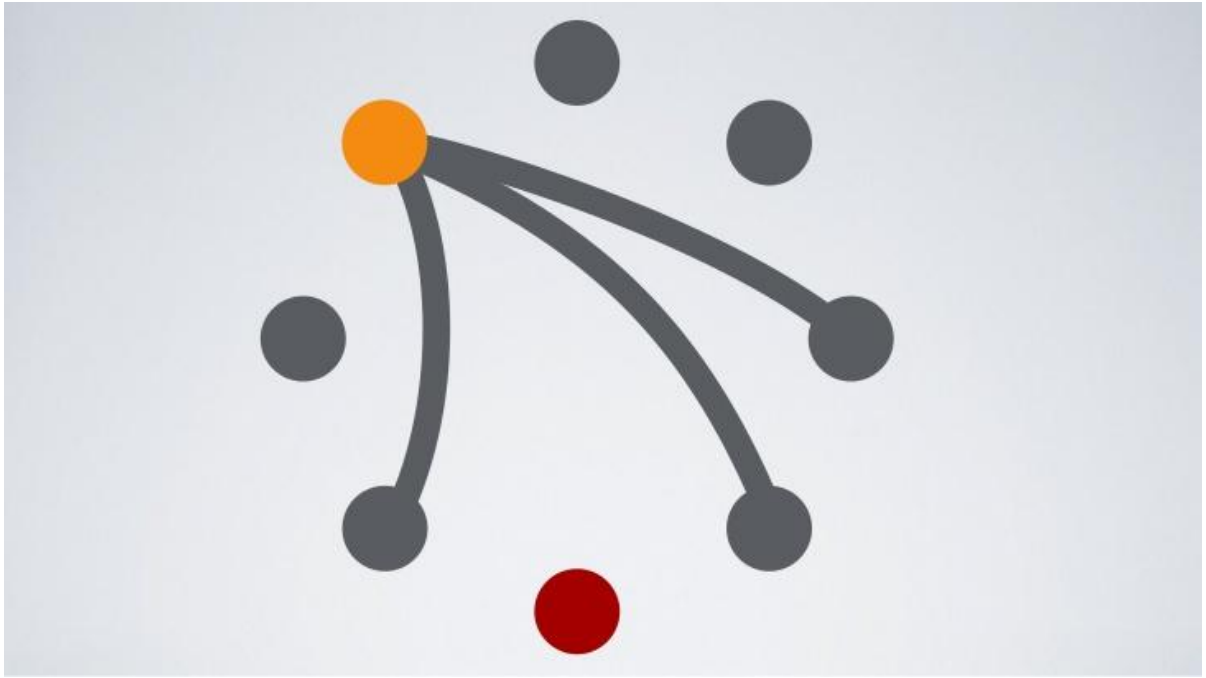
riak

Выход ноды из кластера



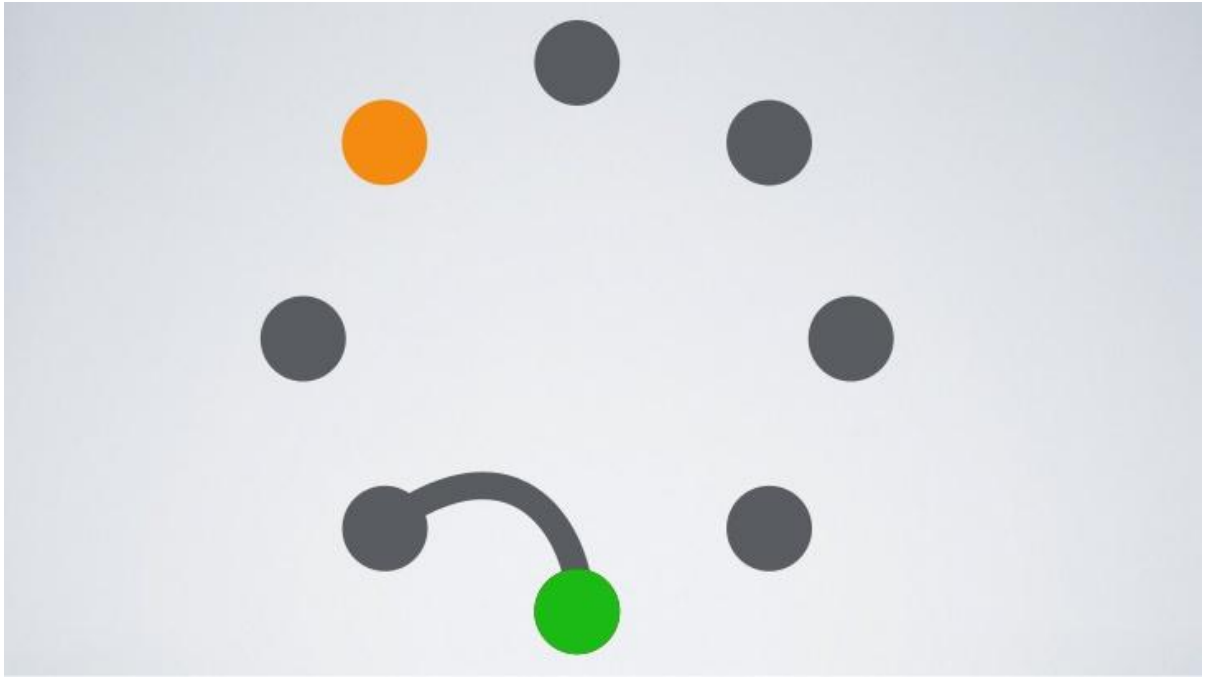
riak

Чтение



riak

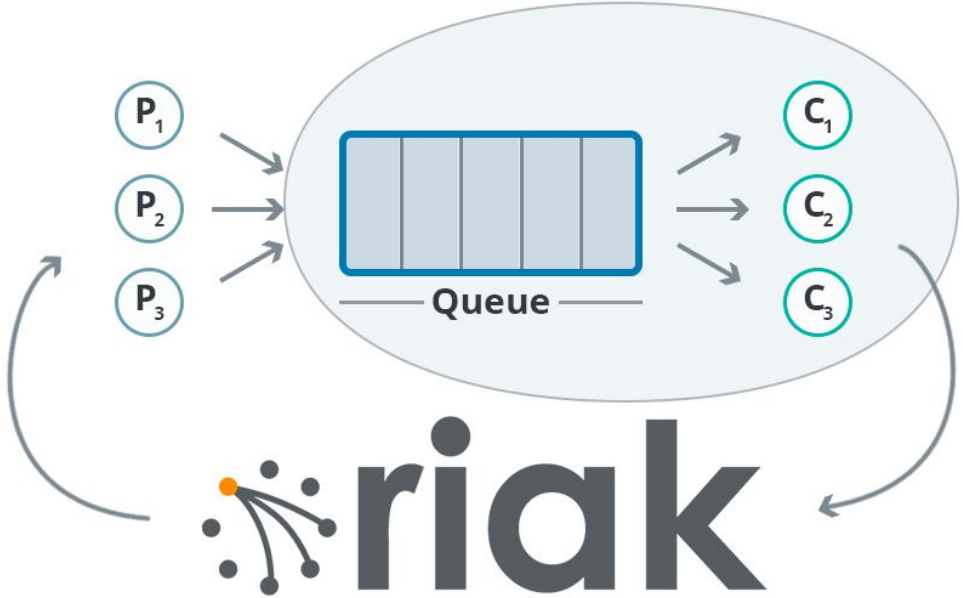
Восстановление ноды



asyncio

- Python модуль, предоставляющий инфраструктуру для написания однопоточного конкурентного кода с использованием корутин;
- PEP-3156;
- Python >= 3.3;
- yield from / await;

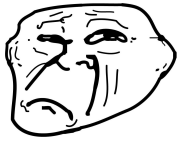
Описание задачи



- P_n - Producer (Генерация сообщений и отсылка их в очередь)
- C_m - Consumer (Обработка сообщений из очереди и сохранение в riak);
- Queue - Очередь с задачами;
- Riak как хранилище результатов;

Проблема

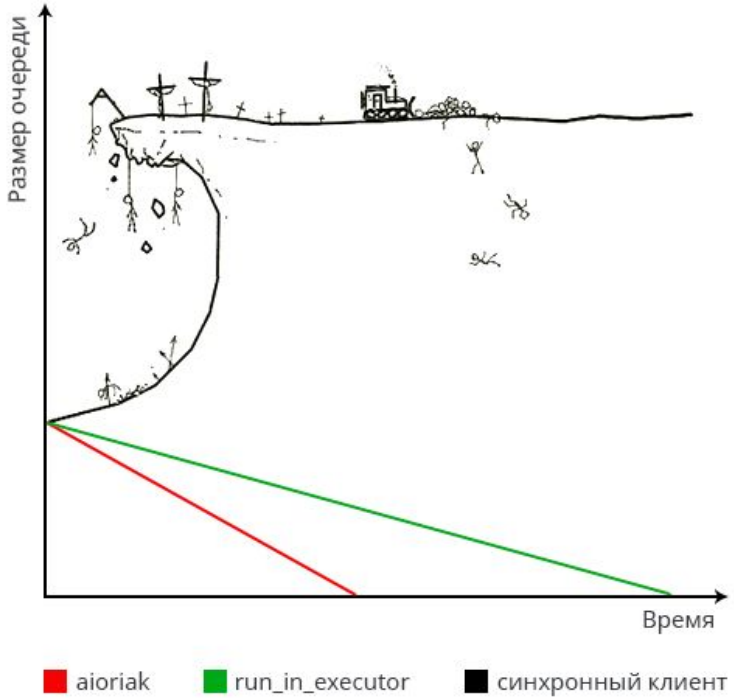
- Библиотека для работы с Riak синхронная;
- Worker’ы будут блокироваться полностью на каждой операции чтения\записи данных;
- Асинхронной библиотеки, подходящей под наши требования, нет;



Возможные решения

- Асинхронный клиент (twisted) (поддержки riak > 2.0 нет);
  - txriakidx — Последние изменения в 2011
  - txRiak — Последние изменения в 2010
  - riakasaurus — Последние изменения в 2014
- REST API - не рекомендуется basho, в пользу protobuf;
- Оставить синхронный код;
- run_in_executor;
- Написать свой клиент;

Синхронный клиент



run_in_executor

```
    import aioamqp
      from concurrent.futures import ThreadPoolExecutor
      import riak


    executor = ThreadPoolExecutor(max_workers=8)


      def store_object():

        client = riak.RiakClient()
          obj = riak.RiakObject('test', 'key')

     obj.store()
     client.close()

async def callback(body, envelope, properties):
    try:
        asyncio.wait_for(loop.run_in_executor(executor, object_store), 1)
    except Exception as e:
        yield from self.channel.basic_client_nack(envelope.delivery_tag)

channel = await protocol.channel()
await channel.basic_consume(callback, queue_name="my_queue")
```

run_in_executor:результаты

Рамблер/ aioriak

- На базе официального python-riak-client от basho;
- Python >= 3.5;
- Поддержка Riak >= 2.0;
- pure asyncio;
- Поддержка:
  - CRUD;
  - BucketTypes;
  - RiakDatatypes;
  - Custom conflict resolver;
- https://github.com/rambler-digital-solutions/aioriak; 

aioriak: пример кода

```
from aioriak import RiakClient


  async def go():
      client = await RiakClient.create(host='127.0.0.1', port=8087)
      bucket = client.bucket_type("foo").bucket("bar")  # Buckettype Datatype Map
      obj = await bucket.get('key')
      obj.registers['deadbeef'].assign('test_string')
      await obj.store()
```

aioriak: результаты

Зачем?

- Решение которое упрощает код;
- Показывает лучшую производительность, избавляя от избыточности PoolExecutor’a;
- Делает ваши волосы мягкими и шелковистыми;
- В данный момент всячески тестируем;
- В ближайшее время ожидается внедрение в production; (уже)
- Буду рад любой обратной связи;
- https://github.com/rambler-digital-solutions/aioriak/issues - Писать туда;