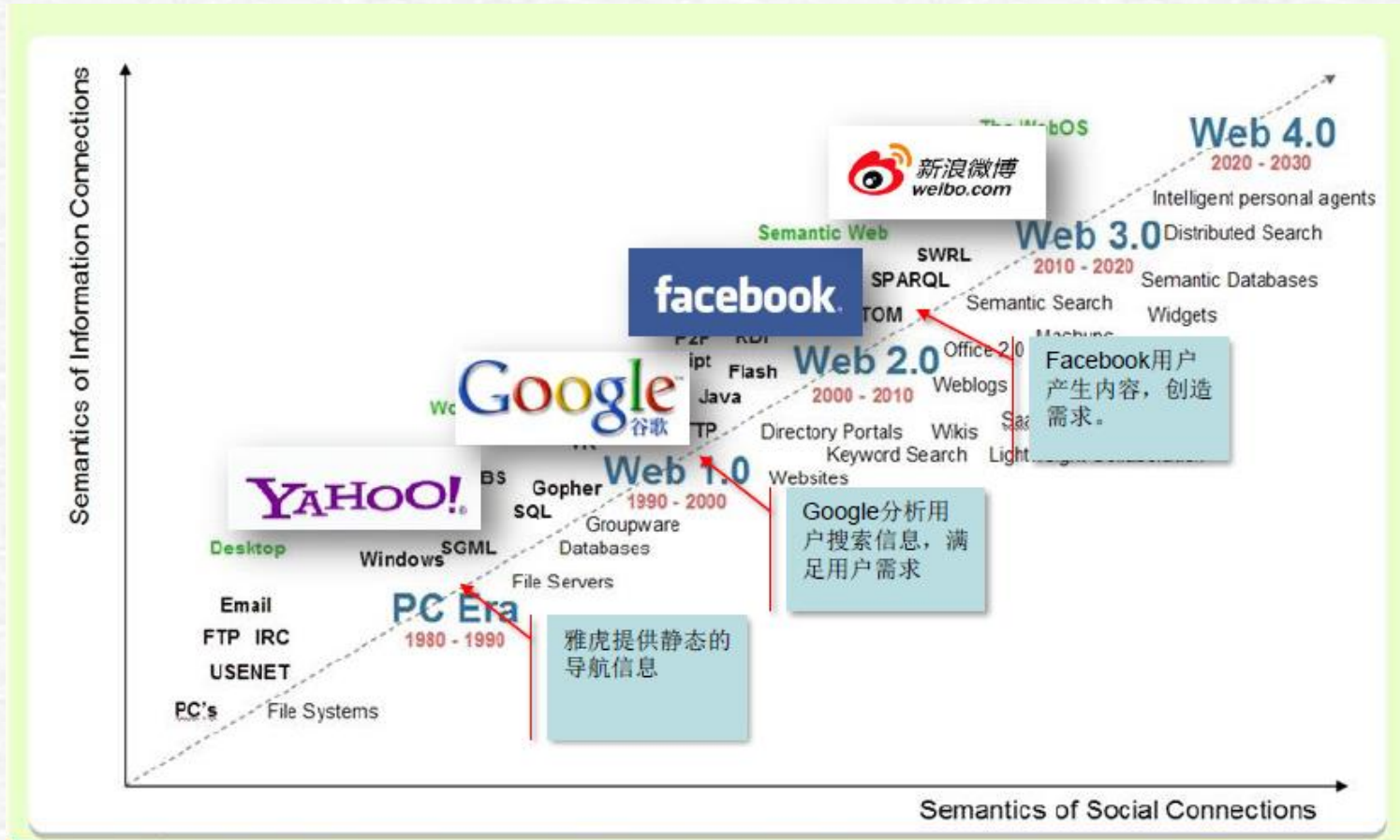




# NoSQL

单 位：重庆大学计算机学院

# 思考



- 随着互联网的发展，大数据时代的到来，互联网更加智能，数据更加复杂，如复杂类型数据（文本、图像、视频）如何存储在数据库中？

# 主要学习目标

- NoSQL数据库



# 思考问题

- Python的字典中key、value数据怎么存储？



# 一 NoSQL数据库

- 复杂数据处理
- Web2.0和移动互联网时代，高并发读写、海量数据处理、数据结构不一致等情况
- 为了解决大规模数据集合和多种数据类型带来的挑战，NoSQL数据库应运而生。
- NoSQL，也称作“Not Only SQL”，“non-relational”。泛指区别于传统的关系型数据库的非关系型的数据库。
- 区别于关系型数据库，不保证关系数据的四大特性：原子性、一致性、隔离性、持久性。

# 一 NoSQL数据库

- 关系型数据库的ACID规则
  - A (Atomicity) 原子性
  - C (Consistency) 一致性
  - I (Isolation) 独立性
  - D (Durability) 持久性

## 二 NoSQL数据库的特点

- 不需要预定义模式：不需要事先定义数据模式，预定义表结构。数据中的每条记录都可能有不同的属性和格式。当插入数据时，并不需要预先定义它们的模式。
- 无共享架构：相对于将所有数据存储在存储区域网络中的全共享架构。NoSQL往往将数据划分后存储在各个本地服务器上。因为从本地磁盘读取数据的性能往往好于通过网络传输读取数据的性能，从而提高了系统的性能。

## 二 NoSQL数据库的特点

- 弹性可扩展：可以在系统运行的时候，动态增加或者删除结点。不需要停机维护，数据可以自动迁移。
- 分区：相对于将数据存放于同一个节点，NoSQL数据库需要将数据进行分区，将记录分散在多个节点上面。并且通常分区的同时还要做复制。这样既提高了并行性能，又能保证没有单点失效的问题。



## 二 NoSQL数据库的特点

- 异步复制：和RAID存储系统不同的是，NoSQL中的复制，往往是基于日志的异步复制。这样，数据就可以尽快地写入一个节点，而不会被网络传输引起迟延。缺点是并不总是能保证一致性，这样的方式在出现故障的时候，可能会丢失少量的数据。
- BASE：相对于事务严格的ACID特性，NoSQL数据库保证的是BASE特性。BASE是最终一致性和软事务。
- 基本可用(Basically Available)、软状态/柔性事务(Soft state)、最终一致性(Eventual consistency)

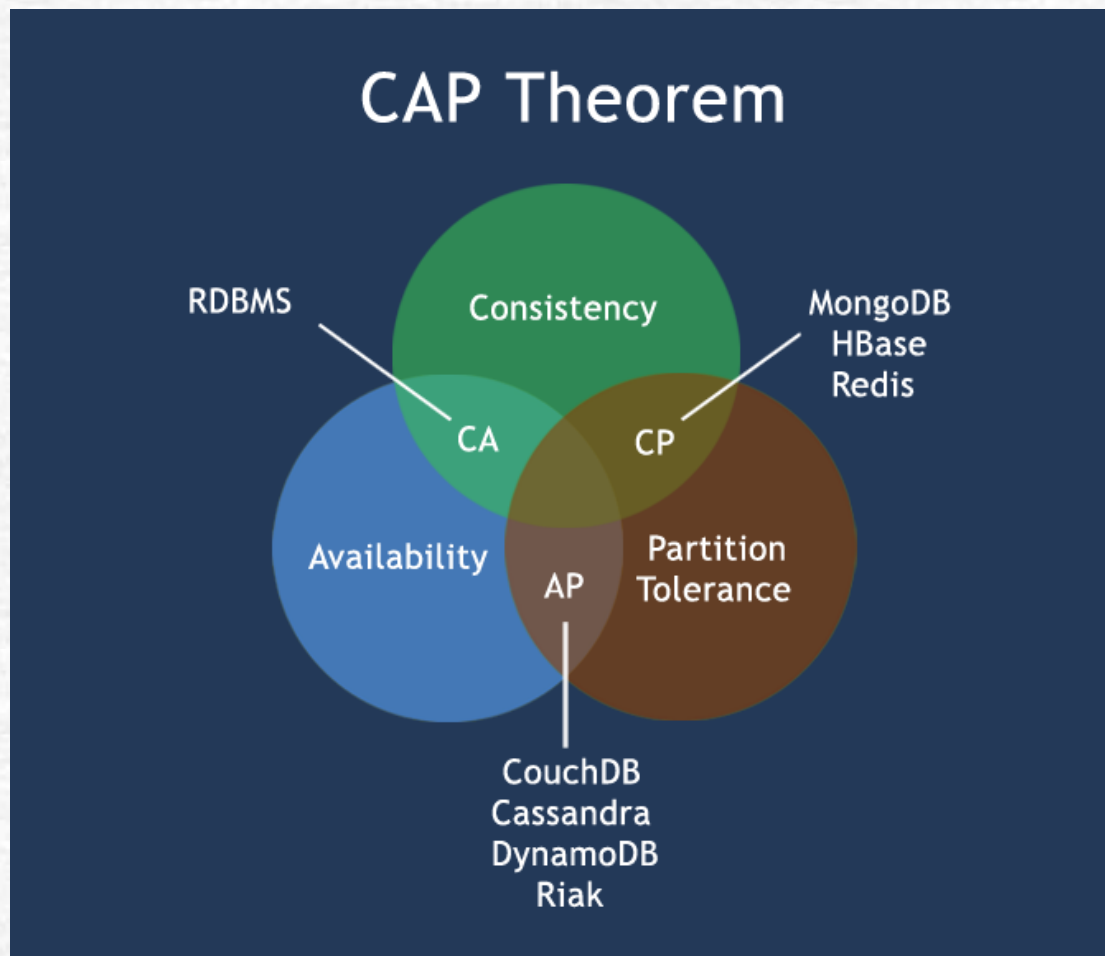
# CAP定理 (CAP theorem)

- 在计算机科学中，CAP定理 (CAP theorem)，又被称作 布鲁尔定理 (Brewer's theorem)，它指出对于一个分布式计算系统来说，不可能同时满足以下三点：
- 一致性 (Consistency) (所有节点在同一时间具有相同的数据)
- 可用性 (Availability) (保证每个请求不管成功或者失败都有响应)
- 分隔容忍 (Partition tolerance) (系统中任意信息的丢失或失败不会影响系统的继续运作)

# CAP定理 (CAP theorem)

- CAP理论的核心是：一个分布式系统不可能同时很好的满足一致性，可用性和分区容错性这三个需求，最多只能同时较好的满足两个。
- 因此，根据 CAP 原理将 NoSQL 数据库分成了满足 CA 原则、满足 CP 原则和满足 AP 原则三 大类：
- CA - 单点集群，满足一致性，可用性的系统，通常在可扩展性上不太强大。
- CP - 满足一致性，分区容忍性的系统，通常性能不是特别高。
- AP - 满足可用性，分区容忍性的系统，通常可能对一致性要求低一些。

# CAP定理 (CAP theorem)



**BASE:** Basically Available, Soft-state, Eventually Consistent。由 Eric Brewer 定义。

CAP理论的核心是：一个分布式系统不可能同时很好的满足一致性，可用性和分区容错性这三个需求，最多只能同时较好的满足两个。

BASE是NoSQL数据库通常对可用性 & 一致性的弱要求原则：

Basically Available — 基本可用  
Soft-state — 软状态/柔性事务。  
"Soft state" 可以理解为"无连接"的，而 "Hard state" 是"面向连接"的

Eventual Consistency — 最终一致性，也是 ACID 的最终目的。



# ACID vs BASE

## ACID vs BASE

ACID	BASE
原子性(Atomicity)	基本可用(Basically Available)
一致性(Consistency)	软状态/柔性事务(Soft state)
隔离性(Isolation)	最终一致性 (Eventual consistency)
持久性 (Durable)	

## 三 NoSQL数据库的分类

### 1 列存储

- HBase
- Cassandra
- Hypertable
- 顾名思义，是按列存储数据的。最大的特点是方便存储结构化和半结构化数据，方便做数据压缩，对针对某一行或者某几行的查询有非常大的IO优势。

### 2 文档存储

- MongoDB
- CouchDB
- 文档存储一般用类似json的格式存储，存储的内容是文档型的。这样也就有机会对某些字段建立索引，实现关系数据库的某些功能。

## 三 NoSQL数据库的分类

### 3 key-value存储

- Tokyo Cabinet / Tyrant
- Berkeley DB
- MemcacheDB
- Redis

可以通过key快速查询到其value。一般来说，存储不管value的格式，照单全收。（Redis包含了其他功能）

### 4 图存储

- Neo4J
- FlockDB

图形关系的最佳存储。使用传统关系数据库来解决的话性能低下，而且设计使用不方便。

### 三 NoSQL数据库的分类

#### 5 对象存储

- db4o
- Versant

通过类似面向对象语言的语法操作数据库，通过对象的方式存取数据。

#### 6 xml数据库

- Berkeley DB XML
- BaseX

高效的存储XML数据，并支持XML的内部查询语法，比如XQuery, Xpath。



# NoSQL的适用场景

- 数据模型比较简单
- 需要灵活性更强的IT系统
- 对数据库性能要求较高
- 不需要高度的数据一致性
- 对于给定key，比较容易映射复杂值的环境

# Nosql数据库实例-Redis

- REmote DIctionary Server(Redis) 是一个由 Salvatore Sanfilippo 写的 key-value 存储系统，是跨平台的非关系型数据库。
- Redis 是一个开源的使用 ANSI C 语言编写、遵守 BSD 协议、支持网络、可基于内存、分布式、可选持久性的键值对(Key-Value)存储数据库，并提供多种语言的 API。
- Redis 通常被称为数据结构服务器，因为值 (value) 可以是字符串(String)、哈希(Hash)、列表(list)、集合(sets)和有序集合(sorted sets)等类型。

# Redis

- Redis 是完全开源的，遵守 BSD 协议，是一个高性能的 key-value 数据库。
- Redis 与其他 key - value 缓存产品有以下三个特点：
  - Redis支持数据的持久化，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用。
  - Redis不仅仅支持简单的key-value类型的数据，同时还提供 list, set, zset, hash等数据结构的存储。
  - Redis支持数据的备份，即master-slave模式的数据备份。

# Redis 的优势

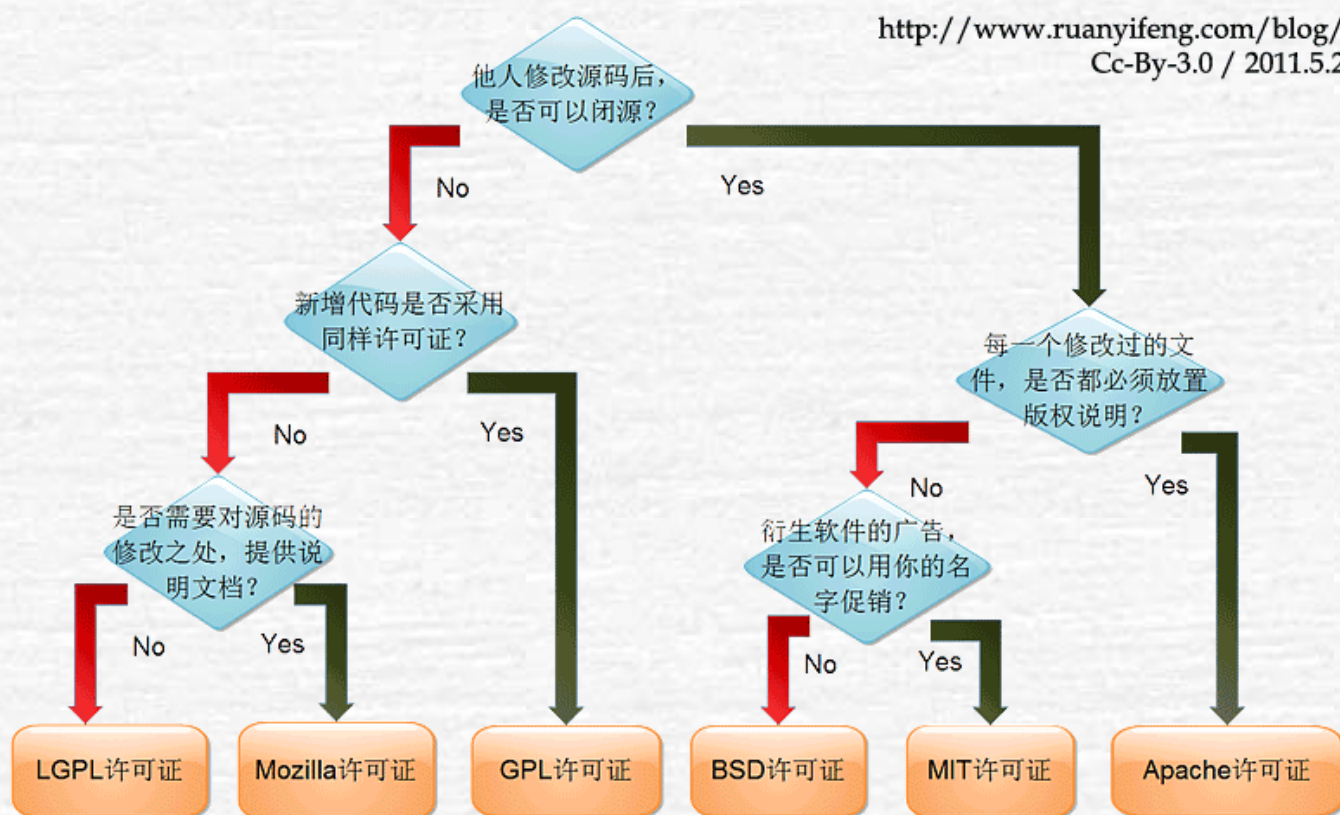
- 性能极高 - Redis能读的速度是110000次/s, 写的速度是81000次/s 。
- 丰富的数据类型 - Redis支持二进制案例的 Strings, Lists, Hashes, Sets 及 Ordered Sets 数据类型操作。
- 原子 - Redis的所有操作都是原子性的，意思就是要么成功执行要么失败完全不执行。单个操作是原子性的。多个操作也支持事务，即原子性，通过MULTI和EXEC指令包起来。
- 丰富的特性 - Redis还支持 publish/subscribe, 通知, key 过期等等特性。



# 插入知识点-开源协议

许可证	版本	包含许可证	包含源代码	链接	状态变化	商业使用	散布	修改	专利授权	私人使用	授权转售	无担保责任	没有商标
Apache许可证	2.0	是			是	是	是	是	是	是	是	是	是
3句版BSD许可证		是				是	是	是		是	是	是	是
2句版BSD许可证		是				是	是	是		是	是	是	
GNU通用公共许可证	2.0	是	是		是	是	是	是	是	是	否	是	
GNU通用公共许可证	3.0	是	是		是	是	是	是	是	是	是	是	
GNU宽通用公共许可证	2.1	是	是	是		是	是	是	是	是	是	是	
GNU宽通用公共许可证	3.0	是	是	是		是	是	是	是	是	是	是	
MIT许可证		是				是	是	是		是	是	是	
Mozilla公共许可证	2.0	是	是			是	是	是	是	是	是	是	是
Eclipse公共许可证	1.0	是	是			是	是	是	是	是	是	是	
Affero通用公共许可证		是	是		是	是	是	是		是	是	是	
一般的著作权 [note 1]		是				是	否	否		是	否		

# 插入知识点-开源协议



# MongoDB

- MongoDB 是由C++语言编写的，是一个基于分布式文件存储的开源数据库系统。
- 在高负载的情况下，添加更多的节点，可以保证服务器性能。
- MongoDB 旨在为WEB应用提供可扩展的高性能数据存储解决方案。
- MongoDB 将数据存储为一个文档，数据结构由键值(key=>value)对组成。MongoDB 文档类似于 JSON 对象。

# MongoDB

## JSON文档

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```

← field: value  
← field: value  
← field: value  
← field: value

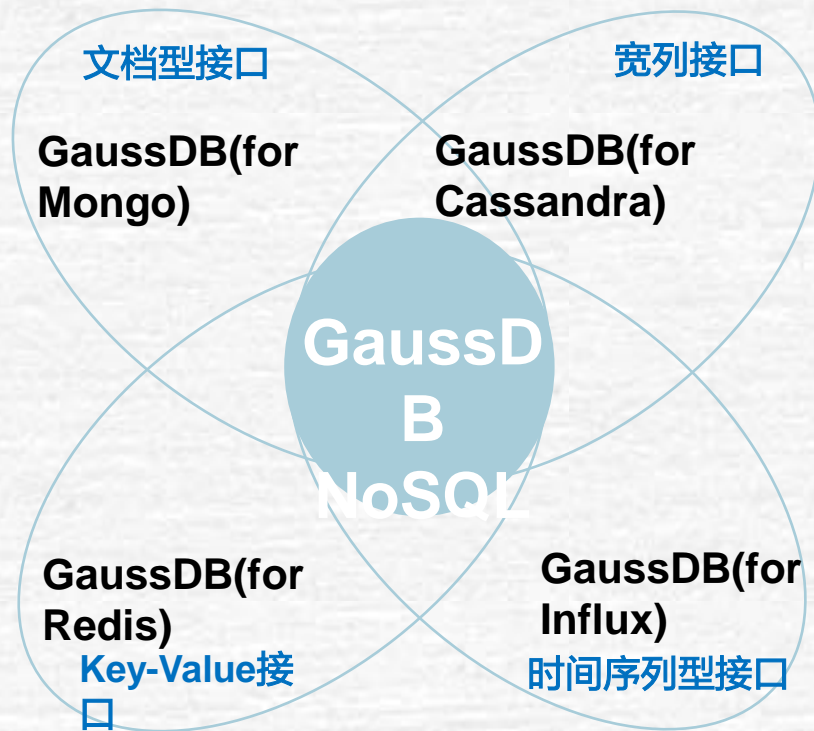


# MongoDB特点

- MongoDB 是一个面向文档存储的数据库，操作起来比较简单和容易。
- 你可以在MongoDB记录中设置任何属性的索引（如：FirstName="Sameer", Address="8 Gandhi Road"）来实现更快的排序。
- 你可以通过本地或者网络创建数据镜像，这使得MongoDB有更强的扩展性。
- 如果负载的增加（需要更多的存储空间和更强的处理能力），它可以分布在计算机网络中的其他节点上这就是所谓的分片。
- Mongo支持丰富的查询表达式。查询指令使用JSON形式的标记，可轻易查询文档中内嵌的对象及数组。
- MongoDB 使用update() 命令可以实现替换完成的文档（数据）或者一些指定的数据字段。
- MongoDB中的Map/reduce主要是用来对数据进行批量处理和聚合操作。
- Map和Reduce。Map函数调用emit(key, value)遍历集合中所有的记录，将key与value传给Reduce函数进行处理。
- Map函数和Reduce函数是使用Javascript编写的，并可以通过db.runCommand或mapreduce命令来执行MapReduce操作。
- GridFS是MongoDB中的一个内置功能，可以用于存放大量小文件。
- MongoDB允许在服务端执行脚本，可以用Javascript编写某个函数，直接在服务端执行，也可以把函数的定义存储在服务端，下次直接调用即可。
- MongoDB支持各种编程语言：RUBY，PYTHON，JAVA，C++，PHP，C#等多种语言。
- MongoDB安装简单。

# 华为Nosql

- 华为自主研发的计算存储分离架构的分布式多模NoSQL数据库服务，包括GaussDB(for Mongo)、GaussDB(for Cassandra)、GaussDB(for Redis)和GaussDB(for Influx)这四款主流NoSQL数据库服务。
- 支持跨3AZ高可用集群，相比社区版具有分钟级计算扩容、秒级存储扩容、数据强一致、超低时延、高速备份恢复的优势。
- 具有高性价比，适用于IoT、气象、互联网、游戏等领域。



# GaussDB (for Mongo) 产品特点

## 存算分离、极致可用、海量存储

- 存算分离：存储层采用DFV高性能分布式存储，计算资源与存储资源按需独立扩展；
- 极致可用：支持3~12节点分布式部署，容忍 $n-1$ 个节点故障，3副本数据存储在，保障数据安全；
- 海量存储：最大支持96 T存储容量；
- 自主可控：支持鲲鹏架构；
- 兼容性：兼容MongoDB协议，开发体验一致。



# GaussDB (for Cassandra)



兼容版本	Cassandra 3.11
备份恢复	备份：支持自动备份（默认保留7天）、手动数据备份； 恢复：支持备份还原到新实例
数据迁移	支持DynamoDB迁移到Cassandra（工具）
弹性扩容	分钟级计算资源扩容、秒级存储节点扩容
监控	节点级监控，包括CPU使用率、内存使用率、网络输入输出吞吐量、活动连接数
安全	多种安全策略保护数据库和用户隐私，例如：VPC、子网、安全组、SSL 等
计费	按需+包周期
性能	超强写入性能，数倍纯读性能提升
高可用	支持3AZ与单AZ
节点规格	4U16G\8U32G\16U64G\32U128G
节点数量	3~200



**集群稳定**  
无Full GC问题



**计算存储分离**  
分钟级节点扩容  
秒级存储扩容



**Active-Active**  
分布式架构  
N-1个节点故障容忍



**高性能**  
性能数倍于社区版



**海量数据**  
单套实例最大100TB数据



**高可靠**  
分钟级备份恢复  
数据强一致性





# 随堂小测试

- 什么是NoSQL?
- NoSQL数据库相比于关系型数据库有什么特点?

# 课堂小结和作业安排

- 基本知识：
  - NoSQL定义
  - NoSQL特点
  - NoSQL应用场景
- 延展性学习：
  - gaussDB（for Mongo）
- 作业  
无。