****

**Ceph Rados**

Contents

[**一.简介** 4](#_Toc528159182)

[**二．集群管理** 4](#_Toc528159183)

[2.1 集群图 4](#_Toc528159184)

[2.2数据放置 4](#_Toc528159185)

[2.3 设备状态 5](#_Toc528159186)

[2.4 Map同步 5](#_Toc528159187)

[**三.智能存储设备** 6](#_Toc528159188)

[3.1 数据复制 6](#_Toc528159189)

[3.2 数据一致性 7](#_Toc528159190)

[3.3 失效校验 7](#_Toc528159191)

[3.4 数据迁移和恢复 7](#_Toc528159192)

[**四．Monitor** 8](#_Toc528159193)

[4.1 Paxos Service 8](#_Toc528159194)

[4.2 工作负载和扩展性 8](#_Toc528159195)

[**五．参考资料** 8](#_Toc528159196)

[**六．附录** 8](#_Toc528159197)

[6.1 API 8](#_Toc528159198)

***\* 版本修订记录 \****

|  |  |  |
| --- | --- | --- |
| ***版本号*** | ***修订时间*** | ***修订内容*** |
| *v1.0* | *2018-08-13* | *初版修订* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

***\* Release Copyleft©free \****

**一.简介**

RADOS:Reliable,Autonomic Distributed Object Store.即可靠、自动化的分布式对象存储。是一个提供高可用、高可靠性、高性能分布式存储的架构，基于RADOS架构原理实现的Ceph系统，被称为“下一代存储“；

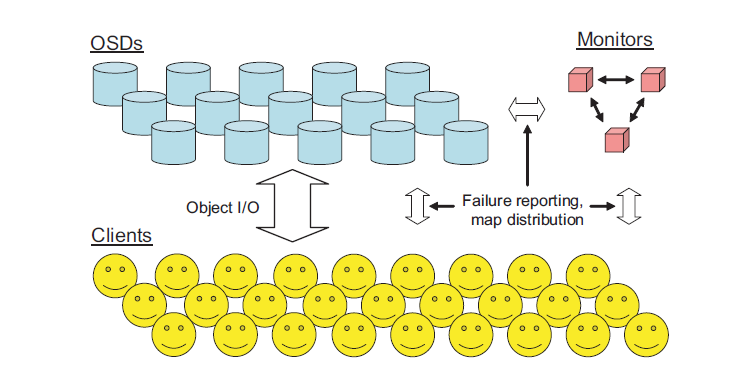


Figure 1 RADOWS架构图

如上图RADOS架构图所示，一个RADOS系统中主要包括两个主要部分，对象数据存储集群可以包含大量的OSD，主要用于监控OSD集群的小规模集群Monitors，

**二．集群管理**

2.1 集群图

Monitors管理集群的唯一方式是操作cluster map(集群图).它指定了集群中的OSD间的关系、OSD的状态以及确定了数据如何分布，集群图会被复制到OSD以及客户端中，每次OSD的状态发生改变或其他的事件导致数据分布发生变化时，集群图会发生改变，此时集群图的版本号(epoch)会增加，集群中的集群图的同步使用了增量惰性同步的方式，这主要时为了减轻Monitors的负载。同步的增量数据为两个相邻版本集群图的变化以及相差多个版本的数据变量的打包，同步的时机为周期性的Monitor和Monitor的信息交互、OSD和Monitor通信或OSD和OSD之间通信时会较集群图版本的信息，集群图版本不一致时会触发同步；

2.2数据放置

RADOS中的数据对象分布是使用的伪随机的策略，当集群中增加新的存储节点时，为了保持集群中数据的均衡性，会有随机的部分数据迁移到新的存储节点上，这使得所有的设备的负载基本上保持均衡，另外也需要保证集群数据分布的稳定性，不能在添加或删除存储节点后导致所有的数据迁移。RADOWS系统中数据的存放分为了两个计算过程：

(1) 数据对象首先被映射到PG(Placement Group)中,PG时对象集合的逻辑单位，每个PG的数据上的数据被分配到相对固定的OSD上，每个OSD上的数据一致的。每个对象放置到那个PG是有对象名称的哈希值决定的：

*pgid = (r, hash(o) & m)*

其中：

* r: 表示复制集群中对象的复制因子；
* o: 表示对象的名称，在Ceph集群中，该值为文件名和条带序号的组合；
* m: 表示PG的数量，理论为2\*\*k -1,Ceph中一般每个OSD上平均分配100个PG；

(2) RADOS系统使用CRUSH算法将PG映射到r(复制因子)个OSD，CRUSH是一种稳定的分布式算法，能够根据OSD的容量或性能进行相对伪随机的PG映射，和HASH算法具有相似的功能；

以上可以看出基于计算的数据存储方式使得集群中取出了笨重、中心化的查询节点，增加了集群的扩展能力；

2.3 设备状态

OSD会周期性向Peering的OSD发出心跳侦测周围的OSD是否发生了状态的变化，并在状态发生变化的情况下告知Monitor,Monitor收到报告后重新生成新的集群图。在RADOS集群中，OSD的状态主要有一下几类：

* in: 代表该OSD处于集群中，可以被PG映射；
* out: 代表该OSD处于集群外，是不会被PG映射的；
* up: 代表OSD网络是可达的；
* down: 代表OSD进程可能关闭了；

集群中几种特殊的混合状态：

* down and in: 代表该OSD已经关闭了，但是该OSD上的数据并未被重新映射到其他的OSD上,Ceph中该模式为降级模式(degreed mode)；
* up and out: 该状态代表OSD是可达的，但是该OSD当前并不会被PG映射，并且该OSD上的数据并不会发生迁移，一般在OSD重启或网络间接性故障的时候OSD会出现该状态；

2.4 Map同步

由于RADOWS集群中可能包含了成千上万的OSD，并且集群图的变化是家常便饭，导致将变化的集群图在集群中进行广播是不切实际的，这会导致集群中网络流量的大量消耗和网络带宽的无实际占用。集群图在OSD和OSD通信或客户端和OSD通信时才显得尤为重要，可以通过比较他们的交互的消息来惰性的更新节点中的进群图，这有效的将Monitor分发任务负载转义到了各个OSD上。

每个OSD都会保存最近的集群图更新的增量数据，并用集群版本号epoch打上标签。当两个通信的OSD的集群图版本不一致时，会将更新的增量的数据进行同步，另外，集群中失效校验的心跳消息保证了集群图快速传输——在有n个OSD的时间复杂度为O(log n)；

**三.智能存储设备**

数据分布的信息主要使用了集群图，这使得在集群中的OSDs可以分布式实施数据冗余策略、失败校验和失效恢复。

客户端在和RADOWS系统交互的时候，只需要将单个的写操作提交到第一个主OSD(*primary OSD*),它会负责所有副本数据的更新和一致性，这将对象复制相关的带宽转移到了存储集群的内部网络，简化了客户端的配置；结合对象版本和PG的短期日志，可以在节点间断性失效的情况下进行对象数据的快速恢复；

3.1 数据复制

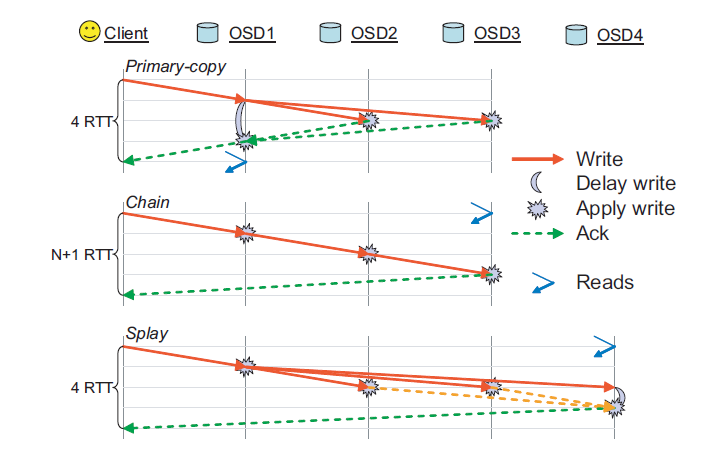


Figure 2 对象复制图

如上图所示，RADOWS系统可以提供三种类型的数据复制：

* Primary-copy: 客户端和主OSD交互后，主OSD将数据并行的更行到其他副本OSD，读写进程都在主OSD上；主OSD接收到所有副本OSD的写完成信号后，再向客户端返回写完成消息；
* Chain: 对象数据被先后顺序的写入各个OSD；
* Splay:该种数据复制方式结合了Primary-copy的并行更新和Chain复制的读写分离；

3.2 数据一致性

所有的RADOWS消息，包括客户端产生的或来自其他OSDs的，都会被打上map版本的标记，这确保了以完全一致的方式进行所有的写入、读取和更新操作；如果一个客户端由于获取了旧版的集群图导致将数据发送给错误的OSD，该OSD会比较客户端的集群图版本并将集群图版本的增量数据返回给客户端，这样它会更新后的集群图将数据重新定向到其他的OSD上；

3.3 失效校验

Peering的存储节点间将会周期性的交换心跳信息来确保检测的设备的失效，在TCP套接字失败并尝试有限次的重连尝试后会见OSD的无法联通的状态发送给Monitor,Monitor收到后会将其标记为down状态。

3.4 数据迁移和恢复

当集群图更新或PG和OSD映射发生改变时，RADOWS数据会发生迁移或失效恢复，集群图发生改变可有是由设备损坏、设备恢复、集群扩张或集群收缩、以及更换新的CRUSH分布式复制策略导致所有对象重分布引起。

在RADOWS系统中，使用了一种稳定的peering算法来为PG的内容建立一致性视图和、复制数据以及恢复数据的正常分布，这种策略依赖于OSD主动复制PG日志和PG当前的内容纪录这个接本设计前提，甚至对象的数据还有可能产生局部缺失，因此，尽管恢复过程是缓慢的，并且对象有时候处于降级模式，但是PG的元数据是被安全的保存的，这简化了恢复算法的设计并允许系统可靠的检测的数据的丢失

* Peering:

当OSD收到集群图更新时，它会遍历所有的增量更新，之后它会检查并有可能调整PG的状态，任何本地存储的PG的活动OSD列表的变化都必须re-peer.和复制一样，peering过程正对集群中的各个PG来说都是独立的；

Peering过程由主OSD发起，对于一个PG中不是主OSD的OSD,都会发送通知消息给主OSD，这个消息包括了：本地存储的PG的基本状态、最近的更新、一定范围内的PG日志、以及其最近的集群图epoch;

* Recovery

Declustered replication一个优势是可以进行并行的故障恢复；在Peering过程中，通过非主OSD的通知的消息就知道了replica缺失的对象数据，它可以将任何对象推送给Replica OSD;

**四．Monitor**

Monitors是一个小的集群，主要维护集群图的主副本，并周期性的更新配置或OSD状态的改变，Monitor集群是基于Paxos算法的，旨在支持可用性和更新延迟的一致性和持久性。需要注意的是,集群中需要有大多数Monitor是可用的；

4.1 Paxos Service

4.2 工作负载和扩展性

**五．参考资料**

【1】RADOS: A Scalable, Reliable Storage Service for Petabyte-scale Storage Clusters

**六．附录**

6.1 API