为 Singularity 容器构建可持久化的 Overlay 二层镜像

张文帅

中国科学技术大学 超级计算中心

在当前使用 Singularity 的集群系统中，用户构建一个应用镜像时，一般需要自己定义一个新的配置文件，然后在具有 root 权限的节点中 build 一个全新的 Singularity 镜像，其中既包含系统，也包含应用。如此，导致不同的应用无法共享同一个底层系统镜像，当在启动镜像或这是做镜像的本地缓存时，占用了大量的节点间带宽或节点内存储空间。

本文中说明如何在超算系统中创建两层镜像，底层为原始系统镜像，二层为用户可写的 Overlay 镜像。如此，底层的系统镜像可以被多个应用共享，用户也可以直接写入自己的应用文件，减少需要使用 ROOT 权限来构建系统镜像的次数，方便了用户在容器内操作的数据持久化，方便了应用容器的构建、存储与迁移。

以下，说明两层镜像的具体构建方法。

1. 载入 singularity 环境

例如：*module load singularity/3.5.3*

2. 测试无 Overlay 层的底层系统镜像

```
$ singularity shell ~/singularity/centos-basic-7.7.sif
Singularity> cat /etc/os-release
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"
CENTOS_MANTISBT_PROJECT="CentOS-7"
CENTOS_MANTISBT_PROJECT_VERSION="7"
REDHAT_SUPPORT_PRODUCT="centos"
REDHAT_SUPPORT_PRODUCT_VERSION="7"
```

3. 生成 Overlay 二层用户镜像

此步中，地球大数据系统 centos7 默认自带的 mkfs.ext3 工具版本较老，没有-d 参数支持，无法创建用户可写的镜像，需要使用新安装的 v1.46.2 版本的 e2fsprogs 工具，其中的 mkfs.ext3 工具位于目录：/public/home/uswszhang5566/singularity/e2fsprogs-1.46.2/install/sbin/mkfs.ext3。

$ dd if=/dev/zero of=overlay.img bs=1M count=500
500+0 records in
500+0 records out
524288000 bytes (524 MB) copied, 0.362755 s, 1.4 GB/s
$ mkdir -p overlay/upper
mkfs.ext3 -d overlay overlay.img

4. **载入用户层镜像，作为用户可写的数据空间**

$ singularity shell --overlay overlay.img ~/singularity/centos-basic-7.7.sif

此时，普通用户已经可以在没有 bind mount 的目录中写入用户的文件了，而且可以将用户文件写入到原本需要 ROOT 权限的目录中，例如/根目录：

Singularity> echo "added by normal user" > /test.txt
Singularity> cat /test.txt
added by normal user
Singularity> exit
$ singularity shell --overlay overlay.img ~/singularity/centos-basic-7.7.sif
Singularity> cat /test.txt
added by normal user

如上演示，所写入的 /test.txt 文件被永久保持在二层镜像 overlay.img 中，可以在重新启动容器后继续使用。

5. **升级系统应用文件**

在前一步写入操作中，用户可写入的文件不能是已存在的非当前用户的文件，也就是无法更新系统镜像中已有的系统文件，无法进一步升级系统程序。

Singularity 为 ROOT 权限设置了较多限制，防止在容器中提权为 ROOT 权限，然后影响宿主系统的安全。

为解决一层系统镜像中的文件的升级问题，用户依然需要登录到一个可以拿到 ROOT 权限的 B 主机中（可以是具有 ROOT 权限的 Docker 容器），在其中，同样加载 Overlay 镜像来升级安装已在系统镜像中存在的程序。

Singularity> yum info git | grep -E "Version|Release|Package"
Installed Packages
Version : 1.8.3.1
Release : 21.el7_7
Summary : Fast Version Control System
Available Packages
Version : 1.8.3.1
Release : 23.el7_8
Summary : Fast Version Control System
Singularity> yum install -y git
[.....]
Singularity> yum info git | grep -E "Version|Release|Package"

Installed Packages
Version : 1.8.3.1
Release : 23.el7_8
Summary : Fast Version Control System

更新安装的系统程序，也会保存在第二层的 Overlay 镜像中，用户可以将此镜像，拷贝回计算集群中，并不需要再拷贝底层系统镜像。

此时，在集群中，结合原底层镜像与新制作的 Overlay 镜像，可以再现 B 主机中制作的完整容器镜像的安装状态。

$ singularity shell --overlay overlay.img ~/singularity/centos-basic-7.7.sif
Singularity> yum info git | grep -E "Version|Release|Package"
Installed Packages
Version : 1.8.3.1
Release : 23.el7_8
Summary : Fast Version Control System

在这种情况下，不同的用户可以基于同一个底层的系统镜像，却可以具有不同的系统软件版本号，以适应自身应用软件的个性化需求。

6. 总结

该方法主要用于交互式的在 Singularity 容器上做持久化的应用修改与部署，可兼顾容器使用上的安全性与制作应用容器的简单便捷性。在高性能计算的特定场景下，避免使用 Docker 导致降低安全性，也避免使用镜像定义文件来制作单一镜像文件的应用容器，增大应用容器制作的困难。