

大数据编程

2-1

大数据核心技术

中央财经大学 商学院
姚凯
2016

主要内容

- ❖ 知识回顾
- ❖ 大数据特点
- ❖ 大数据技术难题
- ❖ 大数据核心技术

大数据相关概念：虚拟化

虚拟硬件

虚拟机

虚拟内存

虚拟桌面

虚拟网络

虚拟存储

大数据相关概念：分布式计算

分布式
存储

分布式
计算模型

分布式
系统

分布式
算法

云计算类型

SaaS (Software-as-a-Service): 软件即服务，厂商将应用软件统一部署在自己的服务器上，客户可以根据自己实际需求，通过互联网向厂商定购所需的应用软件服

PaaS (Platform-as-a-Service): 平台即服务，把服务器平台作为一种服务提供的商业模式，把客户开发的或收购的应用程序部署到供应商的云计算基础设施上去

IaaS (Infrastructure-as-a-Service): 基础设施即服务，消费者通过 Internet 可以从完善的计算机基础设施获得服务

主要内容

- ❖ 知识回顾
- ❖ 大数据特点
- ❖ 大数据技术难题
- ❖ 大数据核心技术

大数据的特点

大数据的特点

Volume 体量

大数据的特点

Volume 体量

Velocity 速度

大数据的特点

Volume 体量

Velocity 速度

Variety 多样

大数据的特点

Volume 体量

Velocity 速度

Variety 多样

大数据的特点

Value (价值)

Veracity (真实性)

大数据的应用

大数据的应用

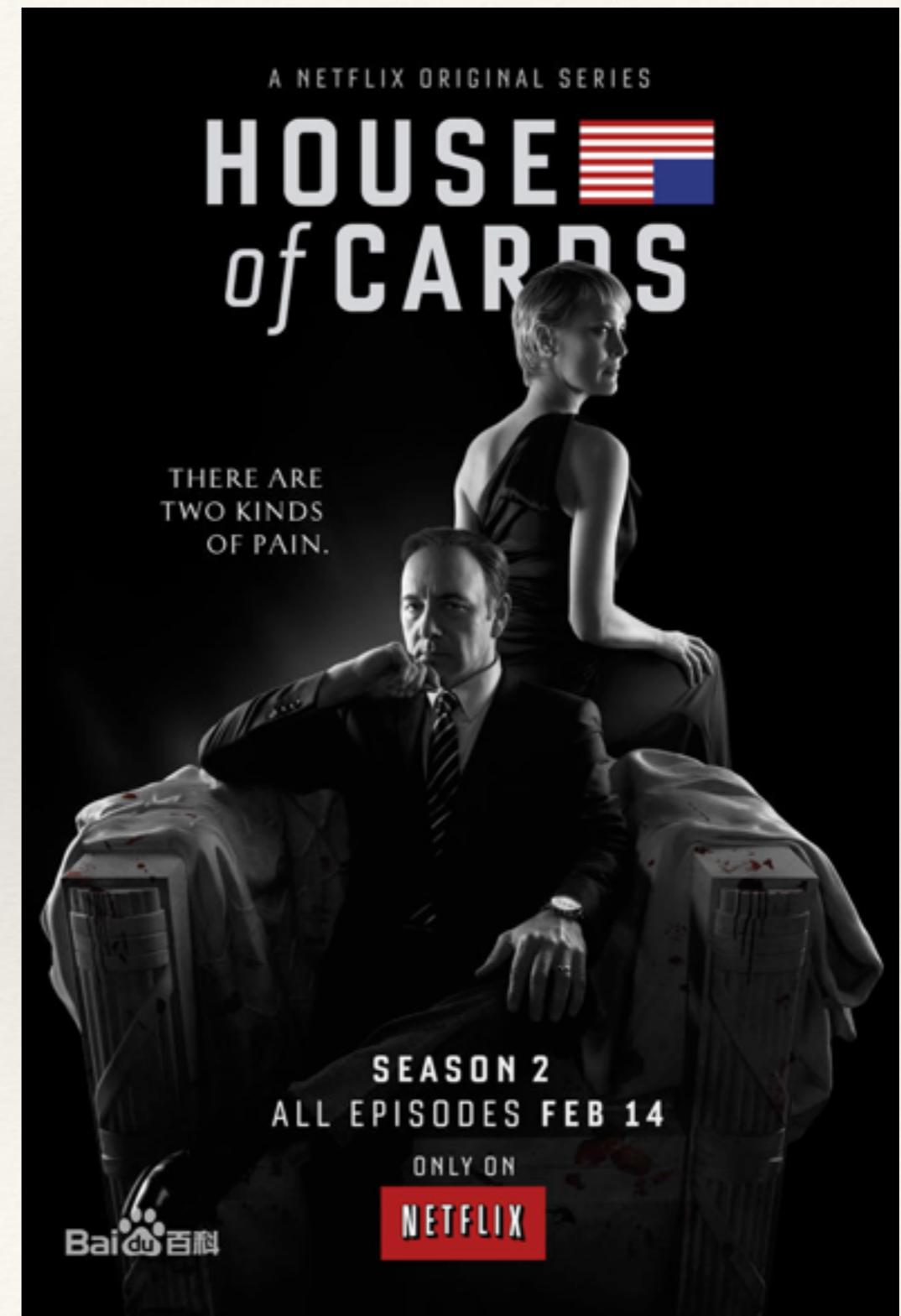
- ❖ 分布式框架
- ❖ 个性化商品推荐
- ❖ 联合分析
- ❖ 精准广告投放

大数据的应用

- ❖ 分布式框架
- ❖ 个性化商品推荐
- ❖ 联合分析
- ❖ 精准广告投放
- ❖ 客户终身价值
- ❖ 个人征信
- ❖ 搜索引擎
- ❖ 时空大数据



Netflix — 纸牌屋



大数据面临的挑战

- ❖ 大数据不能完全替代传统数据
 - ❖ 数据价值
 - ❖ 信息爆炸
 - ❖ 大数据技术实用性

大数据面临的挑战

- ❖ 数据安全及保护
 - ❖ 个人隐私受到威胁
 - ❖ 需要保护的数据量给存储带来压力

大数据面临的挑战

- ❖ 数据安全及保护
 - ❖ 个人隐私受到威胁
 - ❖ 需要保护的数据量给存储带来压力

一天，一个男人冲进了一家位于明尼阿波利斯市郊的塔吉特商店，要求经理出来见他。

他气愤地说：“我女儿还是高中生，你们却给她邮寄婴儿服和婴儿床的优惠券，你们是在鼓励她怀孕吗？”而当几天后，经理打电话向这个男人致歉时，这个男人的语气变得平和起来。他说：“我跟我的女儿谈过了，她的预产期是8月份，是我完全没有意识到这个事情的发生，应该说抱歉的人是我。”

大数据面临的挑战

- ❖ 数据分析与精准预测
 - ❖ 历史数据难以刻画未来需求
 - ❖ 训练数据并不能代表真实数据
 - ❖ 系统缺陷造成重大损失

大数据面临的挑战

- ❖ 数据分析与精准预测
 - ❖ 历史数据难以刻画未来需求
 - ❖ 训练数据并不能代表真实数据
 - ❖ 系统缺陷造成重大损失

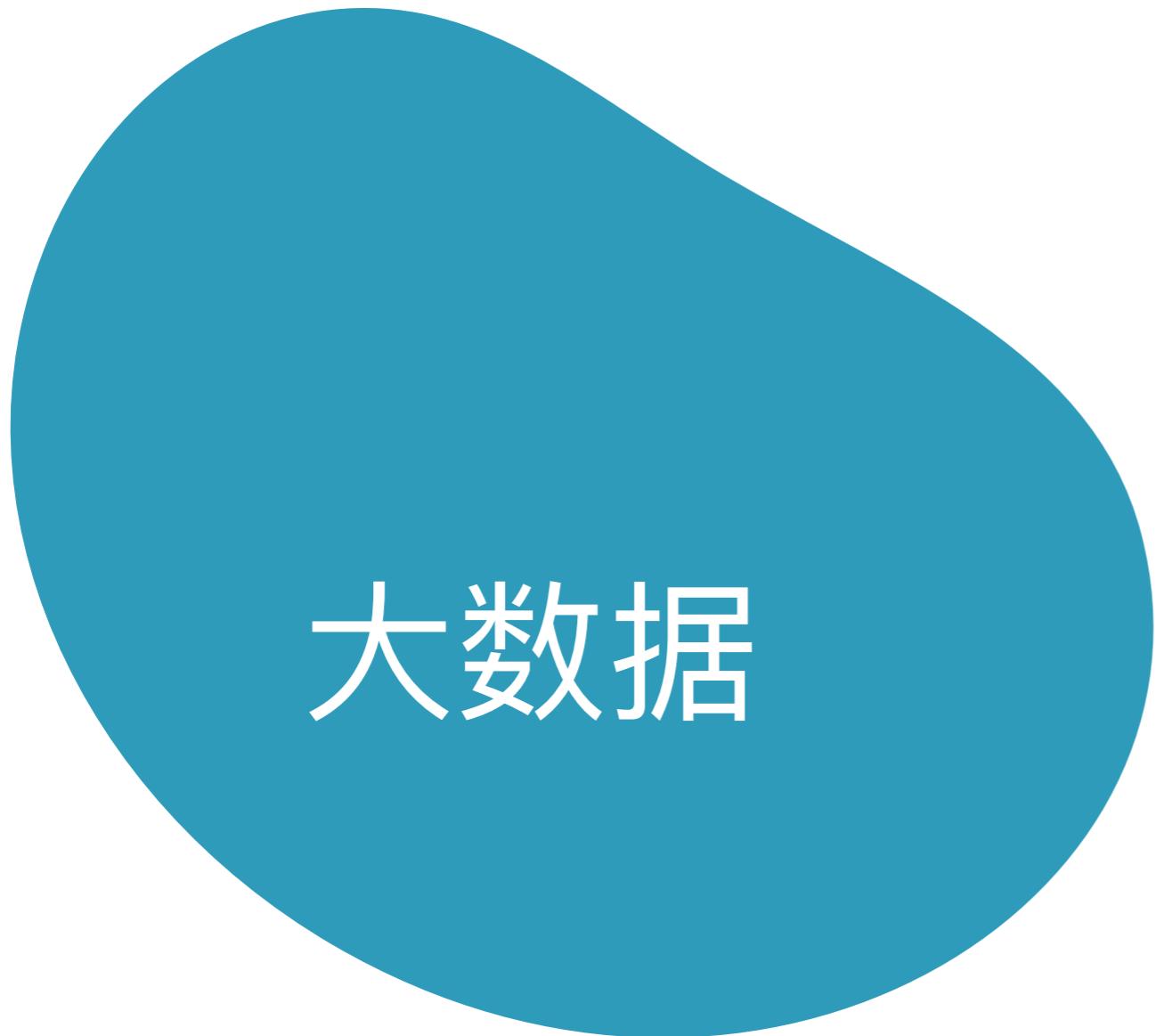
8·16光大证券乌龙指事件：由于订单生成系统存在的缺陷，导致在11时05分08秒之后的2秒内，瞬间重复生成26082笔预期外的市价委托订单；乌龙事件中共下单230亿，成交72亿，涉及150多只股票，损失约为1.94亿元。而光大证券7月实现营业收入2.15亿元，净利润0.45亿元。

主要内容

- ❖ 知识回顾
- ❖ 大数据特点
- ❖ 大数据技术难题
- ❖ 大数据核心技术

大数据难题

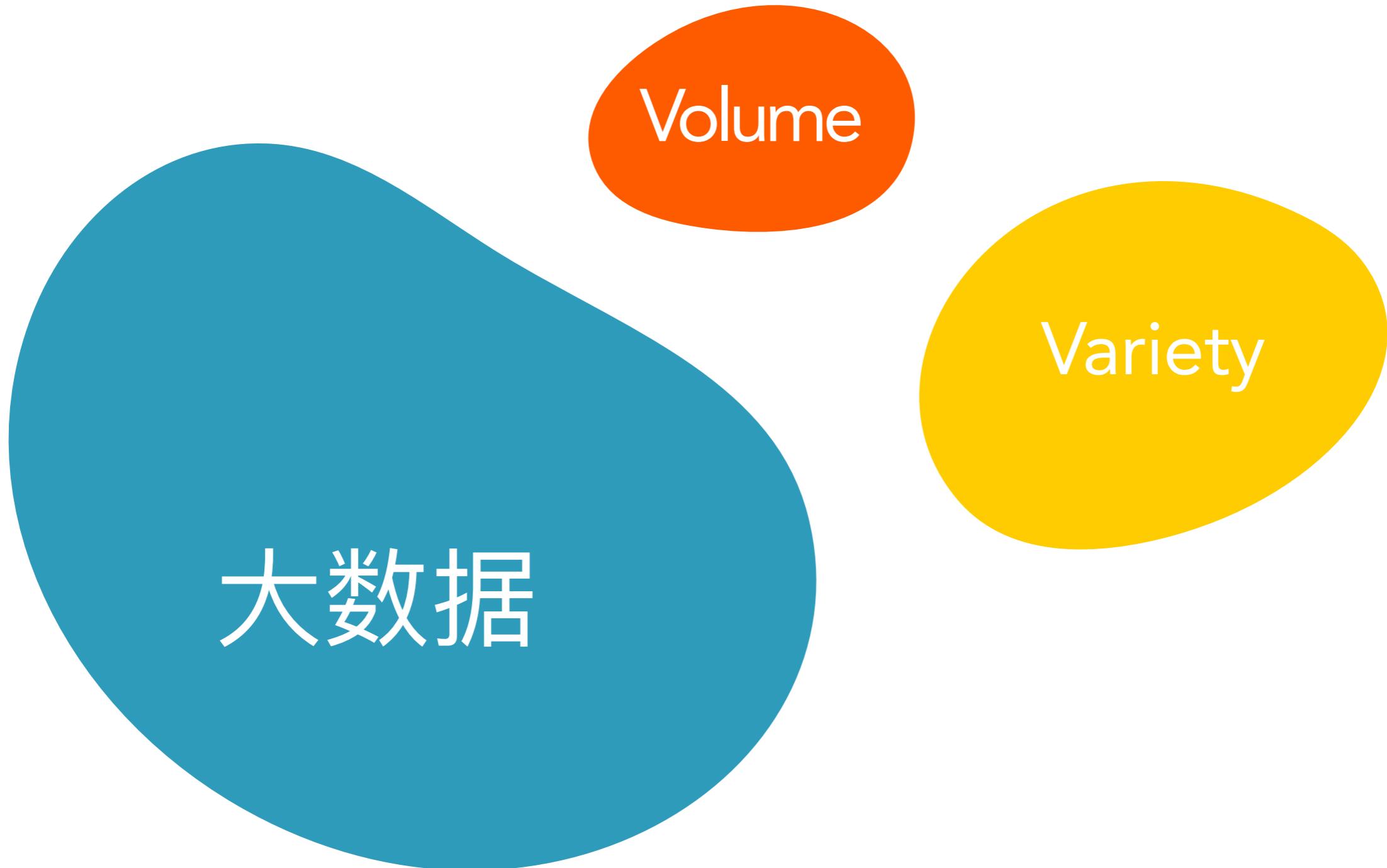
大数据难题



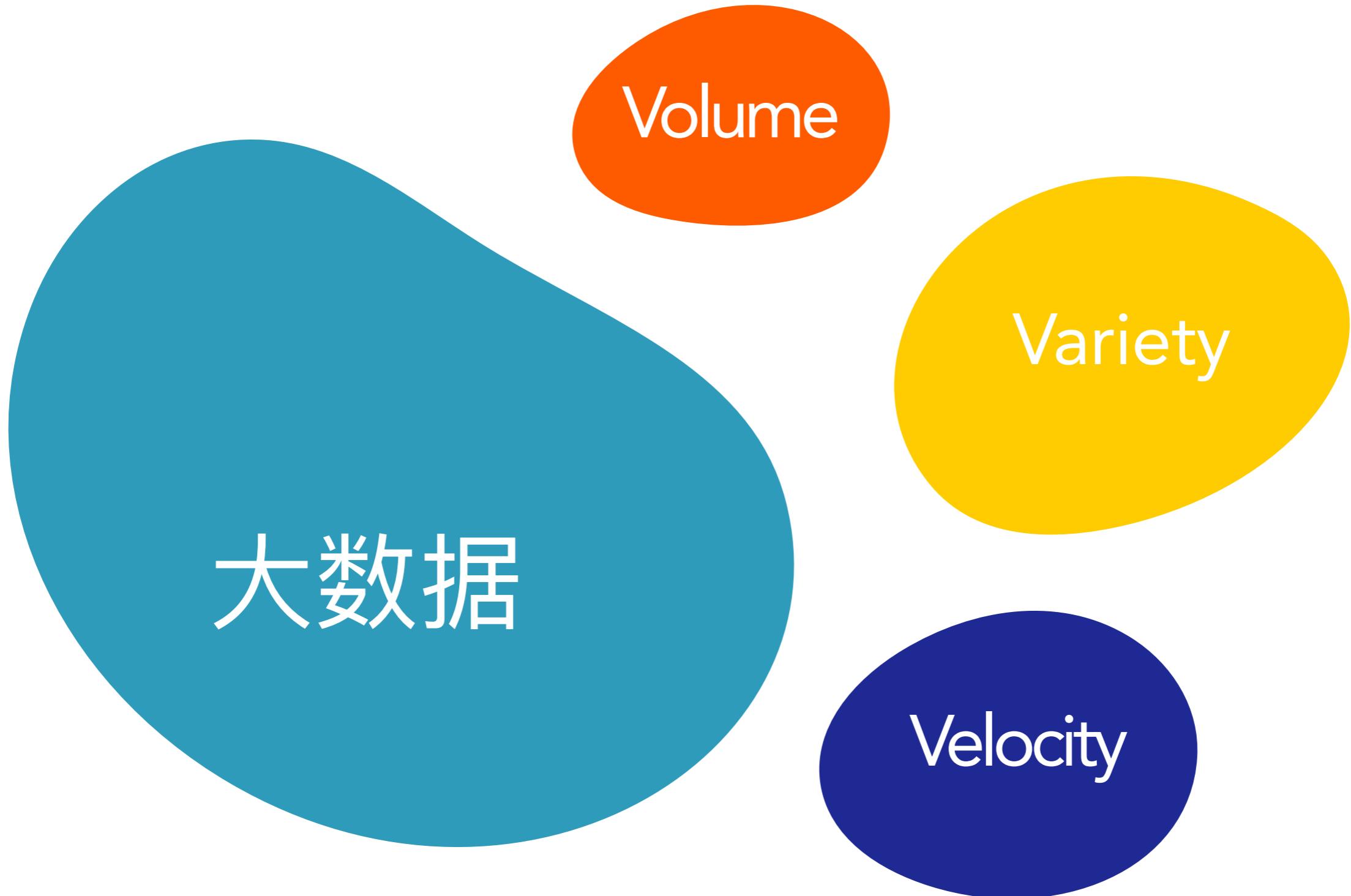
大数据难题



大数据难题



大数据难题



大数据难题

TB



PB

EB

大数据难题

13000+个
iPhone
应用下载

Skype上
37万+分钟的
语音通话

Twitter上发
布98000+新
微博

上传6600张
新照片到
flickr

发出1.68亿+
条Email

Facebook上
更新69.5万+
条新状态

YouTube上
上传600+新
视频

淘宝光棍节
10680+个新
订单

12306出票
1840+张



大数据难题

13000+个
iPhone
应用下载

Skype上
37万+分钟的
语音通话

Twitter上发
布98000+新
微博

上传6600张
新照片到
flickr

发出1.68亿+
条Email

Facebook上
更新69.5万+
条新状态

YouTube上
上传600+新
视频

淘宝光棍节
10680+个新
订单

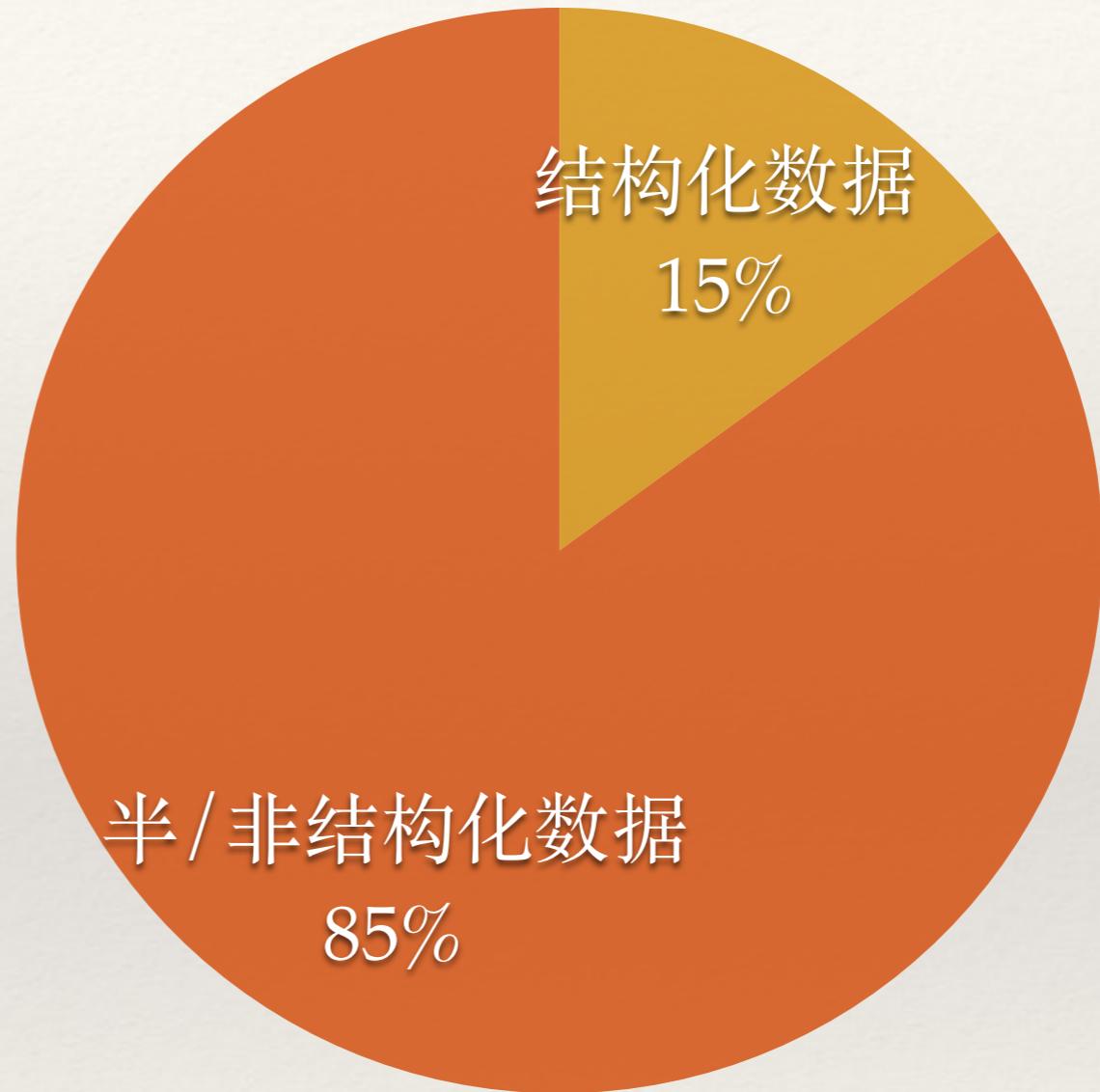
12306出票
1840+张



传统存储方式无法满足海量数据的要求

大数据面临的挑战

Variety



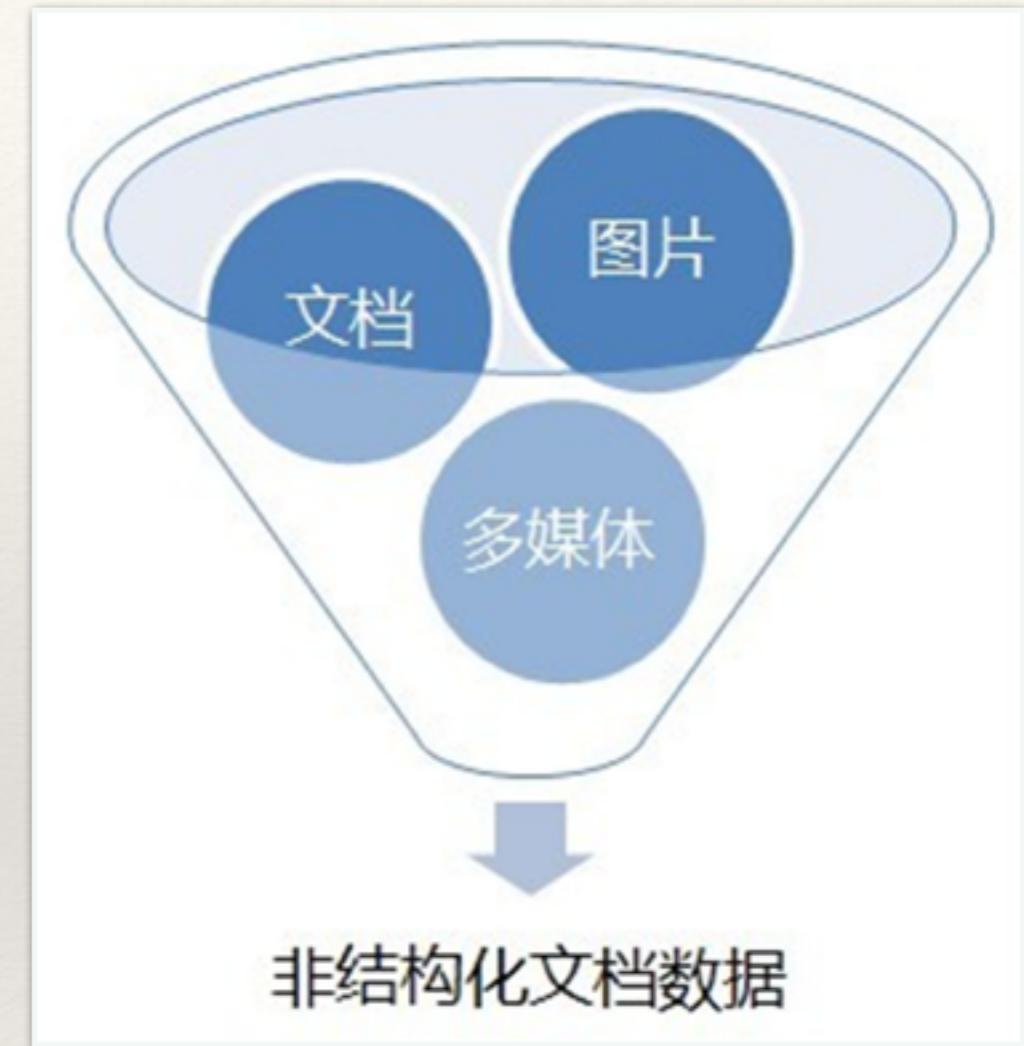
大数据面临的挑战

Variety



大数据面临的挑战

Variety



大数据面临的挑战

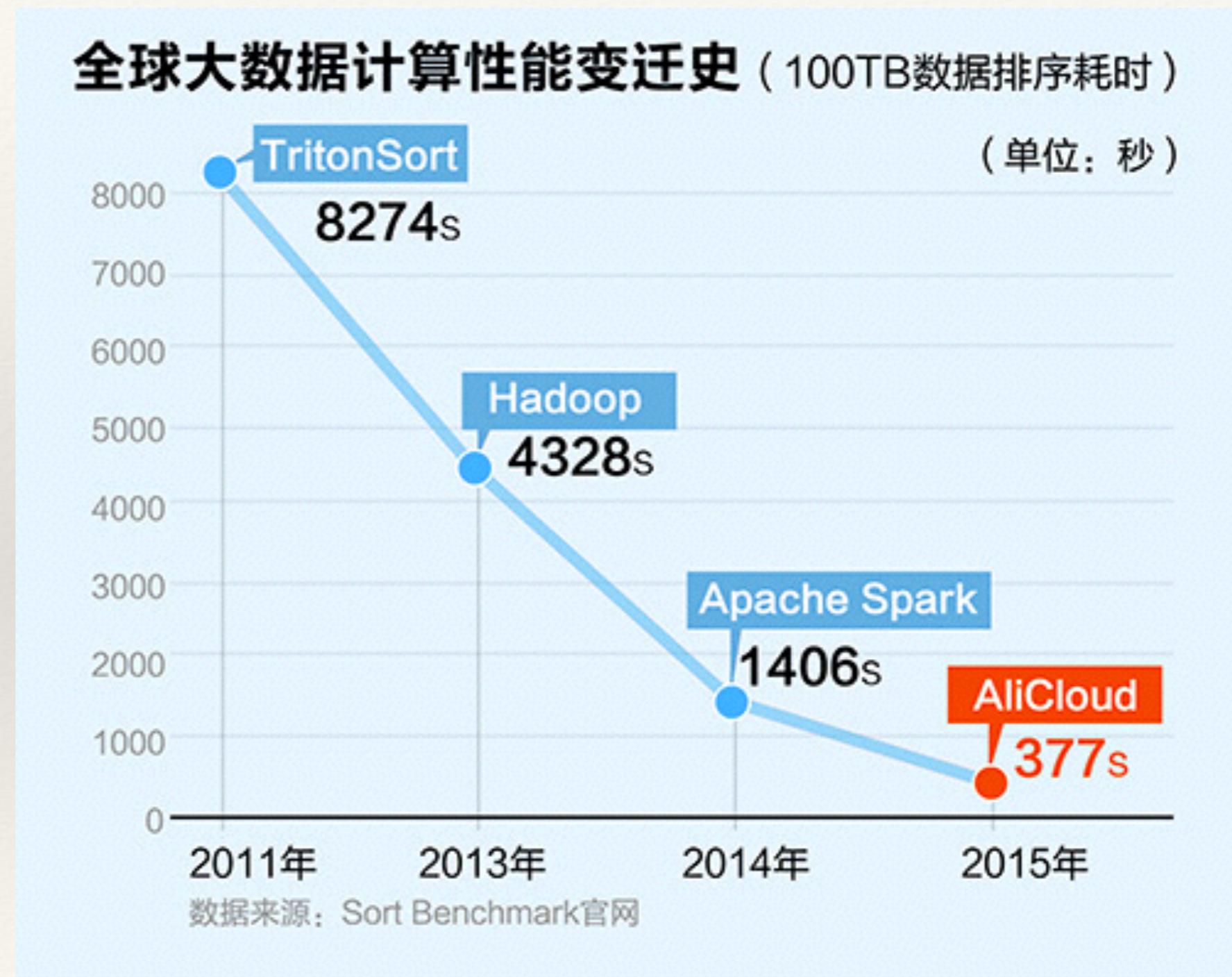


大数据面临的挑战



Velocity

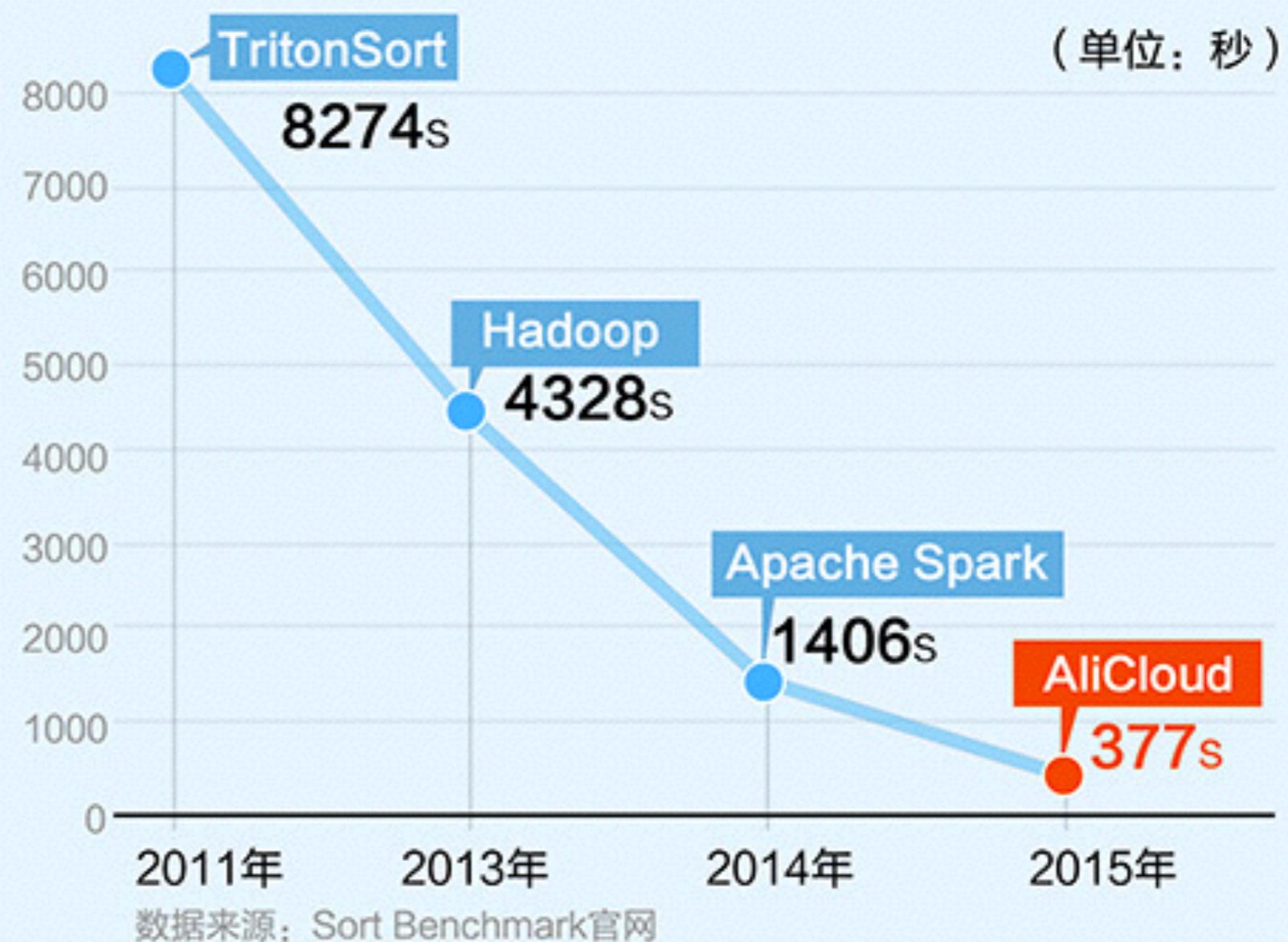
大数据面临的挑战



大数据面临的挑战



全球大数据计算性能变迁史 (100TB数据排序耗时)



大数据面临的挑战



1. 海量存储
2. 数据备份与恢复



1. 新型数据库
2. 新分析方法

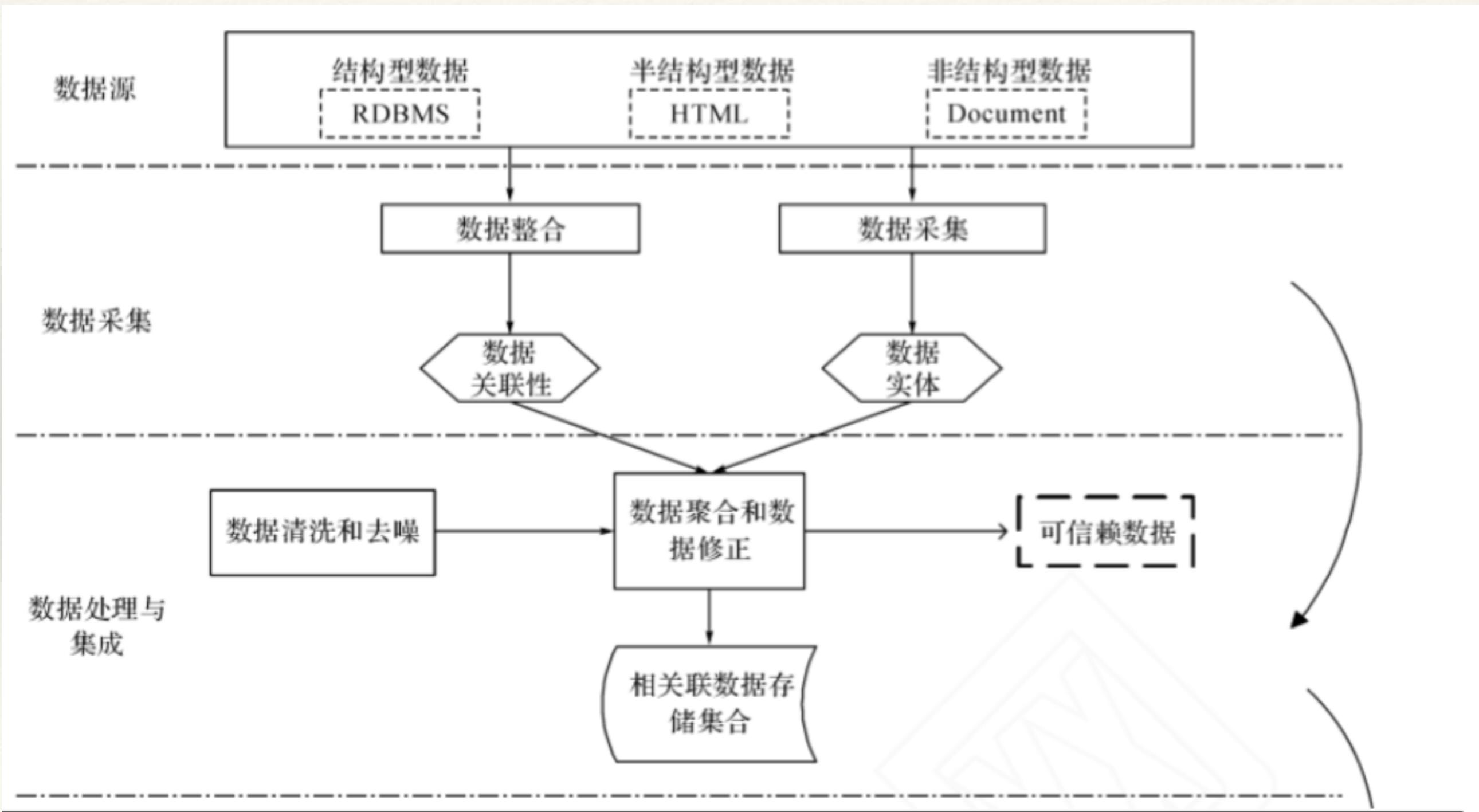


1. 实时处理能力
2. 新处理框架

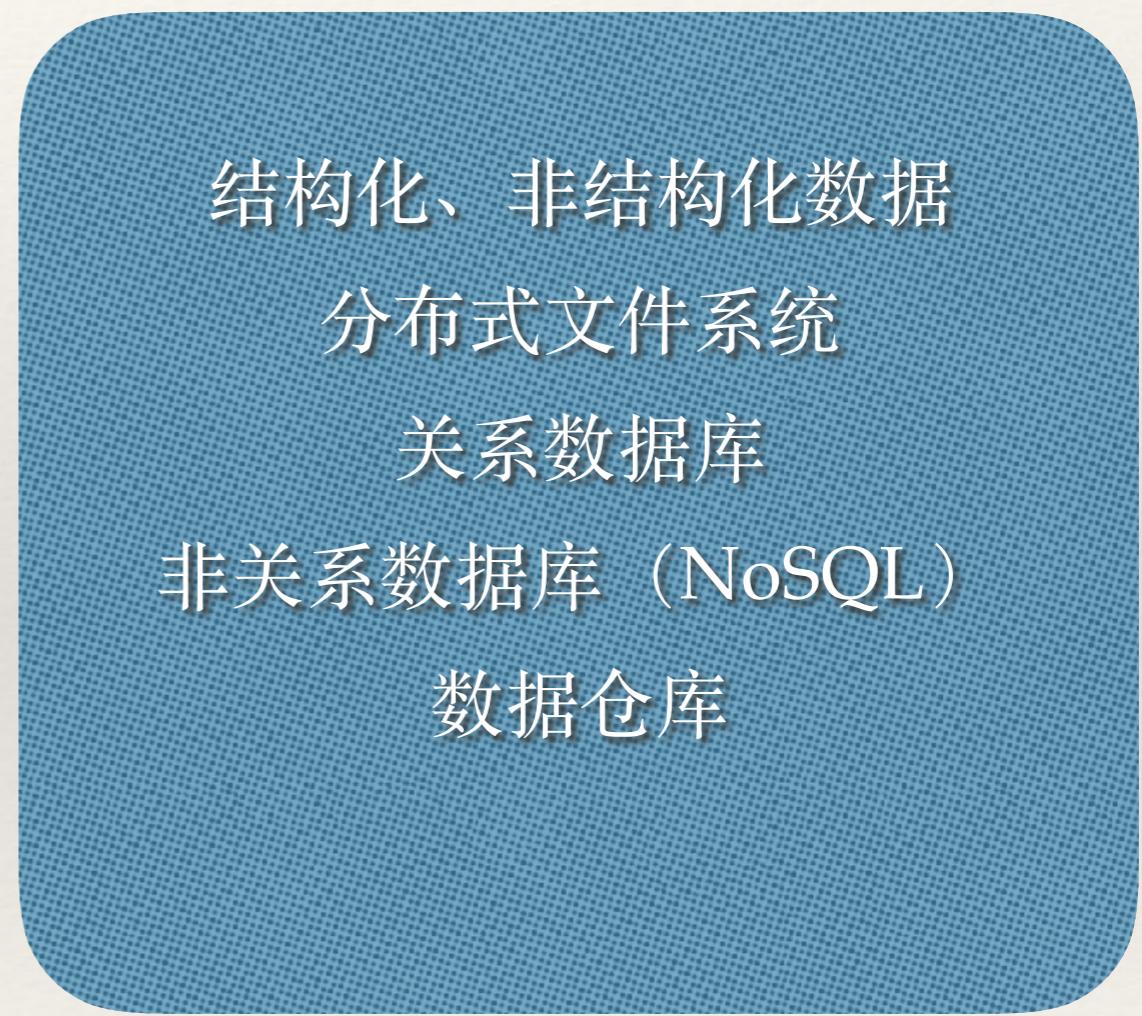
主要内容

- ❖ 知识回顾
- ❖ 大数据特点
- ❖ 大数据技术难题
- ❖ 大数据核心技术

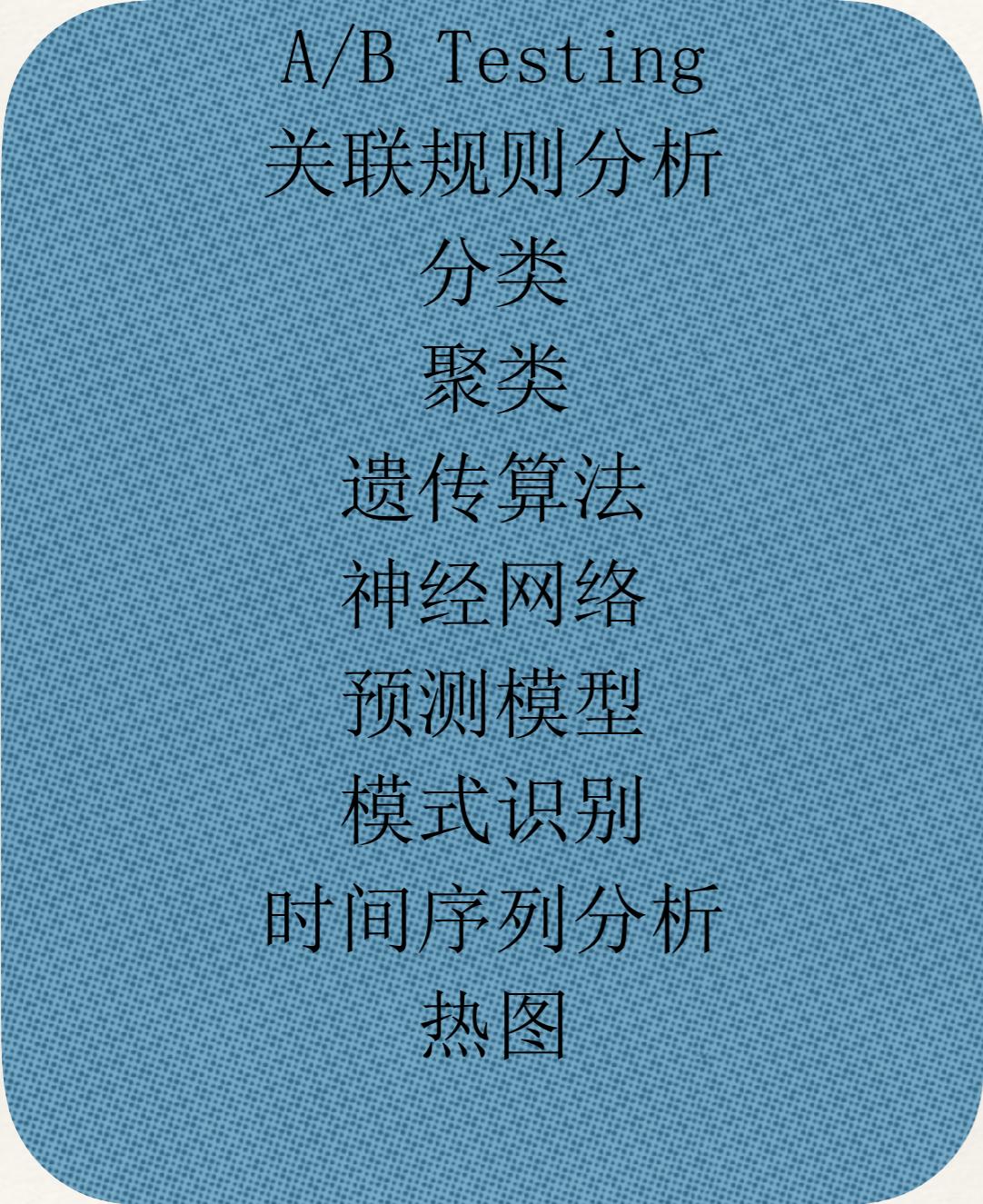
大数据的处理流程



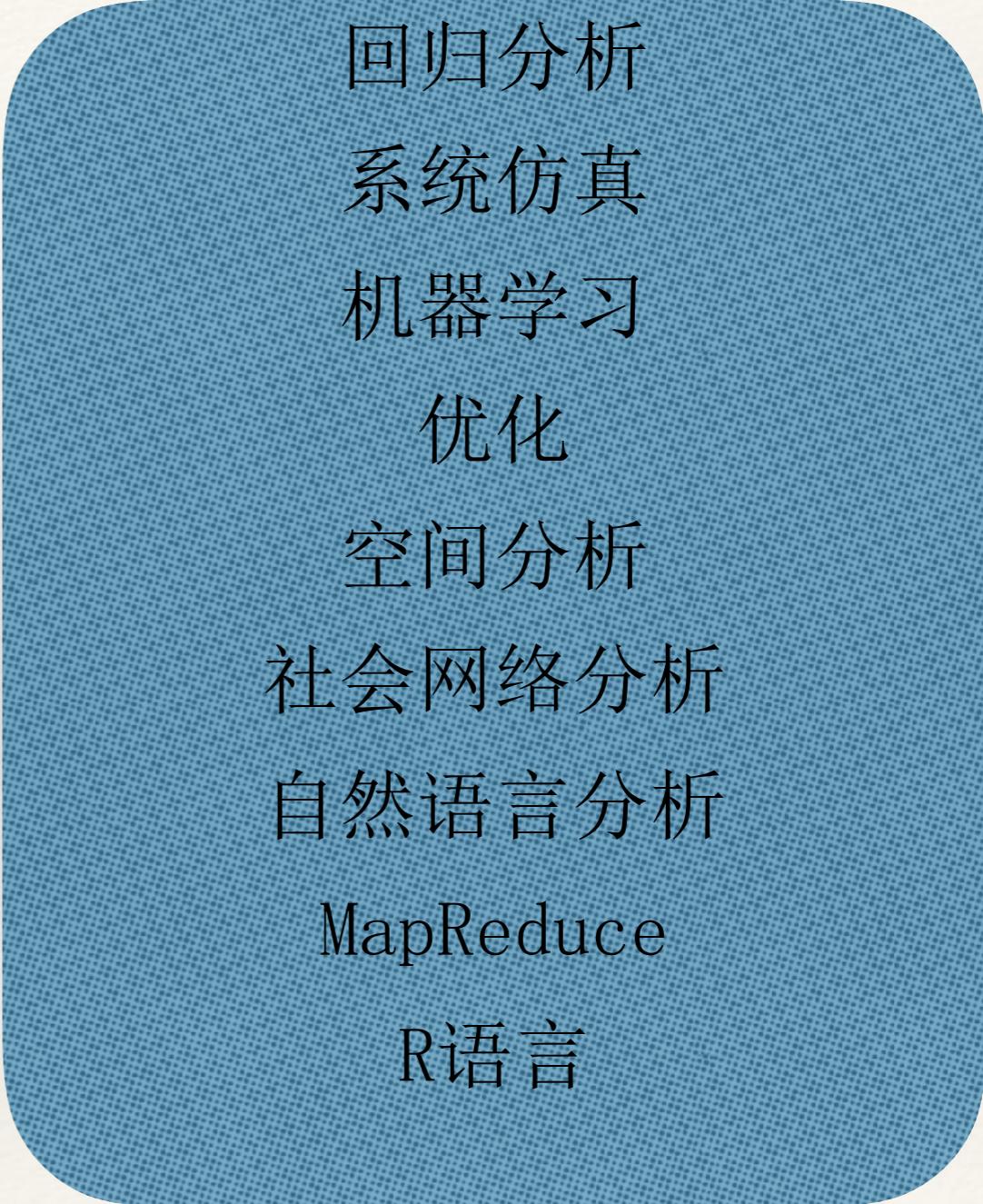
大数据的来源和采集



大数据的分析方法



- A/B Testing
- 关联规则分析
- 分类
- 聚类
- 遗传算法
- 神经网络
- 预测模型
- 模式识别
- 时间序列分析
- 热图



- 回归分析
- 系统仿真
- 机器学习
- 优化
- 空间分析
- 社会网络分析
- 自然语言分析
- MapReduce
- R语言

大数据核心技术

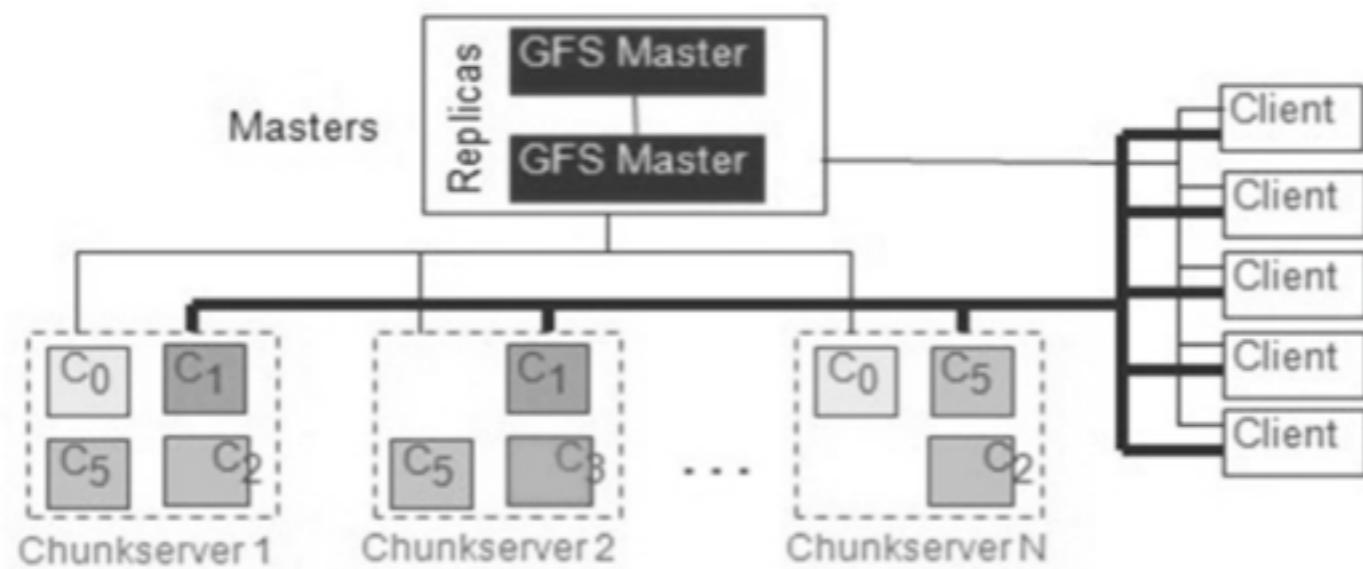
- ❖ 分布式文件系统
- ❖ 非关系型数据库
- ❖ 并行处理和分布式处理
- ❖ Hadoop生态圈
- ❖ 实时计算框架

分布式文件系统



GFS将整个系统分为三类角色：Client（客户端）、Master（主服务器）、Chunk Server（数据块服务器）。

GFS Architecture



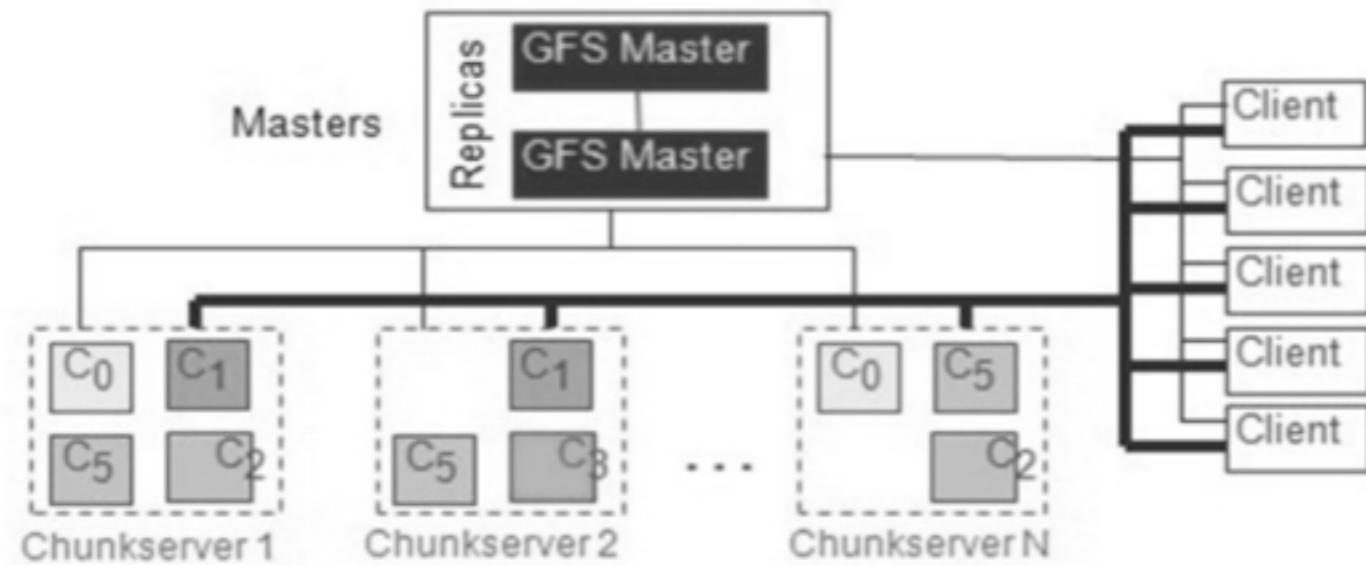
分布式文件系统



GFS将整个系统分为三类角色：Client（客户端）、Master（主服务器）、Chunk Server（数据块服务器）。

GFS Architecture

Heartbeat



大数据核心技术

- ❖ 分布式文件系统
- ❖ 非关系型数据库
- ❖ 并行处理和分布式处理
- ❖ Hadoop生态圈
- ❖ 实时计算框架

NoSQL数据库

NoSQL = Not Only SQL



- 关系型数据库的局限性
 - 难以满足高并发读写的需求
 - 难以满足对海量数据高效率存储和访问的需求
 - 难以满足对数据库高可扩展性和高可用性的需求

NoSQL数据库



- NoSQL数据存储不需要固定的表结构，通常也不存在连接操作。在大数据存取上具备关系型数据库无法比拟的性能优势。

关系型数据库中的表都是存储一些格式化的数据结构，每个元组字段的组成都一样，即使不是每个元组都需要所有的字段，但数据库会为每个元组分配所有的字段。

非关系型数据库以键值对存储，它的结构不固定，每一个元组可以有不一样的字段，每个元组可以根据需要增加一些自己的键值对，这样就不会局限于固定的结构，可以减少一些时间和空间的开销。

NoSQL数据库

- ❖ 键值(Key-Value)存储数据库
- ❖ 列存储数据库
- ❖ 文档型数据库
- ❖ 图形(Graph)数据库

NoSQL 四大家族

NoSQL 数据库类型	代表性产品	性能	扩展性	灵活性	复杂性	优点	缺点
键/值数据库	Redis Riak	高	高	高	无	查询效率高	不能存储结构化信息
列式数据库	HBase Cassandra	高	高	一般	低	查询效率高	功能较少
文档数据库	CouchDB MongoDB	高	可变的	高	低	数据结构灵活	查询效率较低
图形数据库	Neo4J OrientDB	可变的	可变的	高	高	支持复杂的图算法	只支持一定的数据规模

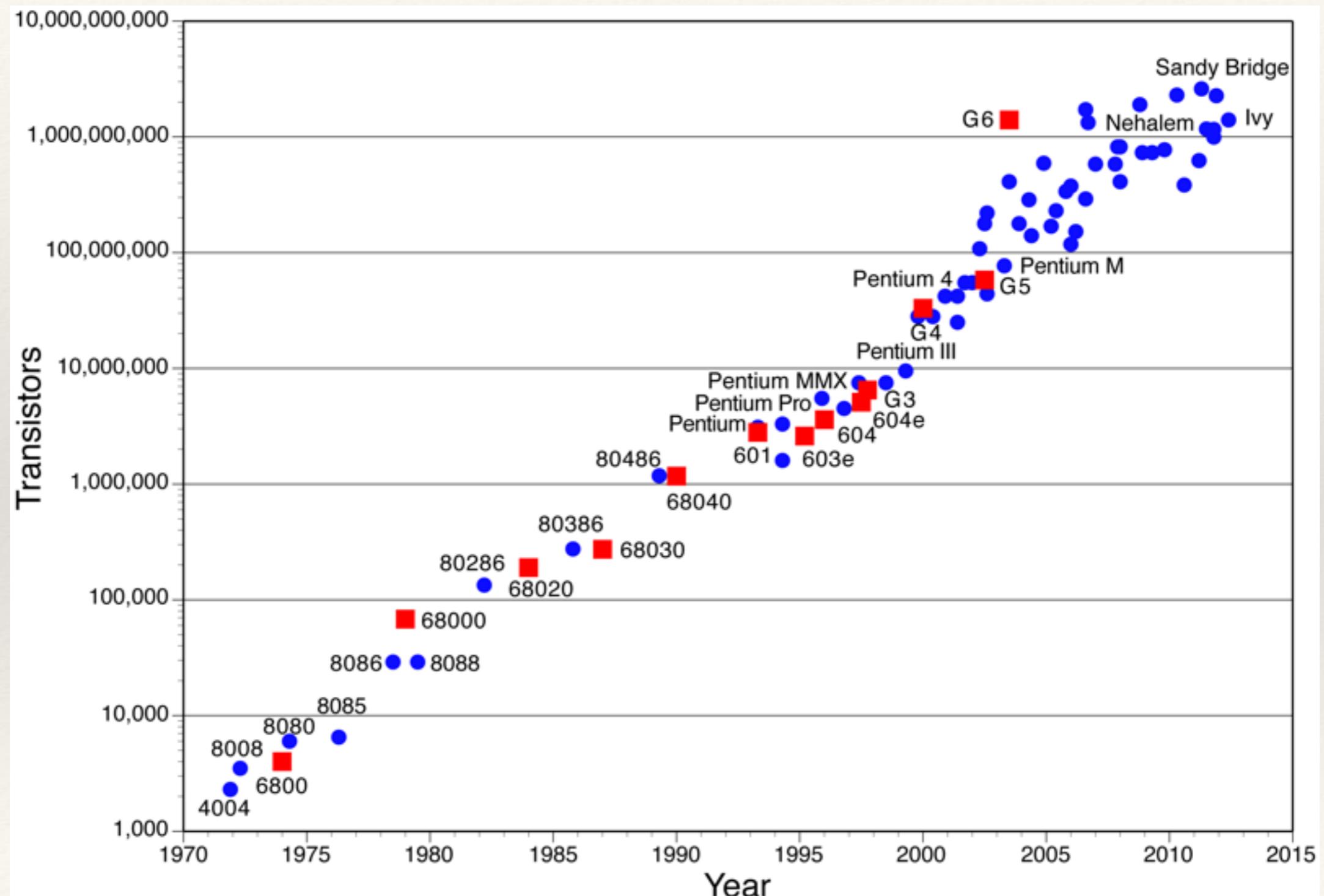
大数据核心技术

- ❖ 分布式文件系统
- ❖ 非关系型数据库
- ❖ 并行处理和分布式处理
- ❖ Hadoop生态圈
- ❖ 实时计算框架

并行处理和分布式处理

- ❖ 并行处理：同时使用多种计算资源解决计算问题的过程，主要可以通过多线程和多进程来实现。
- ❖ 核心问题：资源的分配和竞争的处理机制

并行处理和分布式处理



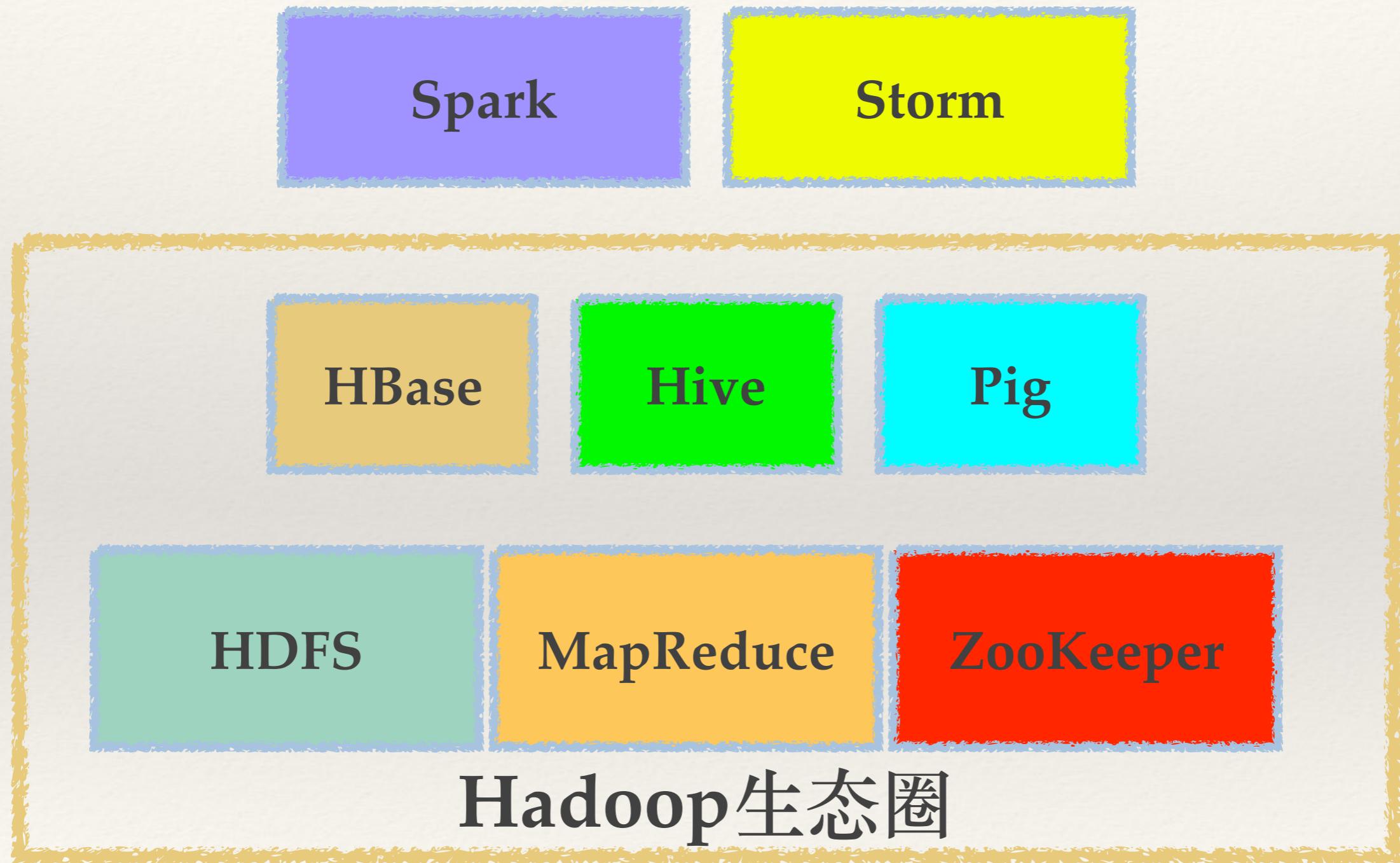
分布式计算

- ❖ 将一个需要非常大计算能力才能完成的问题，分成多个小任务最后将这些计算结果综合起来得到最终的结果。
- ❖ 核心问题：计算机之间的通信需要额外开销，可能会影响最终效率。

大数据核心技术

- ❖ 分布式文件系统
- ❖ 非关系型数据库
- ❖ 并行处理和分布式处理
- ❖ Hadoop生态圈
- ❖ 实时计算框架

大数据核心技术



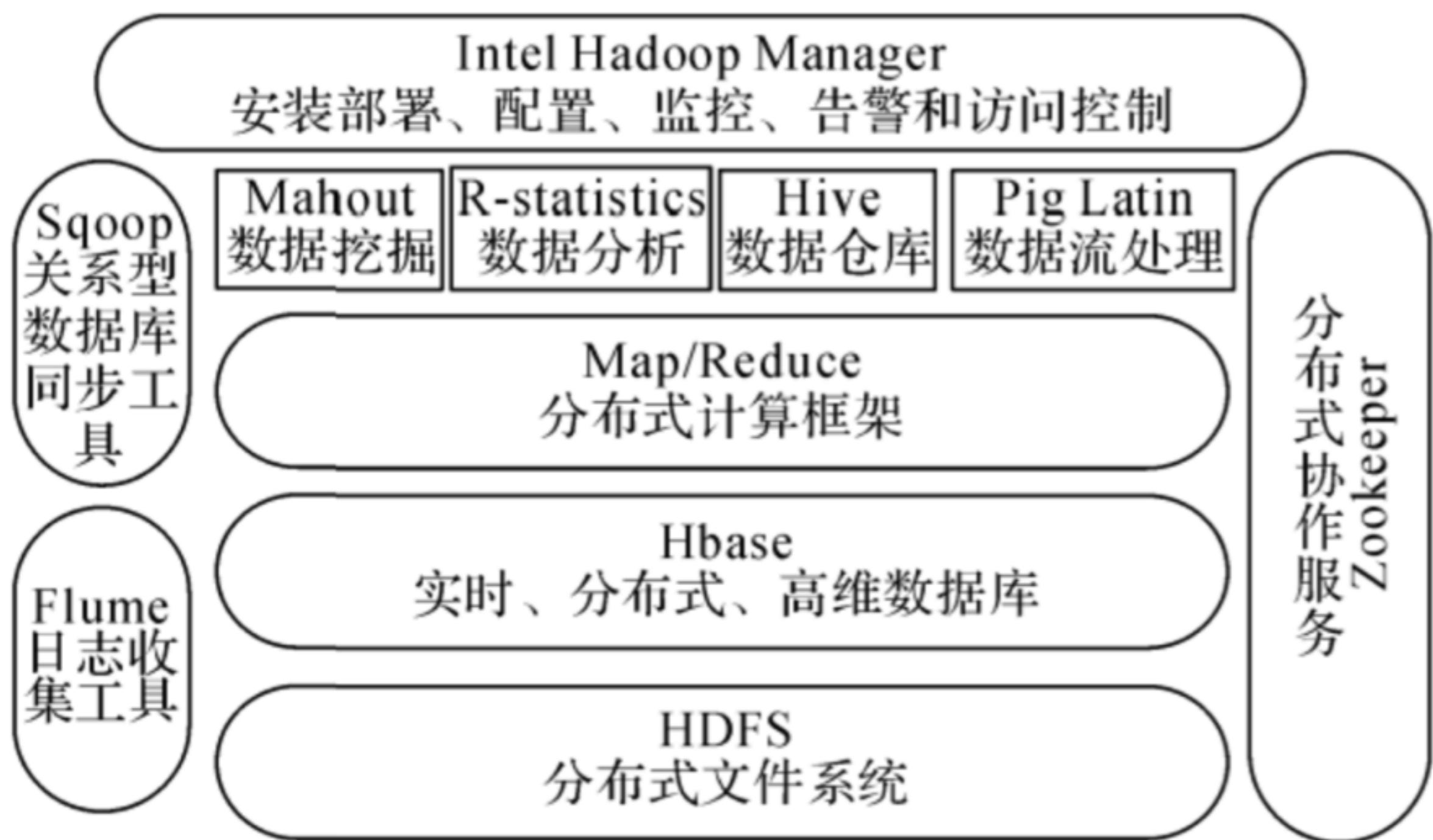
谁在使用Hadoop



为什么使用Hadoop

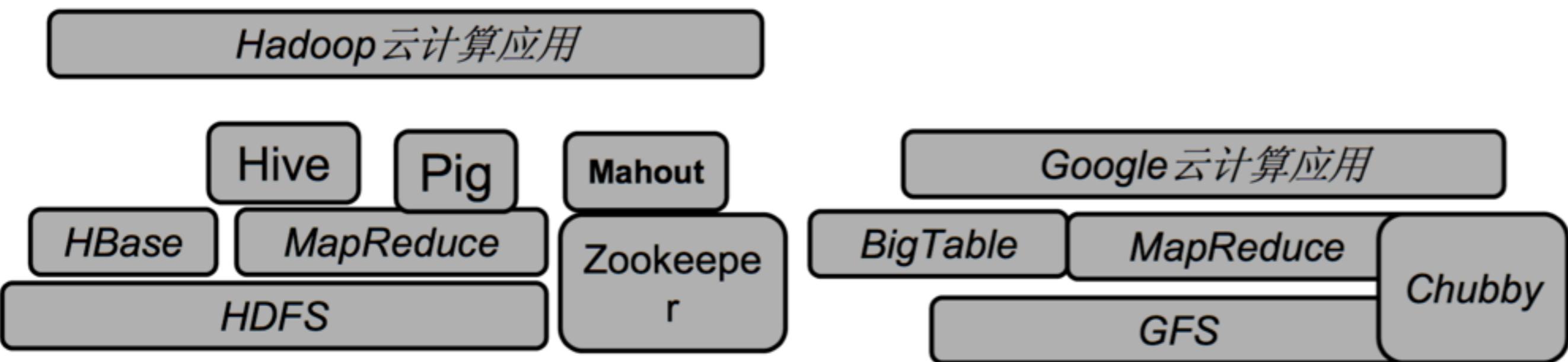
- ❖ 可扩展能力：能够方便地进行扩容，处理PB级
- ❖ 成本低：普通电脑可以
- ❖ 高效率：Hadoop可以在分布式的节点上快速处理
- ❖ 高可靠性：自动将数据复制多份，分布存储在不同的电脑上面

Hadoop技术框架



Hadoop和Google架构比较

- ❖ 并行计算模型: MapReduce->MapReduce
- ❖ 分布式文件系统: HDFS->GFS
- ❖ 数据结构化管理组件: Hbase->BigTable
- ❖ 分布式锁服务: Zookeeper->Chubby

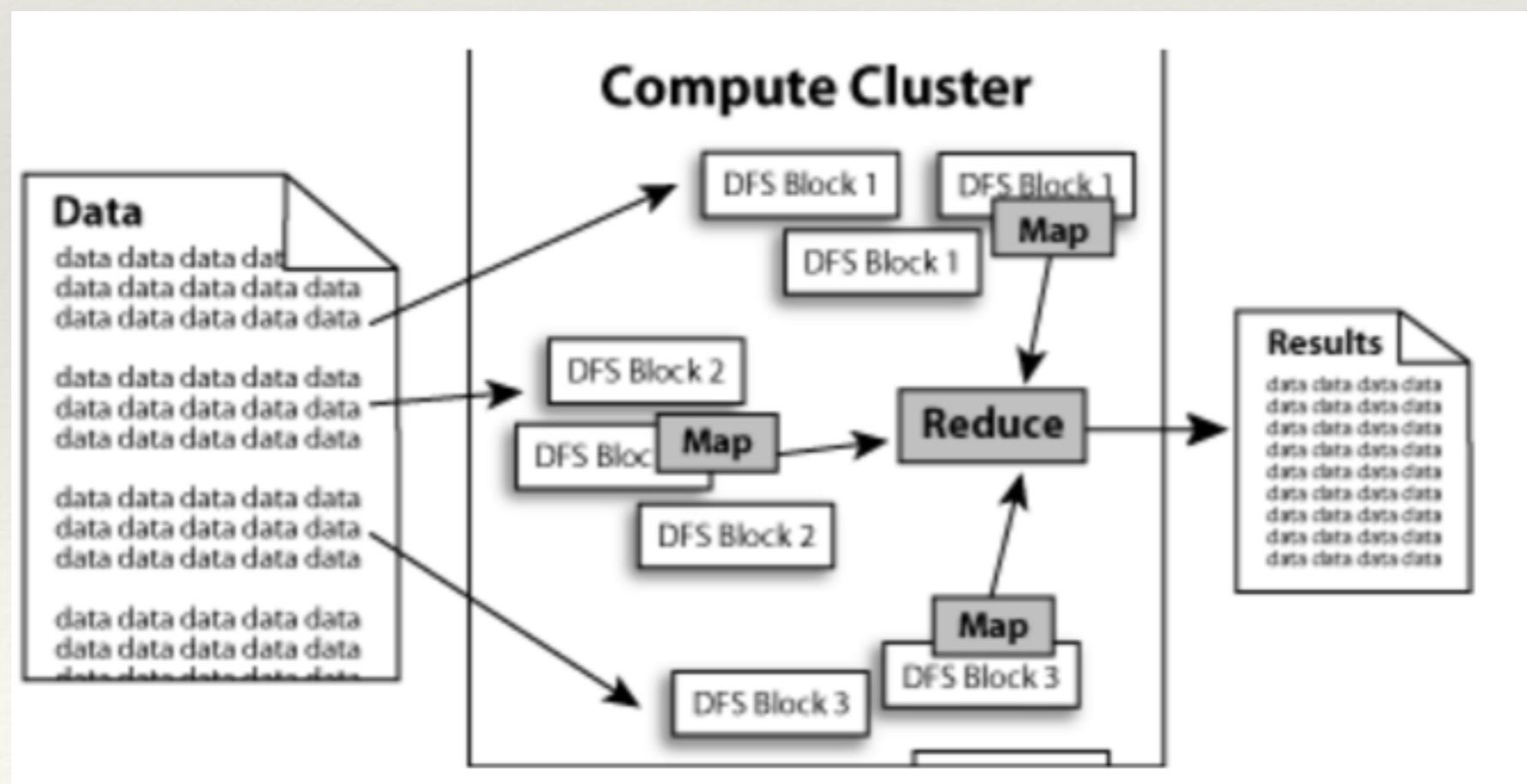


Hadoop核心组件

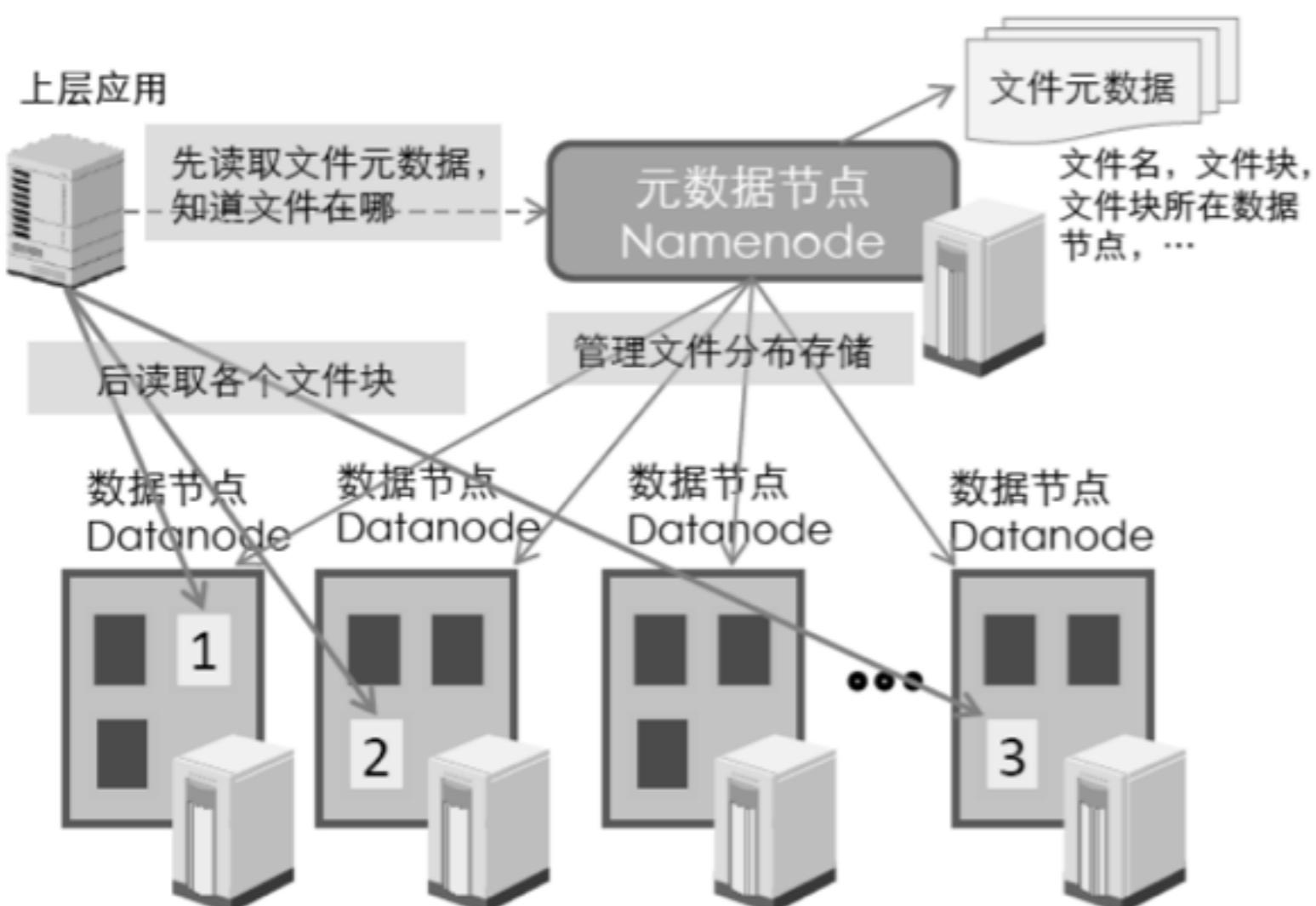
- ❖ HDFS: Hadoop Distributed File System
- ❖ MapReduce
- ❖ ZooKeeper
- ❖ HBase

HDFS

- ❖ HDFS为了做到可靠性(reliability)创建了多份数据块 (data blocks)的复制(replicas),并将它们放置在服务器群的计算节点中(compute nodes),MapReduce就可以在它们所在的节点上处理这些数据了。



HDFS

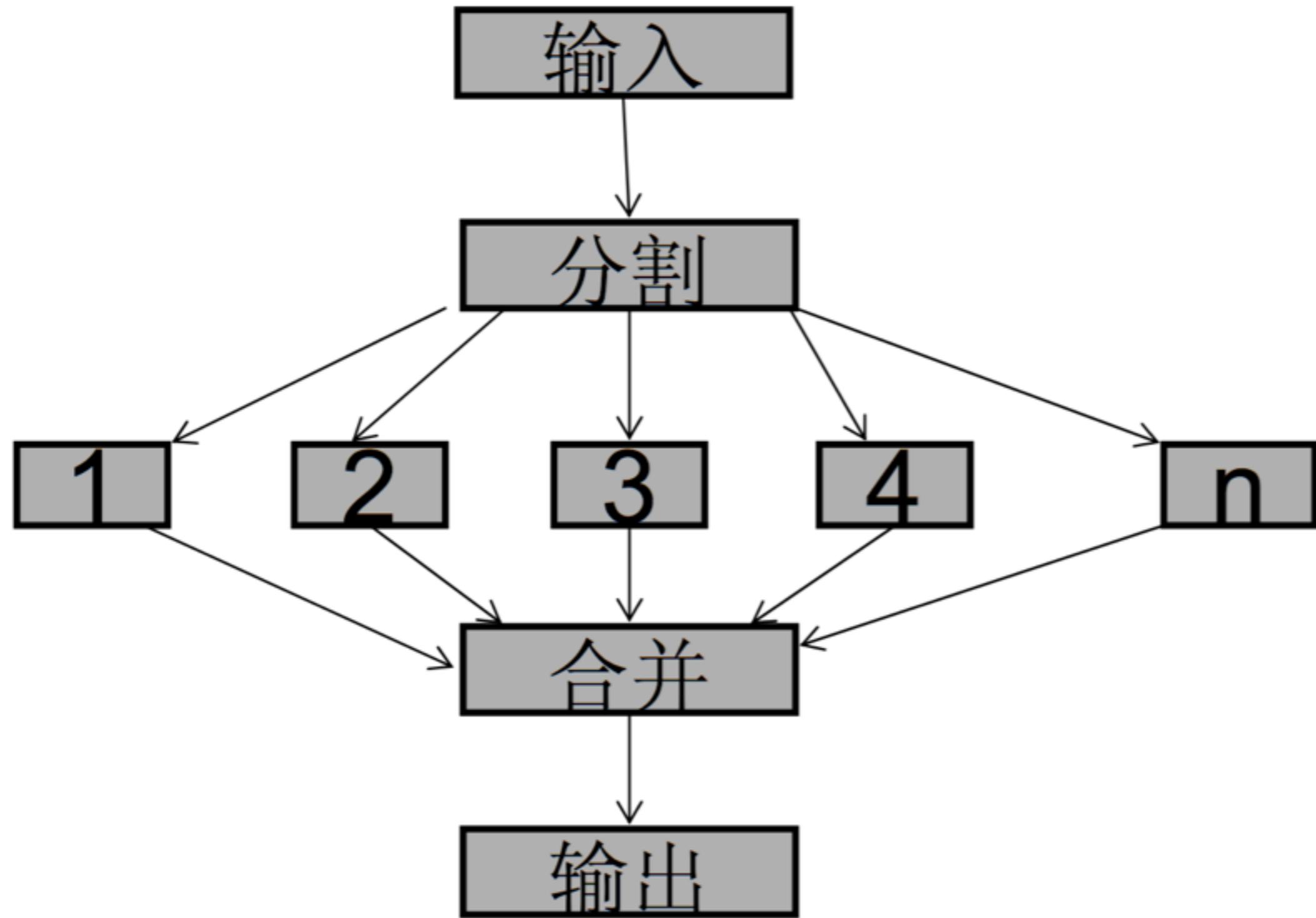


行业应用

- ✓ Yahoo Amazon Facebook Ebay 淘宝 百度
- ✓ 中国移动飞信 中国移动大云

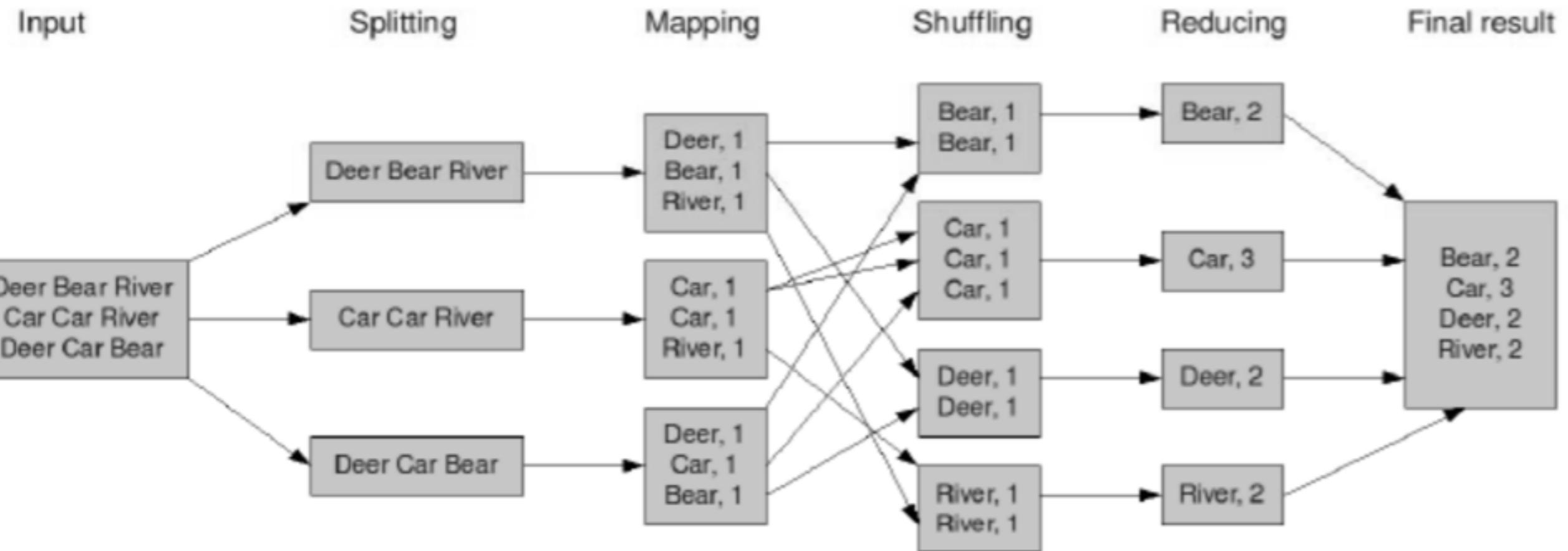
- 采用主从架构，由一个 **Namenode** 和若干个 **Datanode** 组成
- **Namenode**: 负责管理名字空间与客户端访问
- **Datanode**: 管理附带的存储，存储文件的 **block**
- 一个文件分成多个 **block**, **Block** 是 HDFS 最小存储与分配单位, 分布存储, 典型块大小为 64MB 或 128MB
- 一个 **block** 被复制存放于多个 **datanode**

MapReduce



MapReduce 示例

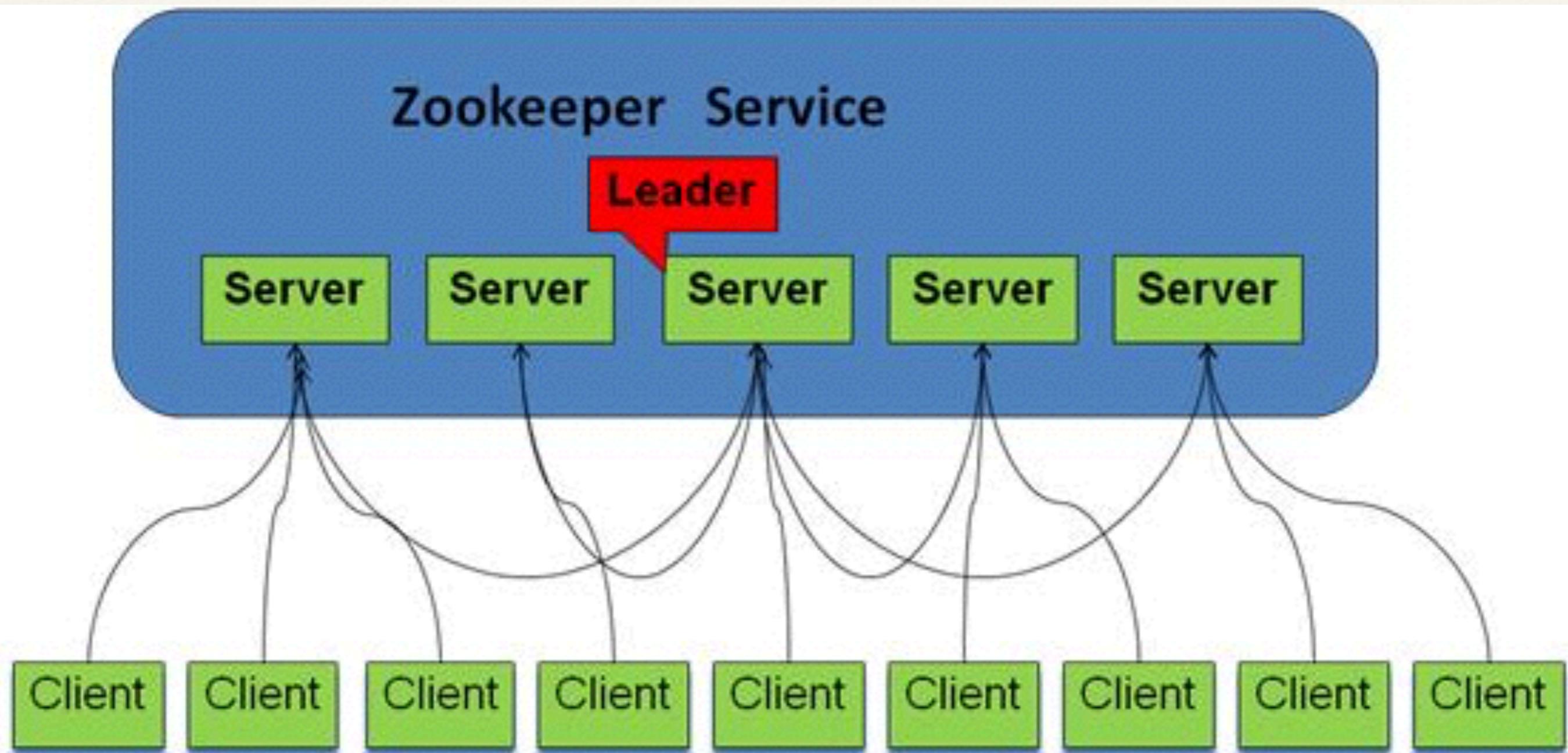
The overall MapReduce word count process



```
public class MaxTemperatureMapper  
extends Mapper<LongWritable, Text, Text, IntWritable> {  
  
private static final int MISSING = 9999;  
  
@Override  
  
public void map(LongWritable key, Text value, Context context)  
throws IOException, InterruptedException {  
  
String line = value.toString();  
  
String year = line.substring(15, 19);  
  
int airTemperature;  
  
if (line.charAt(87) == '+') { // parseInt doesn't like leading plus signs  
airTemperature = Integer.parseInt(line.substring(88, 92));  
}  
else {  
airTemperature = Integer.parseInt(line.substring(87, 92));  
}  
  
String quality = line.substring(92, 93);  
  
if (airTemperature != MISSING && quality.matches("[01459]")) {  
  
context.write(new Text(year), new IntWritable(airTemperature));  
}  
}  
}  
}
```

```
public class MaxTemperatureReducer  
extends Reducer<Text, IntWritable, Text, IntWritable> {  
  
    @Override  
    public void reduce(Text key, Iterable<IntWritable> values,  
                      Context context)  
        throws IOException, InterruptedException {  
        int maxValue = Integer.MIN_VALUE;  
        for (IntWritable value : values) {  
            maxValue = Math.max(maxValue, value.get());  
        }  
        context.write(key, new IntWritable(maxValue));  
    }  
}
```

ZooKeeper



作业1

- ❖ 比较Google和Hadoop大数据框架三个核心组件
- ❖ 解释Spark和Storm的特点与区别
- ❖ 作业： 姓名+作业1.pdf
- ❖ 邮箱： yaokai@cufe.edu.cn