



GOPS 2016
Shanghai

GOPS

全球运维大会

2016

重新定义运维

上海站

会议时间： 9月23日-9月24日

会议地点： 上海·雅悦新天地大酒店

主办单位：



指导单位：



OpenStack Swift对象存储在SSD上的优化 兼谈分布式对象存储的运维

李明宇@OStorage



目录



1

为什么最近对象存储比较火？

2

OpenStack Swift分布式对象存储

3

分布式对象存储的运维需要关注些什么？

4

SSD在OpenStack Swift中的应用

5

使用SSD与磁盘的性能对比

6

再谈EC（Erasure Coding，纠删码）



为什么最近对象存储比较火？

1. 数据量持续快速增长

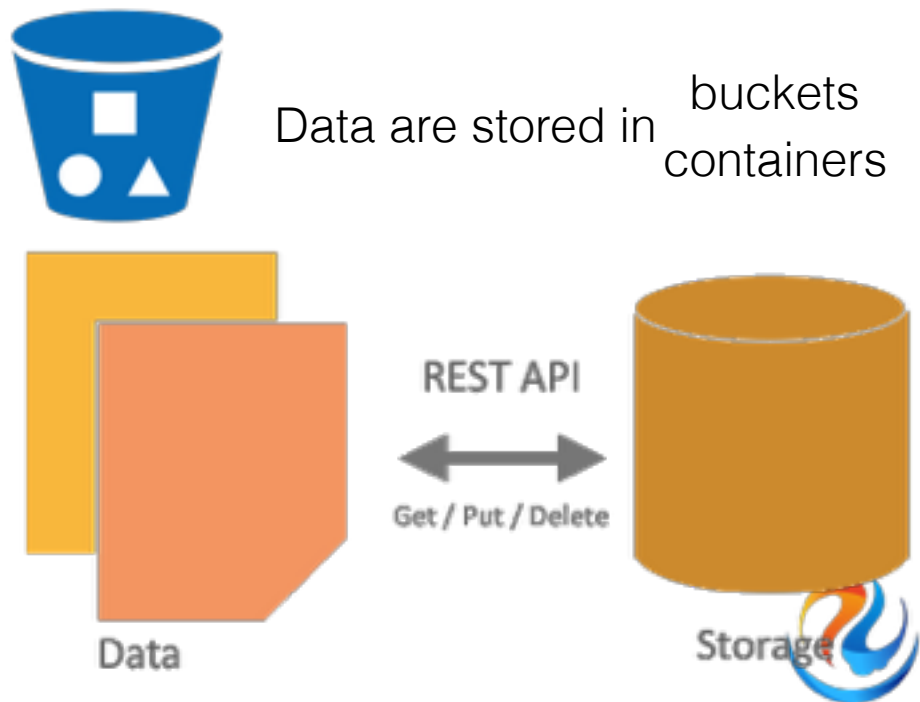
- 90%以上是非结构化数据
- 海量小文件与大体积文件共存

2. 数据访问模式变化

- 虚拟化、云化、互联网访问
- 这些数据不太冷
- 数据共享

3. 数据管理方式变化

Object Store — S3-like storage



目录

1 为什么最近对象存储比较火？



2 OpenStack Swift分布式对象存储

3 分布式对象存储的运维需要关注些什么？

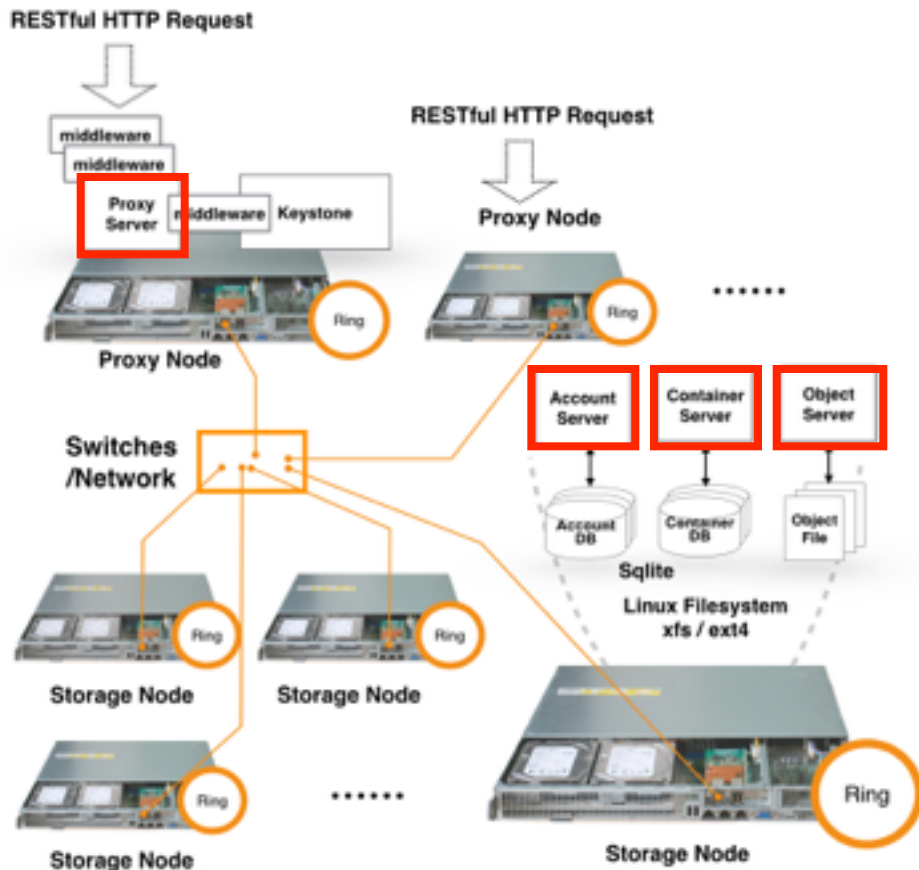
4 SSD在OpenStack Swift中的应用

5 使用SSD与磁盘的性能对比

6 再谈EC（Erasure Coding，纠删码）



OpenStack Swift分布式对象存储



Scalability — Proxy, Storage

WSGI — Extensible

Multi-Region Cluster — 双活、多活

Storage Policy — 分层、分池

Large Object Support

Hadoop Support

Erasure Coding

Object Versioning / Expiring

And Many More ...



OpenStack Swift分布式对象存储

国内典型用户



国外典型客户



目录

1 为什么最近对象存储比较火？

2 OpenStack Swift分布式对象存储

➔ 3 分布式对象存储的运维需要关注些什么？

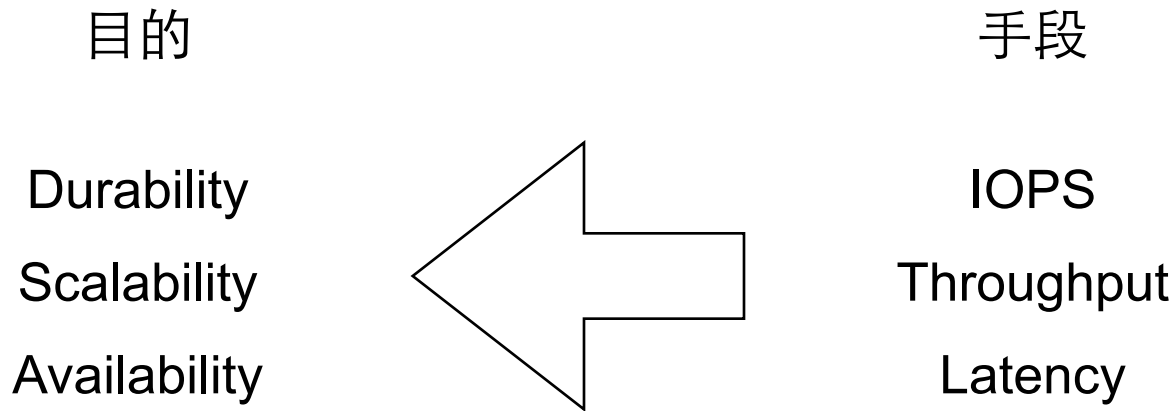
4 SSD在OpenStack Swift中的应用

5 使用SSD与磁盘的性能对比

6 再谈EC（Erasure Coding，纠删码）



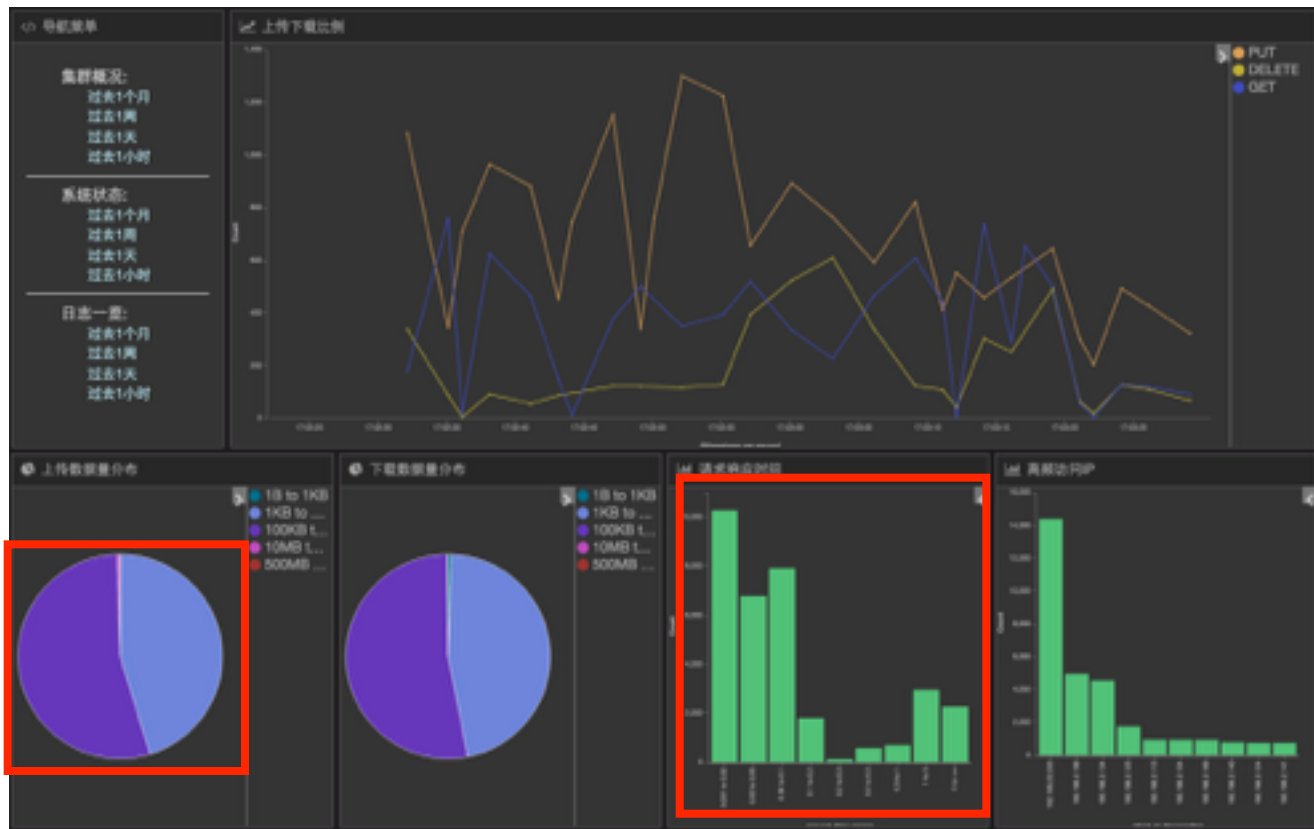
分布式对象存储的运维需要关注些什么？



新存储的诞生是为了更好的实现目的。
新存储的运维也需要新的手段。
对象存储（还/更）应当关注什么？



分布式对象存储的运维需要关注些什么？

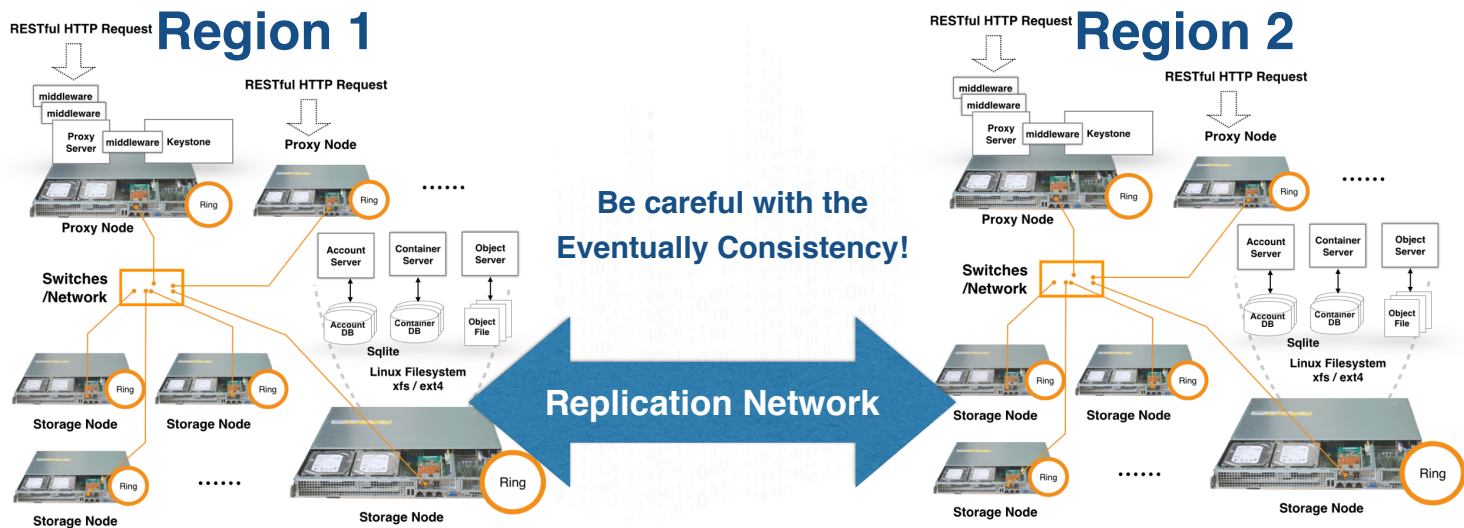


分布式对象存储的运维需要关注些什么？

最终一致性带来了跨地域的优势，

但是我们能否给“最终一致”一个可定量衡量的指标？

——数据实现完全同步的时间。



目录

1 为什么最近对象存储比较火？

2 OpenStack Swift分布式对象存储

3 分布式对象存储的运维需要关注些什么？

➔ 4 SSD在OpenStack Swift中的应用

5 使用SSD与磁盘的性能对比

6 再谈EC（Erasure Coding，纠删码）



SSD在OpenStack Swift中的应用

1. 用来存放Account和Container数据
2. 直接用于存对象数据

	无专用的SSD存放A/C数据	有专用的SSD存放A/C数据	全部使用SSD
100KB 读	1480.5	1530.3	1582
100KB 写	130.7	218.8	617.2
100MB 读		4.0	11.5
100MB 写	1.8	1.9	5.1



SSD在OpenStack Swift中的应用

结论：

1. 优化Account和Container数据存储

- 1) 对于中等规模的集群，配备一定数量的SSD盘，例如8~16块SSD盘来存放Account和Container数据；
- 2) 对于大规模集群，可配备专门的A/C节点，可以考虑与Proxy服务一起部署，称为PAC节点。

2. 对于典型的对象存储应用场景（百TB级容量），现阶段使用SSD加速提高性能的必要性不足，成本较高。



SSD在OpenStack Swift中的应用

3. 用作Cache/分层存储

可看做是前两者的折中方案，结合了二者的优势，但是带来了新的问题：

- 如何移动数据——Container Sync
- 分层策略和分层算法？一些看起来很美算法在实际应用中很难满足应用的期望。
- 在Swift中发挥SSD性能一定要使用负载均衡，否则网络会成为瓶颈，加入负载均衡的Proxy节点数量计算公式：

$$n \geq \text{上行速率} / \text{内部网络速率} * \text{副本数量}$$



目录

1 为什么最近对象存储比较火？

2 OpenStack Swift分布式对象存储

3 分布式对象存储的运维需要关注些什么？

4 SSD在OpenStack Swift中的应用

5 使用SSD与磁盘的性能对比

 **6** 再谈EC（Erasure Coding，纠删码）



再谈EC（Erasure Coding，纠删码）

1. 关于EC的两个误解：

1) EC的计算开销会降低数据读写性能

使用开源编码器，RS(6,8)编码，单节点计算速度达到数GB/s

Swift实测单Proxy节点吞吐率可达700MB/s甚至更高。

2) EC会增加网络开销

相对于副本方案，写入数据时网络流量明显降低。

需接入负载均衡器的节点更少。



再谈EC（Erasure Coding，纠删码）

2. 关于EC的三个真正要注意的问题：

- 1) EC会增加盘的随机IO（SSD优于磁盘）
- 2) 需要更及时准确地判断故障和更换失效设备（运维成本增加）
- 3) 小规模集群中可用区的划分会变得困难，且不支持跨地域。





Thanks

高效运维社区
开放运维联盟

荣誉出品



想第一时间看到高效运维公众号的好文章么？

请打开高效运维公众号，点击右上角小人，并如右侧所示设置即可：



GOPS2016 全球运维大会更多精彩

GOPS2016 全球运维大会·北京站

2016 年12月16日-17日
北京国际会议中心

