ceph

Ceph中国社区 | IT大咖说 知识共享平台

CEPHALOCON APAC 2018

THE FUTURE OF STORAGE

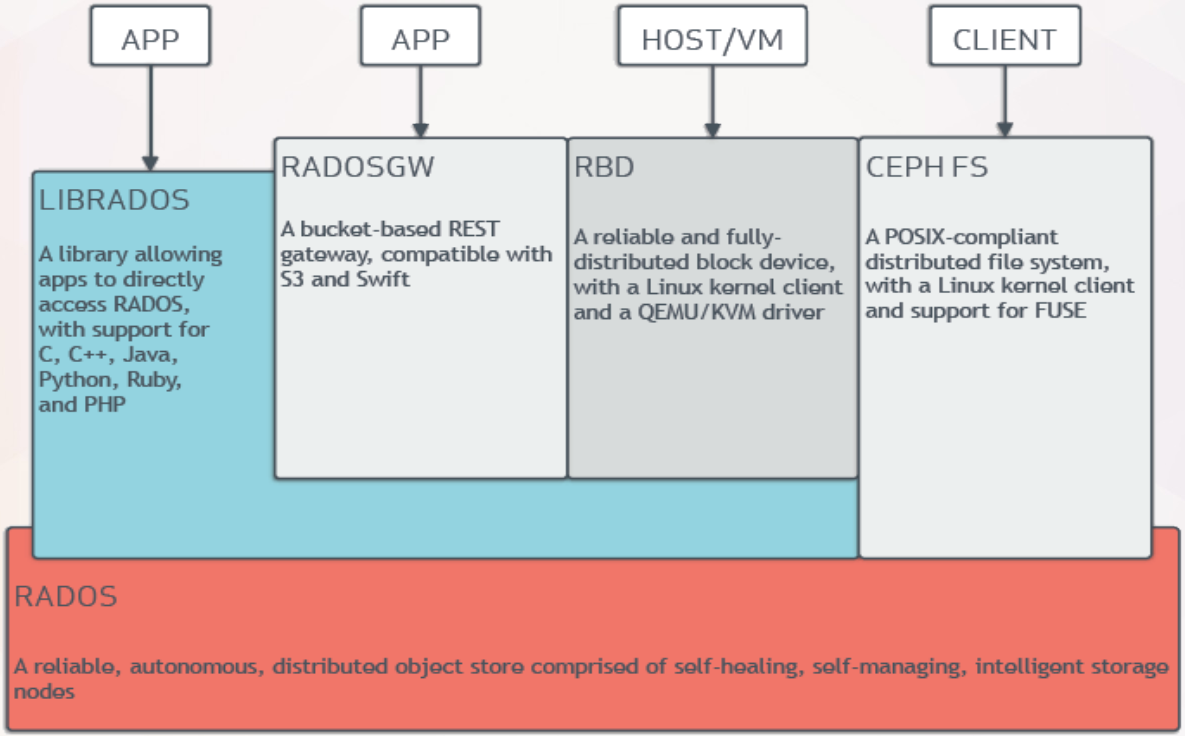
22-23 March 2018 | BEIJING

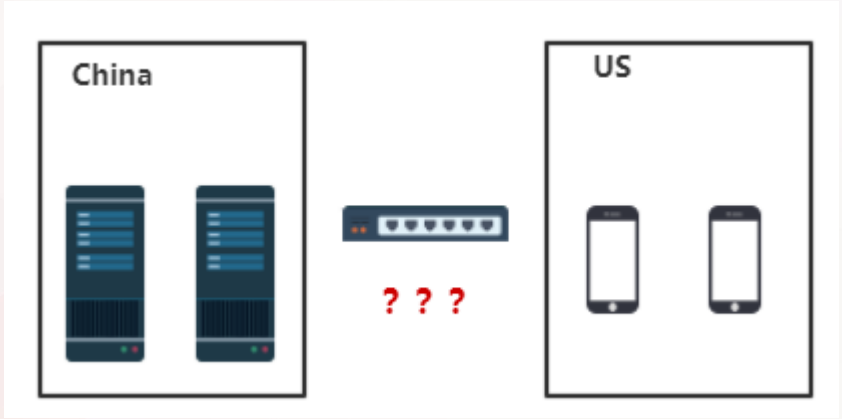
基于Ceph对象存储的混合云

腾讯云 吕珊春

2018-03-22

- 背景介绍
- 发展现状
- 核心机制
- 后续工作


APP
APP
HOST/VM
CLIENT
LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP
RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift
RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver
CEPH FS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE
RADOS
A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes

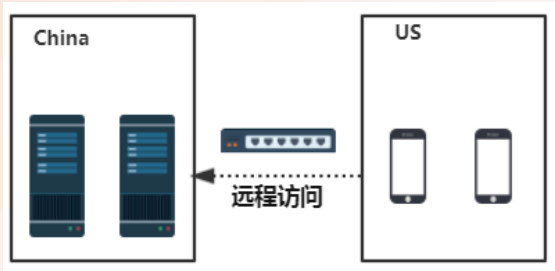


远距离数据访问如何支持？

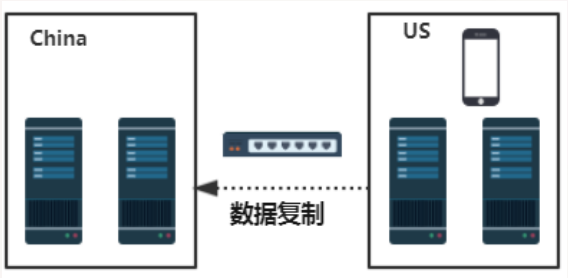
跨地域直接访问

自建数据中心

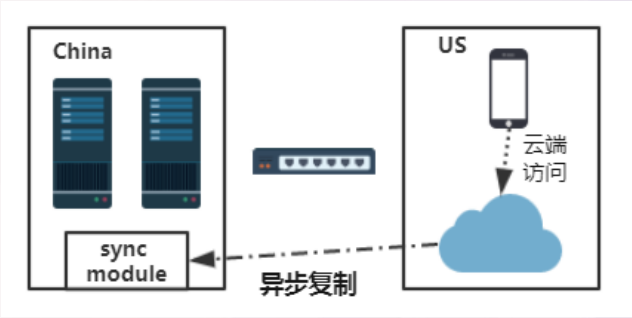
混合云



延迟太高！



成本太高！

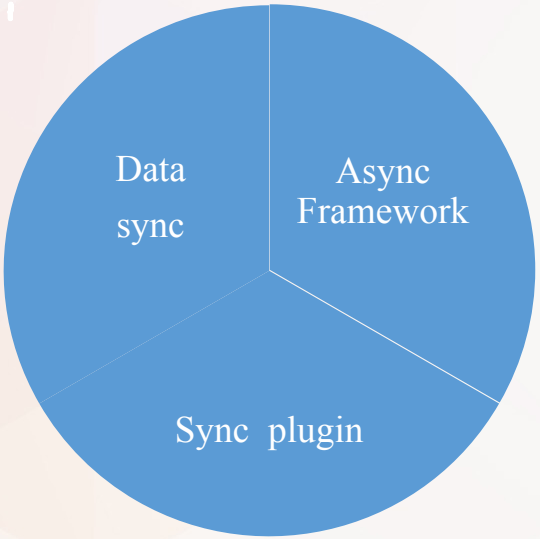


综合成本及快捷性

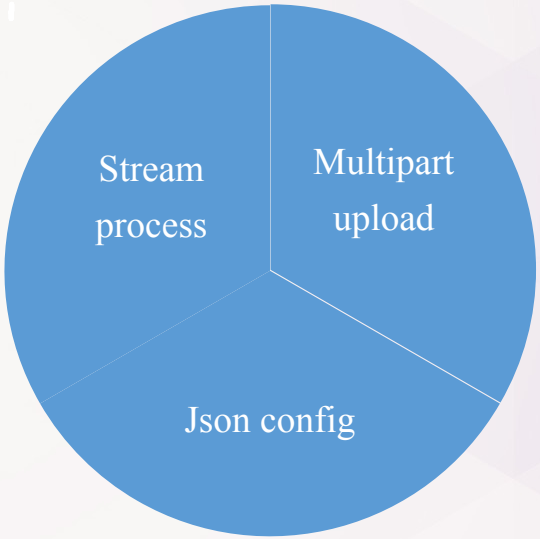
- 背景介绍
- 发展现状
- 核心机制
- 后续工作

- 混合云的好处
  - ✓ 对ceph生态的良好补充
  - ✓ 借助公有云的成本及容量优势
  - ✓ 更加灵活的资源和服务编排
- 发展历史

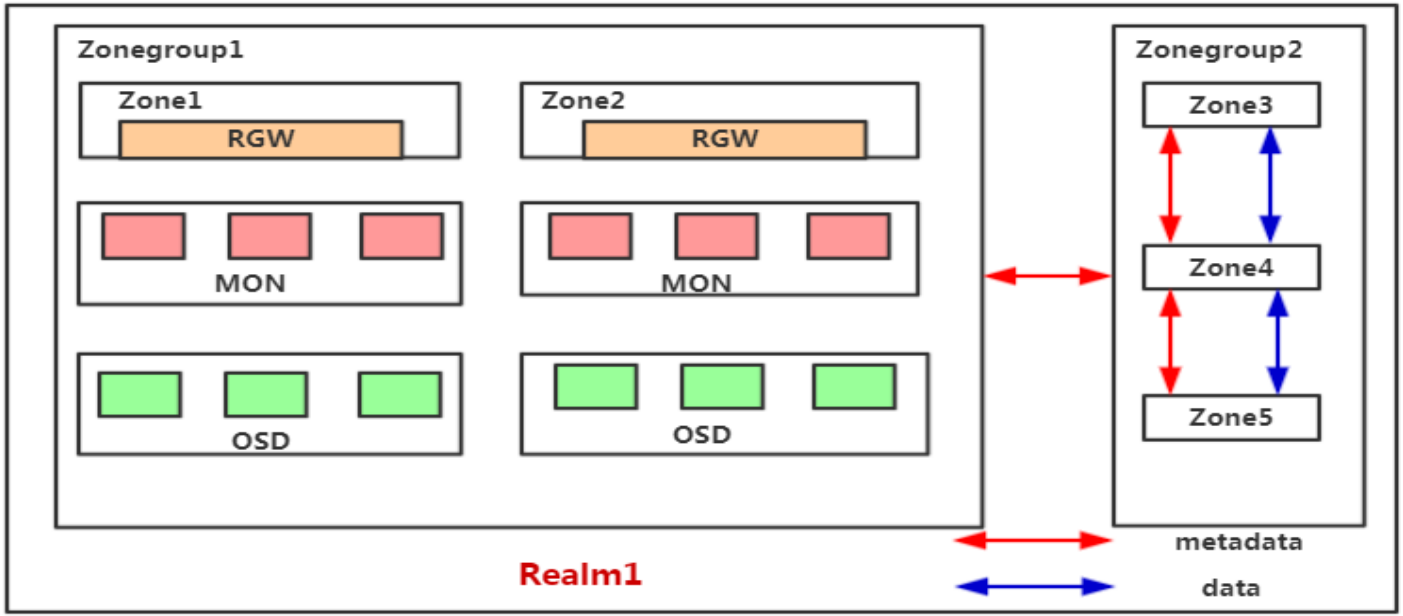
- 背景介绍
- 发展现状
- 核心机制
- 后续工作



Multi-Site



Cloud Sync



- Zone：存在于一个独立的Ceph集群，由一组rgw提供服务，对应一组后台的pool
- Zonegroup：包含至少一个Zone，Zone之间同步数据和元数据
- Realm:一个独立的命名空间，包含至少一个Zonegroup，Zonegroup之间同步元数据

Init

- read remote datalog shard info
- write sync status in local zone(datalog.sync-status.shard*.X)

BuildFull SyncMap

- list bucket instance
- Write sync status in local zone(bucket.sync-status.*{bucket-shard-id})

DataSync

- full sync(fetch remote datalog and write sync_marker locally)
- incremental sync(fetch remote with sync_marker)

Data

```
struct rgw_data_change_log_entry {
  string log_id;
  real_time log_timestamp;
  rgw_data_change entry;
};

struct rgw_data_change {
  DataLogEntityType entity_type;
  string key; //对应的bucket shard名字，有bilog shard与之对应
  real_time timestamp;
};
```

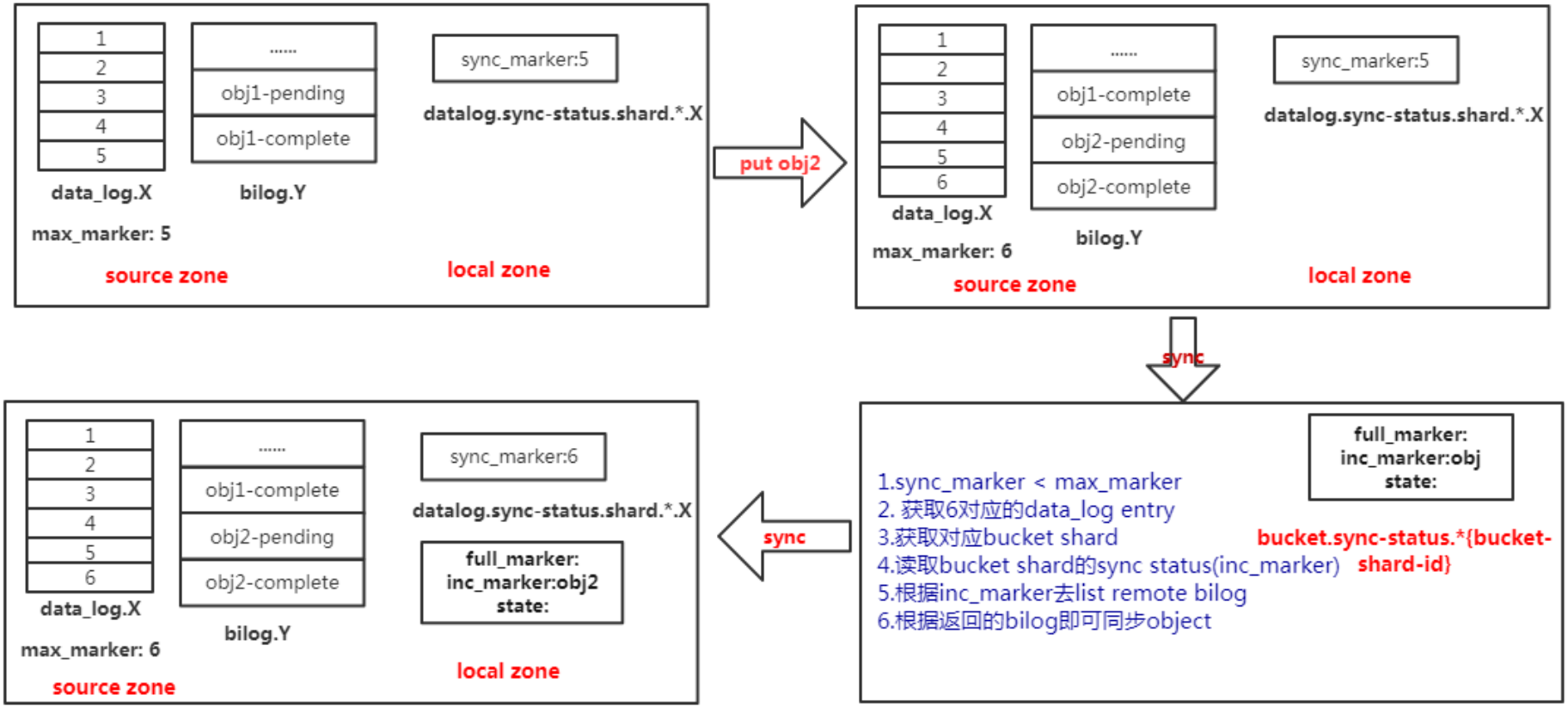
每个bucket shard有一个

g 分片对应的

有数据未同步，则消费

hard，消费bucket shard

对应的bilog则可进行数据同步

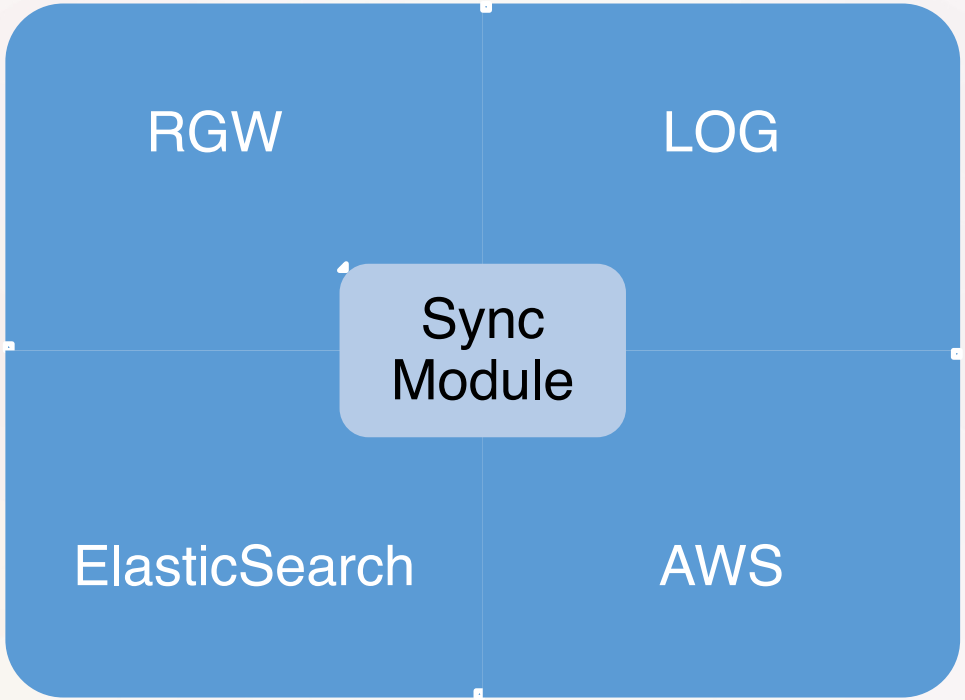

1
2
3
4
5
data_log.X
max_marker: 5
.....
obj1-pending
obj1-complete
bilog.Y
sync_marker:5
datalog.sync-status.shard.*.X
source zone
local zone
put obj2
1
2
3
4
5
6
data_log.X
max_marker: 6
.....
obj1-complete
obj2-pending
obj2-complete
bilog.Y
sync_marker:5
datalog.sync-status.shard.*.X
source zone
local zone
sync
full_marker:
inc_marker:obj
state:
1.sync_marker < max_marker
2. 获取6对应的data_log entry
3.获取对应bucket shard
4.读取bucket shard的sync status(inc_marker)
5.根据inc_marker去list remote bilog
6.根据返回的bilog即可同步object
bucket.sync-status.*{bucket-shard-id}
sync
1
2
3
4
5
6
data_log.X
max_marker: 6
.....
obj1-complete
obj2-pending
obj2-complete
bilog.Y
sync_marker:6
datalog.sync-status.shard.*.X
full_marker:
inc_marker:obj2
state:
source zone
local zone

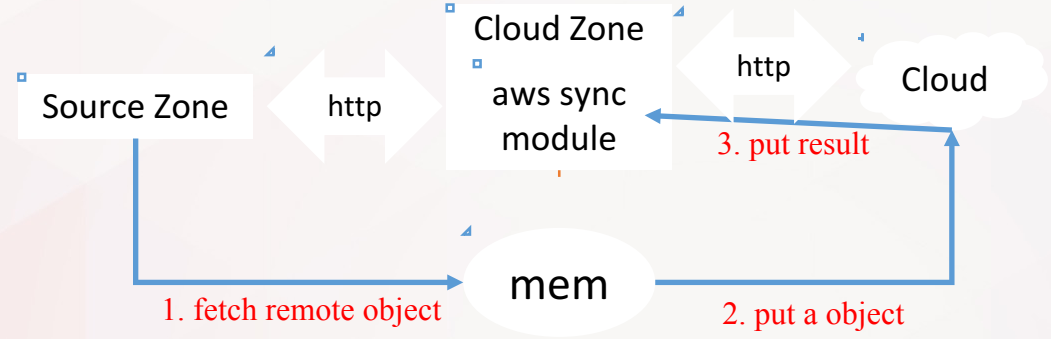
ceph

- boost::asio::coroutine
- stackless coroutine
- RGWCoroutine
- RGWConsumerCR(yield)
- Concurrent Coroutine(spawn)

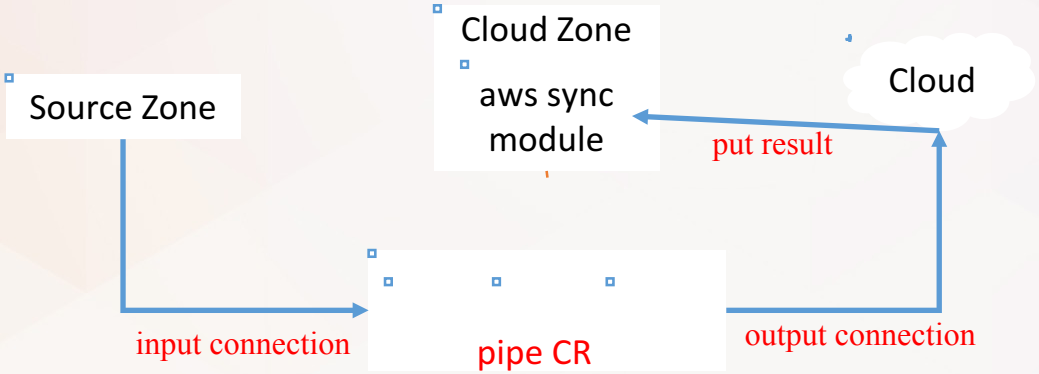
```
/* consumer */
int operate() {
  reenter(this) {
    for (;;) {
      if (!has_product() && going_down) {
        break;
      }
      yield wait_for_product();
      yield {
        string entry;
        while (consume(&entry)) {
... // do something with the new entry
        }
      }
      if (get_ret_status() < 0) {
        return set state(RGWCoroutine Error);
```

```
yield call(new RGWSimpleRadosLockCR(...));
yield {
  for (int i = 0; i < (int)status.num_shards; i++) {
    rgw_meta_sync_marker marker;
    spawn(new RGWSimpleRadosWriteCR<rgw_meta_sync_marker>(..., true);
  }
}
/* unlock */
yield call(new RGWSimpleRadosUnlockCR(...));
```


RGW
LOG
Sync Module
ElasticSearch
AWS

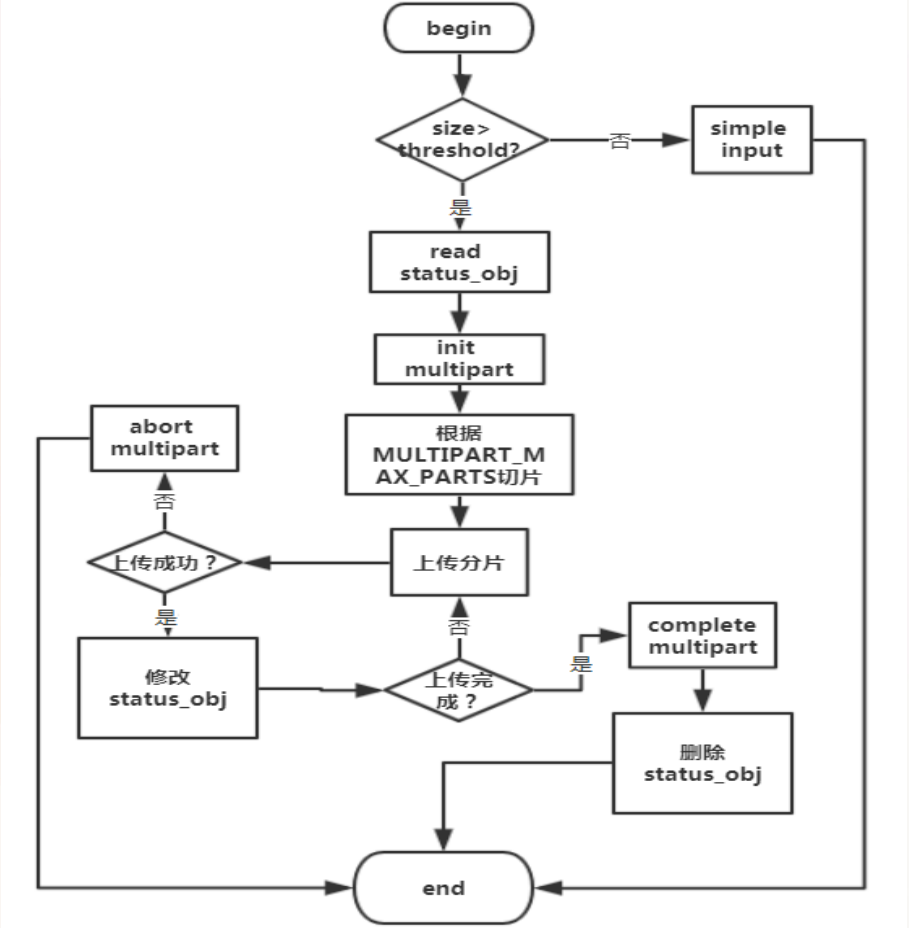


large object?



Streaming process

pipe CR: keep 2 connection, get and put are parallel


begin
size> threshold?
否
simple input
是
read status_obj
init multipart
根据 MULTIPART_MAX_PARTS切片
上传分片
否
上传完成？
是
complete multipart
删除 status_obj
上传成功？
否
abort multipart
是
修改 status_obj
end

```
json configuration definition:

{
  default = {
    "connection": {
        "access_key": <access>,
        "secret": <secret>,
        "endpoint": <endpoint>,
        "host_style" <path | virtual>,
        "acl_mappings": [    # list of source uids and how they map into destination uids in the dest objects acls
        {
          "type" : <id | email | uri>,   #  optional, default is id
          "source_id": <id>,
          "dest_id": <id>
        }
```

```
radosgw-admin  zone create --rgw-zonegroup=test-sync-cos-group --rgw-zone=zone1 --endpoints=127.0.0.1:8005 \
--access-key=accesskey  --secret=secretkey --tier-type=aws \
--tier-config="connection=\"{\"access_key\": \"accesskey\", \"secret\": \"secretkey\",
\"endpoint\":\"127.0.0.1:8006\", \"host_style\": \"path\"},target_path=\"rgw-${zonegroup}-${sid}/${bucket}\""
```

```
        "access_key": <access>,
        "secret": <secret>,
        "endpoint": <endpoint>,
        "acl_mappings": [    # optional, overrides default
        {
          "source_id": <id>,
          "dest_id": <id>
        } ... ]
      } ... ],
  "targets": [
     {
     "source_bucket": <source>, # can specify either specific bucket name (foo), or prefix (foo*)
     "target_path": <dest>,   # (override default)
     "connection_id": <connection_id> # optional, if empty references default connection
      } ... ],
}
```

ceph

Ceph中国社区

IT大咖说 知识共享平台

- 背景介绍
- 发展现状
- 核心机制
- 后续工作

- 同步状态优化，错误信息汇报至MON
- 数据的反向同步
- 支持更多的公有云平台
- 利用RGW支持不同云平台之间的数据同步

Thank You !