数据库选择策略

清华大学软件学院 潘天翔

什么是数据库？

“A **database** is an organized collection of data”. -- Wikipedia

简单地来说，

- 数据库就是对数据的管理
- 业务中包括了对数据的增，删，改，查等操作
- 数据管理中包括用户访问权限，持久化，分布式等不同的方案选择

数据库举例

用户id	用户名	密码	昵称
1	Danna	123456	Helloworld
2	Wang	888888	WangBin
3	Jack	abcyui	Leonardo

假设一个用户管理系统，它的数据库至少应该有以下几个功能：

- 保存这张用户信息表
- 能够通过id，用户名或昵称查找到其对应的所有信息。
- 支持对密码，昵称的修改。
- 支持对用户的添加和删除。

- 尽量少的响应时间
- 尽量少的存储空间

数据库举例

如果存在这样一个超级数据库，它可以存储任意大小的数据以及数据之间的关系，同时提供了最快的增删改查操作，那么它就能解决一切数据问题。

然而理想是美好的，现实往往是残酷的。我们不能没有无穷的存储空间，
也没有光速的响应时间。我们有的仅仅是一台500G磁盘，4G内存
I3 处理器电脑。如何选择合适的数据库？如何最优化输出的结果？

基本问题分析

用户id	用户名	密码	昵称
1	Danna	123456	Helloworld
2	Wang	888888	WangBin
3	Jack	abcyui	Leonardo

假设一个用户管理系统，它的数据库至少应该有以下几个功能：

- 保存这张用户信息表
- 能够通过id，用户名或昵称查找到其对应的所有信息。
- 支持对密码，昵称的修改。
- 支持对用户的添加和删除。

- 数据怎么存？
- 数据怎么查找？
- 怎么修改数据？
- 怎么添加和删除数据？

数据库的基本分类

针对面临的一些问题，现有的数据库往往有以下一些简单的性质分类。

- 数据怎么存？
  - 持久化存储， 内存数据库
  - 单机，分布式
- 数据怎么增删改查？
  - 关系查找，key-value查找
- 操作是否安全？
  - 事务
  - 一致性，最终一致性
- 可用性
  - 故障了怎么恢复？大部分数据库都具有故障恢复功能。

数据库的基本分类

针对面临的一些问题，现有的数据库往往有以下一些简单的性质分类。

- 数据怎么存？

用户id	用户名	密码	昵称
1	Danna	123456	Helloworld
2	Wang	888888	WangBin
3	Jack	abcyui	Leonardo

- 操作是否安全？

用户id	
1	{用户名：Danna,，密码：123456}
2	{用户名：Wang，密码：888888 ，昵称：WangBin}
3	{用户名：Jack，密码：abcyui，邮箱：jack@gmail.com}

-

数据库的基本分类

事务

- A银行转账B：A查询当前余额，A输入转账数额，银行从A账户删除金额，银行给B账户增加金额，B确认金额增加。
- 如果中间某一步出错了？
- 如果同时两个人在用同一个账户转账？
- 事务保证了操作序列的完整执行，在金融业务中尤为重要。

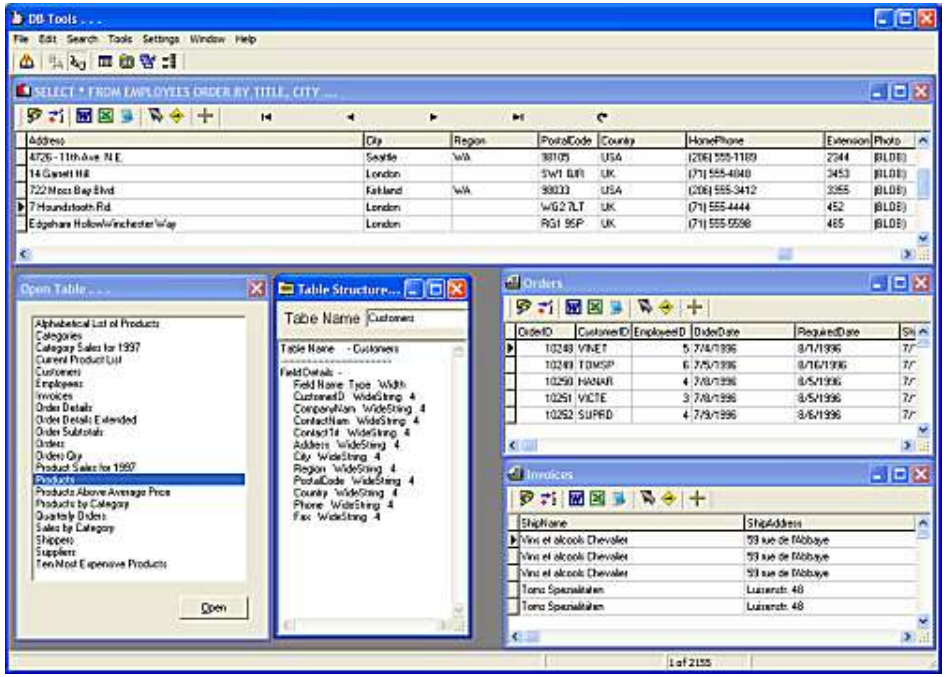
一致性

- 实时一致性，A微信发布了一张照片，同时他的所有好友都能看到。
- 最终一致性，A微信发布了一张照片，其好友总会在未来某个时刻（明天）看到。
- 实时一致性往往消耗计算资源，实际场景中会采用最终一致性进行一定程度的妥协。

常用数据库介绍

Mysql

- 开源的关系型数据库，至今最流行的开源关系型数据库
- 简单易用，拥有大量的第三方插件，社区活跃，文档丰富。
- 关系型数据库支持快速的复杂查询操作。
- 支持完整的事务操作和较高的安全性。



常用数据库介绍

Mongodb

- 模式自由，可以自由地更具需要随时修改文档格式。
- 支持海量数据的查询和插入，支持完全索引
- 自动支持分片等分布式操作，支持故障恢复与备份，学习成本低

需要注意的是，与mysql对比，其有以下几点不同

- Mongo需要占用很大的空间来建立索引
- 不支持事务操作，最终一致性
- 社区尚不成熟，高安全级别无法保证

常用数据库介绍

Mongodb

- 模式自由，可以自由地更具需要随时修改文档格式。
- 支持海量数据的查询和插入，支持完全索引
- 自动支持分片等分布式操作，支持故障恢复与备份，学习成本低

需要注意的是，与mysql对比，其有以下几点不同

- Mongo需要占用很大的空间来建立索引
- 不支持事务操作，最终一致性
- 社区尚不成熟，高

id	（自由的存储模式）
1	{用户名：Danna,，密码：123456}
2	{用户名：Wang，密码：888888 ，昵称：WangBin}
3	{用户名：Jack，密码：abcyui，邮箱：jack@gmail.com}

常用数据库介绍

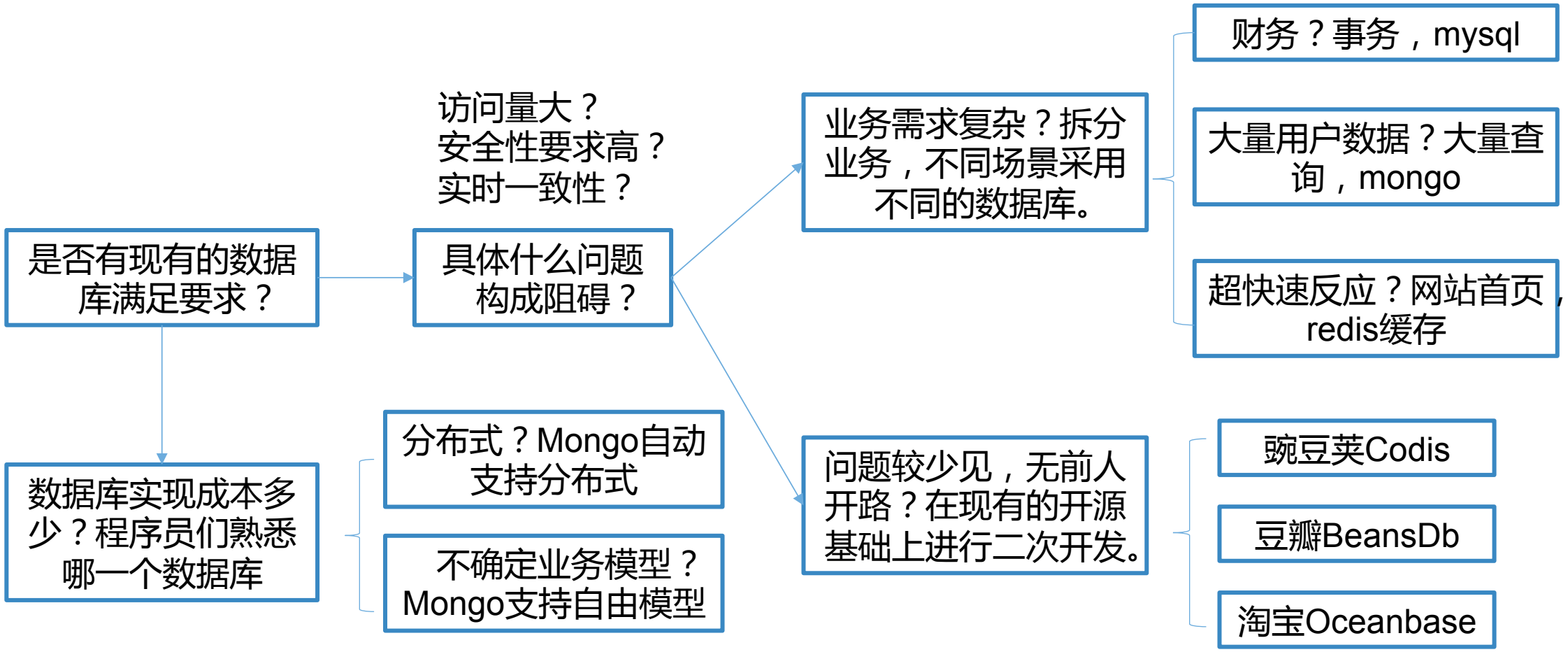
Redis

- 近年兴起的内存数据库，数据在内存之中保证了访问的高效。
- Redis在保证访问速度的同时，也能够进行持久化存储。
- 本身是key-value的存储形式，但是能够支持很多的数据结构，如列表，字典。
- 访问速度非常快！

相较与mysql, mongo其有以下的不同：

- 数据在内存是十分不可靠的，任何的重要数据都不应该存储在内存数据库中。
- 不完整的事务实现，不适合作为安全性高的场景。

数据库选择策略


是否有现有的数据库满足要求？
具体什么问题构成阻碍？
访问量大？
安全性要求高？
实时一致性？
业务需求复杂？拆分业务，不同场景采用不同的数据库。
财务？事务，mysql
大量用户数据？大量查询，mongo
超快速反应？网站首页，redis缓存
问题较少见，无前人开路？在现有的开源基础上进行二次开发。
豌豆荚Codis
豆瓣BeansDb
淘宝Oceanbase
数据库实现成本多少？程序员们熟悉哪一个数据库
分布式？Mongo自动支持分布式
不确定业务模型？Mongo支持自由模型

数据库选择案例分析

考虑一个简单的社交网站案例，要求功能是用户登录，分享等功能。每秒同时在线人数（并发请求）大约为100人。

(1) 用户数据需要持久化存储，不能用内存数据库。

(2) 用户数据安全性不高，不要求事务性，可以使用Mongo

(3) 查询并发量100，较小，mysql(以2000估计)与mongo(以5000估计)均可使用

(4) 业务较复杂，业务模型需要可扩展，mongo修改更方便。

(5) 综合以上几点，我们建议使用Mongo作为数据库。

数据库选择案例分析

考虑一个新闻网站，没有用户登录与交互，仅需要展示新闻内容，但是同时的用户访问量大约为10000人（每秒万人访问）。

(1) 没有用户数据，也没有复杂的关系业务。

(2) 访问量极高，mongo,mysql均不适用。

(3) 内容单一，考虑内存数据库

(4) 安全性要求低，内存数据库可以使用

(5) 综合以上四点，我们建议使用redis等内存数据库。

数据库选择案例分析

考虑一个抢票服务，功能为在某个时刻为大量用户进行抢票服务。要求能够在10s内为5000个用户正确的返回抢票结果。

(1) 用户数据需要持久化存储，考虑 mysql 和 mongo。

(2) 事务性质，不能有一张票被两个人同时抢到，考虑mysql。

(3) 要求快速响应，需要在短时间内返回结果，考虑内存数据库。

考虑到业务场景的复杂性，我们尝试能否对应用场景进行分离，即数据持久化存储用mysql，抢票的时候则使用内存数据库进行响应。然而需要注意的是在进行业务响应时，因为采用内存数据库，那么将不能保证事务性质，怎么办？

数据库选择案例分析

对于抢票业务，我们尝试分离业务场景：

- 抢票业务可以分成（1）非抢票阶段用户登录注册（2）抢票阶段的用户抢票，这部分又包括用户身份验证，用户发出抢票请求，系统验证是否还有余票，系统返回结果

针对以上业务场景的分离，我们对(2)部分的业务请求速度有着极大的需求，因此希望全都在内存之中进行。而(1)因为需要持久化存储，则使用Mysql进行。而在抢票时对用户身份的验证，可以在抢票前将所有用户信息读入到内存数据库之中，来保证访问速度。最后在系统验证余票时，应该有一个事务机制保证不会出现一张票被两个人同时抢到，该机制完全可以通过我们自己的业务代码来进行效率的保证，从而能够将(2)部分的任务全都在内存中进行，保证了反馈的速度。

谢谢大家！

THANKS