

第四周报告-柳杰

一 分布式Redis集群搭建

Redis集群介绍

Redis 集群是一个提供在多个Redis间节点间共享数据的程序集。

Redis集群并不支持处理多个keys的命令,因为这需要在不同的节点间移动数据,从而达不到像Redis那样的性能,在高负载的情况下可能会导致不可预料的错误。

Redis 集群通过分区来提供一定程度的可用性,在实际环境中当某个节点宕机或者不可达的情况下继续处理命令. Redis 集群的优势:

- 自动分割数据到不同的节点上。
- 整个集群的部分节点失败或者不可达的情况下能够继续处理命令。

Redis 集群的主从复制模型

为了使在部分节点失败或者大部分节点无法通信的情况下集群仍然可用, 所以集群使用了主从复制模型,每个节点都会有N-1个复制品。

例如, 在具有A, B, C三个节点的集群,在没有复制模型的情况下,如果节点B失败了, 那么整个集群就会以为缺少5501-11000 (Hash映射) 这个范围的槽而不可用。

然而如果在集群创建的时候 (或者过一段时间) 我们为每个节点添加一个从节点A1, B1, C1,那么整个集群便有三个master节点和三个slave节点组成, 这样在节点B失败后, 集群便会选举B1为新的主节点继续服务, 整个集群便不会因为槽找不到而不可用了

不过当B和B1 都失败后, 集群是不可用的。

Redis 一致性保证

Redis 并不能保证数据的强一致性. 这意味这在实际中集群在特定的条件下可能会丢失写操作。

第一个原因是因为集群是用了异步复制. 写操作过程:

- 客户端向主节点B写入一条命令。
- 主节点B向客户端回复命令状态。
- 主节点将写操作复制给他得从节点 B1, B2 和 B3。

主节点对命令的复制工作发生在返回命令回复之后, 因为如果每次处理命令请求都需要等待复制操作完成的话, 那么主节点处理命令请求的速度将极大地降低 —— 我们必须在性能和一致性之间做出权衡。 注意: Redis 集群可能会在将来提供同步写的方法。 Redis 集群另外一种可能会丢失命令的情况是集群出现了网络分区, 并且一个客户端与至少包括一个主节点在内的少数实例被孤立。

举个例子 假设集群包含 A、B、C、A1、B1、C1 六个节点，其中 A、B、C 为主节点，A1、B1、C1 为 A、B、C 的从节点，还有一个客户端 Z1 假设集群中发生网络分区，那么集群可能会分为两方，大部分的一方包含节点 A、C、A1、B1 和 C1，小部分的一方则包含节点 B 和客户端 Z1。

Z1 仍然能够向主节点 B 中写入，如果网络分区发生时间较短，那么集群将会继续正常运作，如果分区的时间足够让大部分的一方将 B1 选举为新的 master，那么 Z1 写入 B 中的数据便丢失了。

在网络分裂出现期间，客户端 Z1 可以向主节点 B 发送写命令的最大时间是有限制的，这一时间限制称为节点超时时间（node timeout），是 Redis 集群的一个重要的配置选项

Redis 集群的安装和配置测试

- 安装源码编译依赖

```
sudo apt-get install gcc
sudo apt-get install tcl
```

- 下载源码，解压，修改权限

```
wget http://download.redis.io/releases/redis-3.0.2.tar.gz (3.0 以后的版本才支持集群模式)
tar -zxvf redis-3.0.2.tar.gz -C /usr/local
sudo chown -R hadoop ./redis-3.0.2
```

- 编译，安装，测试

```
cd redis-3.0.2/src
make (以root用户运行)
make install
```

- 在每台单机做上面的安装后，修改 redis.conf 配置文件

主机（Master）关键配置修改如下

```
daemonize yes

port 6379

bind 10.1.0.5
```

- 从机配置如下

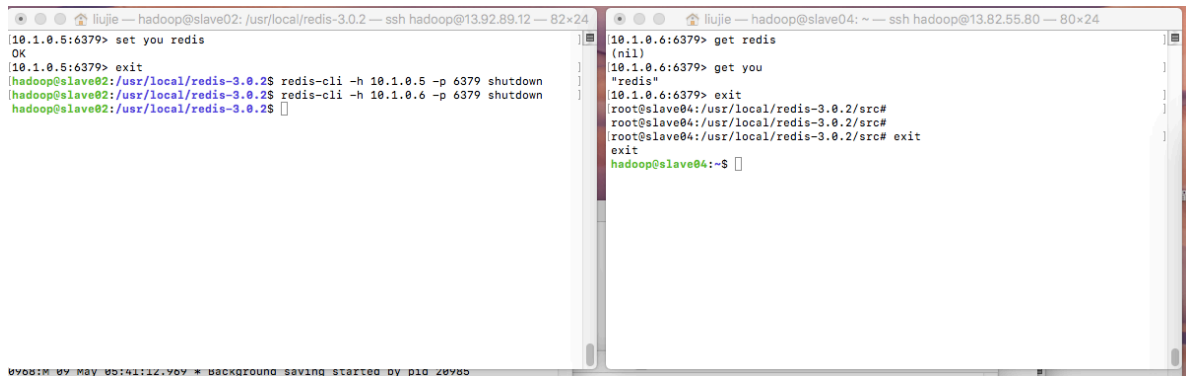
```
daemonize yes
```

```
bind 10.1.0.6
```

```
slaveof 10.1.0.5 6379
```

```
#masterauth 配置机器ssh免密码登录即可
```

- 测试集群



```
liujie — hadoop@slave02: /usr/local/redis-3.0.2 — ssh hadoop@13.92.89.12 — 82x24
10.1.0.5:6379> set you redis
OK
10.1.0.5:6379> exit
[hadoop@slave02: /usr/local/redis-3.0.2$ redis-cli -h 10.1.0.5 -p 6379 shutdown
[hadoop@slave02: /usr/local/redis-3.0.2$ redis-cli -h 10.1.0.6 -p 6379 shutdown
[hadoop@slave02: /usr/local/redis-3.0.2$ ]

liujie — hadoop@slave04: ~ — ssh hadoop@13.82.55.80 — 80x24
10.1.0.6:6379> get redis
(nil)
10.1.0.6:6379> get you
"redis"
10.1.0.6:6379> exit
[root@slave04: /usr/local/redis-3.0.2/src#
[root@slave04: /usr/local/redis-3.0.2/src#
[root@slave04: /usr/local/redis-3.0.2/src# exit
exit
hadoop@slave04:~$ ]
```

二 Hadoop平台上的MapReduce测试

3个节点

- 300MB实验数据

时间点	周转时间(min)
2017/5/5 19:00:00	00:43
2017/5/5 20:00:00	00:37
2017/5/5 21:00:00	00:37
2017/5/5 22:00:00	00:38
2017/5/5 23:00:00	00:40
2017/5/6 00:00:00	00:42
2017/5/6 01:00:00	00:38
2017/5/6 02:00:00	00:47
2017/5/6 03:00:00	00:50
2017/5/6 04:00:00	00:44
2017/5/6 05:00:00	00:45
2017/5/6 06:00:00	00:36
2017/5/6 07:00:00	00:40
2017/5/6 08:00:00	00:53
2017/5/6 09:00:00	00:41
2017/5/6 10:00:00	00:37
2017/5/6 11:00:00	00:38
2017/5/6 12:00:00	00:50
2017/5/6 13:00:00	00:41
2017/5/6 14:00:00	00:42
2017/5/6 15:00:00	00:37
2017/5/6 16:00:00	00:45
2017/5/6 17:00:00	00:48
2017/5/6 18:00:00	00:52

- 3000MB实验数据

时间点	周转时间(min)
2017/5/4 16:00:00	03:52
2017/5/4 17:00:00	03:36
2017/5/4 18:00:00	03:36
2017/5/4 19:00:00	03:35
2017/5/4 20:00:00	03:35
2017/5/4 21:00:00	03:53
2017/5/4 22:00:00	03:34
2017/5/4 23:00:00	03:35
2017/5/5 00:00:00	03:36
2017/5/5 01:00:00	03:37
2017/5/5 02:00:00	03:35
2017/5/5 03:00:00	03:33
2017/5/5 04:00:00	03:35
2017/5/5 05:00:00	03:36
2017/5/5 06:00:00	03:35
2017/5/5 07:00:00	03:53
2017/5/5 08:00:00	03:39
2017/5/5 09:00:00	03:35
2017/5/5 10:00:00	03:38
2017/5/5 11:00:00	03:35
2017/5/5 12:00:00	03:40
2017/5/5 13:00:00	03:37
2017/5/5 14:00:00	03:35
2017/5/5 15:00:00	03:37

- 15880MB

时间点	周转时间(min)
2017/5/7 02:00:00	15:03
2017/5/7 03:00:00	15:12
2017/5/7 04:00:00	15:15
2017/5/7 05:00:00	15:17
2017/5/7 06:00:00	15:15
2017/5/7 07:00:00	15:18
2017/5/7 08:00:00	15:03
2017/5/7 09:00:00	15:02
2017/5/7 10:00:00	15:03
2017/5/7 11:00:00	15:09
2017/5/7 12:00:00	15:25
2017/5/7 13:00:00	15:14
2017/5/7 14:00:00	15:08
2017/5/7 15:00:00	15:22
2017/5/7 16:00:00	15:05
2017/5/7 17:00:00	15:10
2017/5/7 18:00:00	15:25
2017/5/7 19:00:00	15:17
2017/5/7 20:00:00	15:01
2017/5/7 21:00:00	15:14
2017/5/7 22:00:00	15:14
2017/5/7 23:00:00	15:17
2017/5/7 24:00:00	15:08
2017/5/8 01:00:00	15:05

- 6个节点15880MB，在测试中。。。

drwxr-xr-x	hadoop	supergroup	0 B	2017/5/9 下午 12:08:42	0
drwxr-xr-x	hadoop	supergroup	0 B	2017/5/9 下午 1:08:31	0
drwxr-xr-x	hadoop	supergroup	0 B	2017/5/9 下午 2:08:31	0