TECHNISCHE UNIVERSITÄT DRESDEN

SPANK plugin to start root VMs for building Singularity containers

Danny Rotscher (TU Dresden)

ZIH
Zentrum für Informationsdienste und Hochleistungsrechnen

Agenda

1. Overview
2. Infrastructure
3. User friendly scripts
4. Outlook

Overview

```
rotscher@taruslogin5:~> srun -p ml-interactive --cpus-per-task=7 --mem-per-cpu=1443
 --gres=gpu:1 --time=00:30:00 --pty --cloud=kvm bash -l
srun: job 21802568 queued and waiting for resources
srun: job 21802568 has been allocated resources

Die folgenden Module wurden in einer anderen Version erneut geladen:
  1) modenv/scs5 => modenv/ml

rotscher@taurusml2:~> uname -r
4.14.0-115.19.1.el7a.ppc64le
rotscher@taurusml2:~> source /tmp/${SLURM_JOB_USER}_${SLURM_JOB_ID}/activate
Last login: Fri Jun 28 10:19:48 2019 from gateway
[root@rotscher_21802568 ~]# uname -r
3.10.0-957.21.3.el7.ppc64le
[root@rotscher_21802568 ~]# singularity --version
singularity version 3.2.1-1.el7
[root@rotscher_21802568 ~]#
```

Infrastructure

- Slurm configuration
- SPANK plugin
- Prolog- und epilog scripts
- VM start script

Infrastructure - Slurm configuration

slurm.conf:

```
[...]
ProctrackType=proctrack/cgroup
TaskPlugin=task/cgroup
Prolog=/etc/slurm/prolog.sh
Epilog=/etc/slurm/epilog.sh
[...]
```

cgroup.conf:

```
[...]
ConstrainCores=yes
ConstrainRAMSpace=yes
[...]
```

plugstack.conf:

```
[...]
optional /usr/lib64/slurm/spank_cloud.so
[...]
```

Infrastructure - SPANK plugin

```
marie@login$ srun --help
[...]
Options provided by plugins:
      --cloud=[kvm|lxc]       Depending on which parameter was selected, either
                              a container or a vm is created.
      --cloud-create=<yes>    Create vm
      --cloud-type=[kvm|lxc]  Depending on which parameter was selected, either
                              a container or a vm is created.
[...]
```

spank_cloud.c:

```
[...]
int slurm_spank_init_post_opt(spank_t spank, int ac, char *argv[])
{
    spank_context_t context = spank_context();
    if ((context == S_CTX_LOCAL) || (context == S_CTX_ALLOCATOR)) {
        spank_job_control_setenv(spank, "cloud", cloud, 1);
        spank_job_control_setenv(spank, "cloud_create", cloud_create, 1);
        spank_job_control_setenv(spank, "cloud_type", cloud_type, 1);
    }

    return 0;
}
[...]
```

Infrastructure - Prolog scripts

- /etc/slurm/prolog.sh execute all scripts inside /opt/slurm/prolog.d
- 08_update_iplist will update a CSV file, which contain all used IPs

```
#IP;SLURM_JOB_ID;USED
192.168.0.1;21815278;USED
192.168.0.2;;
[...]
192.168.0.252;;
```

- 10_cloud:

```
[...]
if [[ "${SPANK_cloud}" == "kvm" ]]; then
screen -L -S cloud_${SLURM_JOB_ID} -dm bash -c "/opt/slurm/bin/spank_cloud-kvm
screen -ls
PID=$(screen -S cloud_${SLURM_JOB_ID} -Q echo '${PID}')
cgclassify -g cpuset,memory:slurm/uid_${SLURM_JOB_UID}/job_${SLURM_JOB_ID} --s
fi
[...]
```

Infrastructure - Epilog scripts

- /etc/slurm/epilog.sh execute all scripts inside /opt/slurm/epilog.d
- 01_cloud:

```
[...]
if [[ "${SPANK_cloud}" == "kvm" ]]; then
CONTAINER="${SLURM_JOB_USER}_${SLURM_JOB_ID}"
virsh destroy ${CONTAINER}
virsh undefine ${CONTAINER}
rm -rf /tmp/${CONTAINER}
screen -S cloud_${SLURM_JOB_ID} -X quit
screen -ls
echo "### ${SLURM_JOB_ID} EPILOG END ###"
fi
[...]
```

- 02_update_iplist will remove the job id from the CSV file

```
#IP;SLURM_JOB_ID;USED
192.168.0.1;;
192.168.0.2;;
[...]
192.168.0.252;;
```

Infrastructure - VM start script

- Why not put everything in prolog script?
- Copies template image to /tmp directory and modifies them
- Sets resources of VM depending on job memory and cpu allocation
- Start VM
- Create access script for user

User friendly scripts

- startInVM is a wrapper script for starting the VM

```
rotscher@tauruslogin5:~> startInVM --arch=power9
srun: You may only use this partition when allocating GPUs via --gres.
srun: job 21815458 queued and waiting for resources
srun: job 21815458 has been allocated resources
Last login: Fri Jun 28 10:19:48 2019 from gateway
[root@rotscher_21815458 ~]#
```

- buildSingularityImage

```
rotscher@tauruslogin5:~> buildSingularityImage --arch=power9 debian11.sif debian11.def
sbatch: You may only use this partition when allocating GPUs via --gres.
Submitted batch job 21817514
rotscher@tauruslogin5:~> cat debian11.def
Bootstrap: docker
From: debian:11
rotscher@tauruslogin5:~> ll debian11.sif
-rwxr-xr-x 1 rotscher hpcsupport 54829056 11. Dez 21:56 debian11.sif
rotscher@tauruslogin5:~>
```

Outlook

- NEW version in development

```
rotscher@taurusa9:~> srun -A hpcsupport -p hpdlf -w taurusa9 --kvm-create --kvm-image-size=100G -
-kvm-image-load=file:/beegfs/ws/1/rotscher-kvm/packstack_2021-10-24.qcow2  -c12 --mem=90000 --tim
e=08:00:00 --pty bash -l
srun: job 21817524 queued and waiting for resources
srun: job 21817524 has been allocated resources
rotscher@taurusa9:~> source /tmp/${SLURM_JOB_USER}_${SLURM_JOB_ID}/activate
Warning: Permanently added '192.168.0.1' (ECDSA) to the list of known hosts.
Last login: Sat Dec 11 21:51:10 2021
/usr/local/bin/mount_host_data: line 84: sshfs: command not found
/usr/local/bin/mount_host_data: line 84: sshfs: command not found
/usr/local/bin/mount_host_data: line 84: sshfs: command not found
[root@packstack ~]# . keystonerc_admin
[root@packstack ~(keystone_admin)]# openstack server list
+--------------------------------------+-------------+---------+----------------------+---------
--------+-----------+
| ID                                   | Name        | Status  | Networks             | Image
        | Flavor    |
+--------------------------------------+-------------+---------+----------------------+---------
--------+-----------+
| 754b4f95-c309-4a62-a95c-0166dd8151b9 | test_ubuntu | SHUTOFF | external=192.168.0.17 | ubuntu_2
0.04    | m1.medium |
| 14f43717-5127-4438-af9d-d209eb6d3be1 | test_centos | SHUTOFF | external=192.168.0.19 | centos_8
.4.2105 | m1.medium |
+--------------------------------------+-------------+---------+----------------------+---------
--------+-----------+
[root@packstack ~(keystone_admin)]# openstack server start test_ubuntu
[root@packstack ~(keystone_admin)]# grep PRETTY_NAME /etc/os-release
PRETTY_NAME="CentOS Linux 7 (Core)"
[root@packstack ~(keystone_admin)]# ssh ubuntu@192.168.0.17 -i root.pem cat /etc/os-release | gre
p PRETTY_NAME
PRETTY_NAME="Ubuntu 20.04.3 LTS"
[root@packstack ~(keystone_admin)]#
```

Thank you very much for your attention

Questions?

Links

- https://github.com/cea-hpc/pcocc