



**#JPOINT**  
**#HAZELCAST**

**#COMPUTING**

**Распределяй и  
властвуй**

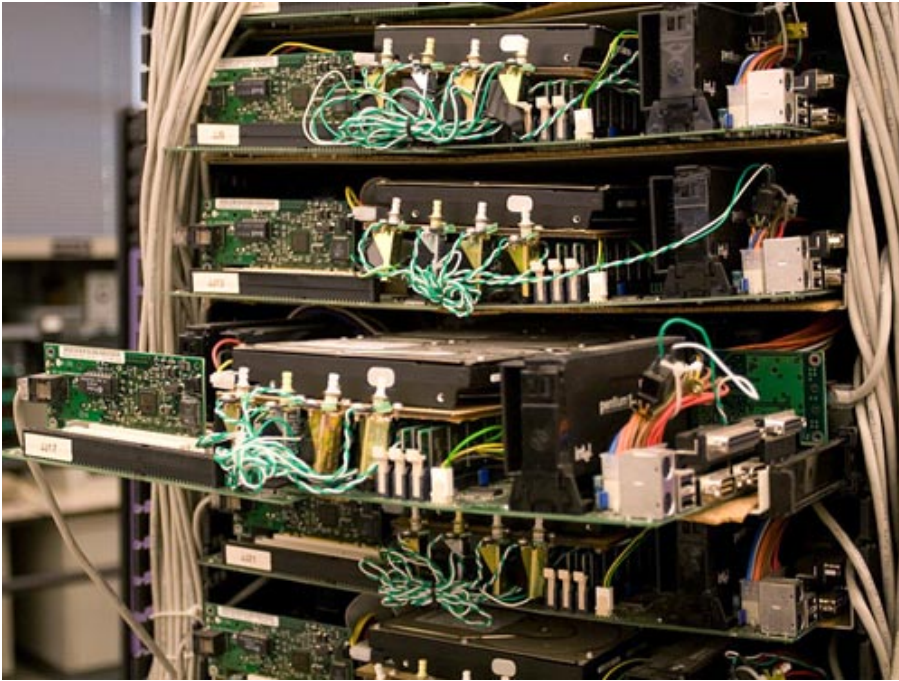
**Введение в  
распределенные  
системы**



**@gamussa**

**Виктор Гамов**  
**Senior Solutions Architect, Hazelcast**

# Сегодня в программе



- Введение в распределенные вычисления
- Распределенный ExecutorService
- MapReduce
- Вопросы и ответы



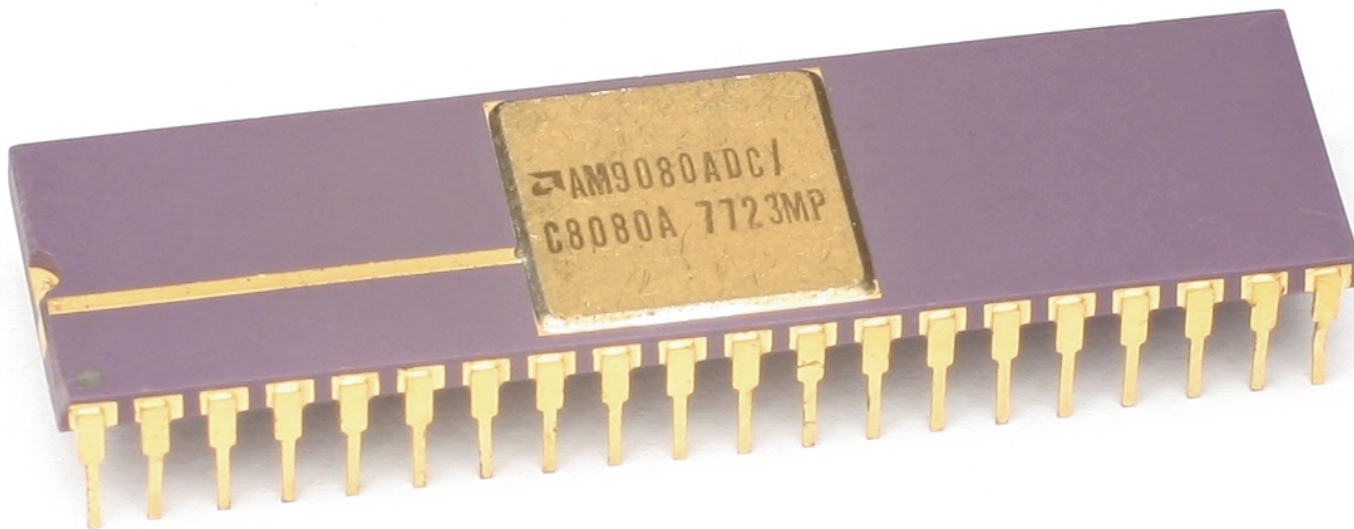
# Распределенные вычисления или многоядерные процессоры на стероидах



# Распределенные вычисления в каждый дом



# А ты помнишь как все начиналось?

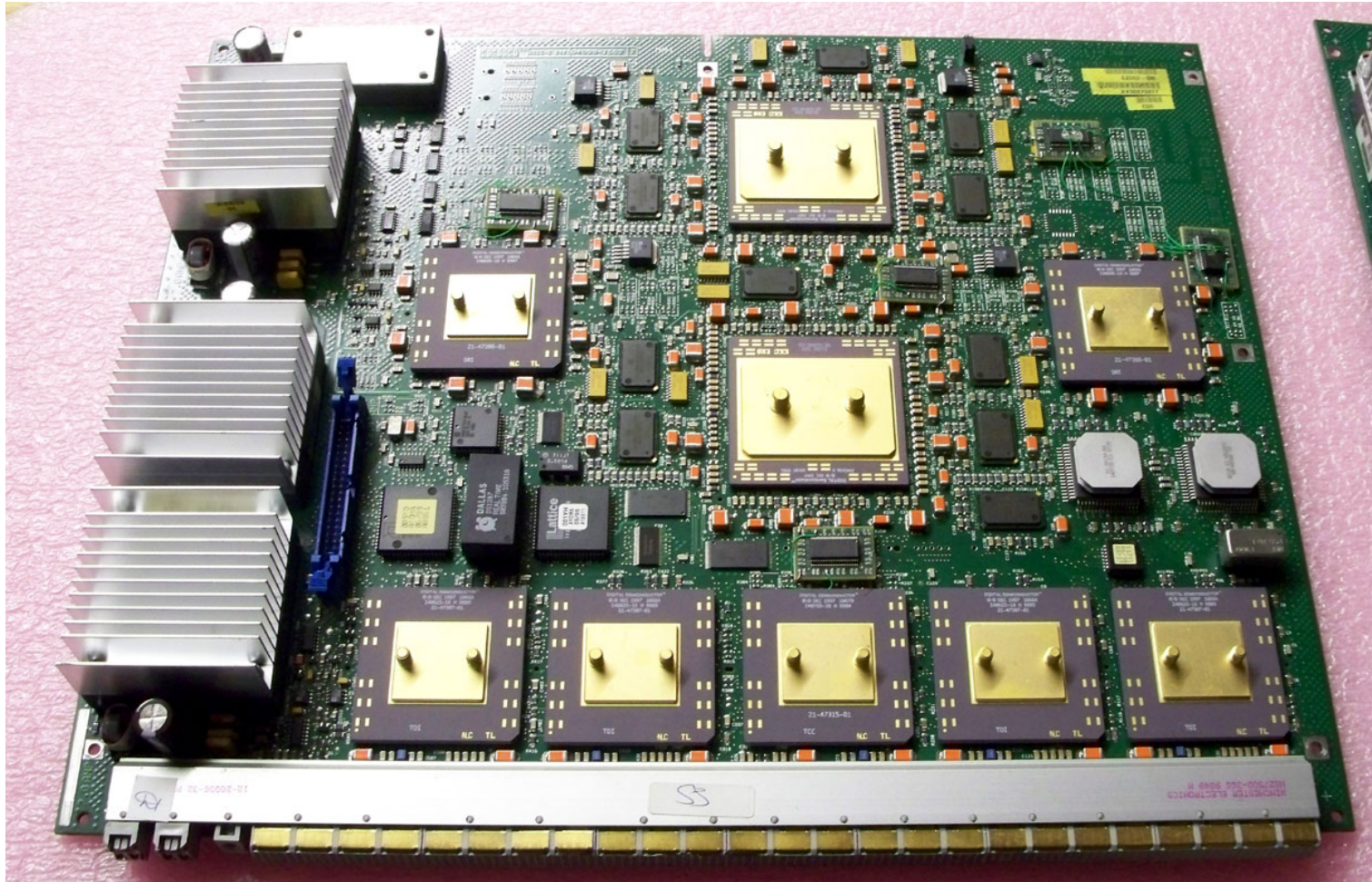


[https://en.wikipedia.org/wiki/Advanced\\_Micro\\_Devices#/media/File:KL\\_Advanced\\_Micro\\_Devices\\_AM9080.jpg](https://en.wikipedia.org/wiki/Advanced_Micro_Devices#/media/File:KL_Advanced_Micro_Devices_AM9080.jpg)

# Многоядерные процессоры - не новинка



# Кластерезуй это!



# Суперкомпьютеры!



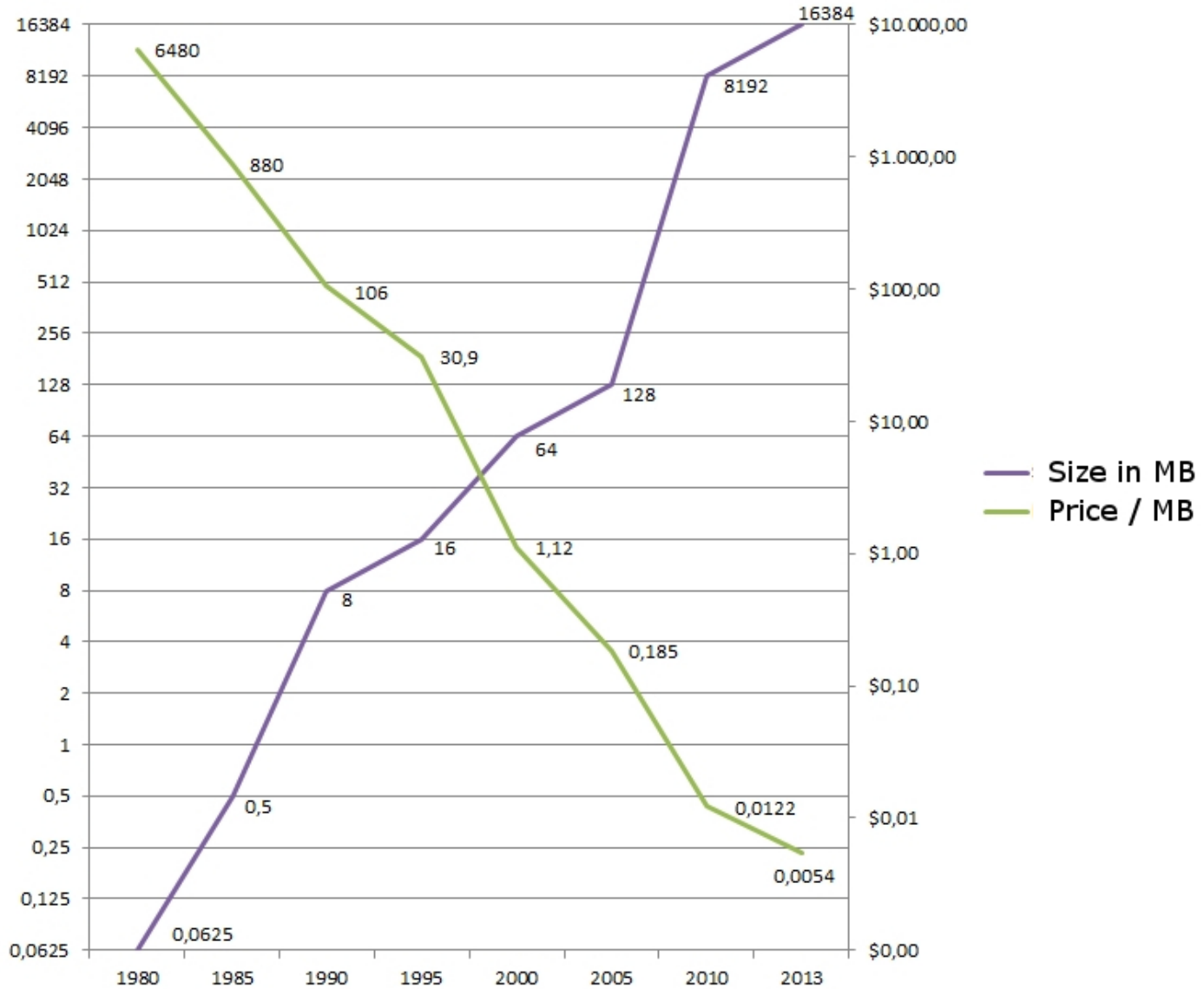


# Вычисления в облаках





# Память дешевет

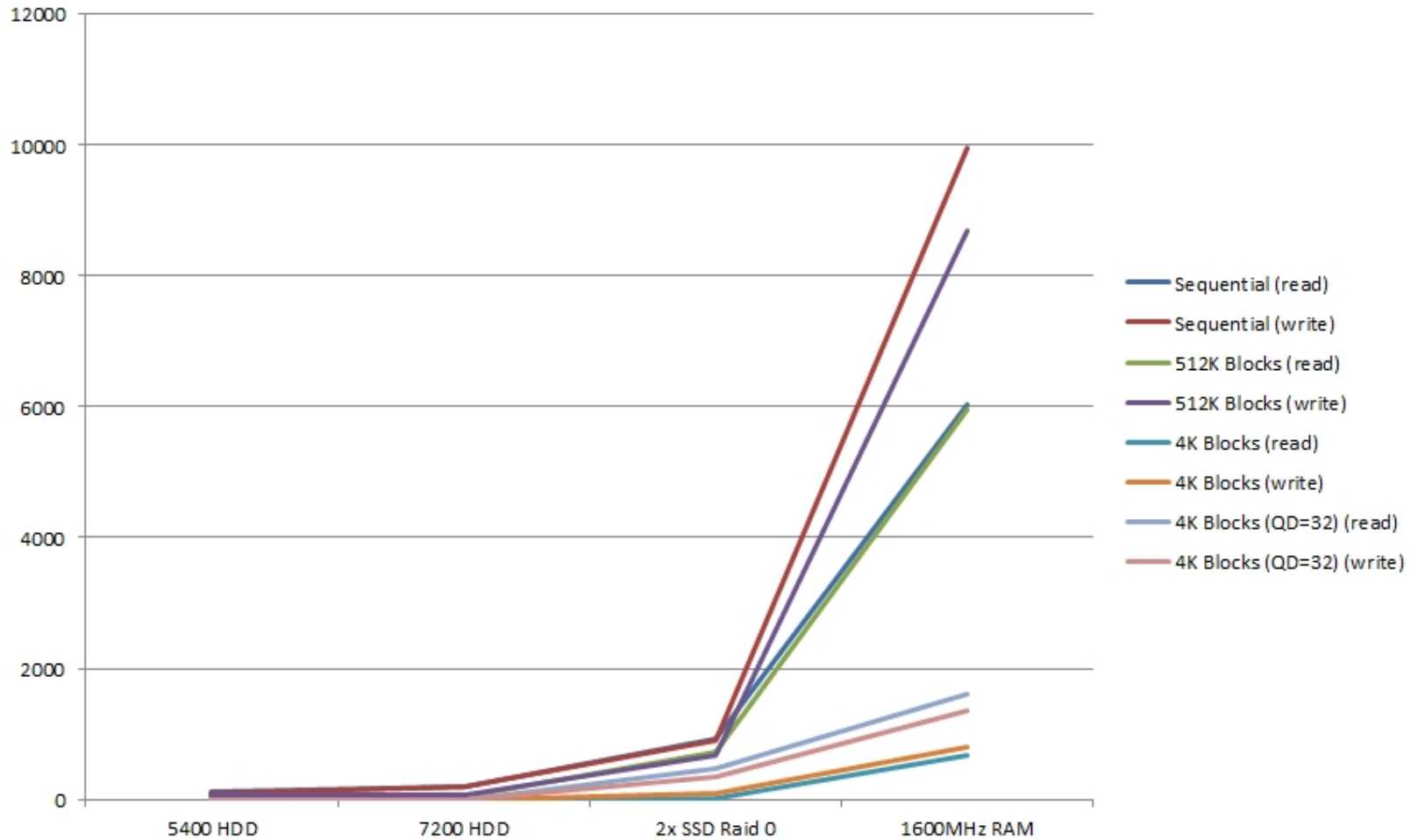




# Вычисления в памяти? А почему бы и нет!



# Сравним скорости





# Распределенный ExecutorService: вся мощь кластера к вашим услугам

## Что такое распределенный ExecutorService

- Расширение `j.u.c.ExecutorService`
- `Runnable` / `Callable` (только `Serializable`)
- Один таск – много нод

## Пример: распечатать адреса всех нод

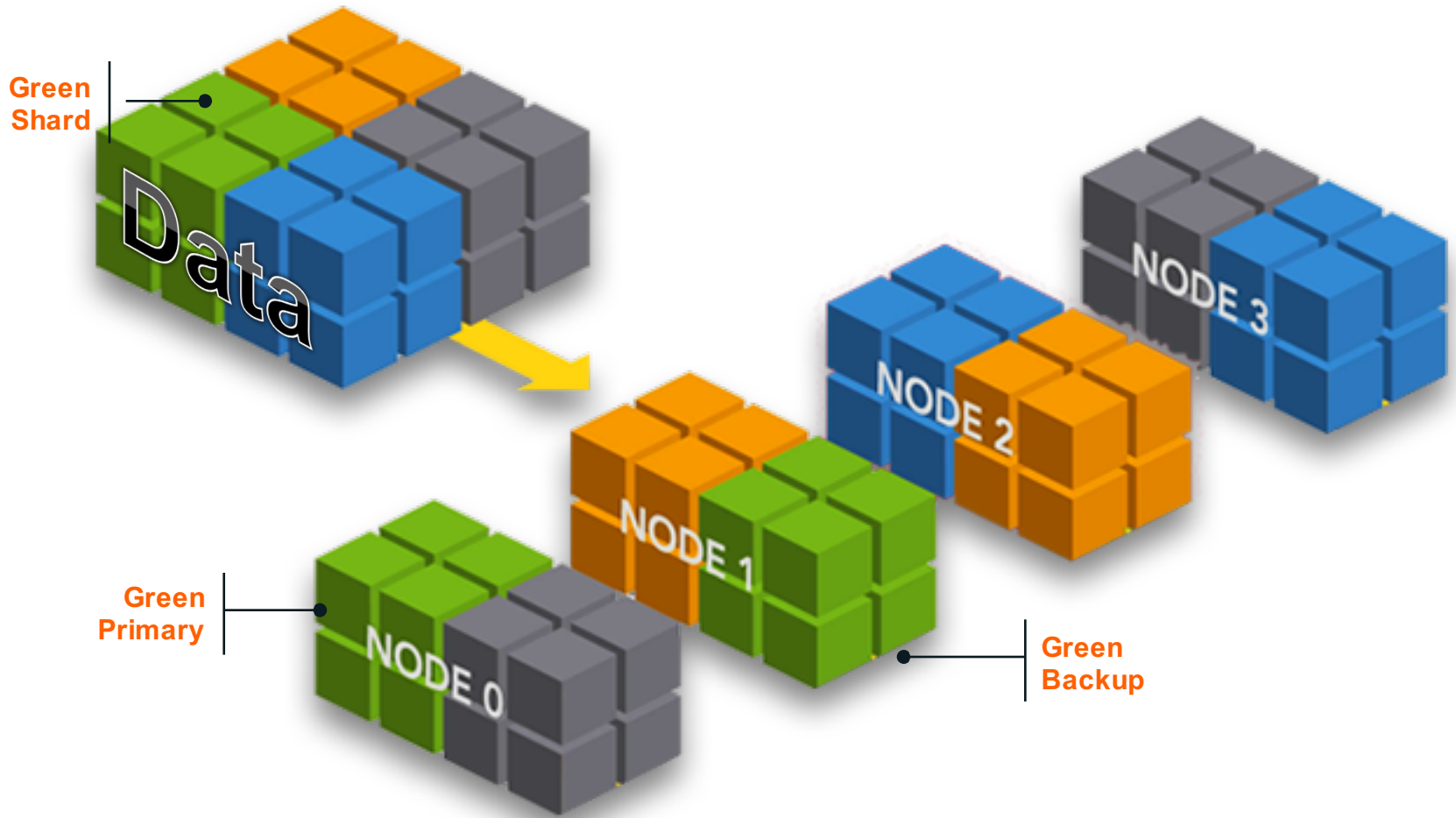
```
public class MemberRunnable
    implements Runnable, HazelcastInstanceAware {

    transient HazelcastInstance hazelcastInstance;

    @Override public void run() {
        final Member localMember =
hazelcastInstance.getCluster().getLocalMember();
        System.out.println(localMember.getAddress());
    }

    @Override public void
setHazelcastInstance(HazelcastInstance hazelcastInstance) {
        this.hazelcastInstance = hazelcastInstance;
    }
}
```

# А чего сразу Hazelcast?







Демо



# MapReduce: Черная магия с планеты Google

## Области применения

- Анализ логов
- Запросы к данным
- Агрегация и сложение
- Распределенная сортировка
- ETL (Extract Transform Load)
- И еще много чего

## Постановка задачи

Показать Top10 слов  
Из романа в стихах  
«Евгений Онегин»

Eugene Onegin: A Romance of Russian  
Life in Verse

Aleksandr Sergeevich Pushkin



Published by the Library of Alexandria

## Основные шаги

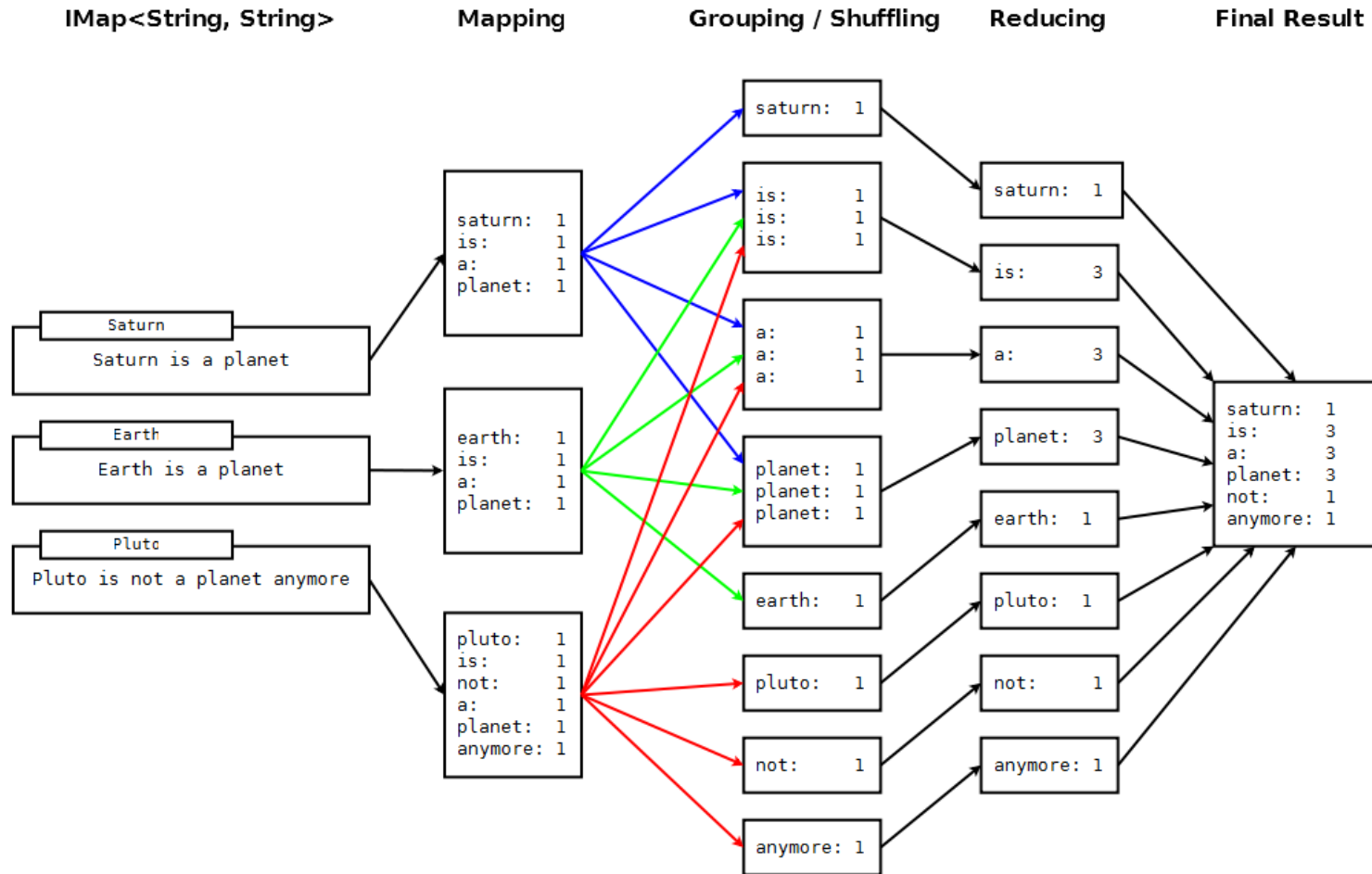
- Считываем данные
- Map / transform
- Reduce

## Основные шаги: расширенная версия

- Считываем данные
- Map / transform
- Combine
- Группировка
- Reduce
- Объединять



# Диаграмма потока данных в MapReduce программе





## Map: Данные трансформируются в набор пар «ключ-значение»

```
public void map(String key, String value, Context<String, Integer> context) {  
    StringTokenizer tokenizer = new StringTokenizer(value);  
  
    while (tokenizer.hasMoreTokens()) {  
        String word = WordCountExample.cleanWord(tokenizer.nextToken());  
  
        if (word.length() >= 4) {  
            context.emit(word.toLowerCase(), ONE);  
        }  
    }  
}
```





**Combine: значения объединяются в промежуточный результат чтобы уменьшить объем передаваемых данных**

```
public void combine(Integer value) {  
    count += value;  
}
```



## Reduce: агрегация данных для получения конечного результата

```
@Override  
public void reduce(Integer value) {  
    count += value;  
}
```

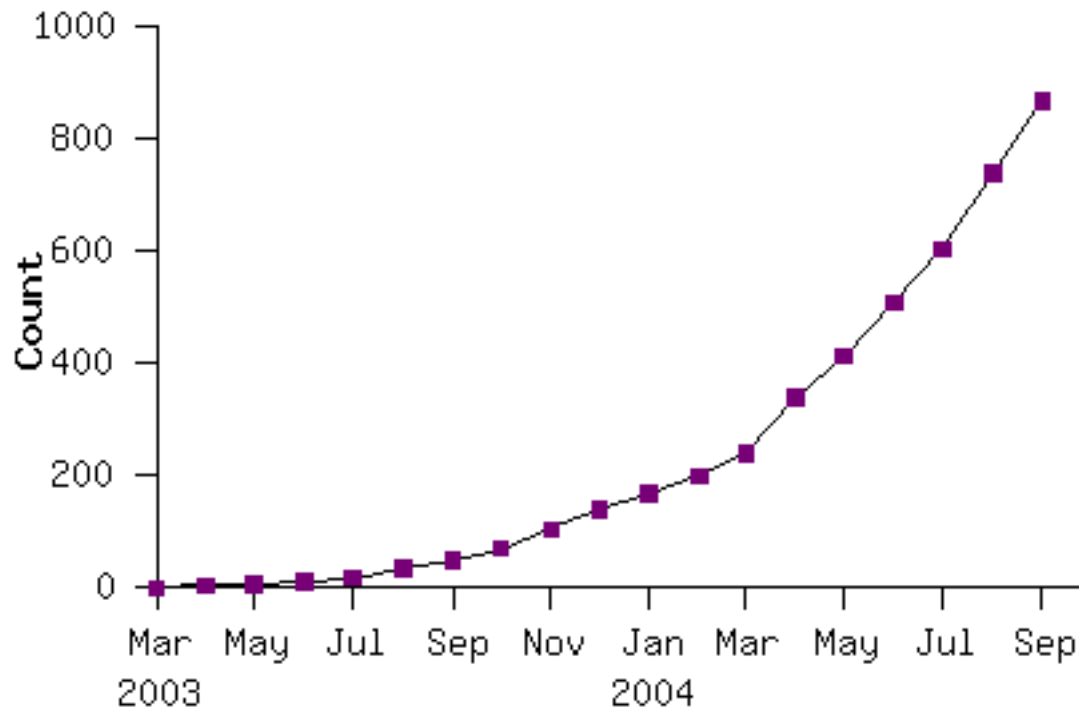
# FOR MATHEMATICIANS

Process:  $(K \times V)^* \rightarrow (L \times W)^* \Rightarrow [(l_1, w_1), \dots, (l_m, w_m)]$

Mapping:  $(K \times V) \rightarrow (L \times W)^* \Rightarrow (k, v) \rightarrow [(l_1, w_1), \dots, (l_n, w_n)]$

Reducing:  $L \times W^* \rightarrow X^* \Rightarrow (l, [w_1, \dots, w_n]) \rightarrow [x_1, \dots, x_n]$

# MapReduce в жизни Google



<https://research.google.com/archive/mapreduce-os-di04-slides/index-auto-0005.html>



Демо

# В качестве заключения

- **Распределенное программирование – это не сложно!**
- **И не дорого**
  - Обычное железо
  - Open Source Software FTW!
- **Учите матчасть**
  - <https://research.google.com/archive/mapreduce-osdi04-slides/index.html>
  - <http://www.hazelcast.org/learn>
- Код <https://github.com/gAmUssA/jpoint-2016-computing-talk>

**ЗАДАВАЙ УМНЫЕ ВОПРОСЫ!**

**#JPOINT #COMPUTING #HAZELC**

Viktor Gamov  
@gamussa

