**目录**

1. 绪论

1.1课题背景及研究意义

1.2 国内外研究现状（MySQL+Memcached）

1.3 研究的主要内容、目标及方法

1.4 本文组织结构

第二章 Redis缓存系统分析

2.1

2.2

2.3

2.4 内部机制及对比

2．

2.x 本章小结

第三章 使用Redis作为 MongoDB缓存的设计

第四章 使用Redis作为 MongoDB缓存的实现

第五章 实验与测试结果分析

**1绪论**

在互联网的发展过程中，经历了两个比较重要的阶段。第一个阶段，web浏览器的网页大部分都是静态网页，这是浏览器发展的初期。在上世纪90年代，由于计算机水平和普及度还不是很高的情况下，一个网站在同一时刻，并不会有很大的访问量。在这个时期，传统的关系数据库有非常好的用武之地。不仅性能优秀，运行稳定，功能很强大，而且可以随处看到许多优秀的案例。在这之中，MySQL做出的巨大的贡献。随着计算机的普及，越来越多的普通人加入到互联网的世界里，在充分享受着信息时代带来的快捷，无界的信息服务时，也对web提出了更高的要求。Web发展进入到第二阶段，网站开始快速发展，动态交互的网站开始占领主导地位。上网人数的快速增长带来的是巨大的网站访问量，动态网站对数据的实时交互提出了很高的要求。随着网站访问量的不断增加，如现在的淘宝网，新浪微博等拥有亿级用户的网站，几乎大部分使用关系型数据库的网站都开始出现性能问题，对用户体验造成的影响将直接决定一个网站的存亡。

市场的需求推动了技术的革新，面对海量数据存储及高性能的数据读写要求，NoSQL概念在2009年被提了出来。由于NoSQL数据库的结构简单和无关系性，使得它弥补了许多关系型数据库的不足，是非常值得研究的技术。NoSQL可以大体上分为4个种类：键值（Key-Value）数据库、面向文档（Document-Oriented）数据库、列存储（Wide Column Store/Column-Family）数据库、 图（Graph-Oriented）数据库.不同种类的数据库有着各自适合的应用场景。

1.1课题背景及研究意义

本文研究的两种数据库MongoDB属于面向文档（Document-Oriented）数据库适用的场景有以下几种：日志。企业环境下，每个应用程序都有不同的日志信息。Document-Oriented数据库并没有固定的模式，所以我们可以使用它储存不同的信息；分析。鉴于它的弱模式结构，不改变模式下就可以储存不同的度量方法及添加新的度量。Redis属于键值（Key-Value）数据库，适用的场景是储存用户信息，比如会话、配置文件、参数、购物车等等。这些信息一般都和ID（键）挂钩，这种情景下键值数据库是个很好的选择。据此，MongoDB可以满足大量数据的存储，Redis是内存数据库，适合Key-Value形式的快速读写，适合做缓存，本文就将基于此展开研究。

在缓存技术方面，MySQL使用Query Cache，每次表的更新Cache就失效，是一种大粒度的Cache，在针对web2.0的交互频繁的应用，Cache性能不高。而NoSQL的Cache是记录级的，是一种细粒度的Cache，所以NoSQL在这个层面上来说就要性能高很多了。MongoDB本身使用的是内存映射存储引擎，它会把数据文件映射到内存中，如果是读操作，内存中的数据起到缓存的作用，如果是写操作，内存还可以把随机的写操作转换成顺序的写操作，总之可以大幅度提升性能。MongoDB并不干涉内存管理工作，而是把这些工作留给操作系统的虚拟内存管理器去处理，这样做的好处是简化了MongoDB的工作，但坏处是你没有方法很方便的控制MongoDB占多大内存，而且操作系统的虚拟内存管理器会按照LRU算法淘汰冷数据，这种单一的缓存策略并不能满足更多种场景的需求。

1.2 国内外研究现状（MySQL+Memcached）

1.3 研究的主要内容、目标及方法

1.4 本文组织结构