GNU/Linux-Ceph

分布式文件系统-Ceph

GNU/Linux-Ceph

Ceph

Ceph 是一种为优秀的性能、可靠性和可扩展性而设计的统一的、分布式文件系统。

GNU/Linux-Ceph

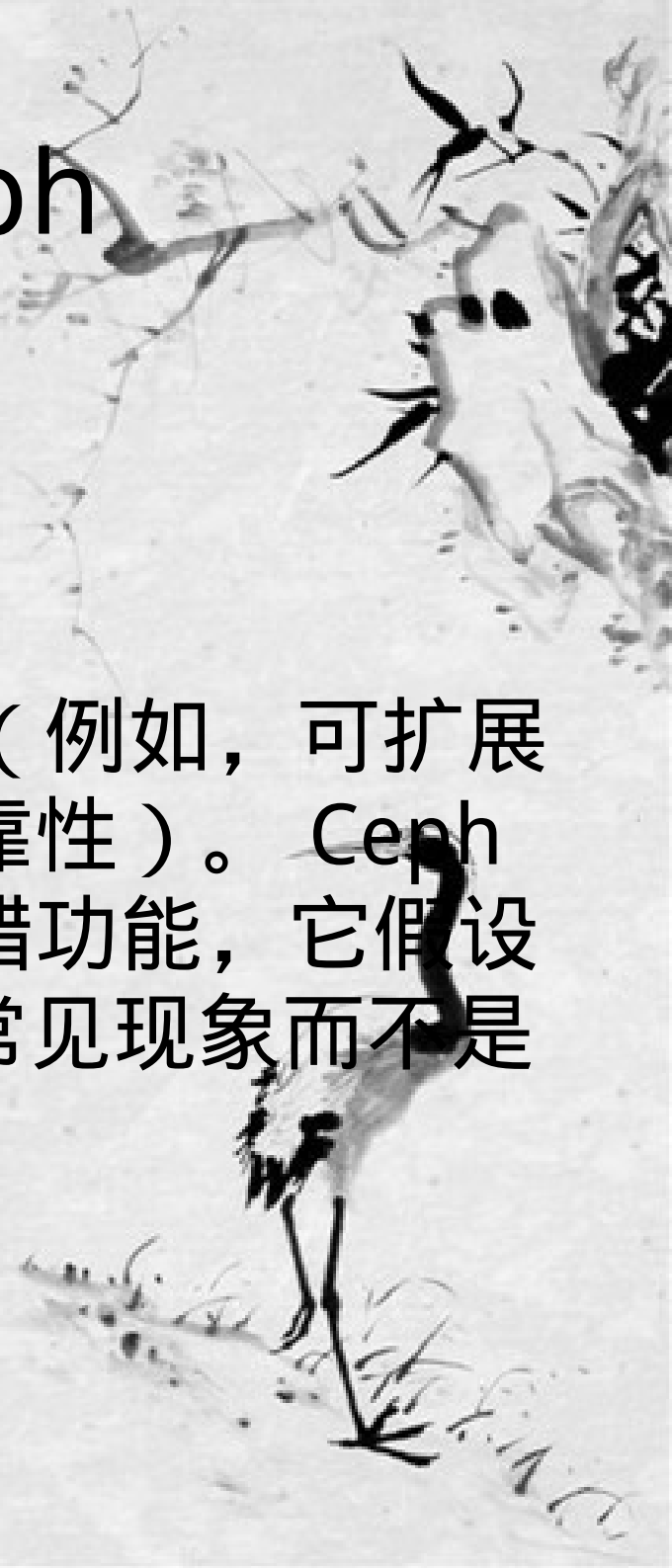
Ceph

简单定义为以下 3 项：
1. 可轻松扩展到数 PB 容量
2. 支持多种工作负载的高性能（每秒输入 / 输出操作 [IOPS] 和带宽）
3. 高可靠性

GNU/Linux-Ceph

Ceph

但是，这些目标之间会互相竞争（例如，可扩展性会降低或者抑制性能或者影响可靠性）。 Ceph 的设计还包括保护单一点故障的容错功能，它假设大规模（ PB 级存储）存储故障是常见现象而不是例外情况。



GNU/Linux-Ceph

Ceph

它的设计并没有假设某种特殊工作负载，但包括了适应变化的工作负载，并提供最佳性能的能力。它利用 POSIX 的兼容性完成所有这些任务，允许它对当前依赖 POSIX 语义（通过以 Ceph 为目标的改进）的应用进行透明的部署。

GNU/Linux-Ceph

Ceph

Ceph 生态系统架构可以划分为四部分：

1. Clients： 客户端（数据用户）

2. cmds： Metadata server cluster，元数据服务器（缓存和同步分布式元数据）

GNU/Linux-Ceph

Ceph

3. cosd： Object storage cluster，对象存储集群（将数据和元数据作为对象存储，执行其他关键职能）

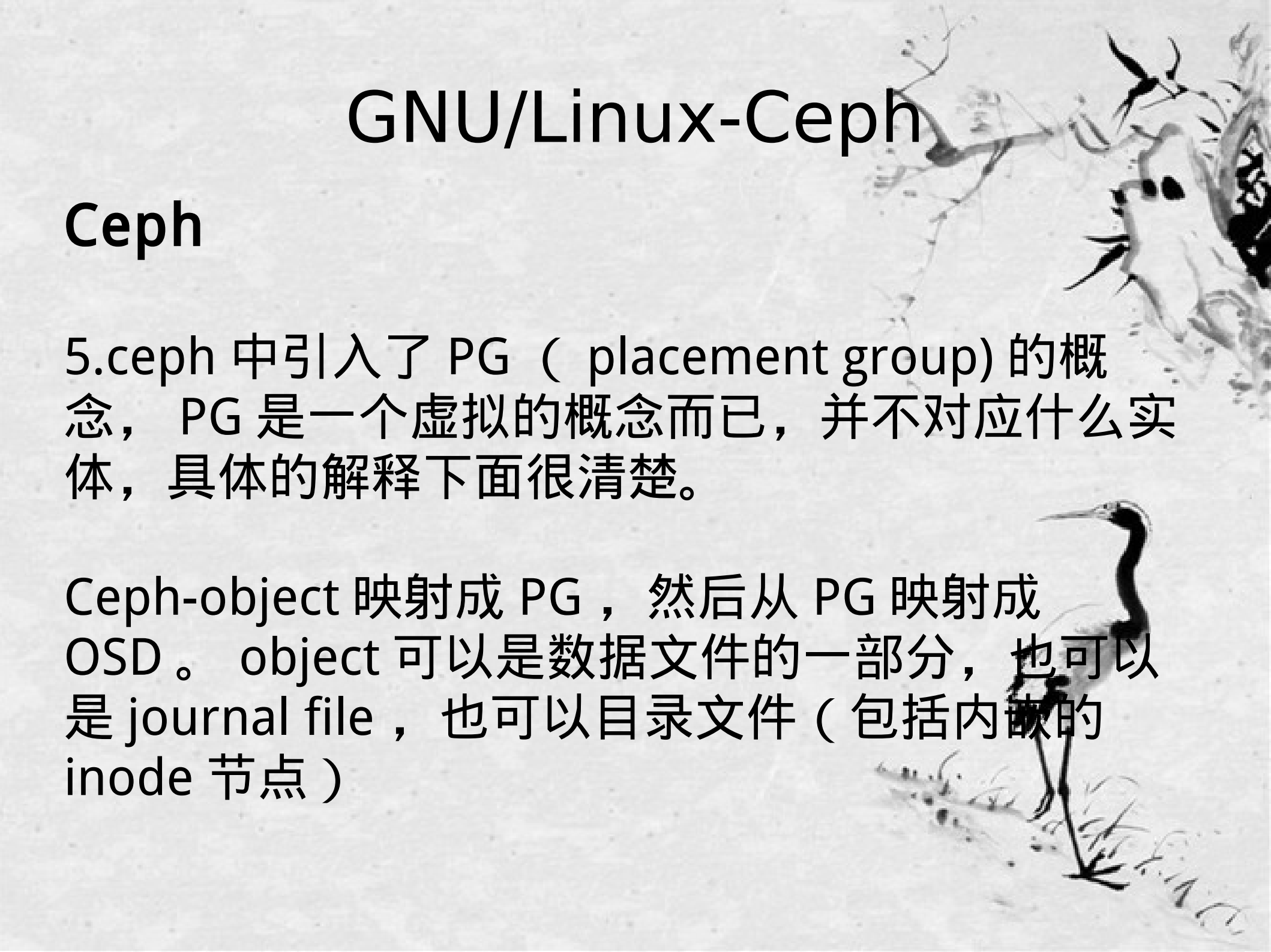
4. cmon： Cluster monitors，集群监视器（执行监视功能）

GNU/Linux-Ceph

Ceph

5.ceph 中引入了 PG（ placement group) 的概念， PG 是一个虚拟的概念而已，并不对应什么实体，具体的解释下面很清楚。

Ceph-object 映射成 PG ，然后从 PG 映射成 OSD 。 object 可以是数据文件的一部分，也可以是 journal file ，也可以目录文件（包括内嵌的 inode 节点）



GNU/Linux-Ceph

Ceph

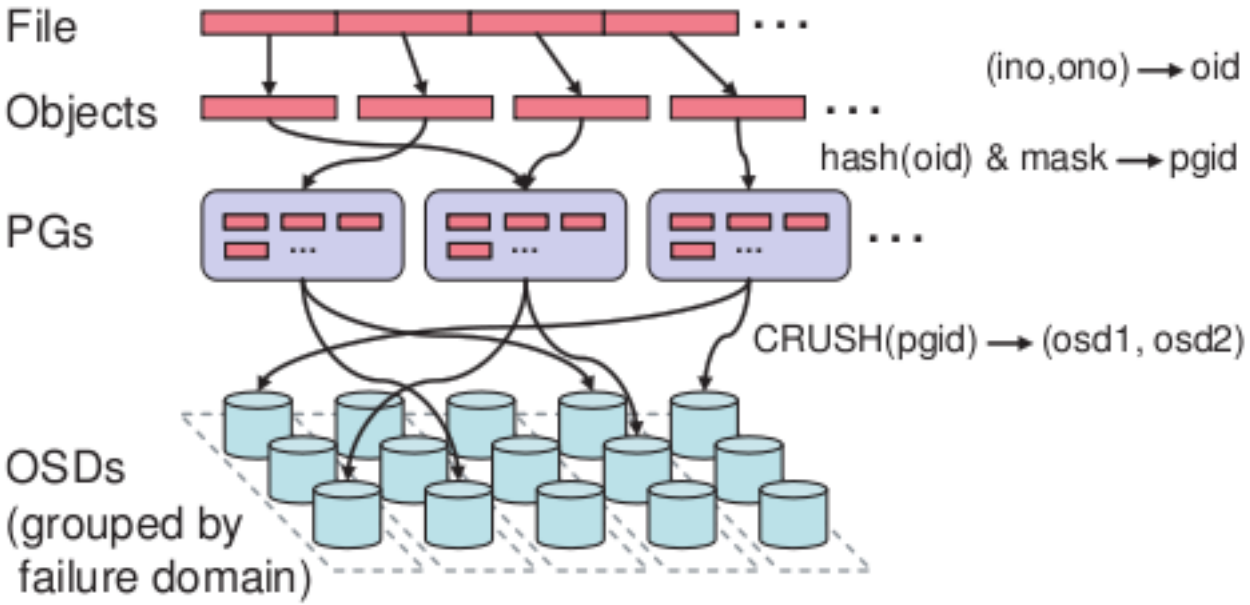


Figure 3: Files are striped across many objects, grouped into *placement groups* (PGs), and distributed to OSDs via CRUSH, a specialized replica placement function.

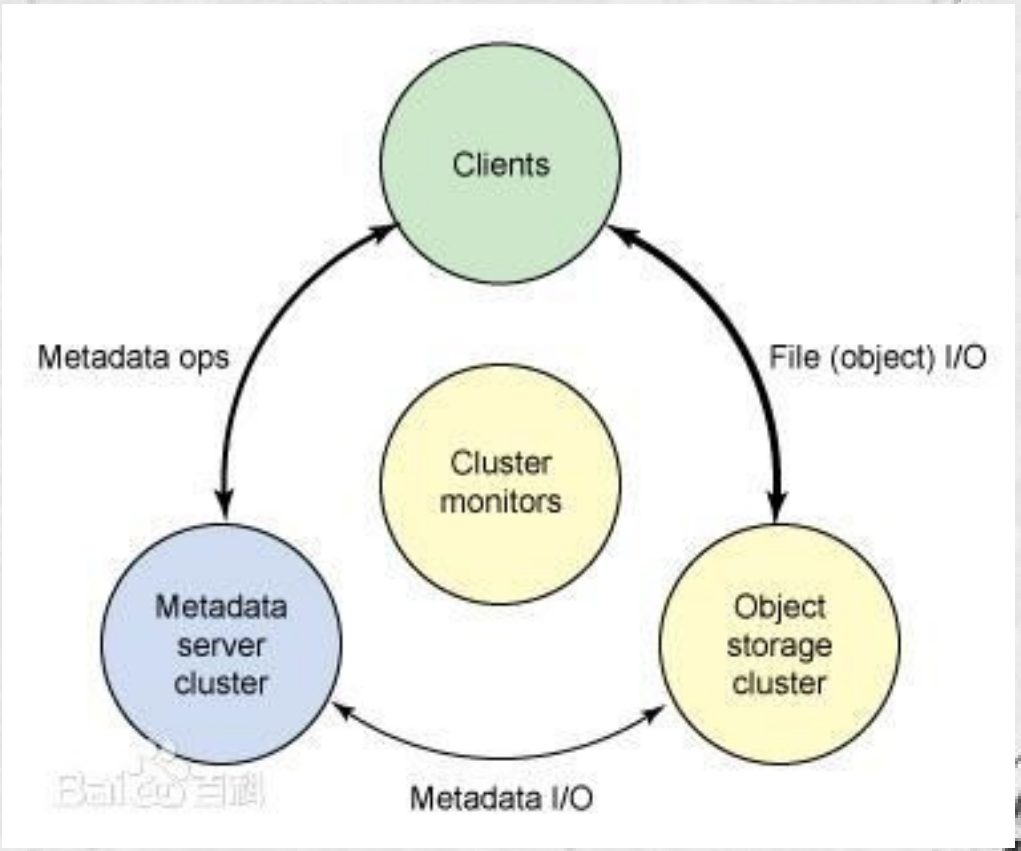
GNU/Linux-Ceph

Ceph

如果有一个 OSD，默认有 192 个 PG。如果有 2 各 OSD 则默认有 2*192=384 个 PG.

GNU/Linux-Ceph

Ceph 生态系统的框架



GNU/Linux-Ceph

Ceph 未来发展

作为分布式文件系统，其能够在维护 POSIX 兼容性的同时加入了复制和容错功能。从 2010 年 3 月底，您可以在 Linux 内核（从 2.6.34 版开始）中找到 Ceph 的身影，作为 Linux 的文件系统备选之一， Ceph.ko 已经集成入 Linux 内核之中。虽然目前 Ceph 可能还不适用于生产环境，但它对测试目的还是非常有用的。

GNU/Linux-Ceph

Ceph 未来发展

Ceph 不仅仅是一个文件系统，还是一个有企业级功能的对象存储生态环境。

现在， Ceph 已经被集成在主线 Linux 内核中，但只是被标识为实验性的。在这种状态下的文件系统对测试是有用的，但是对生产环境没有做好准备。但是考虑到 Ceph 加入到 Linux 内核的行列，不久的将来，它应该就能用于解决海量存储的需要了。

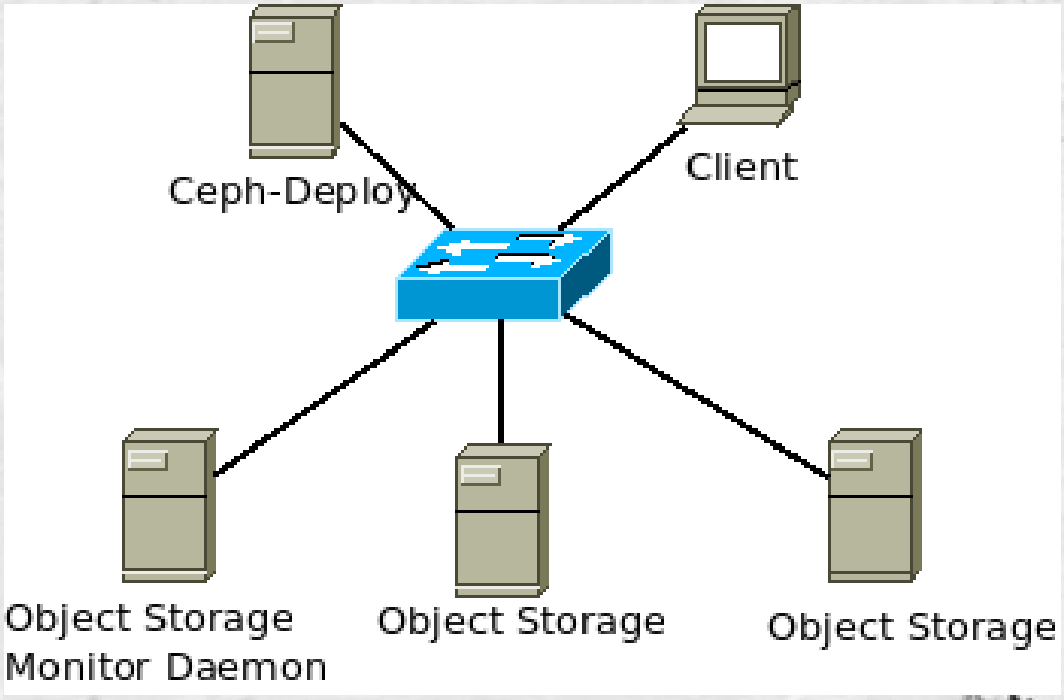
GNU/Linux-Ceph

Ceph 未来发展

一些开源的云计算项目已经开始支持 Ceph，事实上 Ceph 是目前 OpenStack 生态系统中呼声最高的开源存储解决方案。这些项目都支持通过 libvirt 调用 Ceph 作为块设备进行读写访问。

GNU/Linux-Ceph

Ceph 试验拓扑



GNU/Linux-Ceph

实现 Ceph

1. 在所有节点创建一个账户，所谓 ceph 的管理账户

2. 所有节点上为创建的账户授予 root 权限

```
#cat /etc/sudoers.d/ceph
Defaults:snow !requiretty
snow ALL = (root) NOPASSWD:ALL

#chmod 440 /etc/sudoers.d/ceph
```

GNU/Linux-Ceph

实现 Ceph

3. 在 Deploy 服务器生成 ssh 秘钥
#su - snow
$ssh-keygen

4. 在 Deploy 服务器建立 ssh 独立配置文件
$vi ~/.ssh/config

GNU/Linux-Ceph

实现 Ceph

Host cephsrv
    Hostname cephsrv.niliu.edu
    User snow
Host node01
    Hostname node01.niliu.edu
    User snow

GNU/Linux-Ceph

实现 Ceph

Host node02
    Hostname node02.niliu.edu
    User snow
Host node03
    Hostname node03.niliu.edu
    User snow

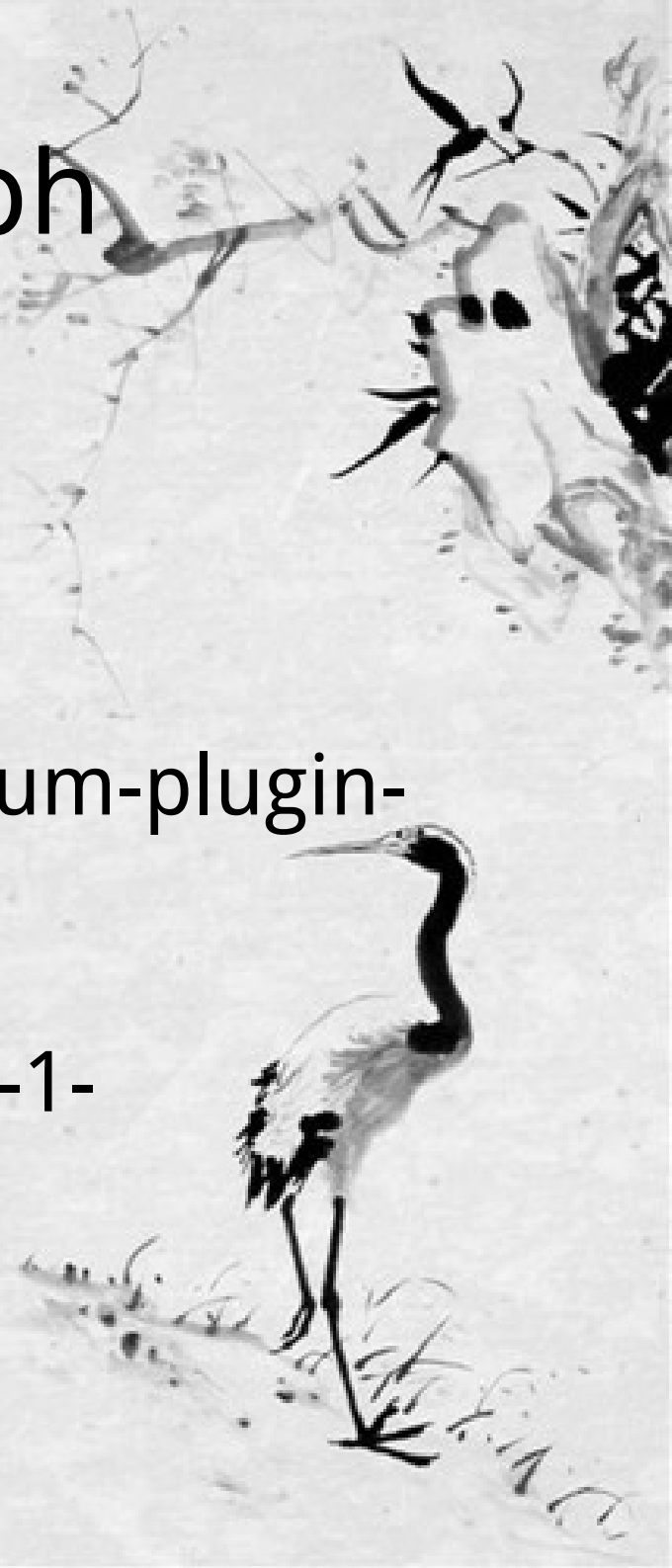
GNU/Linux-Ceph

实现 Ceph

5. 设定权限
$ chmod 600 ~/.ssh/config

6. 向其他节点传输 ssh-key
$ssh-copy-id node01
$ssh-copy-id node02
$ssh-copy-id node03

GNU/Linux-Ceph



实现 Ceph

7. 安装 Ceph 管理节点及其他节点

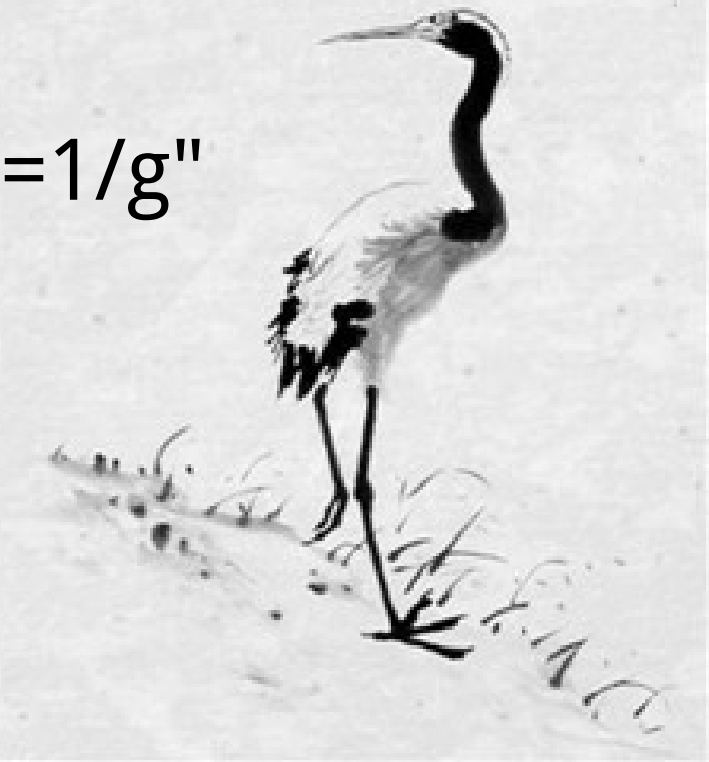
```
$sudo yum -y install epel-release yum-plugin-priorities \
http://download.ceph.com/rpm-infernalis/el7/noarch/ceph-release-1-1.el7.noarch.rpm
```

GNU/Linux-Ceph

实现 Ceph

7. 安装 Ceph 管理节点及其他节点

```
$sudo sed -i -e
"s/enabled=1/enabled=1\npriority=1/g"
/etc/yum.repos.d/ceph.repo
```



GNU/Linux-Ceph

实现 Ceph

7. 安装 Ceph 管理节点及其他节点
$sudo yum install ceph-deploy -y

8. 建立 ceph 目录
$ mkdir ceph
$ cd ceph

GNU/Linux-Ceph

实现 Ceph

9. 部署 ceph, 生成监控节点信息
$ceph-deploy new node01

10. 定义对象存储资源
$vi ./ceph.conf
//* 于最后追加
osd pool default size = 2

//*Object Storage Device， 提供存储资源。

GNU/Linux-Ceph

实现 Ceph

11. 修改 20 行 , 设定 ceph 版本
$exit
vi /usr/lib/python2.7/site-packages/ceph_deploy/install.py

改为 ceph 现行版本
args.release = 'infernalis'

#su - snow

GNU/Linux-Ceph

实现 Ceph

12. 将 ceph 安装到所有节点
$ceph-deploy install dlp node01 node02 node03

13. 初始化监控及秘钥
$ ceph-deploy mon create-initial

GNU/Linux-Ceph

实现 Ceph

14. 在所有节点上创建存储目录
Node1:
#mkdir -v /storage01
#chown ceph. /storage01

Node2:
#mkdir -v /storage02
#chown ceph. /storage02

GNU/Linux-Ceph

实现 Ceph

14. 在所有节点上创建存储目录

Node3:
#mkdir -v /storage03
#chown ceph. /storage03

GNU/Linux-Ceph

实现 Ceph

15. 准备资源池

```
$ ceph-deploy osd prepare node01:/storage01
node02:/storage02 node03:/storage03
```

GNU/Linux-Ceph

实现 Ceph

16. 激活资源池

```
$ ceph-deploy osd activate node01:/storage01
node02:/storage02 node03:/storage03
```

GNU/Linux-Ceph

实现 Ceph

17. 传输 ceph 配置文件

```
$ ceph-deploy admin cephsrv node01 node02 node03
$ sudo chmod 644 /etc/ceph/ceph.client.admin.keyring
```

GNU/Linux-Ceph

实现 Ceph

18. 查看 ceph 资源池状态

$ ceph health

GNU/Linux-Ceph

实现 Ceph

//* 如果打算重新建立资源池

1. 移除 ceph 软件包

$ ceph-deploy purge cephsrv node01 node02 node03

//* 移除配置

$ ceph-deploy purgedata cephsrv node01 node02 node03

$ ceph-deploy forgetkeys

GNU/Linux-Ceph

使用 Ceph

//* 对客户端进行安装及配置管理

```
$ ceph-deploy install client
$ ceph-deploy admin client
```

GNU/Linux-Ceph

使用 Ceph

//* 位于客户端操作

#su - snow
$ sudo chmod 644
/etc/ceph/ceph.client.admin.keyring

GNU/Linux-Ceph

使用 Ceph

//* 位于客户端操作

//* 创建一个磁盘为 disk01, 大小为 10G
$ rbd create disk01 --size 10240

$ rbd ls -l

//* 将磁盘映射为 rdb 设备
$ sudo rbd map disk01
$ rbd showmapped

GNU/Linux-Ceph

使用 Ceph

//* 位于客户端操作

```
$ sudo mkfs.xfs /dev/rbd0

$ sudo mount /dev/rbd0 /mnt

$ df -hT
```

GNU/Linux-Ceph

基于网络系统挂载 Ceph

//* 位于 ceph-deploy 操作

1. 在指定节点创建元数据服务器
$ ceph-deploy mds create node01

GNU/Linux-Ceph

基于网络系统挂载 Ceph

//* 元数据（ Metadata ），又称中介数据、中继数据，为描述数据的数据（ data about data ），主要是描述数据属性（ property ）的信息，用来支持如指示存储位置、历史数据、资源查找、文件记录等功能。元数据算是一种电子式目录，为了达到编制目录的目的，必须在描述并收藏数据的内容或特色，进而达成协助数据检索的目的。

GNU/Linux-Ceph

基于网络系统挂载 Ceph

//* 位于 node1 操作

2. 在指定节点创建 pools
$ sudo chmod 644
/etc/ceph/ceph.client.admin.keyring

$ ceph osd pool create cephfs_data 128
$ ceph osd pool create cephfs_metadata 128

GNU/Linux-Ceph

基于网络系统挂载 Ceph

//* 位于 node1 操作

3. 启用 pools
$ ceph fs new cephfs cephfs_metadata cephfs_data

$ ceph fs ls
$ ceph mds stat

GNU/Linux-Ceph

基于网络系统挂载 Ceph

//* 客户端挂载资源池

1. 安装 ceph 源

```
#yum  install http://download.ceph.com/rpm-infernalis/el7/noarch/ceph-release-1-1.el7.noarch.rpm -y
```

GNU/Linux-Ceph

基于网络系统挂载 Ceph

//* 客户端挂载资源池

2. 安装 ceph 挂载工具
yum install ceph-fuse -y

GNU/Linux-Ceph

基于网络系统挂载 Ceph

//* 客户端挂载资源池

3. 获取 admin 秘钥
ssh snow@node01.niliu.edu "sudo ceph-authtool -p /etc/ceph/ceph.client.admin.keyring" > admin.key

chmod 600 admin.key

GNU/Linux-Ceph

基于网络系统挂载 Ceph

//* 客户端挂载资源池

4. 挂载
mount -t ceph node01.niliu.edu:6789:/ /mnt -o name=admin,secretfile=admin.key

5. 测试
#df -hT