



Redis 3.0 集群容器化管理与控制

林喆@HBS

大纲

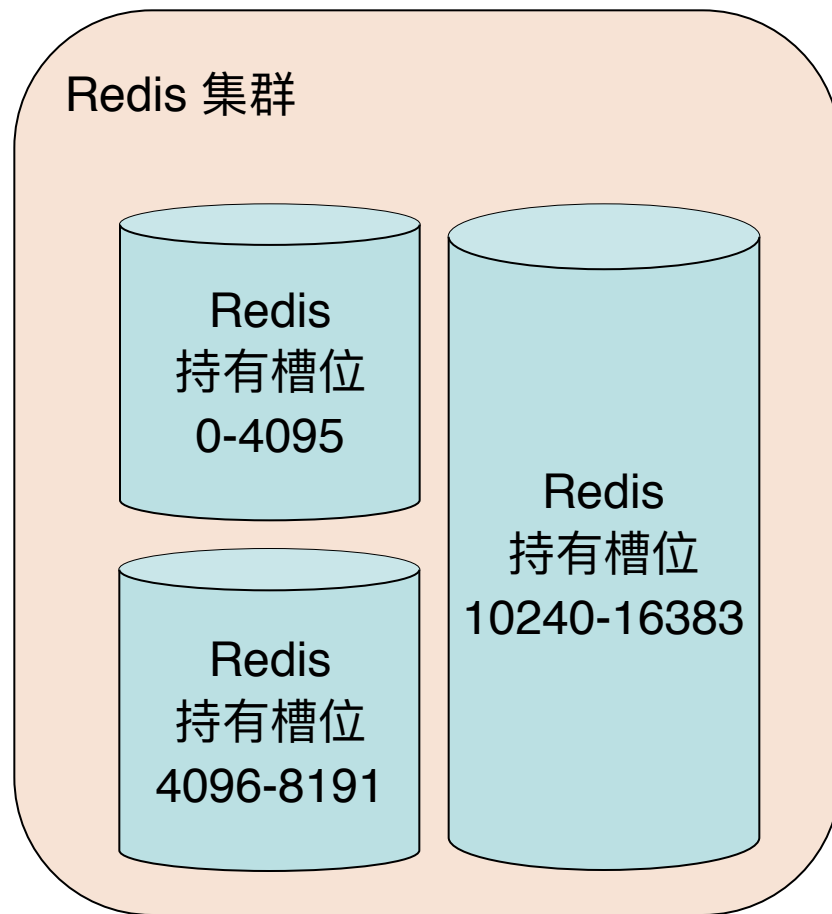
- Redis 3.0 集群基本特性
- 如何用 docker 运维 Redis
- 线上业务用况

Redis 3.0 集群功能特性概述

- 数据散列
 - 集群中的每个节点持有一些 **hash 槽位**, 一个集群总共有 16384 个槽位, 可以以任意方式分布在各个主节点上
 - 存储时, 按 hash 槽进行数据分片
 - 扩容或缩容时, 按 hash 槽迁移数据, 槽位迁移的过程中能保证原子性 (牺牲一定的可用性)

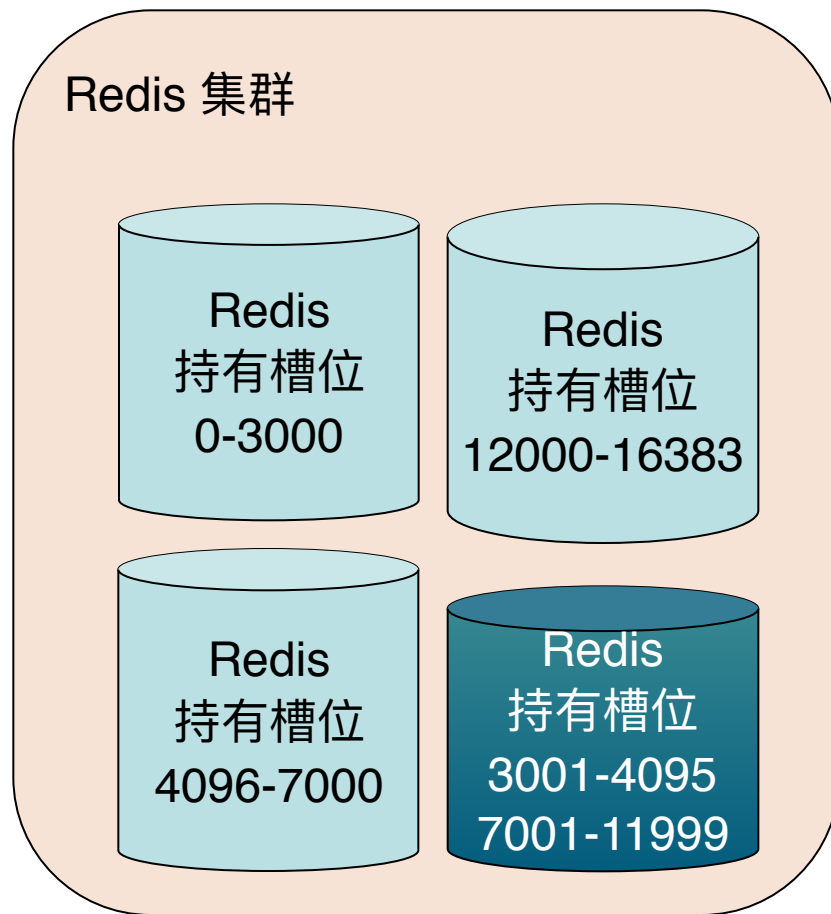
Redis 3.0 集群示意图

比如有三个 Redis 实例
分别持有
4096
4096
8192
个槽位中的数据



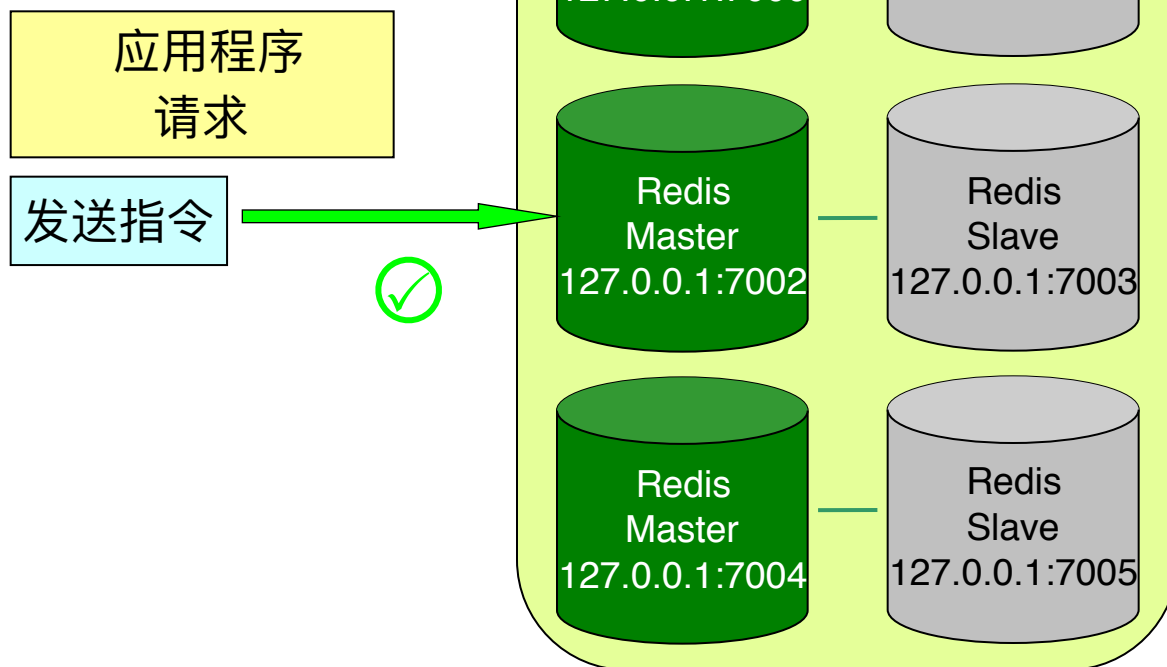
Redis 3.0 集群示意图

扩容时, 新加一个 Redis 实例进来
然后将原有的槽位迁移一部分过去



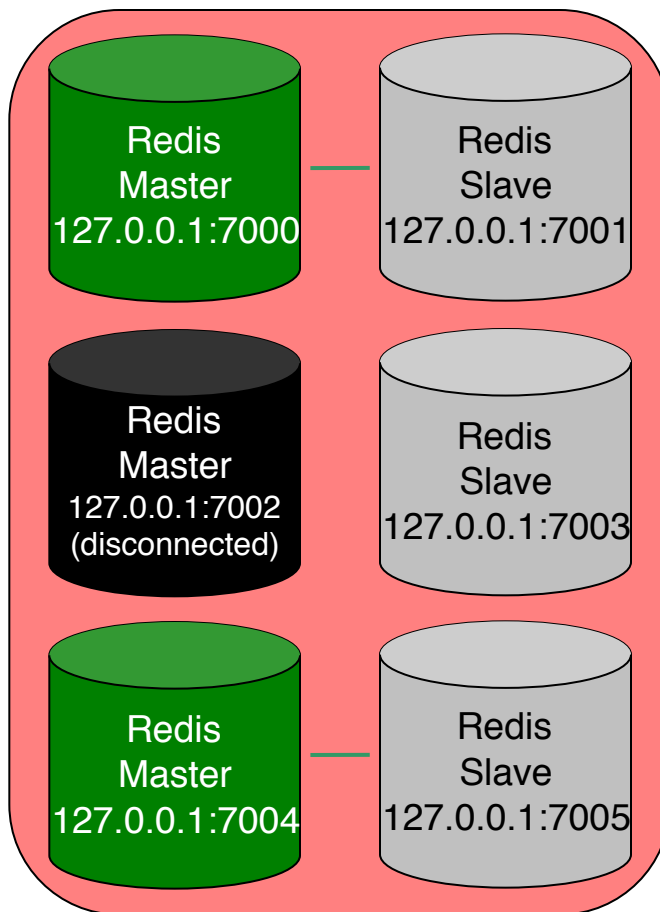
Redis 3.0 主从替换

Redis 主从节点可以
运行时配置



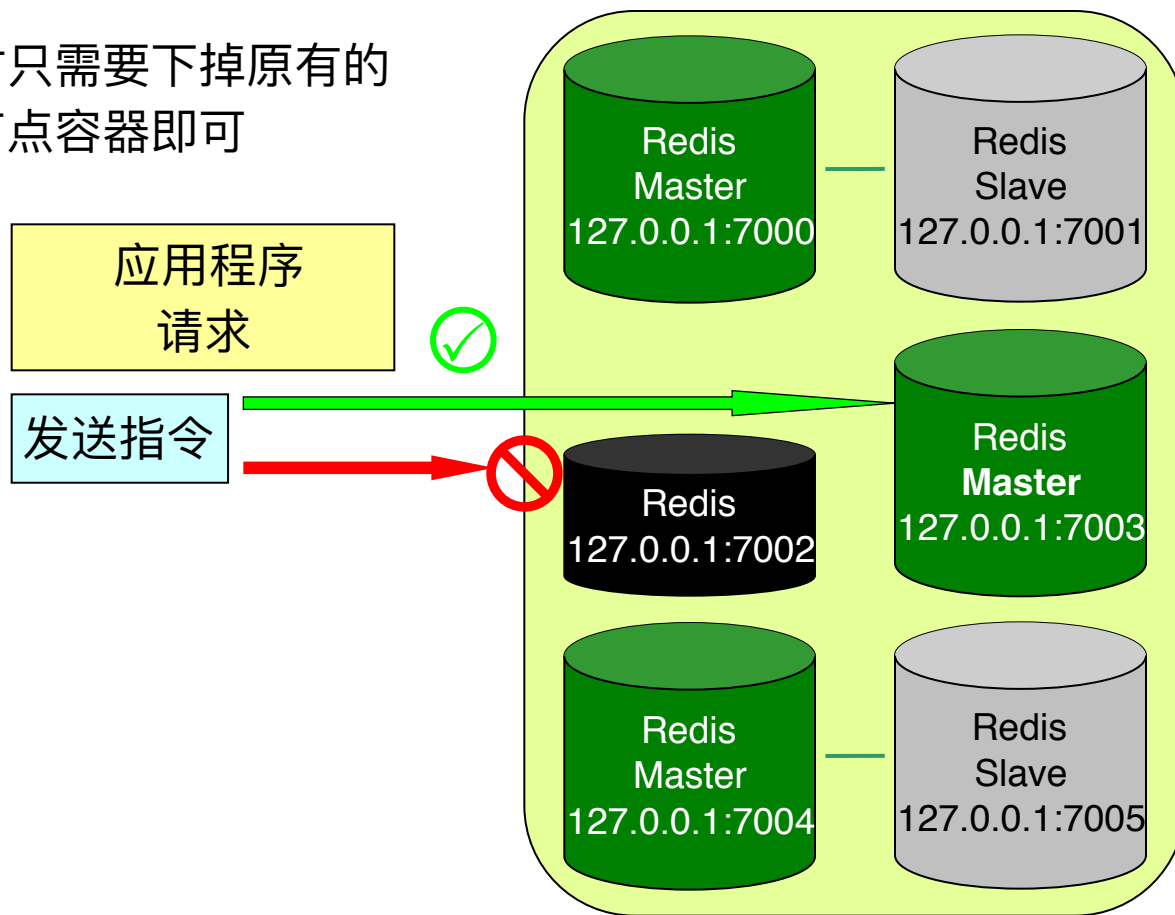
Redis 3.0 主从替换

主节点挂掉后
从节点会自动替补



Redis 3.0 主从替换

这时只需要下掉原有的主节点容器即可



实施方案

- 构建一个 Redis 3.0 的 docker 镜像
 - 以 ubuntu 为基础镜像, 安装 gcc/make 等之后, 下载源码编译
 - 将配置文件中 daemon 设置为 no
- 启动一个容器
 - 采用 ERU 项目通过 HTTP API 来启动容器, 设置容器网络
 - <https://github.com/projecteru>
 - 启动时根据需要加参数, 这一点 Redis 的设计很方便
 - 比如, 需要打开 AOF, 则运行时加上 --appendonly yes
- 将容器加入集群
 - 使用 Redis-ctl 项目管理
 - <https://github.com/HunanTV/redis-ctl>

Redis Ctl 主页面

集群信息									
#	描述	节点数	代理	SLA					
16		1	没有代理						
17	1	3	10.1.1.8889	OK	✈	0.7.2-2015-09-18	100.0%	状态历史	📈
18	ds	1	10.1.1.8889	OK	✈	0.7.2-2015-09-18	100.0%	状态历史	📈

节点信息										显示全部节点	
节点	内存	SLA	集群信息	基本操作							
10.1.1.6379	3.24M / 1.0 GB 0.33%	100.0%	在役 #16 主	状态历史	📈	关闭报警	🛑				
10.2.1.6379	4.24M / 1.0 GB 0.43%	100.0%	在役 #17 主	状态历史	📈	关闭报警	🛑				
10.1.1.969	65.15M / 1.0 GB 6.67%	100.0%	就绪	状态历史	📈	关闭报警	🛑				
10.2.1.6379	2.81M / 1.0 GB 0.29%	100.0%	就绪	状态历史	📈	关闭报警	🛑				
10.1.1.6379	3.18M / 1.0 GB 0.33%	100.0%	在役 #17 主	状态历史	📈	关闭报警	🛑				
10.2.1.5:6379	3.20M / 1.0 GB 0.33%	100.0%	在役 #17 从	状态历史	📈	关闭报警	🛑				
10.2.1.37:6379	3.18M / 1.0 GB 0.33%	100.0%	在役 #18 主	状态历史	📈	关闭报警	🛑				

整合 ERU 自动部署 Redis 容器

Eru地址: http://10.10.10.1000/

app - 机器数 2

指定机器

kili11-10.10.10.10:2376

HOST 网络模型部署

创建节点

启用 AOF

集群模式

指定端口 6379

创建 ERU 节点

创建代理

选择集群

线程数

只读

指定端口 8889

创建 ERU 代理

新增节点

[查看 Eru 节点列表](#)

容器 ID	镜像版本	宿主地址	创建时间	地址	集群模式	下线容器
9fd5c8cc-9c31	9649e98	10.10.10.10	2016-03-07 10:29:45	10.222.10.10		下线容器

新增代理

[查看 Eru 代理列表](#)

容器 ID	镜像版本	宿主地址	创建时间	地址	下线容器
-------	------	------	------	----	------

在控制面板上将 Redis 加入集群

内存

2.9 MB / 1.0 GB [设置](#)

部署

ERU 容器

ERU 容器 ID 0fe8f6c09151bb3a75793af7ce4de7c9898861c5c0dee4d671fbb6190

ERU 容器创建时间 2015-11-26 15:25:59

ERU 宿主机地址 10.200.8.34

统计

CPU	
用户态累计	2555.77
内核态累计	8079.15
总运行时间	8795269
平均 CPU 占用率	0.12%

存储	
Keyspace hits	338
Keyspace misses	144
Expired keys	0
Evicted keys	0

内存	
已使用内存	2.9 MB
常驻集	11.6 MB

[状态历史](#)

关闭报警

发送指令

INFO

CLUSTER NODES

集群自动发现

启动集群

加入集群

m 1 个节点

加入成为主节点

-- 或 --

10.2 :6379

以此主节点的副本加入

自动扩容

- 无人值守的 Redis 集群扩容
 - 当监测进程发现内存使用量超过阈值会自动通过 ERU 创建新实例加入集群并自动迁移数据
 - 可定制扩容方案参数

设置自动扩容

ERU 地址: <http://10.200.8.33:5000/> 集群自动扩容在集群节点内存容量使用超过 90% 时自动申请新节点并迁移数据. 启用自动扩容

用于自动申请节点的机房: d 机器数 2 指定分配容器的机器 kill6 - 10: 34:2376

配备从节点 1 个

从节点 #0 kill3 - 10: 29:2376

所有节点启用 AOF

自动扩容任务

任务 #23

任务步骤 (共 3 项)

ID	命令	参数	开始时间	执行状态
27	添加主节点	10.222.84.127:6379	2015-06-03 10:28:27	完成 2015-06-03 10:28:28
28	添加从节点	从节点为 10.222.189.112:6379 主节点为 10.222.84.127:6379	2015-06-03 10:28:31	正在执行
29	迁移槽位	从 10.222.153.164:6379 迁移 7192 个槽位至 10.222.84.127:6379		等待

线上业务用况

•QPS

- 2015 小年夜晚会直播 峰值 > 50,000 QPS (Redis 3.0 rc2)
- 2015 元宵晚会直播 峰值 > 30,000 QPS (Redis 3.0 rc2)
- 2015 年 <<我是歌手>> 直播 峰值 > 80,000 QPS (Redis 3.0 正式版)
- 2015 年双十一晚会直播 峰值 > 30,000 QPS (Redis 3.0.4)
- 2016 元旦晚会直播 峰值 > 100,000 QPS (Redis 3.0.5)

• 资源消耗

- 8 台服务器
- 超过 70 个 Redis 实例 (70G 内存)



谢谢