

加餐（五）-Redis有哪些好用的运维工具？

你好，我是蒋德钧。

今天的加餐，我来给你分享一些好用的Redis运维工具。

我们在应用Redis时，经常会面临的运维工作，包括Redis的运行状态监控，数据迁移，主从集群、切片集群的部署和运维。接下来，我就从这三个方面，给你介绍一些工具。我们先来学习下监控Redis实时运行状态的工具，这些工具都用到了Redis提供的一个监控命令：INFO。

最基本的监控命令：INFO命令

Redis本身提供的INFO命令会返回丰富的实例运行监控信息，这个命令是Redis监控工具的基础。

INFO命令在使用时，可以带一个参数section，这个参数的取值有好几种，相应的，INFO命令也会返回不同类型的监控信息。我把INFO命令的返回信息分成5大类，其中，有的类别当中又包含了不同的监控内容，如下表所示：

类别	子类别	对应INFO命令的section参数
实例本身配置信息	无	server
运行状态统计信息	客户端统计信息	client
	通用统计信息	stat
	数据库整体统计信息	keyspace
	不同类型命令的调用统计信息	commandstats
资源使用统计信息	CPU使用情况	cpu
	内存使用情况	memory
关键功能运行状态	RDB、AOF运行情况	persistence
	主从复制的运行情况	replication
	切片集群的运行情况	cluster
扩展模块信息	无	modules

在监控Redis运行状态时，INFO命令返回的结果非常有用。如果你想了解INFO命令的所有参数返回结果的详细含义，可以查看Redis官网的介绍。这里，我给你提几个运维时需要重点关注的参数以及它们的重要返回结果。

首先，无论你是运行单实例或是集群，我建议你重点关注一下stat、commandstat、cpu和memory这四个参数的返回结果，这里面包含了命令的执行情况（比如命令的执行次数和执行时间、命令使用的CPU资源），内存资源的使用情况（比如内存已使用量、内存碎片率），CPU资源使用情况等，这可以帮助我们判断实例的运行状态和资源消耗情况。

另外，当你启用RDB或AOF功能时，你就需要重点关注下persistence参数的返回结果，你可以通过它查看到RDB或者AOF的执行情况。

如果你在使用主从集群，就要重点关注下replication参数的返回结果，这里面包含了主从同步的实时状态。

不过，INFO命令只是提供了文本形式的监控结果，并没有可视化，所以，在实际应用中，我们还可以使用一些第三方开源工具，将INFO命令的返回结果可视化。接下来，我要讲的Prometheus，就可以通过插件将Redis的统计结果可视化。

面向Prometheus的Redis-exporter监控

Prometheus是一套开源的系统监控报警框架。它的核心功能是从被监控系统中抽取监控数据，结合**Grafana**工具，进行可视化展示。而且，监控数据可以保存到时序数据库中，以便运维人员进行历史查询。同时，Prometheus会检测系统的监控指标是否超过了预设的阈值，一旦超过阈值，Prometheus就会触发报警。

对于系统的日常运维管理来说，这些功能是非常重要的。而Prometheus已经实现了使用这些功能的工具框架。我们只要能从被监控系统中获取到监控数据，就可以用Prometheus来实现运维监控。

Prometheus正好提供了插件功能来实现对一个系统的监控，我们把插件称为exporter，每一个exporter实际是一个采集监控数据的组件。exporter采集的数据格式符合Prometheus的要求，Prometheus获取这些数据后，就可以进行展示和保存了。

Redis-exporter就是用来监控Redis的，它将INFO命令监控到的运行状态和各种统计信息提供给Prometheus，从而进行可视化展示和报警设置。目前，Redis-exporter可以支持Redis 2.0至6.0版本，适用范围比较广。

除了获取Redis实例的运行状态，Redis-exporter还可以监控键值对的大小和集合类型数据的元素个数，这个可以在运行Redis-exporter时，使用check-keys的命令行选项来实现。

此外，我们可以开发一个Lua脚本，定制化采集所需监控的数据。然后，我们使用scripts命令行选项，让Redis-exporter运行这个特定的脚本，从而可以满足业务层的多样化监控需求。

最后，我还想再给你分享两个小工具：**redis-stat**和**Redis Live**。跟Redis-exporter相比，这两个都是轻量级的监控工具。它们分别是用Ruby和Python开发的，也是将INFO命令提供的实例运行状态信息可视化展示。虽然这两个工具目前已经很少更新了，不过，如果你想自行开发Redis监控工具，它们都是不错的参考。

除了监控Redis的运行状态，还有一个常见的运维任务就是数据迁移。接下来，我们再来学习下数据迁移的工具。

数据迁移工具Redis-shake

有时候，我们需要在不同的实例间迁移数据。目前，比较常用的一个数据迁移工具是**Redis-shake**，这是阿里云Redis和MongoDB团队开发的一个用于Redis数据同步的工具。

Redis-shake的基本运行原理，是先启动Redis-shake进程，这个进程模拟了一个Redis实例。然后，Redis-shake进程和数据迁出的源实例进行数据的全量同步。

这个过程和Redis主从实例的全量同步是类似的。

源实例相当于主库，Redis-shake相当于从库，源实例先把RDB文件传输给Redis-shake，Redis-shake会把RDB文件发送给目的实例。接着，源实例会再把增量命令发送给Redis-shake，Redis-shake负责把这些增量命令再同步给目的实例。

下面这张图展示了Redis-shake进行数据迁移的过程：



Redis-shake的一大优势，就是支持多种类型的迁移。

首先，它既支持单个实例间的数据迁移，也支持集群到集群间的数据迁移。

其次，有的Redis切片集群（例如Codis）会使用proxy接收请求操作，Redis-shake也同样支持和proxy进行数据迁移。

另外，因为Redis-shake是阿里云团队开发的，所以，除了支持开源的Redis版本以外，Redis-shake还支持云下的Redis实例和云上的Redis实例进行迁移，可以帮助我们实现Redis服务上云的目标。

在数据迁移后，我们通常需要对比源实例和目的实例中的数据是否一致。如果有不一致的数据，我们需要把它们找出来，从目的实例中删除，或者是再次迁移这些不一致的数据。

这里，我就要再给你介绍一个数据一致性比对的工具了，就是阿里云团队开发的[Redis-full-check](#)。

Redis-full-check的工作原理很简单，就是对源实例和目的实例中的数据进行全量比对，从而完成数据校验。不过，为了降低数据校验的比对开销，Redis-full-check采用了多轮比较的方法。

在第一轮校验时，Redis-full-check会找出在源实例上的所有key，然后从源实例和目的实例中把相应的值也都查找出来，进行比对。第一次比对后，redis-full-check会把目的实例中和源实例不一致的数据，记录到sqlite数据库中。

从第二轮校验开始，Redis-full-check只比较上一轮结束后记录在数据库中的不一致的数据。

为了避免对实例的正常请求处理造成影响，Redis-full-check在每一轮比对结束后，会暂停一段时间。随着Redis-shake增量同步的进行，源实例和目的实例中的不一致数据也会逐步减少，所以，我们校验比对的轮数不用很多。

我们可以自己设置比对的轮数。具体的方法是，在运行redis-full-check命令时，把参数comparetimes的值设置为我们想要比对的轮数。

这里有个地方需要注意下，Redis-full-check提供了三种比对模式，我们可以通过comparemode参数进行设置。comparemode参数有三种取值，含义如下：

- KeyOutline，只对比key值是否相等；
- ValueOutline，只对比value值的长度是否相等；
- FullValue，对比key值、value长度、value值是否相等。

我们在应用Redis-full-check时，可以根据业务对数据一致性程度的要求，选择相应的比对模式。如果一致性要求高，就把comparemode参数设置为FullValue。

好了，最后，我再向你介绍一个用于Redis集群运维管理的工具CacheCloud。

集群管理工具CacheCloud

CacheCloud是搜狐开发的一个面向Redis运维管理的云平台，它实现了主从集群、哨兵集群和Redis Cluster的自动部署和管理，用户可以直接在平台的管理界面上进行操作。

针对常见的集群运维需求，CacheCloud提供了5个运维操作。

- 下线实例：关闭实例以及实例相关的监控任务。
- 上线实例：重新启动已下线的实例，并进行监控。
- 添加从节点：在主从集群中给主节点添加一个从节点。
- 故障切换：手动完成Redis Cluster主从节点的故障转移。
- 配置管理：用户提交配置修改的工单后，管理员进行审核，并完成配置修改。

当然，作为运维管理平台，CacheCloud除了提供运维操作以外，还提供了丰富的监控信息。

CacheCloud不仅会收集INFO命令提供的实例实时运行状态信息，进行可视化展示，而且还会把实例运行状态信息保存下来，例如内存使用情况、客户端连接数、键值对数据量。这样一来，当Redis运行发生问题时，运维人员可以查询保存的历史记录，并结合当时的运行状态信息进行分析。

如果你希望有一个统一平台，把Redis实例管理相关的任务集中托管起来，CacheCloud是一个不错的工具。

小结

这节课，我给你介绍了几种Redis的运维工具。

我们先了解了Redis的INFO命令，这个命令是监控工具的基础，监控工具都会基于INFO命令提供的信息进行二次加工。我们还学习了3种用来监控Redis实时运行状态的运维工具，分别是Redis-exporter、redis-stat和Redis Live。

关于数据迁移，我们既可以使用Redis-shake工具，也可以通过RDB文件或是AOF文件进行迁移。

在运维Redis时，刚刚讲到的多数开源工具，已经可以满足我们的不少需求了。但是，有时候，不同业务线

你基于开源工具进行二次开发或是自研，从而更好地满足业务使用需求。

每课一问

按照惯例，我给你提个小问题：你在实际应用中还使用过什么好的运维工具吗？

欢迎在留言区写下你的思考和答案，我们一起交流讨论。如果你觉得今天的内容对你有帮助，也欢迎你分享给你的朋友或同事。我们下节课见。

精选留言：

- Kaito 2020-11-09 00:10:06

老师这节课讲的工具很实用。

平时我们遇到的 Redis 变慢问题，有时觉得很难定位原因，其实是因为我们没有做好完善的监控。

Redis INFO 信息看似简单，但是这些信息记录着 Redis 运行时的各种状态数据，如果我们把这些数据采集到并监控到位，80% 的异常情况能在第一时间发现。

机器的 CPU、内存、网络、磁盘，都影响着 Redis 的性能。

监控时我们最好重点关注以下指标：

- 1、客户端相关：当前连接数、总连接数、输入缓冲大小、OPS
- 2、CPU 相关：主进程 CPU 使用率、子进程 CPU 使用率
- 3、内存相关：当前内存、峰值内存、内存碎片率
- 4、网络相关：输入、输出网络流量
- 5、持久化相关：最后一次 RDB 时间、RDB fork 耗时、最后一次 AOF rewrite 时间、AOF rewrite 耗时
- 6、key 相关：过期 key 数量、淘汰 key 数量、key 命中率
- 7、复制相关：主从节点复制偏移量、主库复制缓冲区

能够查询这些指标的当前状态是最基本的，更好的方案是，能够计算出这些指标的波动情况，然后生成动态的图表展示出来，这样当某一刻指标突增时，监控能帮我们快速捕捉到，降低问题定位的难度。

目前业界比较主流的监控系统，都会使用 Prometheus 来做，插件也很丰富，监控报警也方便集成，推荐用起来。[21赞]