讲师介绍

Hash　　QQ：805921455

从事Java软件研发十年。

前新浪支付核心成员、

咪咕视讯(中国移动)项目经理、

对分布式架构、高性能编程有深入的研究。

明天，你一定会感谢今天奋力拼搏的你

分布式一致性hash算法在缓存架构中的应用

分布式高并发—缓存技术

目录

课程安排

01

hash算法的缓存组件应用

在Memcached、Redis中的应用

02

高性能体系之二级缓存

Java本地缓存与分布式缓存绝佳组合

03

高并发系统缓存架构方案

剖析从浏览器缓存到数据库缓存

04

总结

会总结会学习

目录

hash算法的缓存组件应用

Hash+取模

分布式缓存集群

示例： 3个节点的集群，数据 World:888

假设： hash(World) = 200

则数据放到服务器2上 200%3 = 2

- 场景：高并发场景，集群临时扩容，加一台机器！

hash(World) = 200；200%4 = 0 再次读取，key对应的节点发生了变化

直接导致数据缓存命不中！

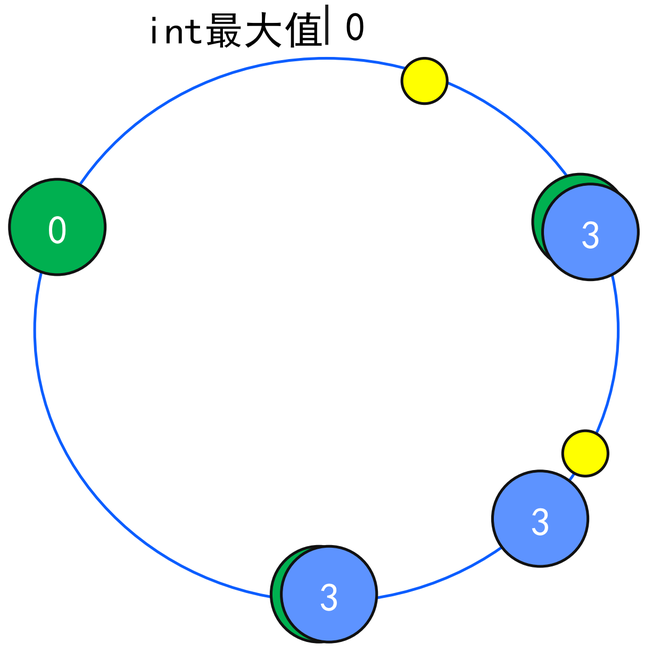
- 增加一个节点后，有多大比例的数据缓存命不中？

3台变4台，导致3/4的映射发生变动

99台服务器扩容到100台，导致99%的映射会发生变动

一致性Hash算法

分布式缓存集群



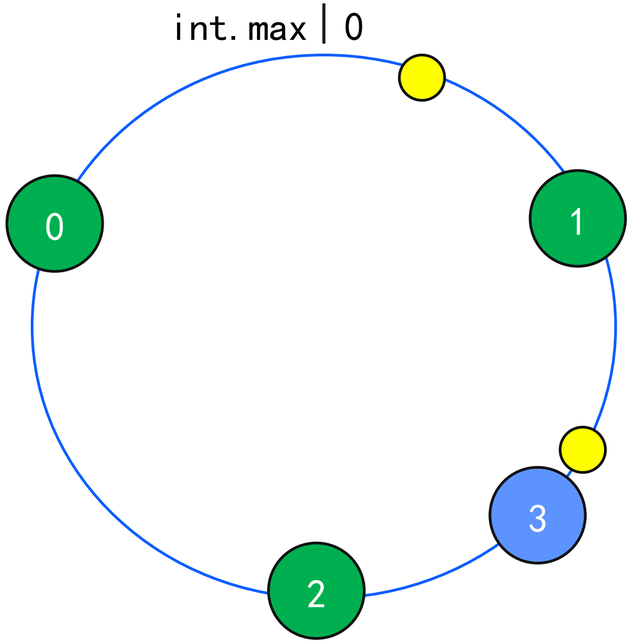
- hash值一个非负整数，把非负整数的值范围做成一个圆环；
- 对集群的节点的某个属性求hash值（如节点名称），根据hash值把节点放到环上；
- 对数据的key求hash，一样的把数据也放到环上，按顺时针方向，找离它最近的节点，就存储到这个节点

增加一个节点影响几何？

影响部分数据，(0~1)/3，取个中值 1/6 。

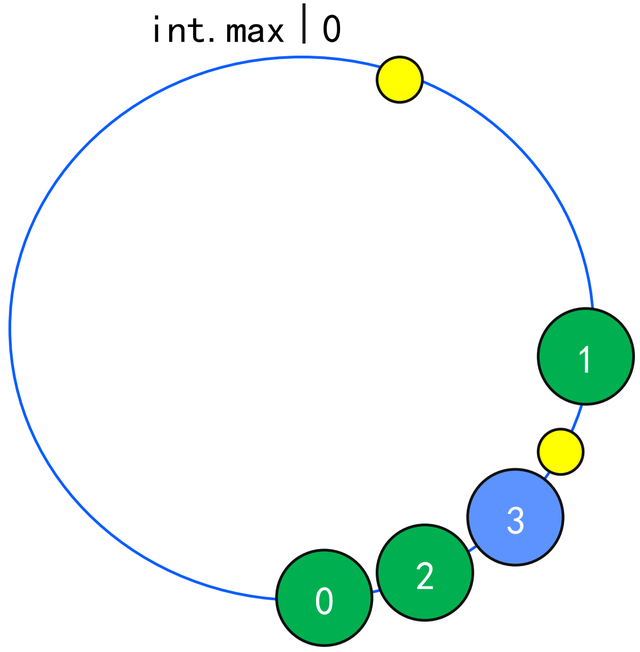
一致性Hash算法

新增节点能均衡缓解原有节点的压力吗？



不能

集群的节点一定会均衡分布在环上吗？



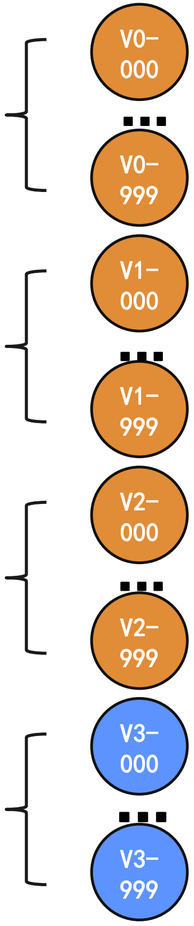
不一定，哈希有倾斜

如何做到均衡分布，均衡缓解？

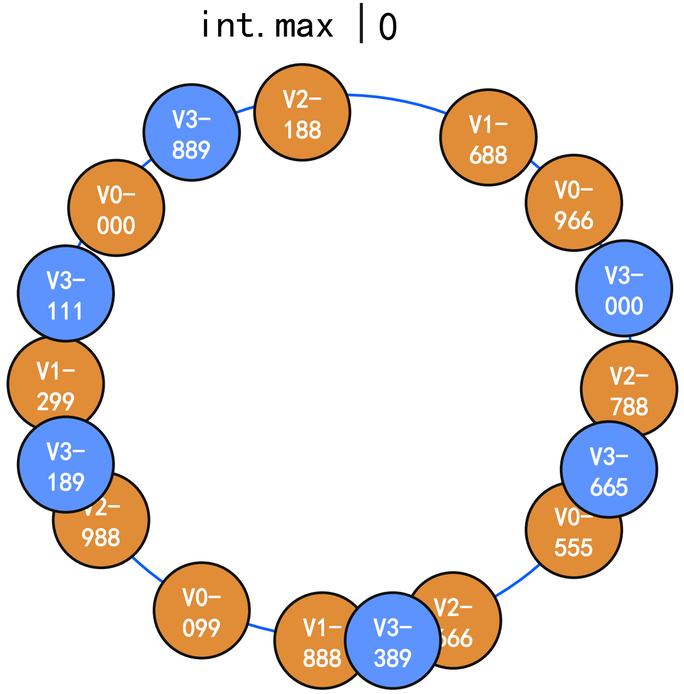
一致性Hash算法

分布式缓存集群

提前分配虚拟节点



虚拟节点上环，来分布数据

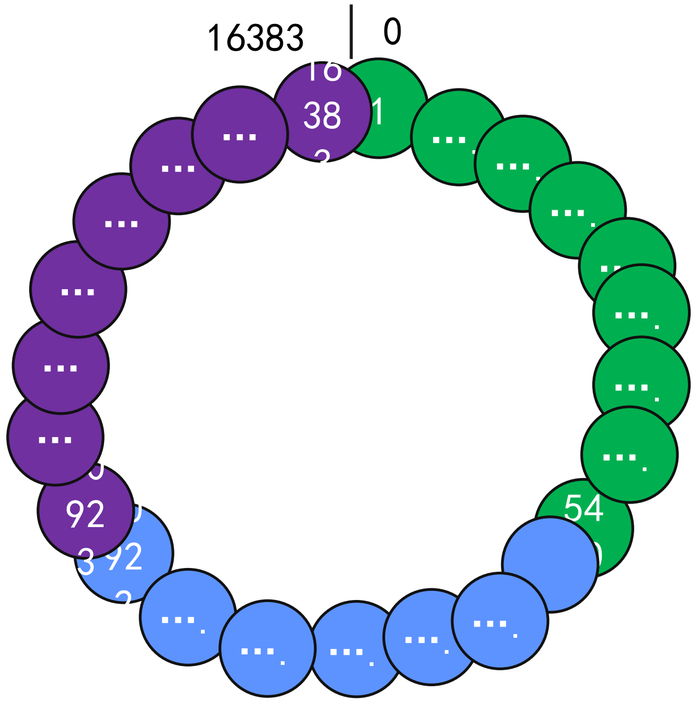


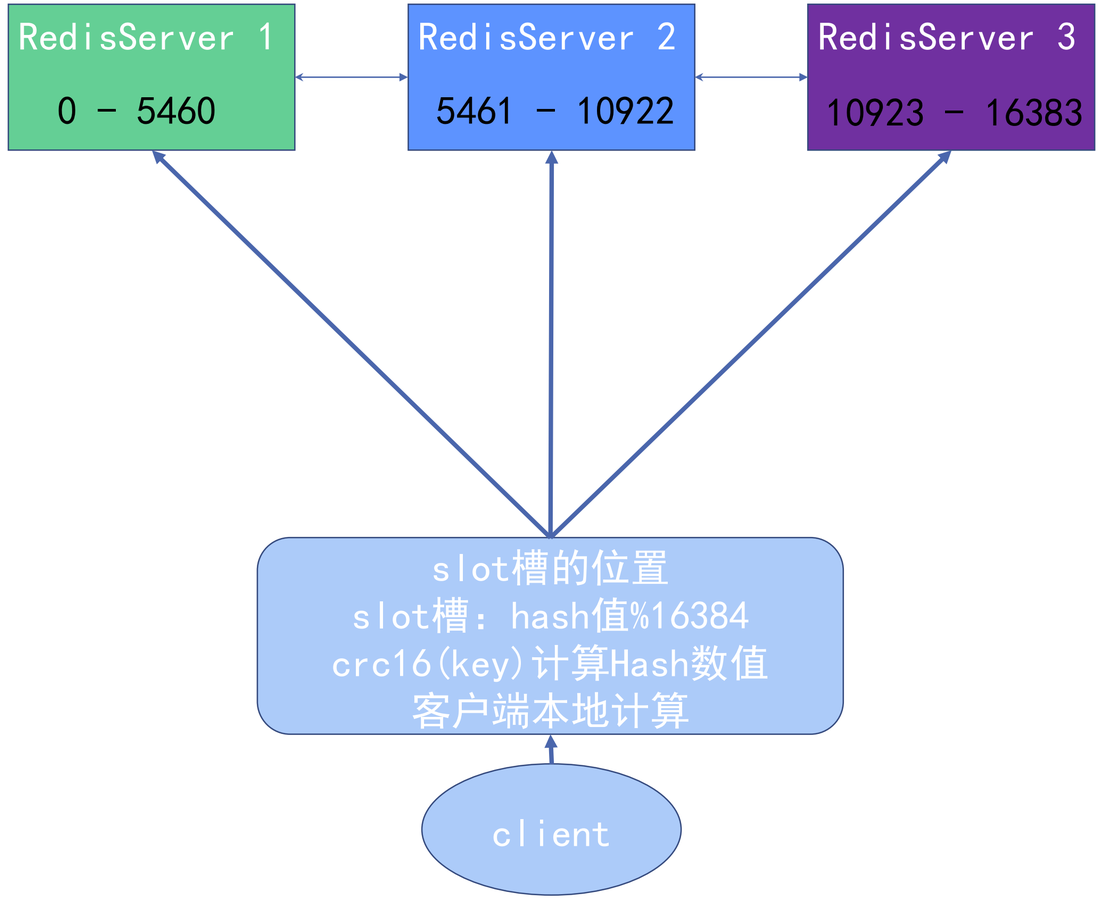
虚拟节点越多，分布越均衡。

虚拟节点越多，新增节点对原有节点影响越均衡。

典型实例：redis的slot机制!

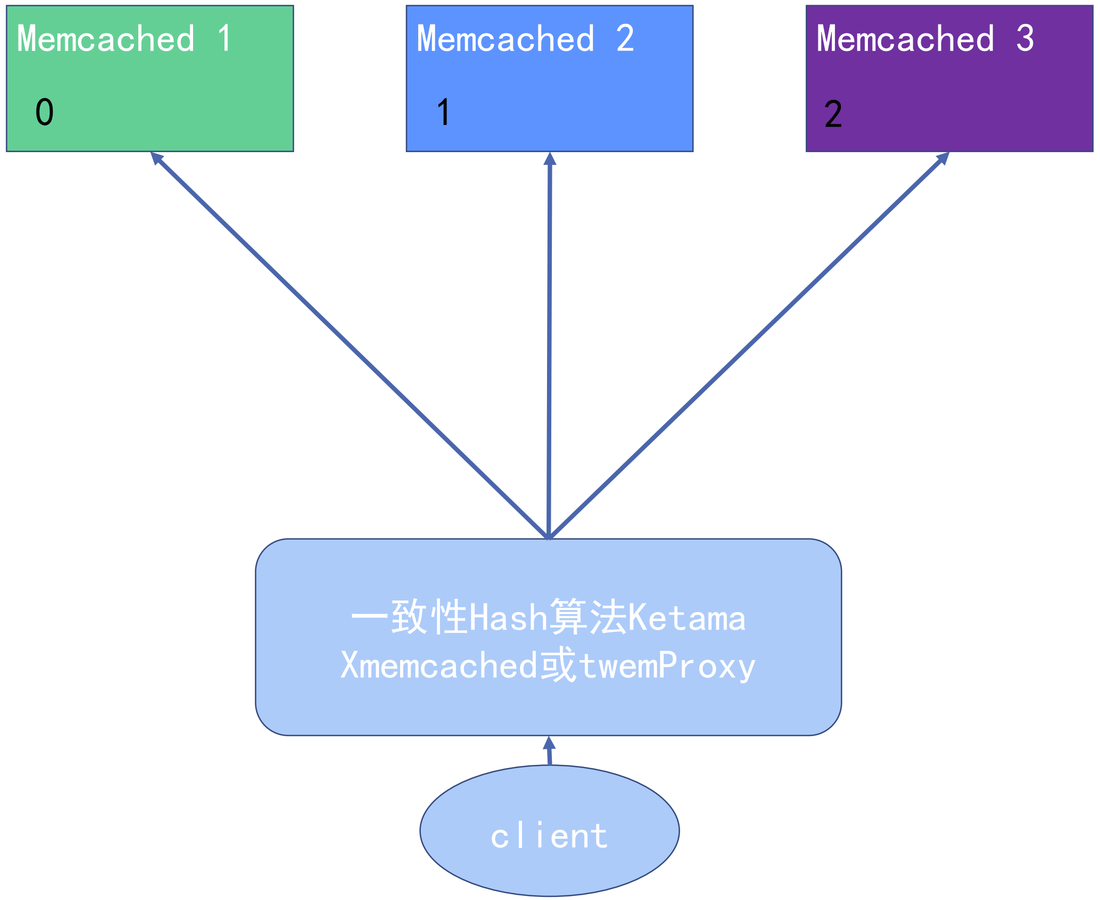
Redis的Slots





使用的范围是0-16383，默认情况下slot是连续排列的

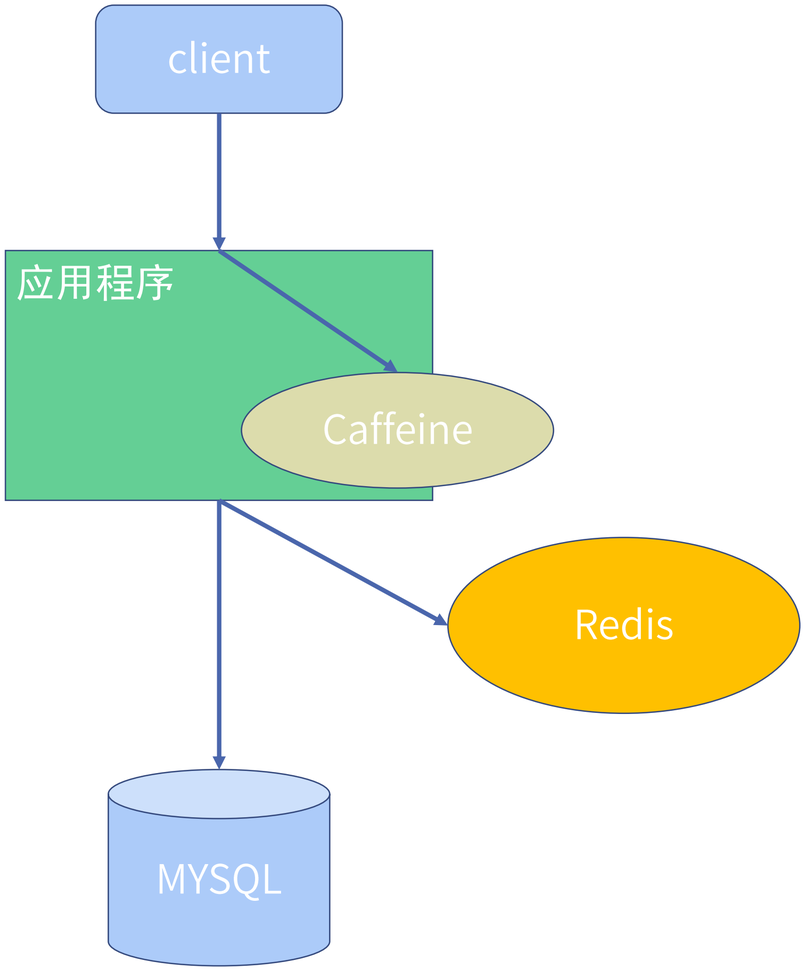
Memcached集群客户端实现



目录

高性能体系之二级缓存

两级缓存技术-J2Cache



1. 客户端向服务端发起请求
2. 服务端先从本地缓存获取
3. 本地缓存未命中则从分布式缓存Redis获取
4. Redis未命中则从数据库查询

解决的问题

1. 使用应用本地存缓存时，一旦应用重启后，由于缓存数据丢失，缓存雪崩，给数据库造成巨大压力，导致应用堵塞
2. 使用应用本地存缓存时，多个应用节点无法共享缓存数据
3. 使用分布式缓存，由于大量的数据通过缓存获取，导致缓存服务的数据吞吐量太大，带宽跑满。现象就是 Redis 服务负载不高，但是由于机器网卡带宽跑满，导致数据读取非常慢

目录

高并发系统缓存架构方案

数据库缓存

Variable_name	Value
have_query_cache	YES
query_cache_limit	1048576
query_cache_min_res_unit	4096
query_cache_size	1048576
query_cache_type	OFF
query_cache_wlock_invalidate	OFF

show variables like 'query_cache';

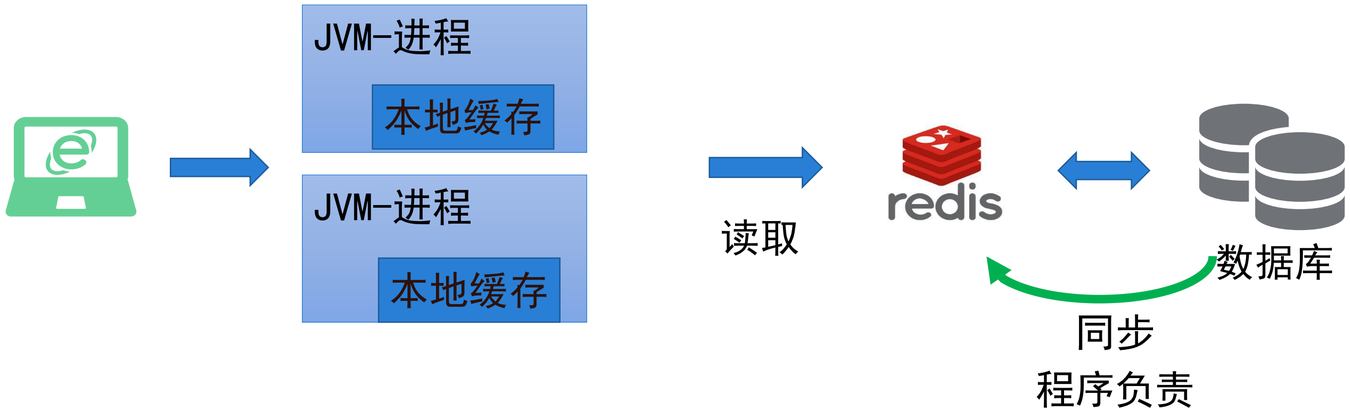
数据库查询缓存

Variable_name	Value
Qcache_free_blocks	1
Qcache_free_memory	1031832
Qcache_hits	0
Qcache_inserts	0
Qcache_lowmem_prunes	0
Qcache_not_cached	584587
Qcache_queries_in_cache	0
Qcache_total_blocks	1

show status like '%Qcache%' ;

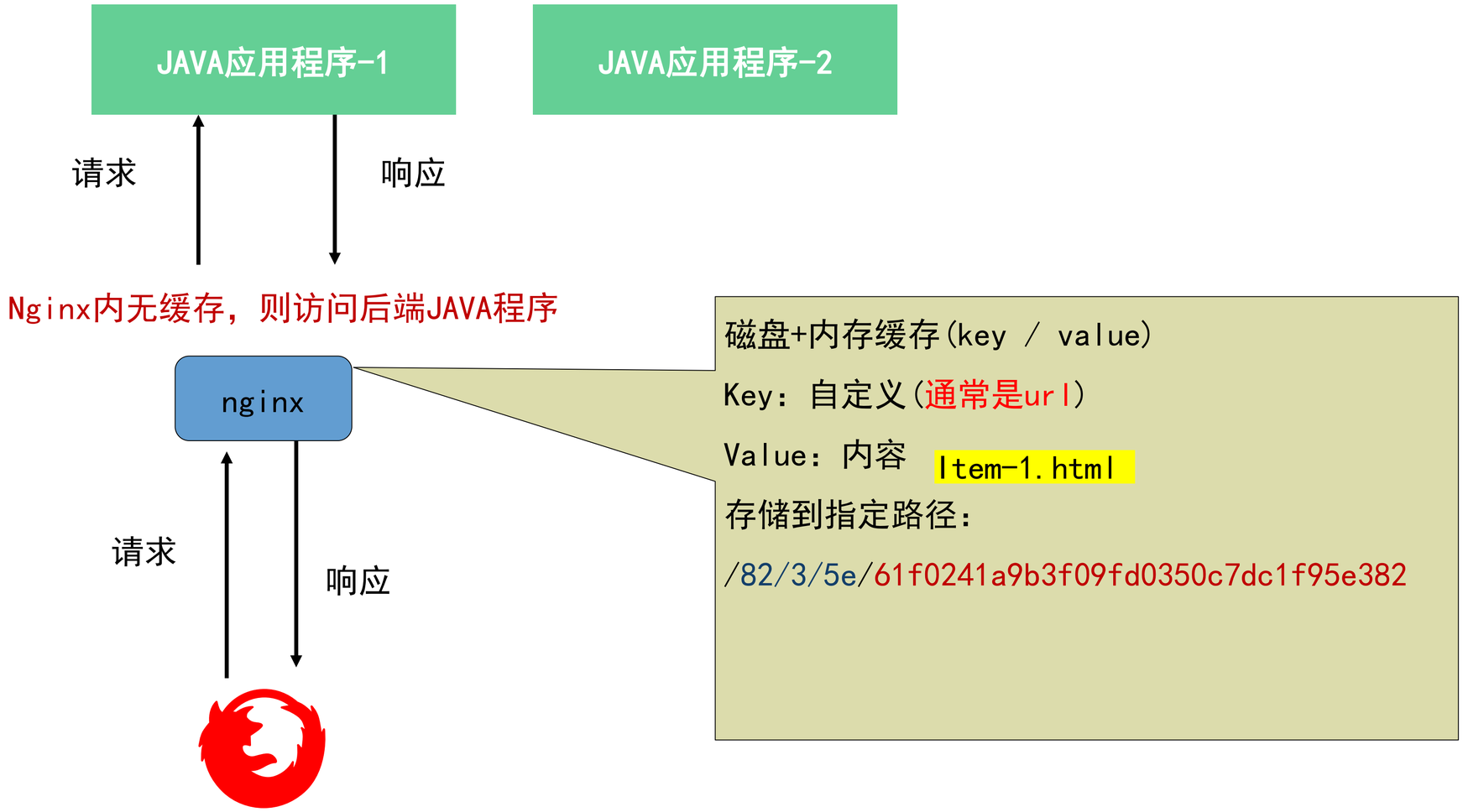
缓存命中率

应用层数据缓存



比较项	单机缓存	分布式缓存中间件
运行方式	嵌入单一进程内部的存储区域	独立启动的进程
交互方式	通过内存直接访问	通过网络进行交互
读取速度	纳秒级别	毫秒级别
实现	HashMap、Guava、Ehcache..	Redis、Memcached..
优缺点	速度最快。只能单进程使用，同样的数据存储到不同的进程，导致缓存维护成本高内存占用率高	速度快，多进程共用，节省内存占用和缓存维护成本

代理服务器缓存


JAVA应用程序-1
JAVA应用程序-2
请求
响应
Nginx内无缓存，则访问后端JAVA程序
nginx
磁盘+内存缓存(key / value)
Key：自定义(通常是url)
Value：内容
Item-1.html
存储到指定路径：
/82/3/5e/61f0241a9b3f09fd0350c7dc1f95e382
请求
响应

目录

总结

总结

Hash算法的应用

应用程序的两级缓存

从浏览器到数据库的缓存架构分析

谢谢观看