

# Платформа Сбербанка

## Архитектурные развилки

07.04.2017

# Докладчик



Алексей Курагин

Архитектор

Центр развития ядра платформы  
Сбербанк-Технологии  
Новосибирск

# План доклада

- Введение в GridGain
- Переход на IMDG
- Размещение данных
- Исполнение кода
- Георезервирование
- Обновление кластера

Краткий ликбез

# **ВВЕДЕНИЕ В GRIDGAIN(IGNITE)**

# Что такое GridGain?



Хранение

Вычисление

База данных

Кэш L2

# Работа с данными

```
public static void main(String[] args){
    Ignite ignite = Ignition.start();
    IgniteCache<Integer,String> cache =
        ignite.getOrCreateCache("myCache");

    for(int i=0;i<10;i++)
        cache.put(i, "hello"+i);

    for(int i=0;i<10;i++)
        System.out.println(cache.get(i));
}
```

# Вычислительные задачи

```
public static void main(final String[] args) {
    Ignite ignite = Ignition.start();
    Collection calls = new ArrayList<IgniteCallable<Integer>>();

    for(final String arg:args) {
        calls.add(new IgniteCallable<Integer>() {
            public Integer call() throws Exception {
                return arg.length();
            }
        });
    }
    Collection<Integer> results = ignite.compute().call(calls);
    for(Integer res:results)
        System.out.println(res);
}
```

# Persistent Data Store



Java Objects / Binary format



Java Objects / Binary format



Persistent Data Store



Архитектурные развилки

# ПЕРЕХОД НА IMDG



# Территориальные банки

10 лет назад...

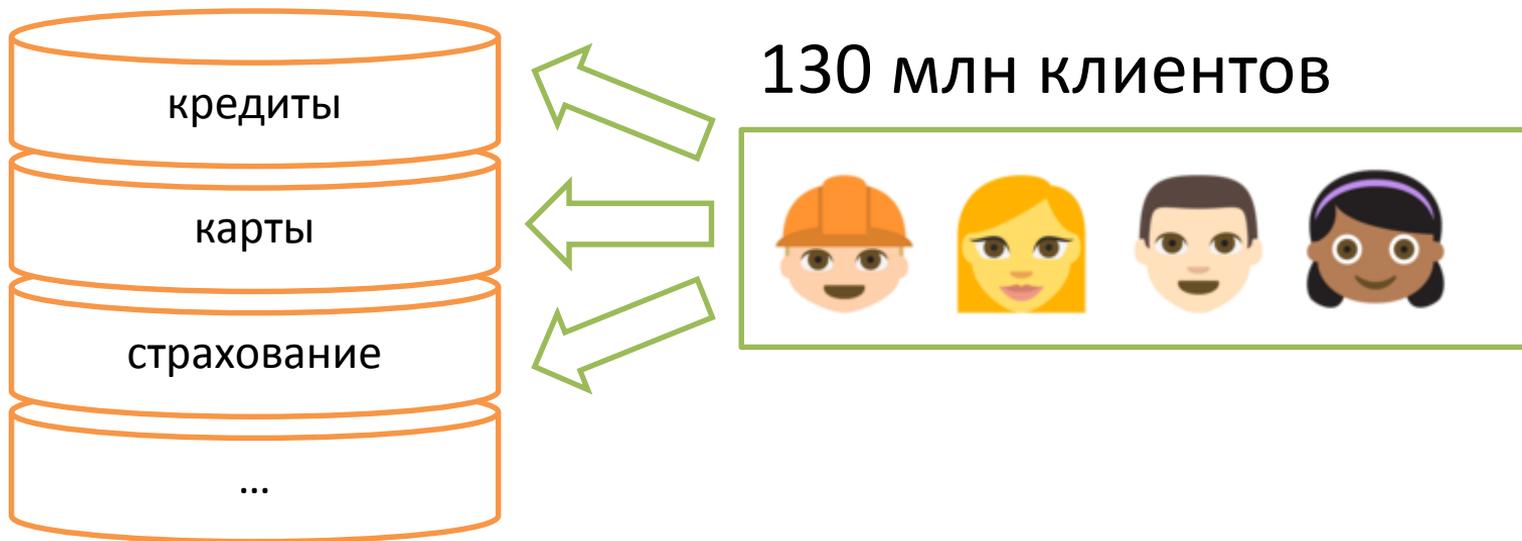


# Территориальные банки

5 лет назад...



# Синхронизация



# Масштабирование

Вертикальное



Горизонтальное



# Типичный менеджер



Мы перешли на GridGain, значит наши приложения будут быстрее, надёжнее и функциональнее чем на Oracle!

Архитектурные развилки

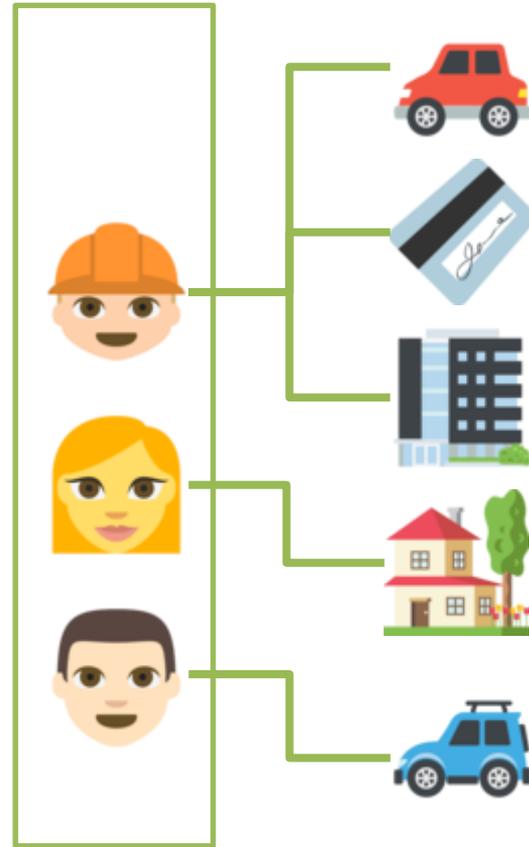
# РАЗМЕЩЕНИЕ ДАННЫХ



# Группировка данных



VS

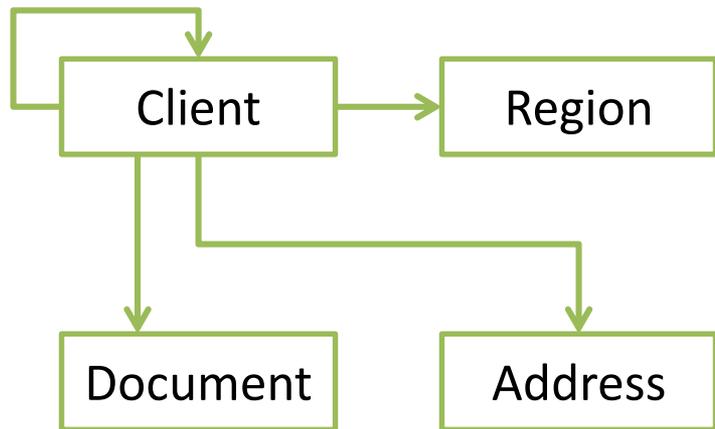


# Типы сущностей

- Справочники
- Партицируемые
- Связанные (ссылки)
  - Обратные
  - Прямые
    - Разделённые
    - Вложенные

# Партицируемая сущность

```
// Партицируемая сущность  
public class Client {  
  
    private Key<Long, Region> id;  
  
    private String name;  
  
    private Address address;  
  
    private Link<Client> spouse;  
  
    private DirectLink<Document> passport;  
  
}
```



# Партицируемая сущность

*// Партицируемая сущность*

```
public class Client {  
  
    private Key<Long, Region> id;  
  
    private String name;  
  
    private Address address;  
  
    private Link<Client> spouse;  
  
    private DirectLink<Document> passport;  
  
}
```

*//Справочник*

```
public class Region {  
  
    private Long id;  
  
    private String name;  
  
}
```

# Вложенная сущность

*// Партицируемая сущность*

```
public class Client {  
  
    private Key<Long, Region> id;  
  
    private String name;  
  
    private Address address;  
  
    private Link<Client> spouse;  
  
    private DirectLink<Document> passport;  
  
}
```

*//Вложенная сущность*

```
public class Address {  
  
    private String address;  
  
    private String postIndex;  
  
}
```

# Прямая связь

```
// Партицируемая сущность  
public class Client {  
  
    private Key<Long, Region> id;  
  
    private String name;  
  
    private Address address;  
  
    private Link<Client> spouse;  
  
    private DirectLink<Document> passport;  
  
}
```

```
//Связанная сущность  
public class Document {  
  
    private Long id;  
  
    private String serialNo;  
  
    private Date issueDate;  
  
}
```

# Обратная связь

*// Партицируемая сущность*

```
public class Client {  
  
    private Key<Long, Region> id;  
  
    private String name;  
  
    private Address address;  
  
    private Link<Client> spouse;  
  
    private DirectLink<Document> passport;  
  
}
```

*//Связанная сущность*

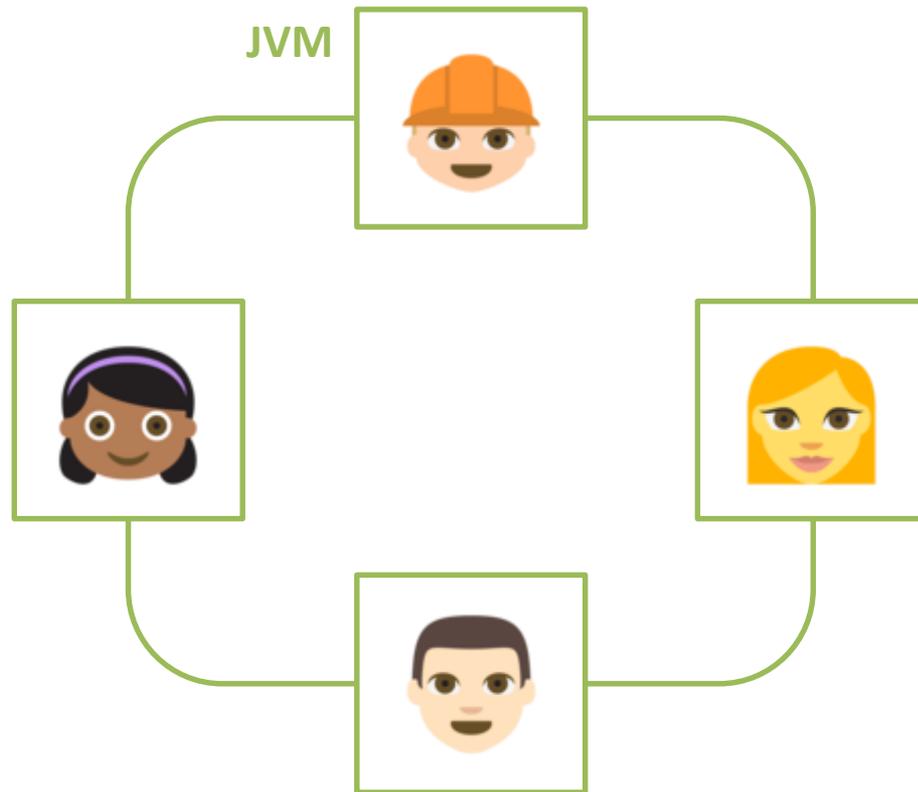
```
public class Refill {  
  
    private Long id;  
  
    private ReverseLink<Client> client;  
  
    private Currency value;  
  
}
```

Архитектурные развилки

# ИСПОЛНЕНИЕ КОДА

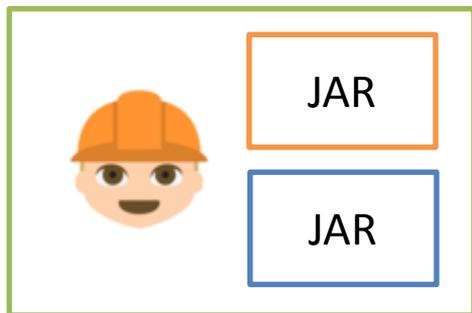


# Кластер



# Размещение кода

JVM



данные + код

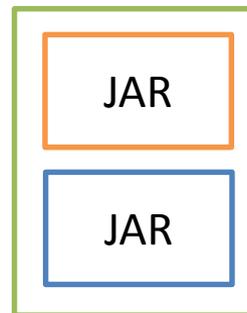
VS

JVM



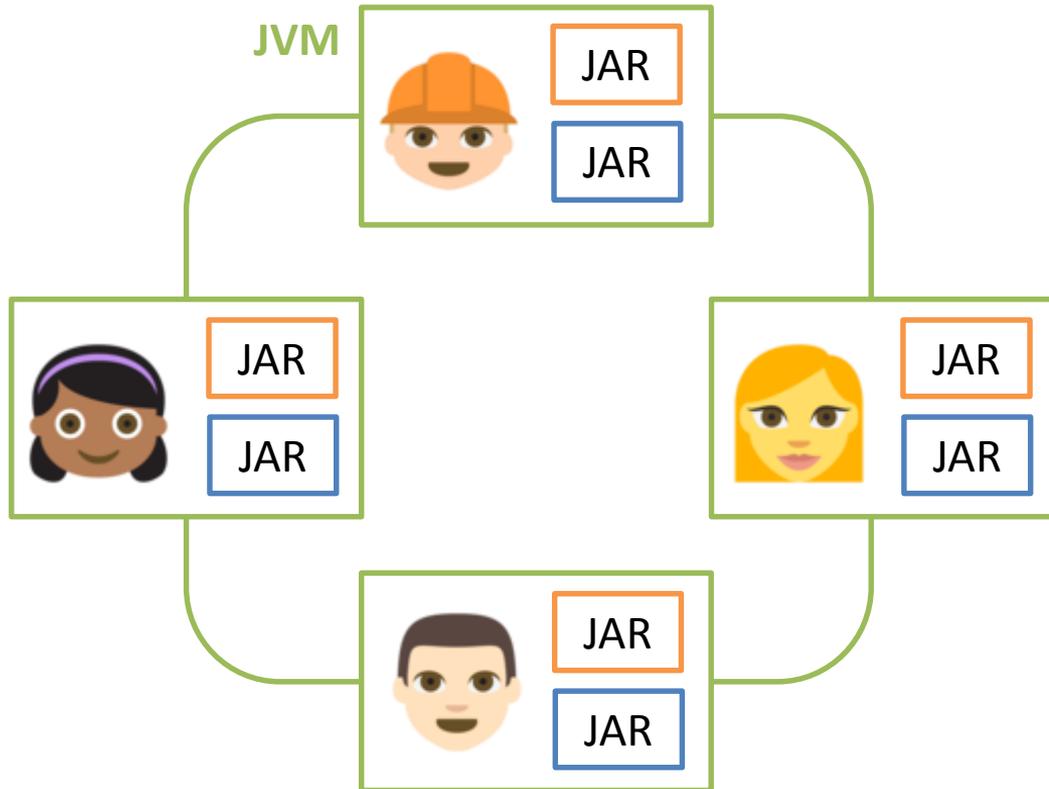
данные

JVM

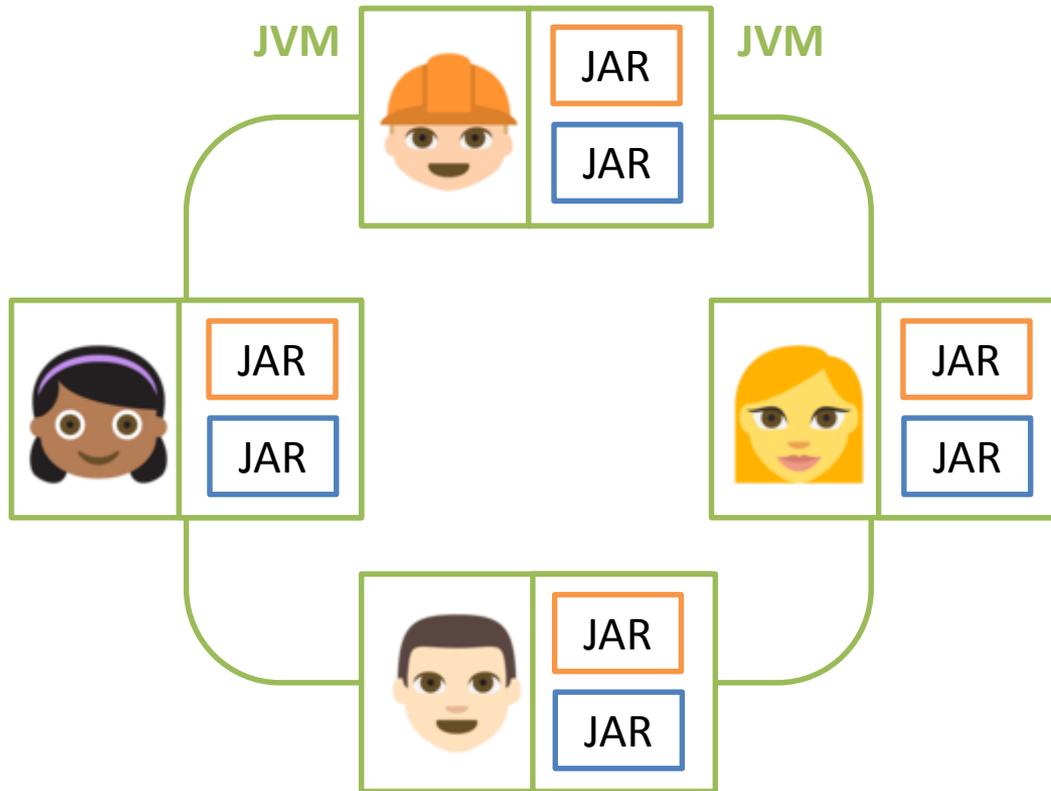


код

# Код вместе с данными



# Код отдельно от данных



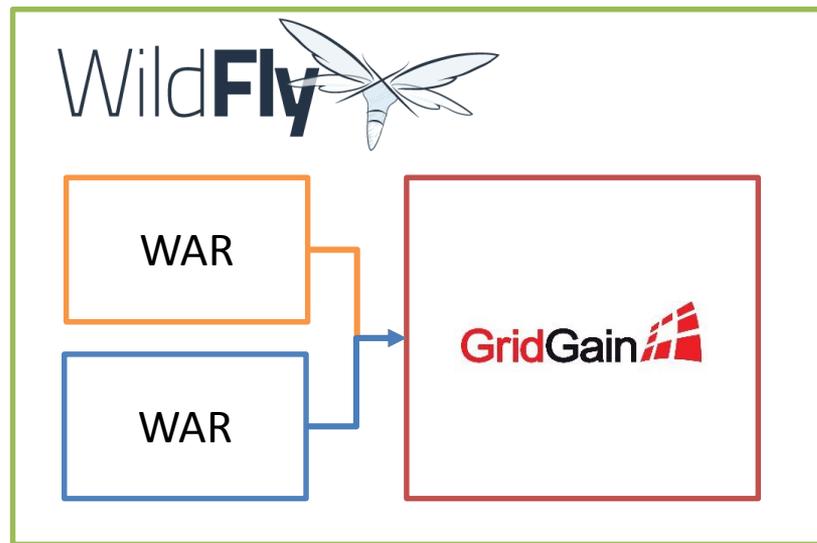
# Проблемы клиентских узлов Ignite

- Распознавание медленных узлов
- Потеря событий при переподключении
- Переполнение памяти
- Блокировка операций серверного узла

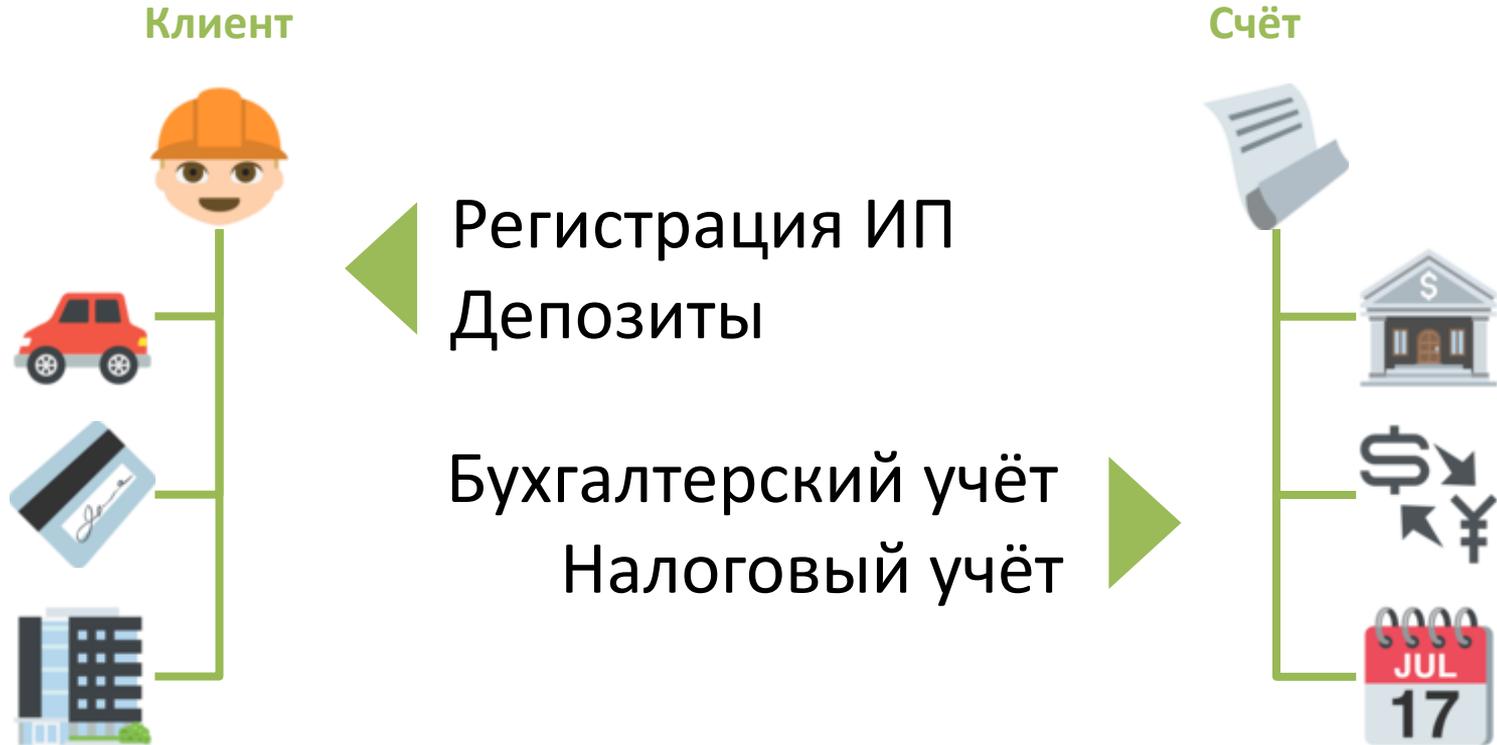
# Общий клиентский узел GridGain



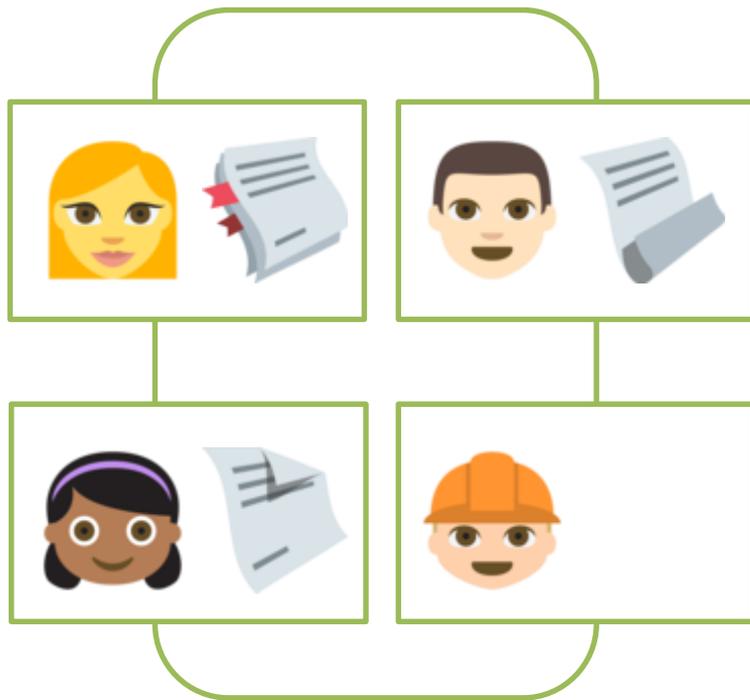
VS



# Партицируемые объекты

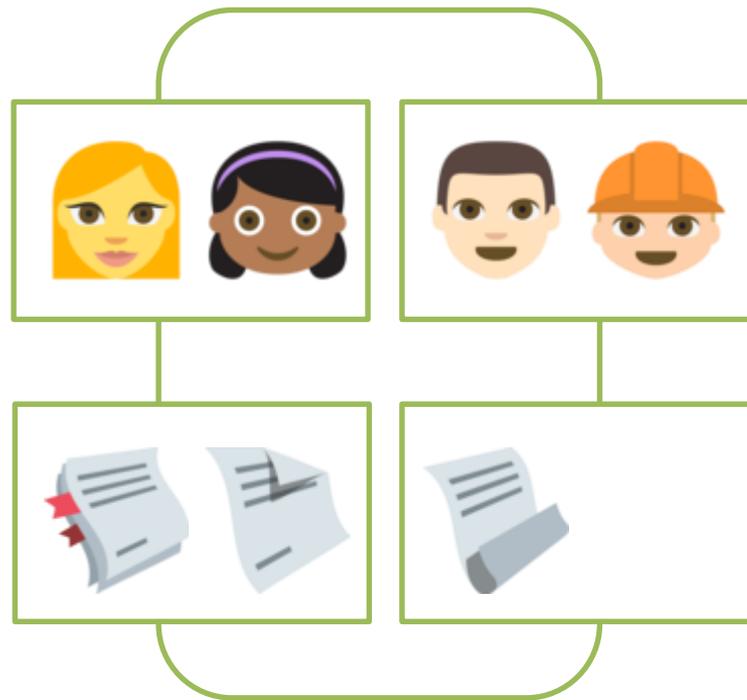


# Островная модель



Смешанный

VS



«Островной»

# Островная модель

- Независимое управление мощностью
- Минимизация конкуренции
- Гибкость в эксплуатации
- Изолированность

Архитектурные развилки

# ГЕОРЕЗЕРВИРОВАНИЕ

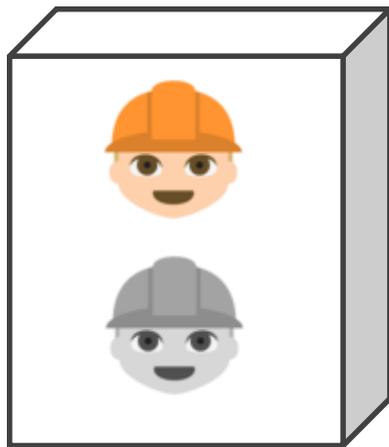


# Требования к размещению

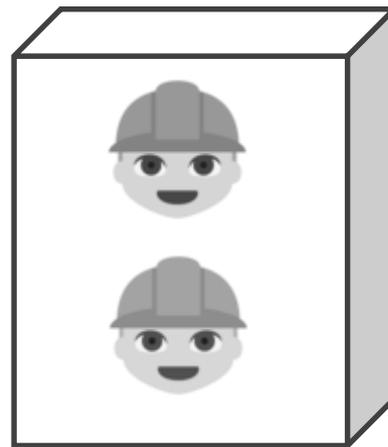
➡ 3-х кратное резервирование

➡ Гео-реплика

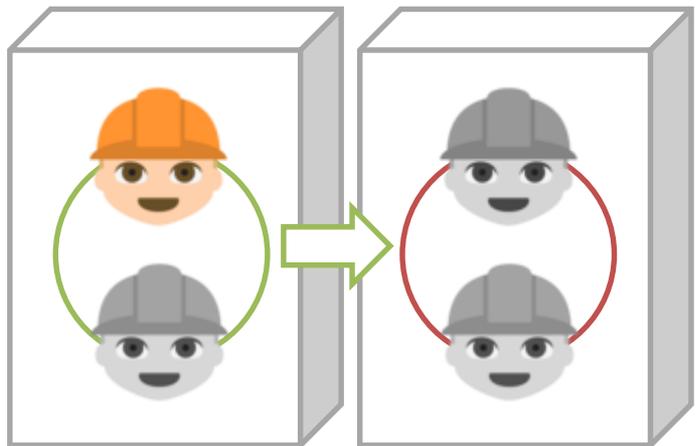
ЦОД



ЦОД

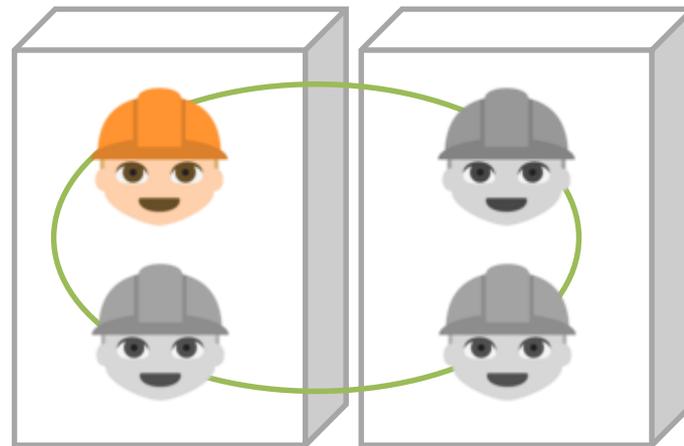


# Георезервирование



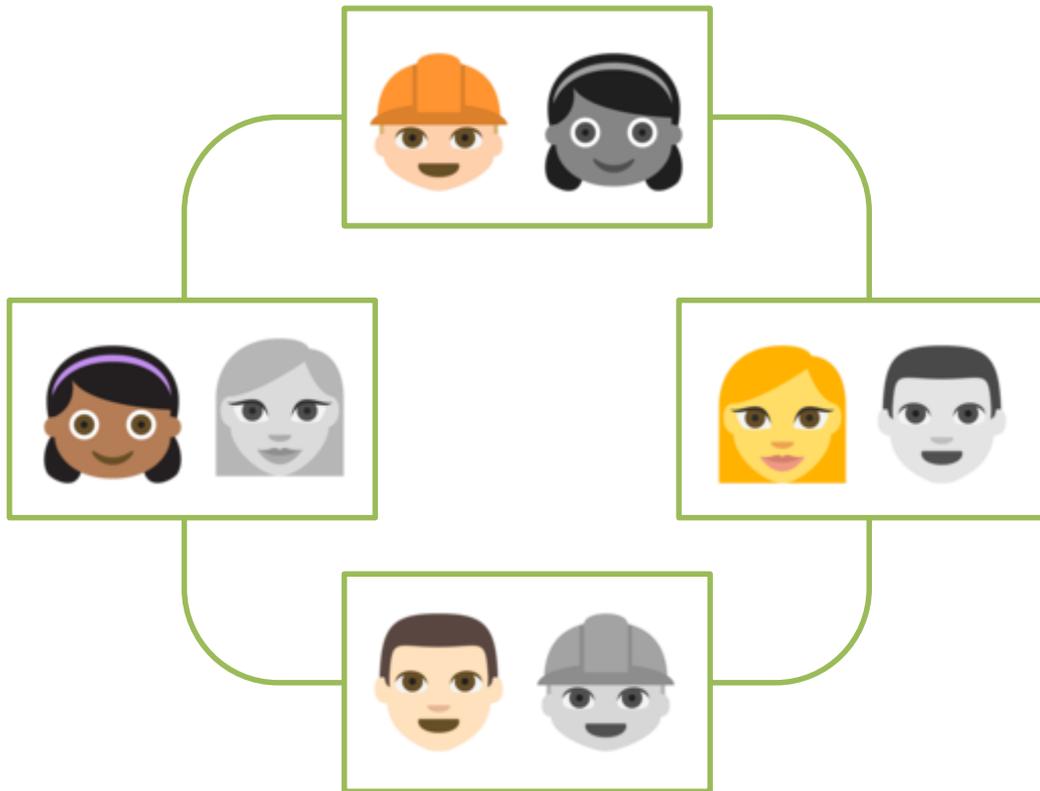
Основной + Резервный

VS

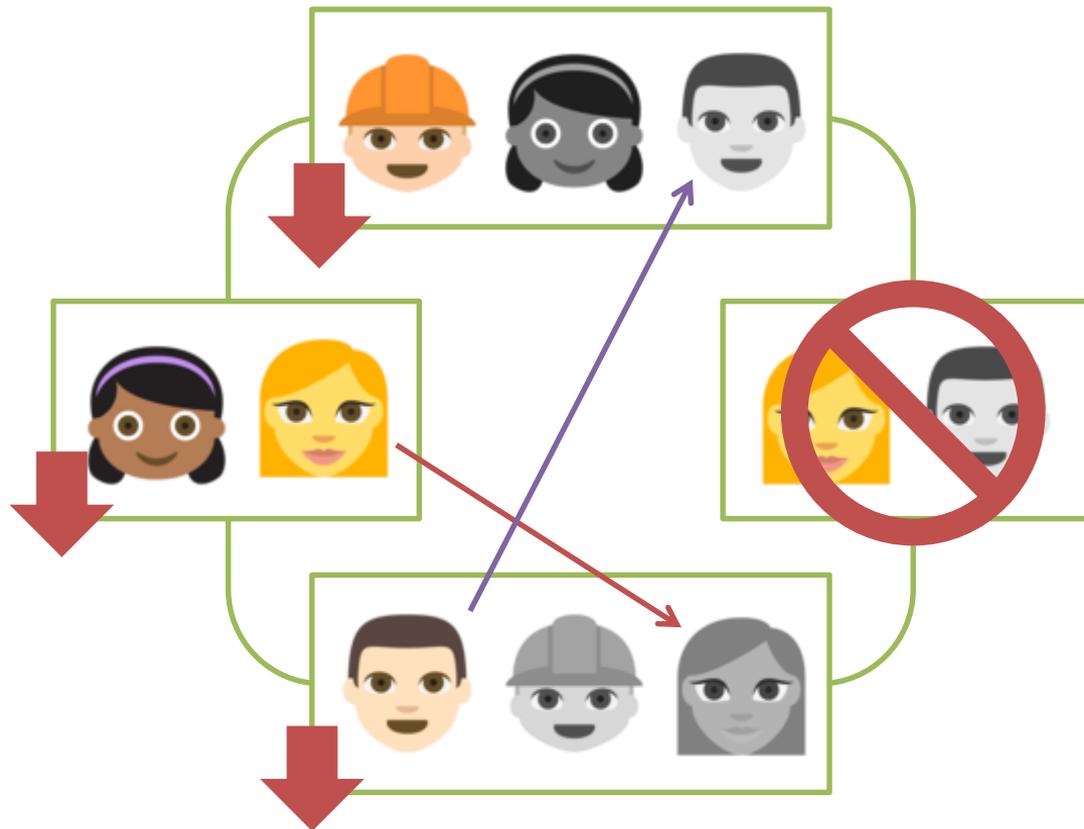


Общий

# Резервирование



# Ребалансировка



# Зоны и ячейки

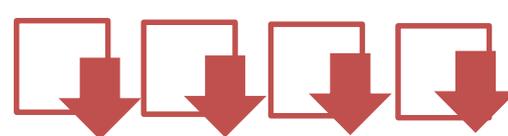
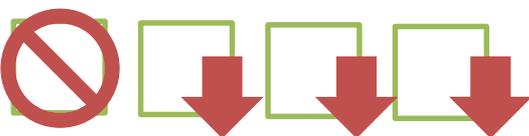
ЦОД

ЦОД

Ячейка



Ячейка



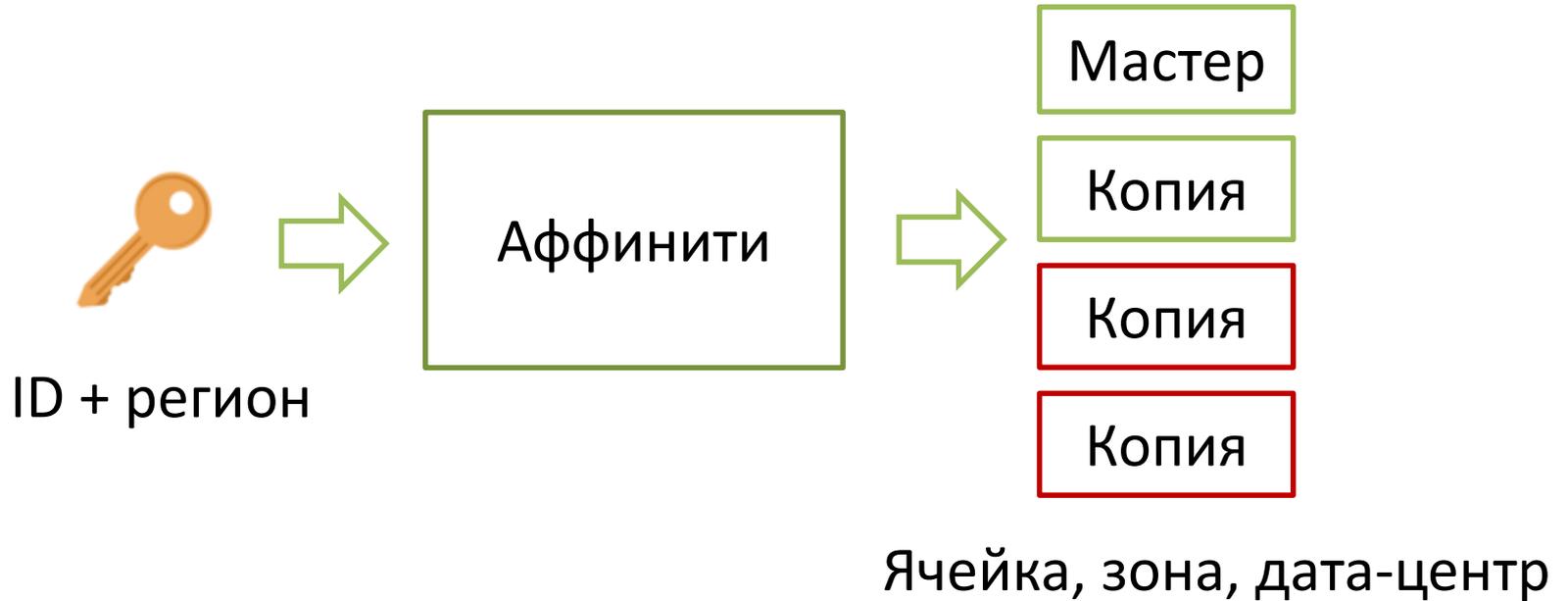
Зона

Ячейка



Зона

# Аффинити функция



# Аффинити функция

```
public interface AffinityFunction extends Serializable {  
  
    List<List<ClusterNode>> assignPartitions  
        (AffinityFunctionContext affCtx);  
  
    int partition(Object key);  
  
    int partitions();  
  
    void reset();  
  
    void removeNode(UUID nodeId);  
}
```

# Аффинити функция: assignPartitions

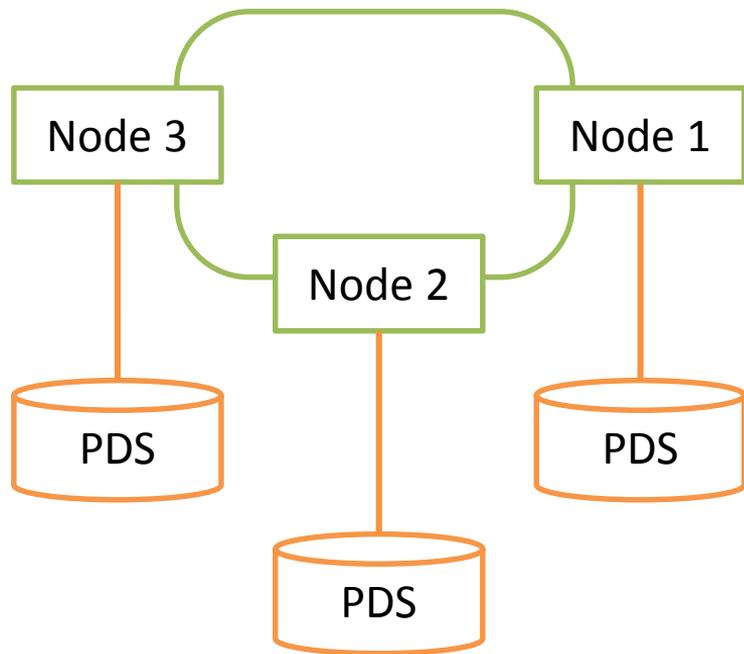
|          |   | Мастер | Копия 1 | Копия 2 | Копия 3 |
|----------|---|--------|---------|---------|---------|
| Партиции | 0 | 0      | 1       | 4       | 5       |
|          | 1 | 1      | 2       | 5       | 6       |
|          | 2 | 2      | 3       | 6       | 7       |
|          | 3 | 7      | 4       | 3       | 0       |
|          | 4 | 4      | 5       | 0       | 1       |

Архитектурные развилки

# ОБНОВЛЕНИЕ GRIDGAIN



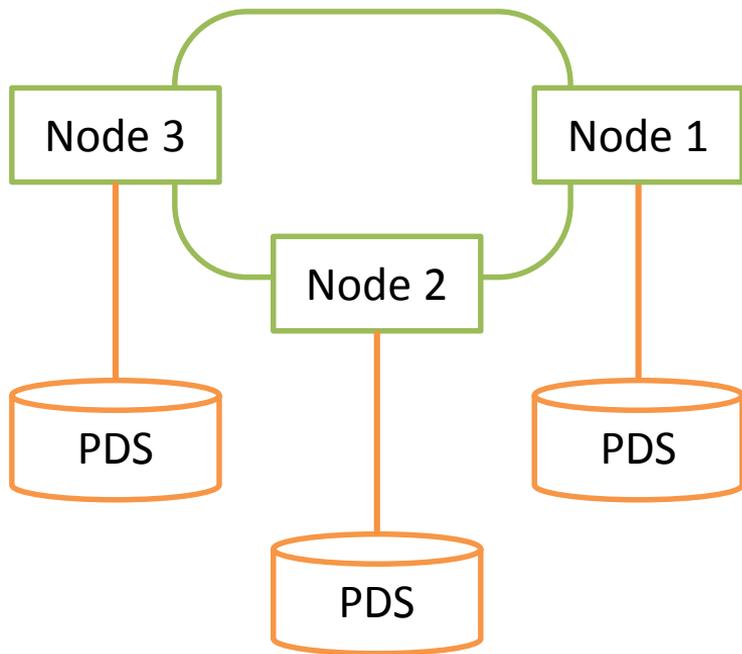
# Обновление без остановки кластера



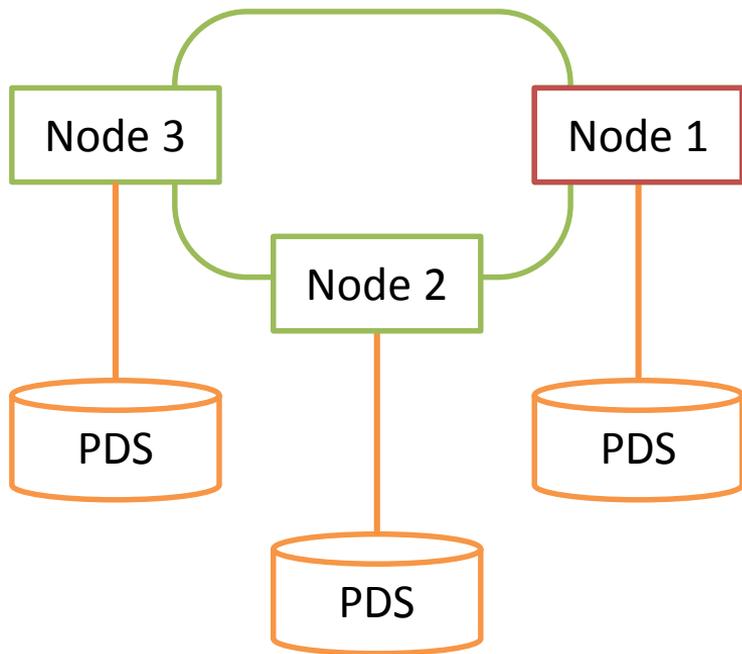
Вы пробовали выключить  
и снова включить?



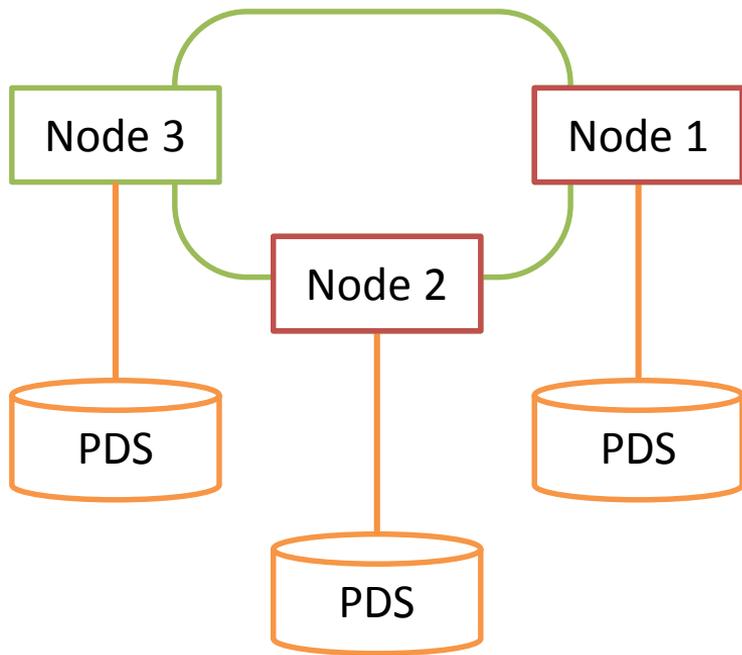
# Поочерёдное обновление узлов



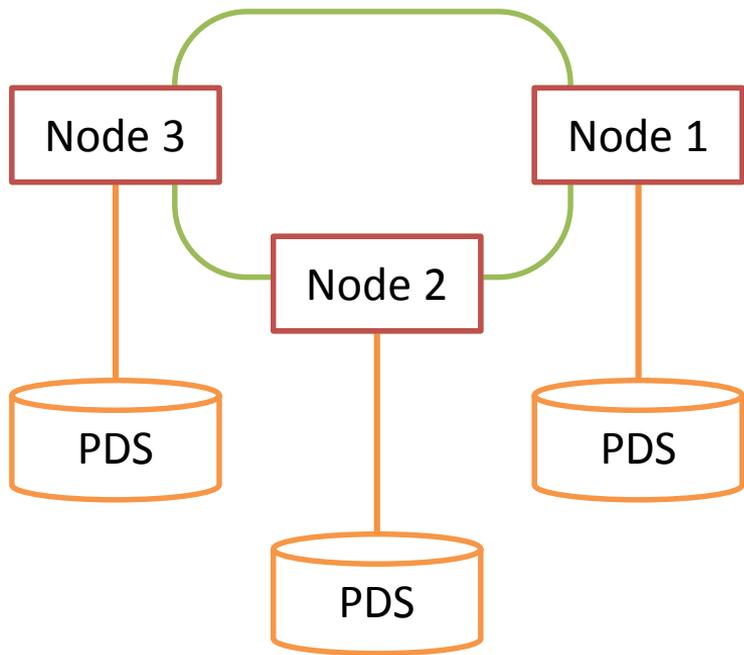
# Поочерёдное обновление узлов



# Поочерёдное обновление узлов



# Поочерёдное обновление узлов



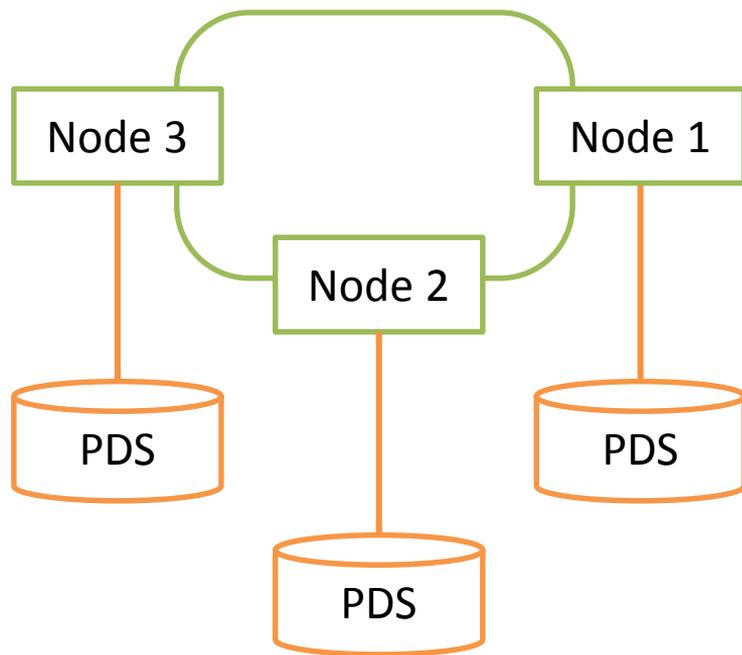
👍 Технологически просто

👎 Только минорные версии

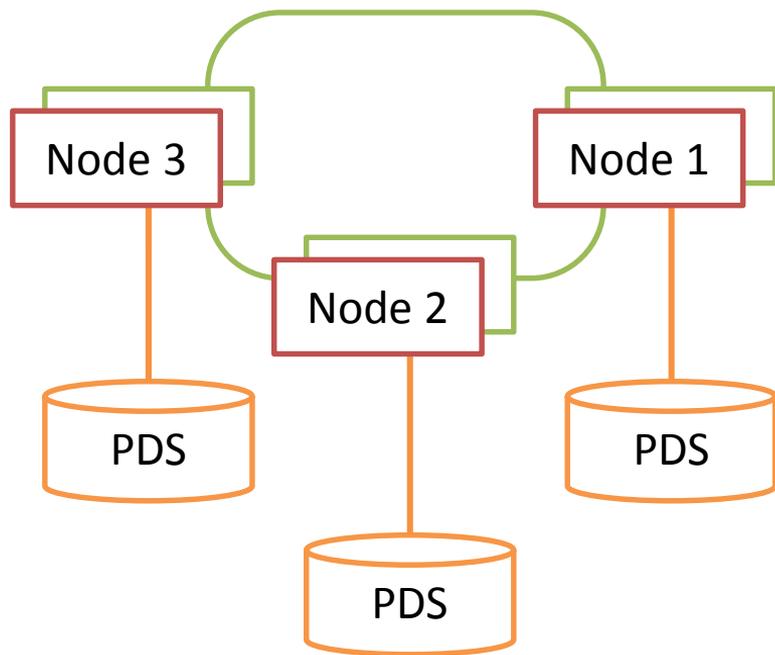
👎 Нагрузка на кластер

👎 «Дело жизни»

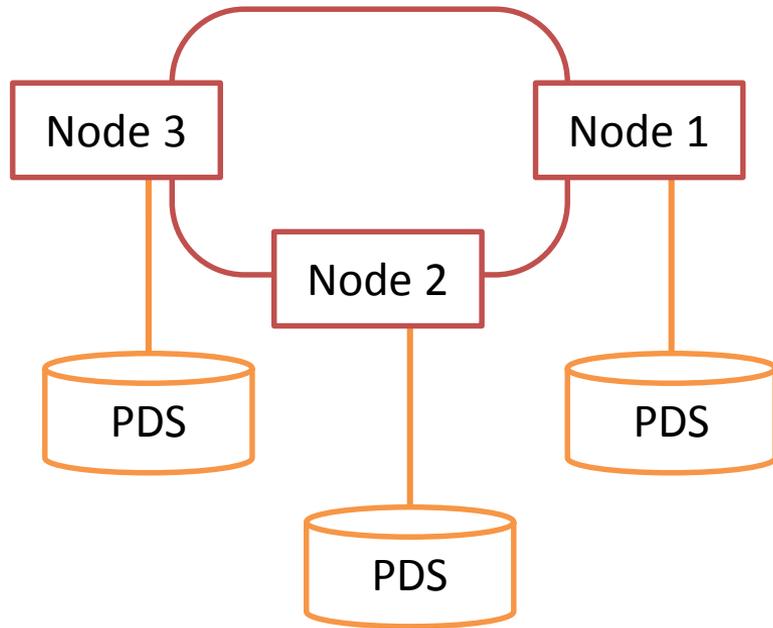
# «Холодное» обновление



# «Холодное» обновление



# «Холодное» обновление



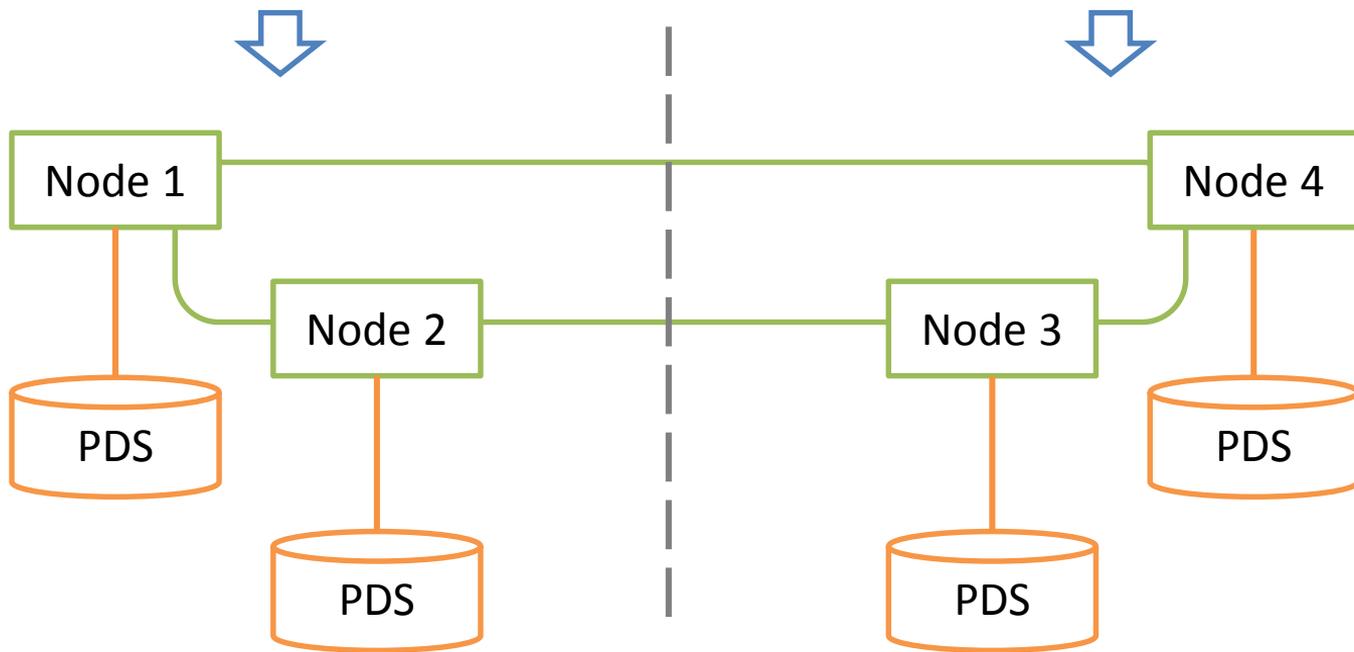
👍 Кластер обновляется быстро

👎 Нужна совместимость PDS

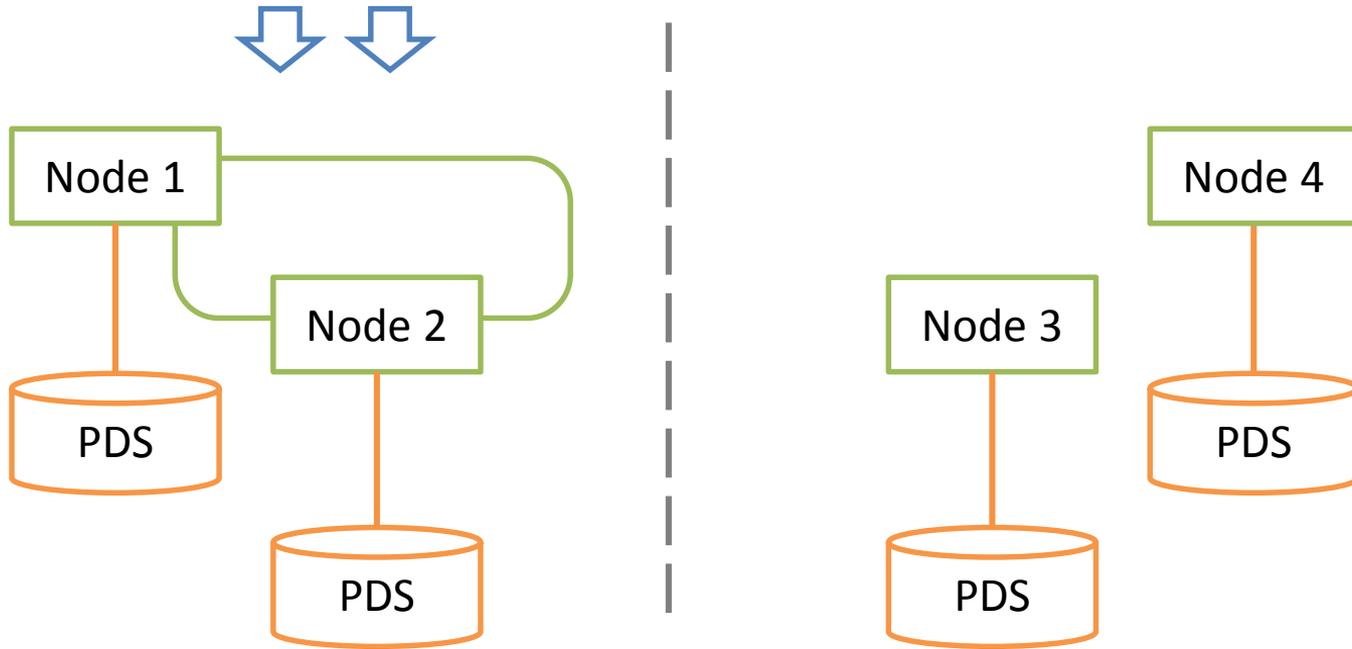
👎 Необходимы ресурсы на узле

👎 Технологически сложно

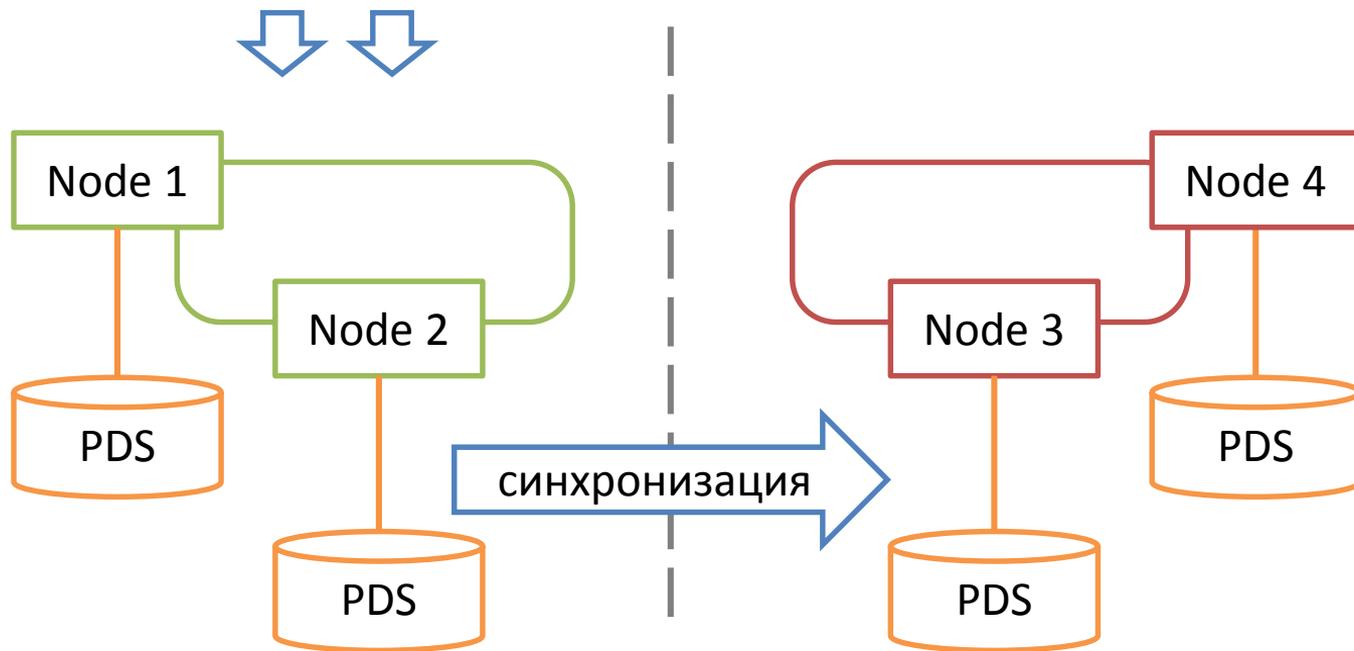
# Геообновление



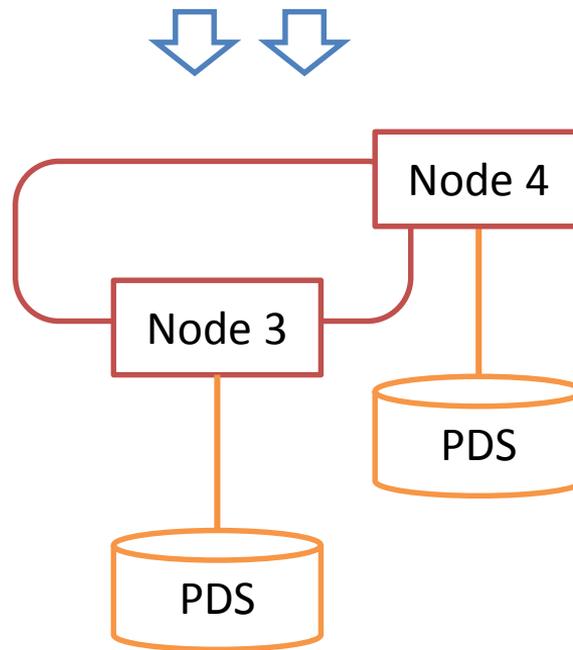
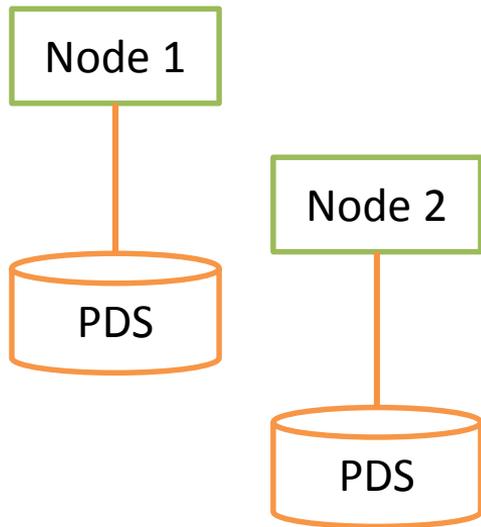
# Геообновление



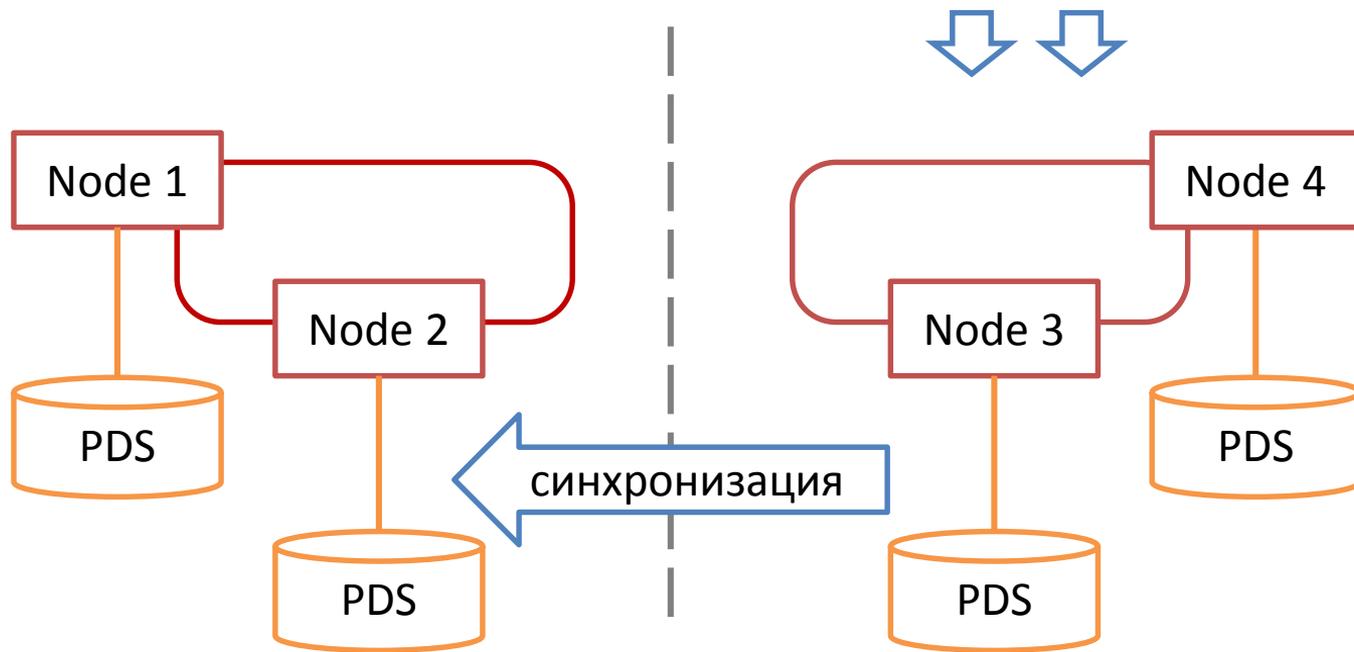
# Геообновление



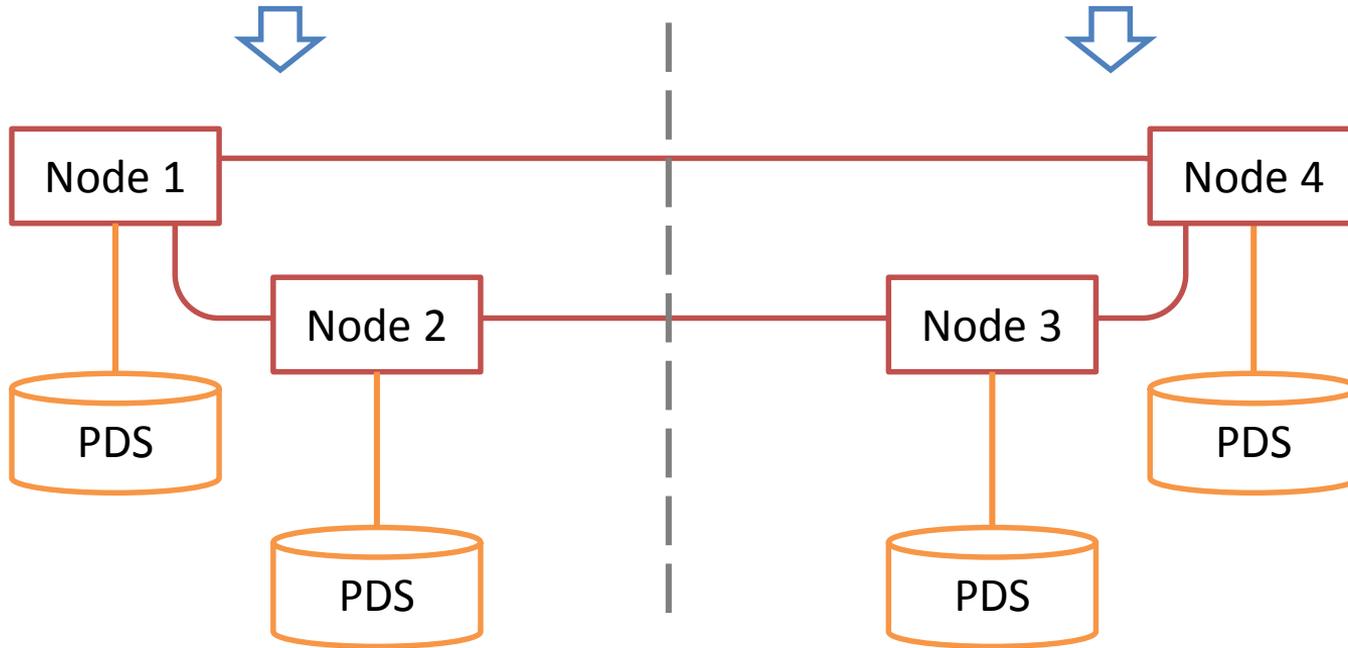
# Геообновление



# Геообновление



# Геообновление



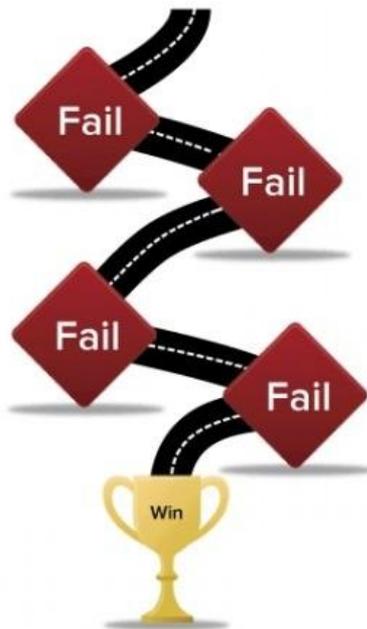
👍 Универсальность      👎 Сложно      👎 Нагрузка

# Заключение

ЧТО ДУМАЕТ  
БОЛЬШИНСТВО  
ЛЮДЕЙ



ЧТО ЗНАЮТ  
УСПЕШНЫЕ ЛЮДИ



@douglaskarr



Алексей Курагин

Сбербанк-Технологии

[AVKuragin.SBT@sberbank.ru](mailto:AVKuragin.SBT@sberbank.ru)

