

厦门大学计算机科学系研究生课程

《大数据技术基础》

第10章 NoSQL数据库 (2013年新版)

林子雨

厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn ▶▶

主页: <http://www.cs.xmu.edu.cn/linziyu>

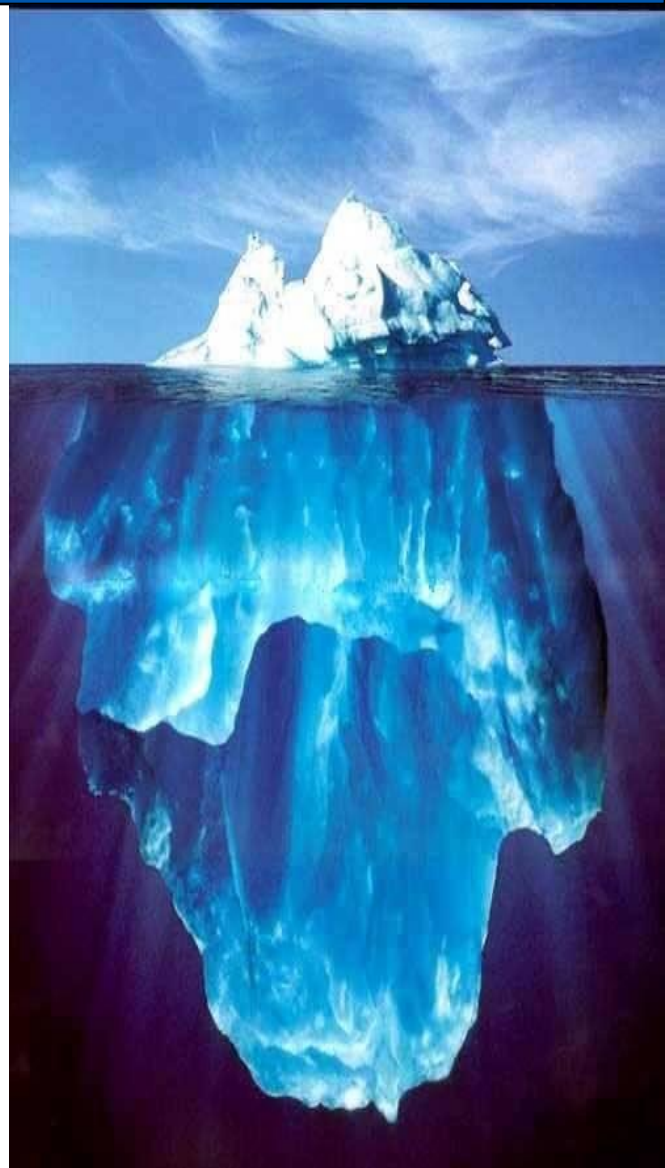




提纲

- NoSQL数据库概念
- 为什么要使用NoSQL数据库
- NoSQL特点和挑战
- 对NoSQL的质疑
- CAP理论
- 最终一致性
- NoSQL数据库开源软件
- 参考文献
- 附件

本讲义PPT存在配套教材，由林子雨通过大量阅读、收集、整理各种资料后编写而成
下载配套教材请访问《大数据技术基础》2013
班级网站：<http://dblab.xmu.edu.cn/node/423>





NoSQL数据库概念

NoSQL概念

•NoSQL：意即反SQL运动，是一项全新的数据库革命性运动。NoSQL数据库，指的是非关系型的数据库。NoSQL项目的名字上看不出什么相同之处，但是，它们通常在某些方面相同：它们可以处理超大量的数据。

在NoSQL运动的最新一次聚会中，来自世界各地的150人挤满了CBS Interactive的一间会议室。分享他们如何推翻缓慢而昂贵的关系数据库的暴政，怎样使用更有效和更便宜的方法来管理数据。

关系数据库的缺陷

- 关系数据库并不适合所有的数据模型
- 关系数据库扩展难度大
- 标准化通常会伤害到性能



NoSQL的目标

- 反标准化，通常是无模式的，文档型存储
- 以key/value为基础，支持通过key进行查找
- 水平扩展
- 内置复制
- HTTP/REST或很容易编程的API
- 支持MapReduce风格的编程
- 最终一致性



为什么要使用NoSQL数据库



Web2.0网站的“三高”需求

- 对数据库高并发读写的性能需求
- 对海量数据的高效率存储和访问的需求
- 对数据库的高可扩展性和高可用性的需求

RDBMS在Web2.0无用武之地

- 数据库事务一致性需求
- 数据库写实时性和读实时性需求
- 对复杂的SQL查询，特别是多表关联查询的需求



NoSQL特点和挑战

NoSQL特点

- 灵活的可扩展性
- 大数据
- 降低管理难度
- 经济
- 灵活的数据模型

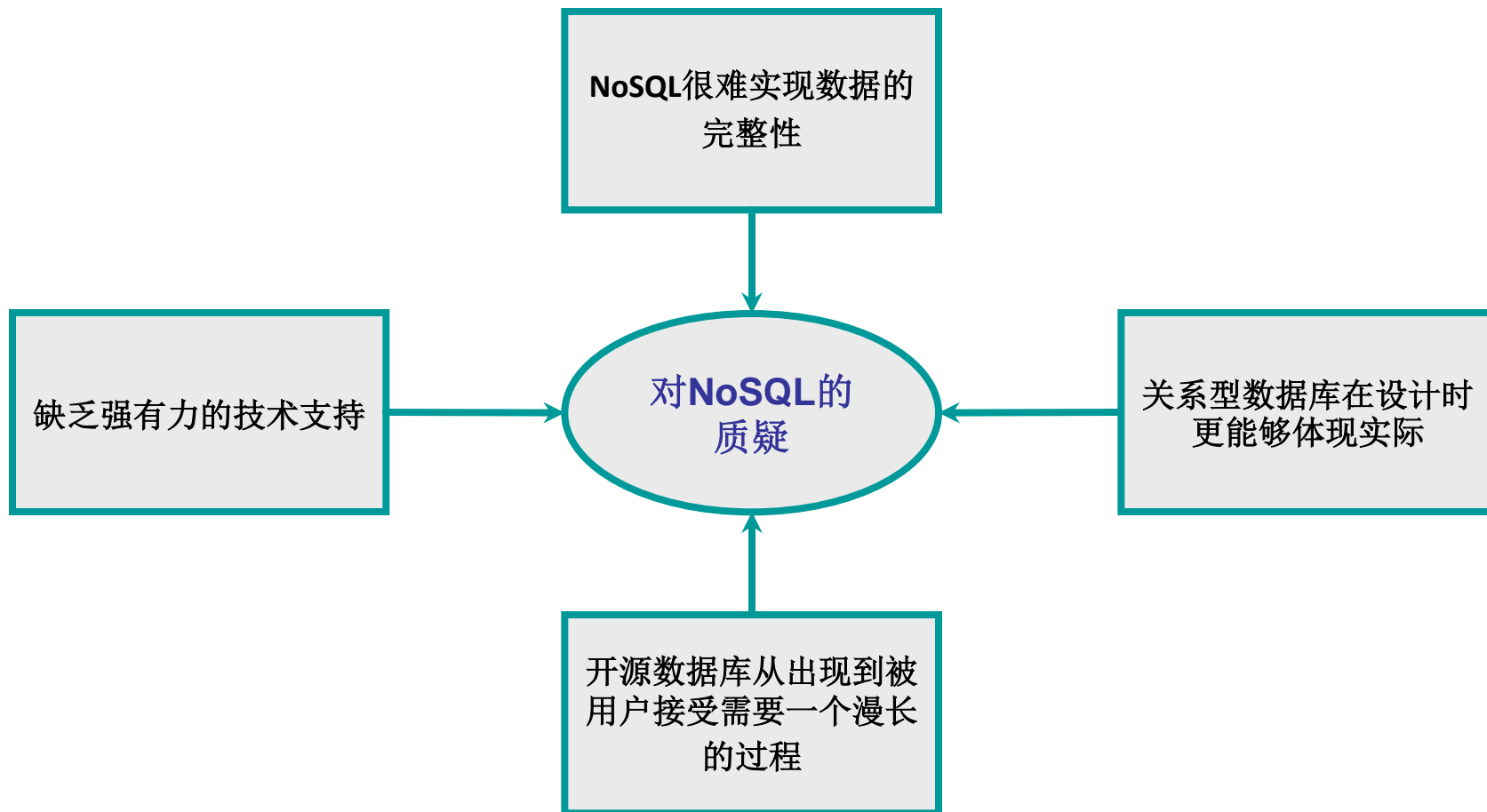
NoSQL

NoSQL五大挑战

- 成熟度
- 支持
- 分析和商业智能化
- 管理
- 专业知识

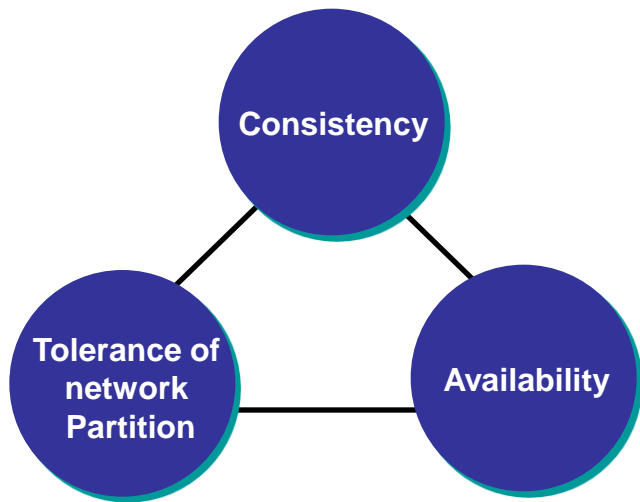


对NoSQL的质疑





CAP理论



CAP理论的启示

- CAP理论告诉我们，一个分布式系统不可能满足一致性，可用性和分区容错性这三个需求，最多只能同时满足两个。
- 在一个系统中，可以对某些数据做到CP，对另一些数据做到 AP，就算是对同一个数据，调用者可以指定不同的算法，某些算法可以做到CP，某些算法可以做到AP。

当处理CAP的问题时，可以有4个选择：

1

□ 放弃Partition Tolerance

2

放弃Availability

3

□ 放弃Consistency

4

引入BASE

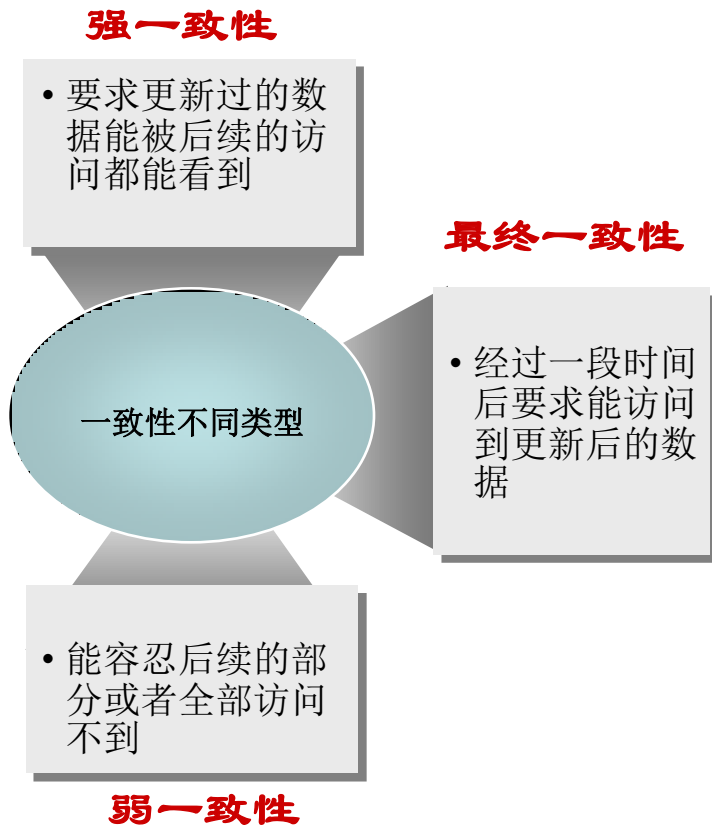
BASE: Basically Available, Soft-state, Eventual Consistency

- BASE模型反ACID模型，完全不同ACID模型，牺牲高一致性，获得可用性或可靠性
- BASE思想主要强调基本的可用性，如果你需要高可用性，也就是纯粹的高性能，那么就要以一致性或容错性为牺牲



最终一致性

从客户端角度



- **因果一致性**：如果进程A通知进程B它已更新了一个数据项，那么进程B的后续访问将返回更新后的值。与进程A无因果关系的进程C的访问遵守一般的最终一致性规则。
- **“读己之所写 (read-your-writes)” 一致性**：当进程A自己更新一个数据项之后，它总是访问到更新过的值，绝不会看到旧值。这是因果一致性模型的一个特例。
- **会话 (Session) 一致性**：这是上一个模型的实用版本，它把访问存储系统的进程放到会话的上下文中。只要会话还存在，系统就保证“读己之所写”一致性。
- **单调 (Monotonic) 读一致性**：如果进程已经看到过数据对象的某个值，那么任何后续访问都不会返回在那个值之前的值。
- **单调写一致性**：系统保证来自同一个进程的写操作顺序执行。要是系统不能保证这种程度的一致性，就非常难以编程了。



最终一致性

从服务器端角度

- N — 数据复制的份数;
- W — 更新数据是需要保证写完成的节点数;
- R — 读取数据的时候需要读取的节点数;

如果 $W+R>N$ ，写的节点和读的节点重叠，则是强一致性。例如对于典型的一主一备同步复制的关系型数据库， $N=2$ ， $W=2$ ， $R=1$ ，则不管读的是主库还是备库的数据，都是一致的。

如果 $W+R\leq N$ ，则是弱一致性。例如对于一主一备异步复制的关系型数据库， $N=2$ ， $W=1$ ， $R=1$ ，则如果读的是备库，就可能无法读取主库已经更新过的数据，所以是弱一致性。



最终一致性

从服务器端角度（续）

- 对于分布式系统，为了保证高可用性，一般设置 $N \geq 3$ 。不同的 N 、 W 、 R 组合，是在可用性和一致性之间取一个平衡，以适应不同的应用场景。
- 如果 $N=W$ ， $R=1$ ，任何一个写节点失效，都会导致写失败，因此可用性会降低，但是由于数据分布的 N 个节点是同步写入的，因此可以保证强一致性。
- 如果 $N=R$ ， $W=1$ ，只需要一个节点写入成功即可，写性能和可用性都比较高。但是，读取其他节点的进程可能不能获取更新后的数据，因此是弱一致性。这种情况下，如果 $W < (N+1)/2$ ，并且写入的节点不重叠的话，则会存在写冲突。



NoSQL与关系数据库的比较

比较标准	RDBMS	NoSQL	备注
数据库原理	完全支持	部分支持	RDBMS有数学模型支持、NoSQL则没有
数据规模	大	超大	RDBMS的性能会随着数据规模的增大而降低；NoSQL可以通过添加更多设备以支持更大规模的数据
数据库模式	固定	灵活	使用RDBMS都需要定义数据库模式，NoSQL则不用
查询效率	快	简单查询非常高效、较复杂的查询性能有所下降	RDBMS可以通过索引，能快速地响应记录查询(point query)和范围查询(range query)；NoSQL没有索引，虽然NoSQL可以使用MapReduce加速查询速度，仍然不如RDBMS
一致性	强一致性	若一致性	RDBMS遵守ACID模型；NoSQL遵守BASE (Basically Available、soft state、Eventually consistent)模型
扩展性	一般	好	RDBMS扩展困难；NoSQL扩展简单
可用性	好	很好	随着数据规模的增大，RDBMS为了保证严格的一致性，只能提供相对较弱的可用性；NoSQL任何时候都能提供较高的可用性
标准化	是	否	RDBMS已经标准化 (SQL)；NoSQL还没有行业标准
技术支持	高	低	RDBMS经过几十年的发展，有很好的技术支持；NoSQL在技术支持方面不如RDBMS
可维护性	复杂	复杂	RDBMS需要专门的数据库管理员(DBA)维护；NoSQL数据库虽然没有DBMS复杂，也难以维护



典型的NoSQL数据库分类

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



NoSQL数据库开源软件

Couchbase
Membase



{name: "mongo", type: "DB"}

MongoDB



Cassandra

Apache Cassandra



Hypertable



主讲教师和助教



主讲教师：林子雨

单位：厦门大学计算机科学系

E-mail: ziyulin@xmu.edu.cn

个人网页: <http://www.cs.xmu.edu.cn/linziyu>

数据库实验室网站: <http://dblab.xmu.edu.cn>



助教：赖明星

单位：厦门大学计算机科学系数据库实验室2011级硕士研究生（导师：林子雨）

E-mail: mingxinglai@gmail.com

个人主页: <http://mingxinglai.com>

欢迎访问《大数据技术基础》2013班级网站: <http://dblab.xmu.edu.cn/node/423>
本讲义PPT存在配套教材《大数据技术基础》，请到上面网站下载。

The background of the slide features several faint, light-blue silhouettes of people. At the top, there are two groups of people standing and holding hands. On the right side, a person is shown in profile, looking towards the center. At the bottom left, two people are shown in profile, facing each other as if in conversation. The overall scene suggests a community or a group of people.

Thank You!