# 数据库场景选型

目录

[数据库场景选型 1](#_Toc2583693)

[一、 关系型数据库场景选择 1](#_Toc2583694)

[二、 Nosql数据库选择场景 2](#_Toc2583695)

## 关系型数据库场景选择

本文不讨论sqlite，这个数据库主要是充当XML配置使用的，就是软件的一些配置可以用它来存，虽然它速写速度很快，但是它很多数据库的必要性（完整事务）没有支持，衡量数据库的指标也不只是读写性能。

Sqlite劣势：

1. 写入粒度粗，写入时全排他锁，会锁定其他连接的写操作直到完成写入，不适合高并发。
2. 基于文件I/O的，直接读写db文件，缺乏管理和优化，没有网络连接数据库这层，缺乏安全性。

Sqlite优势：

1. 基于文件I/O，速度快。
2. 简单、不用配置，移动端和嵌入式端最适合。

其他数据库基于网络连接，所以瓶颈主要卡在网络I/O上，所以觉得慢。

首先，数据库选型，应当考虑一下问题：

1. 对数据一致性（ACID）的要求如何，（要求越高，越不适合使用mysql），如果是支付相关，更应该是oracle
2. 对数据保护的要求如何（要求越高，越不适合使用mysql）
3. 数据读写比如何（越高，越需要考虑使用mysql）
4. 数据的规模如何（规模较小时使用mysql即可，中等规模时，oracle可能更加经济，规模很大时需要从整个架构层面来考虑）
5. 业务变化的频繁程度如何（频繁变换的业务，意味着频繁的数据结构变更，意味着使用mysql的相对成本越高）
6. 自身的技术能力如何（oracle的第三方服务更好，自身没有比较NB的技术能力，还是使用成熟产品oracle比较合适）。

当然，关系型数据库的诞生之初，主要是为了企业设计的，因为企业内部系统需求变换不快，所以用基于二维表结构的关系型数据库（RDBMS）很适合，而随着今年互联网大数据等级web应用，才催生了一堆nosql数据库（非关系型），因为数据结构更加灵活，更适合需求变化快的场景。

例子：

如果是内网项目，每天同时就几百个终端，那么mysql就是不错的选择。当下火热的创业创业web应用互联网公司也多用mysql，用户不算多也是显而易见的，当然，也有基于成本考虑的原因。

## Nosql数据库选择场景

虽然sql数据库是非常有用的工具，但经历了15年的一支独秀之后的垄断即将被打破。这只是时间问题：被迫使用关系数据库，但最终发现不能适应需求的情况不胜枚举。

但是nosql数据库之间的不同，远超过两个sql数据之间的差别。这意味着软件架构师更应该的项目开始时就选择好一个合适的nosql数据库。这里针对这种情况，对Cassandra，MongoDB，Couchdb，Redis，Riak，Membase，Neo4j和Hbase进行了比较：

注：nosql是一项全新的数据库革命性运动，nosql的拥护者们提倡运用非关系型的数据存储。现金的计算机体系结构在数据存储方面的要求具备庞大的水平扩展性，而nosql致力于改变这一现状。目前Google的BigTable和Amazon的Dynamo使用的就是nosql数据库。

1. CouchDB

所用语言：Erlang

特点：DB一致性，易于使用

使用许可：Apache

协议：Http/REST

双向数据复制，持续进行或临时处理，处理时带冲突检查，因此，采用的时master-master复制（见编注2）

MVCC-写操作不阻塞读操作

可保存文件之前的版本

Crash-only（可靠的设计）

需要不时的进行数据压缩

视图：嵌入式 映射/减少

格式化视图：列表显示

支持进行服务端文档验证

支持认证

根据变化实时更新

支持附件处理

因此，CouchApps（独立的js应用程序）需要jQuery程序库

2、