并行文件系统的使用场景

并行文件系统以其可以承载巨大量级的数据及方便扩展的性能，并且通常支持统一的命名空间，广泛地使用在具有并行读写 ，高吞吐，高带宽的场景中，典型的HPC (High Performance Computing) 高性能运算场景，如气象预测，基因分析，媒体渲染农场等需要多个计算节点同时对文件系统进行文件并发操作。

为什么选择 BeeGFS？

1.并行文件系统场景，使用分布式服务器分散存储压力

2.快速进行容量与性能的扩展与提升

3.管理相对方便，有图形化管理页面

4.相对于很多商用方案，BeeGFS是开源产品，源码下载地址参考：https://www.beegfs.io/content/source-code/

BeeGFS 是什么？

BeeGFS ( www.beegfs.io ) 是行业领先的并行集群文件系统，设计时优先考虑性能强大的同时，安装和管理非常方便，是I / O密集型工作负载的优质存储解决方案。

BeeGFS 简单介绍

1.BeeGFS 的系统架构如下：组件角色示意图如下：

上图来自BeeGFS官方

上图右半部分可以看出 BeeGFS 核心服务包含3个存储相关角色的组件：客户端，元数据服务，存储服务。同时包括管理节点与图形管理和监控服务，每个服务的功能介绍如下：

BeeGFS客户端 ( Client ) 介绍与特性

作用：客户端服务就是把后端的存储空间，提供给前端计算服务的组件。在计算节点或者需要并行存储空间的服务器安装客户端，就可以方便运用并行文件系统来扩展空间和性能。

兼容性：BeeGFS 客户端支持将 BeeGFS系统再次导出为标准的 NFS，通过 SAMBA 导出为 Windows 共享，或者做为 Hadoop HDFS 系统的替代。在 BeeGFS 即将发布的新版本将计划支持原生的 Windows 客户端。

启动挂载：当客户端以服务形式加端后，存储会被自动挂载到 bee-mounts.conf 里面定义的路径，通常情况我们修改 Linux 环境的 /etc/fstab 来开机挂载文件系统的操作也适用，但并不为官方推荐。

存储服务组件 ( Storage service ) 介绍与特性

扩展性：存储服务 ( 有时也称为 “对象存储服务” (Object Storage Service ) 是用于存储条带化用户文件内容 ( 也称为数据块文件 ) 的主要服务。BeeGFS存储服务基于横向扩展 ( Scale-Out ) 设计。这意味着，每个BeeGFS文件系统实例可以具有一个或多个存储服务组件，，方便提高性能与空间。一个存储服务实例具有一个或多个存储服务组件。

缓存优化：由于 BeeGFS 自动使用存储服务器上的所有可用 RAM 自动进行缓存，因此它还可以在将数据写入磁盘之前将较小的 IO 请求聚合到较大的块中。此外，如果最近被另一个客户端访问过，则数据会从高速缓存中加载，来有效针对突发流量。

数据保护：传统会在存储服务组件的物理存储使用 RAID 方式进行数据保护，但 AWS EBS 服务原生多份保存，所以如无必要，无需对本地 EBS 做软件 RAID 保护。

元数据服务组件 ( Metadata Services ) 介绍与特性

作用：元数据服务组件保存着例如：目录结构，文件与目录权限，在存储上文件数据分片信息等存储的元数据信息。

索引：元数据服务在客户端打开文件时向客户端提供有关单个用户文件在存储服务上的位置 ( 所谓的“条带模式” ) 的信息，帮忙客户端找到要访问的数据位置之后，元数据服务将不参与数据的读写操作访问。

横向扩展：BeeGFS 元数据服务支持横向扩展，这意味着 BeeGFS 文件系统中可以有一个或多个元数据服务。它负责全局名称空间的专有部分，因此拥有更多的元数据服务器可提高整体系统性能。

性能：在元数据目标上，BeeGFS 为每个用户创建的文件创建一个元数据文件。这是 BeeGFS 的一项重要设计决策，它避免了将所有元数据存储在单个数据库中可能会损坏的情况。因为低元数据访问延迟提高了文件系统的响应速度。BeeGFS 元数据非常小，并且随着用户创建的文件数量线性增长。512GB 的可用元数据容量通常可容纳约 5 亿个用户文件。

管理服务：

可以将管理服务视为 BeeGFS 客户端，元数据，数据存储的“汇聚点”。它轻巧，对性能没有要求，也不存储任何用户数据，通常不需要在独占的虚拟实例上运行。 管理服务监视所有已注册的服务并检查其状态。 因此，在新环境部署是时也第一个部署。管理服务维护所有其他 BeeGFS 服务及其状态情况。

可选的图形管理组件 Admon ( Administration and Monitoring service ) ，其包括2个部分：

Admon 后端服务：该服务在任何具有对元数据和存储服务的网络访问权限的计算机上运行。该服务收集其他BeeGFS服务的状态信息并将其存储在数据库中。

基于 Java 的图形客户端，这个可以安装在您的工作站上。 它通过 http 连接远程 admon 守护程序方便管理和监控

环境部署

部署架构图

部署说明

可用区：如果没有可用区级别的高可用部署需要，可以使用单可用区部署来提供整体架构的性能。

子网划分：可以对于整体网络进行公有子网与私有子网的划分，将对外部通信的 Admon 与堡垒机放到公有子网方便管理员直接访问。将其它 BeeGFS 相关的应用组件放到私有子网内来通过减少对外部的暴露，保证系统的访问安全

公有子网： 外部可以通过 IGW ( Internet Gateway ) 对 EC2 进行直接访问

私有子网 ( 外部不能直接通过 IGW ( Internet Gateway ) 直接访问，内部可以通过添加 NAT Gateway 方式访问外网安装应用文件等

安全组设置：由于各个角色之间需要互相通信，所以要在安全组中把互相的端口访问打开。默认不同角色使用 8000 ~ 8010 等高位端口。如果不确认，可以将私有子网的组件实例都加到一个内部互相通信全部允许的安全组中。对外只开放到 bastion 堡垒机的通信

置放群组：对于性能交互强的BeeGFS核心3大组件，可以把组件对应的EC2实例放到集群置放群组，来保证通信质量。

Client协议：BeeGFS 客户端支持将mount点重新导出为NFS协议或者通过 Samba 与 Windows 系统文件共享。本例中将展示导出为 SAMBA 与 NFS 协议。

元数据节点：本例中只做了一台 Metadata 元数据节点，实际生产中可以使用 BeeGFS Buddy Mirroring 功能做镜像元数据节点来增强高可用性。

存储节点： BeeGFS 中保存数据分片的节点，本例中为展示 BeeGFS 的使用了 3 个存储节点保存数据分片扩展性能与容量。如果需要对数据进行保护，同样请开始 Buddy Mirroring 功能

S3 资料库：对于安装文件，以及要使用并行文件系统处理的数据等，可以存放在 S3 中，利用 S3 与 EC2 之间的高速网络来提高初始数据加载速度。

主机名：本例中使用 node{n} 来约定主机名，对于各个角色，请参考架构图旁边的表格，可以使用自建 DNS 或者AWS Route 53 私有托管区域来做内部域名解析。否则请修改 /etc/hosts 来加入节点名字方便解析，不推荐此方法，因为当 BeeGFS 扩展节点时，多数节点 /etc/hosts 都要进行修改，工作比较繁琐容易出现逻辑错误。

组件系统配置

本例中安装系统配置如下表，软件系统 Ubuntu 18.04，Windows 客户端使用 Windows 2019

角色 主机名 系统配置 vCPU 内存 (GB) 存储

Admon / Bastion node5 c5.large 2 4 EBS 8GB

Management node1 c5.large 2 4 EBS 8GB

Metadata node2 c5d.xlarge 4 8 1 x 100GB NVMe SSD

Storage-I node301 d2.xlarge 4 30.5 3 x 2TB HDD

Storage-II node302 d2.xlarge 4 30.5 3 x 2TB HDD

Storage-III node303 d2.xlarge 4 30.5 3 x 2TB HDD

Client node4 c5.large 2 4 EBS 8GB

表1：服务角色与配置

实例存储：

本案例中元数据组件与存储组件实例都使用了附加在主机上的实例存储 ( Instance Store ) ，它的优势可以方便使用主机通信优势来增加 IOPS 与带宽，但同时需要考虑的是，实例存储在实例停止或者终止状态下，内容将丢失。所以使用实例存储一定要做好数据的保护。定时同步到 S3 或者使用 Buddy Mirroring 做保护。

\* 以上为本案例实际部署使用的配置，官方的最小配置请参考：https://www.beegfs.io/wiki/FAQ#min\_reqs

安装路径与名称

BeeGFS 应用组件包角色与相应的组件包名称如下：

组件角色 组件包名称

管理服务组件 beegfs-mgmtd

元数据服务组件 beegfs-meta

存储数据服务组件 beegfs-storage

客户端组件 beegfs-client / beegfs-helperd

Admon 组件 beegfs-admon

命令行组件 beegfs-utils

通用组件 beegfs-common

表2：组件包名称

对应组件的默认安装路径如下：

文件类型 路径

应用程序文件 /opt/beegfs

启动脚本 /etc/init.d/beegfs-{pack-name}

日志文件 /var/log/beegfs-{pack -name}.log

编译器 The GNU C compiler (gcc)

配置文件 /etc/beegfs/beegfs-{pack-name}.conf

表3：安装路径

\* { pack-name } 意为将 { } 内容替换为对应的组件名称，例如，存储节点的日志 /var/log/beegfs-{ pack-name } 代表其保存于 /var/log/beegfs-storage.log

安装过程

下列如无特殊说明，将需要在每个组件节点上执行

安装源 ( Repository ) 更新

$ sudo wget https://www.beegfs.io/release/latest-stable/dists/beegfs-deb9.list

$ sudo cp beegfs-deb9.list /etc/apt/sources.list.d/

$ sudo apt update

$ sudo wget -q https://www.beegfs.io/release/latest-stable/gpg/DEB-GPG-KEY-beegfs -O- | apt-key add -

图形方式安装

对于 BeeGFS 安装步骤不熟悉可以使用图形 ( GUI ) 安装步骤，对于进阶用户或者想了解更多 BeeGFS 组件安装与配置，可以使用手动安装方式，下面步骤在 admon 节点安装

$ sudo apt-get install beegfs-admon # 安装图形管理组件beegfs-admon

$ /etc/init.d/beegfs-admon start # 开启 Admon 守护进程

Admon 图形化管理应用是一个 .jar 文件，保存在 admon节点的 /opt/beegfs/beegfs-admon-gui/ 路径下，复制其到管理员的笔记本或者工作站上面，通过 java -jar beegfs-admon.jar 打开 Admon 客户端进行图像化配置与安装，默认用户名是 Administrator , 密码是 admin 当在本地笔记本访问进，请在安全组加入工作站或者笔记本的公网IP，开放访问。

图形界面如下图, 通过左边列表栏最下方 Installation 进行安装，图形界面同时具有监控管理等功能

图: 图形管理界面

手动安装 BeeGFS 组件

1.安装源更新，与图形界面安装过程一致

2.更改主机名称，根据上表1名称，在每个组件实例中运行，下面以 node1 管理节点为例，

$ sudo hostnamectl set-hostname node1 # 更改机器名为node1

$ sudo sh -c 'echo 127.0.1.1 $(hostname) >> /etc/hosts' # 更改机器名为node1

$ sudo sh -c 'echo Private\_IP $(hostname) >> /etc/hosts' # 将这里的Private\_IP改为实例的私有IP地址

$ sudo reboot # 重新启动重新

3.配置管理节点 node1 到每个节点的无密码登陆，在node1 生成密钥对，把公钥加到其它节点的 /root/.ssh/authorized\_keys 文件中，将 id\_rsa 文件保存到 /root/.ssh/id\_rsa, ssh 登陆管理节点 node1，详细参考 https://docs.aws.amazon.com/zh\_cn/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html

$ ssh -i keypair.ssh ubuntu@node1 # ssh登陆到 node1 shell

$ sudo su # 切换到root界面

# vi /etc/ssh/sshd\_config # 修改 PermitRootLogin 值是 yes ，允许 root 登陆

# cd ~/.ssh

# ssh-keygen -t rsa # 生成公钥与密钥，密钥会保存到 ~/.ssh/id\_rsa，公钥文件 ~/.ssh/id\_rsa.pub

# cat id\_rsa.pub >> authorized\_keys # 公钥加入授权文件，登陆时自动验证

# cp authorized\_keys authorized\_keys\_node1 # 更名，准备复制到其它节点

# scp authorized\_keys\_node1 node{n}:/root/.ssh/ # 将授信公钥文件使用 scp 拷贝到其它节点 node{n}

# scp id\_rsa node{n}:/root/.ssh/ # 将授信密钥文件使用 scp 拷贝到其它节点 node{n}

4.登陆其它节点node{n}，以node2为例

$ ssh -i keypair.pem ubuntu@node2 # 先从堡垒机，登陆到其它节点，以node2为例

$ sudo su # 切换到root状态

# cat /root/ssh/authorized\_keys\_node1 >> /root/ssh/authorized\_keys

# vi /etc/ssh/sshd\_config # 修改 PermitRootLogin 值是 yes ，允许 root 登陆

# ssh root@node1 # 测试是否无密码输入可以登陆 node1，并加入 node1 到known\_hosts 配置文件当中，具体格式可以使用 man 8 sshd 进行查看

# ssh root@node2 # 以 root 方式登陆回 node2 进行双向认证，并加入known\_hosts

# exit # 退出 node2，再次 exit 退出 node1，再次 exit 退回到堡垒机

5.登陆到不同的节点，并在每个组件实例中安装与配置各自的服务：

登陆到管理节点 node1 进行配置：

$ ssh root@node1

# apt install beegfs-mgmtd # 安装管理服务应用

# /opt/beegfs/sbin/beegfs-setup-mgmtd -p /data/beegfs/beegfs\_mgmtd # 在对应目录创建管理服务

# systemctl start beegfs-mgmtd # 启动服务

登陆到元数据管理节点 node2 ，安装配置元数据服务：

$ ssh root@node2 # 登陆 node2

# apt install beegfs-meta # 安装元数据服务组件

## 以下格式化 NVMe 卷成xfs文件系统，并挂载到 /data/lun\_meta 路径，文档请参考https://docs.aws.amazon.com/zh\_cn/AWSEC2/latest/UserGuide/ebs-using-volumes.html

$ lsblk # 查看NVMe卷的名字，假设为 /dev/nvme0n1

$ sudo mkfs -t xfs /dev/nvme0n1 # 格式化成 xfs 文件系统

$ sudo mkdir -p /data/lun\_meta # 创建挂载点

$ sudo mount /dev/nvme0n1 /data/lun\_meta # 将格式化后的新卷挂载到 /data/lun\_meta

## 设置重新启动，注意是 restart 后自动挂载，Stop 后 NVMe 数据将丢失，如果需要 STOP 操作，请使用 EBS 卷做为存储

$ sudo lsblk -o +UUID # 检查卷的 UUID，假设为abcd-efgh-1234-5678

$ sudo vi /etc/fstab # 修改自动加载配置文件

添加：UUID= abcd-efgh-1234-5678 /data/lun\_meta xfs defaults,nofail 0 2

## 在对应目录创建元数据管理服务，并指定元数据服务 sid 为2，管理节点名称为 node1, 如下：

# /opt/beegfs/sbin/beegfs-setup-meta -p /data/lun\_meta a -s 2 -m node1

# systemctl start beegfs-meta # 启动服务

这里要注意 node1 域名要可以被 node2 节点进行解析，下面步骤中也一样的要求

6.登陆到数据存储节点 node301, node302, node303 进行配置，参考文档 https://www.beegfs.io/wiki/FAQ#add\_storage\_target

$ ssh root@node301 # 未来同样登陆 302,303 执行相应操作

# apt install beegfs-storage # 安装存储管理服务

## 到此需要格式化 NVMe 卷，其过程与配置自己加载请参考元数据服务的卷配置过程，假设挂载路径为/mnt/lun\_storage\_1

## 加入本地存储卷到 BeeGFS 存储中，如下，301 是存储节点代号，3011是目前存储节点代码，在 301 实例上如果增加另外的卷，可以选择 3012,3013 等 301{n} 形式进行存储扩容。

# /opt/beegfs/sbin/beegfs-setup-storage -p /mnt/lun\_storage\_1 -s 301 -i 3011 -m node1

# systemctl start beegfs-storage # 启动服务

登陆到客户端节点 node4 进行配置：

$ ssh root@node4

# apt install beegfs-client beegfs-helperd beegfs-utils # 安装客户端，命令行工具盒

# /opt/beegfs/sbi­­n/beegfs-setup-client -m node1 # 指定管理节点 node1

# vim /etc/beegfs/beegfs-mounts.conf

# 修改beegfs-mounts.conf，第一项是挂载目录，第二项是配置文件路径，如下:

/mnt/beegfs /etc/beegfs/beegfs-client.conf

# systemctl start beegfs-helperd # 启动heplerd服务，提供日志与 DNS 解析等服务

# systemctl start beegfs-client # 启动客户端服务

登陆到 Admon 节点 node5 进行配置，这个组件不是核心组件，所以这一步可以不做

$ ssh root@node5

# apt install beegfs-admon # 安装 admon service

# vim /etc/beegfs/beegfs-admon.conf # 找到选项 "sysMgmtdHost" 并进行配置 "sysMgmtdHost=node1"

# systemctl start beegfs-admon # 启动Admon服务

安装完成，在客户端检查连接性

$ ssh root@node4 # 登陆客户端节点

# beegfs-ctl --listnodes --nodetype=meta –nicdetails # 检查元数据服务情况

# beegfs-ctl --listnodes --nodetype=storage –nicdetails # 检查数据存储服务情况

# beegfs-ctl --listnodes --nodetype=client –nicdetails # 检查客户端服务情况

# beegfs-net # 显示客户端的实际节点连接情况

# beegfs-check-servers # 显示客户端的配置连接情况

# beegfs-df # 显示可用空间与存储情况

7.未来如果需要删除元数据服务节点或者数据存储节点:

## 使用 BeeGFS 命令行客户端 beegfs-ctl 来进行操作：

## 下面命令为去掉元数据节点 <nodeID> :

# beegfs-ctl --removenode --nodetype=meta <nodeID>

## 下面命令为去掉存储数据节点 <nodeID> :

$ beegfs-ctl --removenode --nodetype=storage <nodeID>

数据安全：请注意，删除节点不会自动迁移此节点上存储的文件，因此请谨慎使用此选项。 如果要在删除文件之前将文件从存储服务器移动到其他存储服务器，请参考官方：https://www.beegfs.io/wiki/FAQ#migrate

使用BeeGFS并行文件系统空间

1.直接通过 BeeGFS 客户端使用并行文件系统：

在上面的安装步骤中，我们对 node4 中的 beegfs-mounts.conf 进行了修改，挂载点是 /mnt/beegfs ，对 /mnt/beegfs 的文件修改会相应提交到元数据管理节点 node2 与存储节点 node301, node302, node303. 同时其它客户端也会收到 node4 的文件修改。

2.通过 Exports 共享给 Linux 环境使用

BeeGFS 客户端配置：如上架构图 ( 图1 ) ，本例将在 BeeGFS 客户端 ( node4 ) 安装 nfs-kernel-server ，通过使用 nfs 把本地并行文件系统分享给其它 Linux 系统使用，使用 root 模式，步骤如下：

## 保证beegfs-client 配置支持把文件并行文件系统挂载点再次导出为 nfs 修改如下：

# vi /etc/beegfs/beegfs-client.conf

tuneRefreshOnGetAttr = true

# apt install nfs-kernel-server # 安装nfs-kernel 服务器端

## 修改配置文件 exports, 加入相应路径信息

# vi /etc/exports

/mnt/beegfs \*(rw,async,fsid=0,no\_subtree\_check,no\_root\_squash)

# exportfs -a # 导出文件系统

# systemctl restart nfs-kernel-server # 重新启动 nfs 客户端

NFS 客户端配置: 登陆node-linux, 进入root模式，安装 nfs 客户端支持

# apt-get update

# apt install nfs-common

# mkdir -p /mnt/node4nfs

# mount -t nfs node4.beegfs:/mnt/beegfs /mnt/node4nfs # 挂载 node4 的导出文件系统

# touch /mnt/node4nfs/a.txt # 生成文件进行测试

# ls /mnt/node4nfs # 检查是否生成

3.通过 Samba 共享给 Windows 系统使用：

BeeGFS 客户端配置：如上架构图 ( 图1 ) ，本例将在 BeeGFS 客户端 ( node4 ) 安装samba ，通过使用 samba 把本地并行文件系统分享给其它 Windows系统使用，使用 root 模式，步骤如下：

# apt install samba # 安装 samba 服务器端

# vi /etc/samba/smb.conf # 加入以下配置，这里假设客户端已经将文件系统绑定在 /mnt/beegfs

[beegfs]

comment = BeeGFS file system

public = yes

path = /mnt/beegfs

browsable = yes

writable = yes

read only = no

# 重新启动 samba 服务来生效配置

systemctl restart smbd

systemctl status smbd

Windows 端配置：

使用“开始” – “运行“ 打开路径 \\node4\mnt\beegfs 看是否可以正常运行，请确认安全组已经把 node-win 加入到允许列表

高版本Windows有组策略禁止insecure guest logons， 运行 msc 修改项目如下：

Computer Configuration -> Administrative Templates -> Network -> Lanman Workstation 然后选择Enable insecure guest logons

总结

上面我们了解了并行文件系统 BeeGFS 的功能，作用，并且解释了安装步骤。同时对于并行文件系统的客户端，NFS，SMB 共享与配置进行了安装说明。希望大家可以轻松在 AWS 环境上开启对 BeeGFS 的使用。

参考材料

1.BeeGFS 参考材料：

官方：https://www.beegfs.io/content/

BeeGFS白皮书：https://www.beegfs.io/docs/whitepapers/Introduction\_to\_BeeGFS\_by\_ThinkParQ.pdf

BeeGFS WIKI: https://www.beegfs.io/wiki/FAQ#add\_node

2.SAMBA部署；https://wiki.samba.org/index.php/Operating\_System\_Requirements

3.NFS安装: https://vitux.com/install-nfs-server-and-client-on-ubuntu/