Atitit 存储与数据库性能调优流程

目录

[1. 数据库出现性能瓶颈，对外表现有几个方面： 1](#_Toc15328)

[1.1. 大量请求阻塞 1](#_Toc17963)

[1.2. SQL 操作变慢 1](#_Toc19168)

[1.3. 存储io出现问题 2](#_Toc3380)

[2. 数据库优化方案很多，主要分为两大类：软件层面、硬件层面。 2](#_Toc25429)

[2.1. 软件层面包括：SQL 调优、表结构优化、读写分离、数据库集群、分库分表等； 2](#_Toc7752)

[2.2. SQL 调优 索引调优 hint强制索引 2](#_Toc14168)

[2.3. 读写分离 约提升5倍 2](#_Toc18590)

[2.4. 表结构优化 冗余字段。 json字段 2](#_Toc4316)

[2.5. 架构优化 集群 cache 分库 3](#_Toc20387)

[3. 配置优化 3](#_Toc27064)

[3.1. Query cache 3](#_Toc367)

[3.2. Wreite cache区 3](#_Toc3286)

[3.3. 事务落盘机制 改为每秒 提升10倍 3](#_Toc9644)

[3.4. 优化io 大表分区 3](#_Toc20990)

[3.5. 优化io大表单独表空间 3](#_Toc12475)

[3.6. 优化io 防止大文件 3](#_Toc20696)

[3.7. 延时updt索引与外置索引 3](#_Toc17710)

[3.8. 同实例端口数据库分库也可减少io文件体积 3](#_Toc3354)

[4. 升级数据库版本，提升一个数量级 4](#_Toc7617)

[4.1. Msyql 5.6---到8.0 4](#_Toc27895)

[4.2. 小型数据库升级到大型数据库 4](#_Toc3201)

[5. 分库提升无限被 4](#_Toc5202)

[6. Ref 4](#_Toc24513)

# 数据库出现性能瓶颈，对外表现有几个方面：

## 大量请求阻塞

　　在高并发场景下，大量请求都需要操作数据库，导致连接数不够了，请求处于阻塞状态。

## SQL 操作变慢

　　如果数据库中存在一张上亿数据量的表，一条 SQL 没有命中索引会全表扫描，这个查询耗时会非常久。

## 存储io出现问题

　　业务量剧增，单库数据量越来越大，给存储造成巨大压力。

　　从机器的角度看，性能瓶颈无非就是CPU、内存、磁盘、网络这些，要解决性能瓶颈最简单粗暴的办法就是提升机器性能，但是通过这种方法成本和收益投入比往往又太高了，不划算，所以重点还是要从软件角度入手。

# 数据库优化方案很多，主要分为两大类：软件层面、硬件层面。

## 软件层面包括：SQL 调优、表结构优化、读写分离、数据库集群、分库分表等；

　　硬件层面主要是增加机器性能。

## SQL 调优 索引调优 hint强制索引

　　SQL 调优往往是解决数据库问题的第一步，往往投入少部分精力就能获得较大的收益。

　　SQL 调优主要目的是尽可能的让那些慢 SQL 变快，手段其实也很简单就是让 SQL 执行尽量命中索引。

**开启慢 SQL 记录**

## 读写分离 约提升5倍

## ****表结构优化**** 冗余字段。 json字段

## 架构优化 集群 cache 分库

　　当单台数据库实例扛不住时，我们可以增加实例组成集群对外服务。

　　当发现读请求明显多于写请求时，我们可以让主实例负责写，从实例对外提供读的能力；

　　如果读实例压力依然很大，可以在数据库前面加入缓存如 redis，让请求优先从缓存取数据减少数据库访问。

缓存分担了部分压力后，数据库依然是瓶颈，这个时候就可以考虑分库分表的方案了，后面会详细介绍。

# 配置优化

## Query cache

## Wreite cache区

## 事务落盘机制 改为每秒 提升10倍

## 优化io 大表分区

## 优化io大表单独表空间

## 优化io 防止大文件

## 延时updt索引与外置索引

也可改为定时索引

## 同实例端口数据库分库也可减少io文件体积

# 升级数据库版本，提升一个数量级

## Msyql 5.6---到8.0

## 小型数据库升级到大型数据库

# 分库提升无限被

## 多机负载均衡

# Ref

我们为什么要分库分表？|sql|mysql|单库\_网易订阅