一、集群环境介绍

1、操作系统

集群的操作系统为 64 位 Centos 7.9，提供标准的 64 位 Linux 操作系统环境。

2、并行环境

Intel oneapi， openmpi

3、数学库

本集群采用 intel one api 编译器自带的 fftw mkl 数学库。

4、编程语言环境

C/C++/Fortran 编译器

5、集群调度管理软件

集群采用的是 slurm 调度软件，浪潮 ClusterEngineV5.2 集群管理系统。

6、集群资源介绍

浪潮高性能集群目前有 1 个管理登录节点，28 个计算节点。

28 个计算节分别配置了 15 个瘦计算节点,9 个 GPU 节点和 4 个胖节点，分成三个分区，分别是 cpu、gpu、 fat 分区。

登录节点配置 24 个 CPU 核心，128G 内存。

cpu 分区配置了 15 个瘦计算节点，每个计算节点分别配置 52 Intel_5320 2.2GHz 个 CPU 核心，256G 内存；15 个 cpu 计算节点共 780 个 CPU 核心。

gpu 分区配置了 9 个 GPU 计算节点，每个计算节点分别配置 48 个 Intel_5318Y 2.1GHz CPU 核心，512G 内存，4 张 NVIDIA A800-80G PCIE 的 GPU 卡； 9 个节点共 432 个 CPU 核心，36 张 A800 的 GPU 卡。

fat 分区配置了 4 个胖计算节点，每个节点分别配置 96 个 Intel_6348H 2.3GHz CPU 核心，1024G 内存。

集群存储采用宏杉存储，可用空间约为 500T，挂载到所有节点的 home 目录，home 目录为共享目录。

用户的应用软件可安装在用户自己的家目录下面。常用的并行库及编译器及部分通用软件一般是集群管理员安装在/home/software 目录下，用户可直接调用/home/software 目录下的所有软件，/home/software 目录下的软件环境变量添加方法参考/home/sourcecode/env_file 文件调用即可。

二、集群登录与快速使用

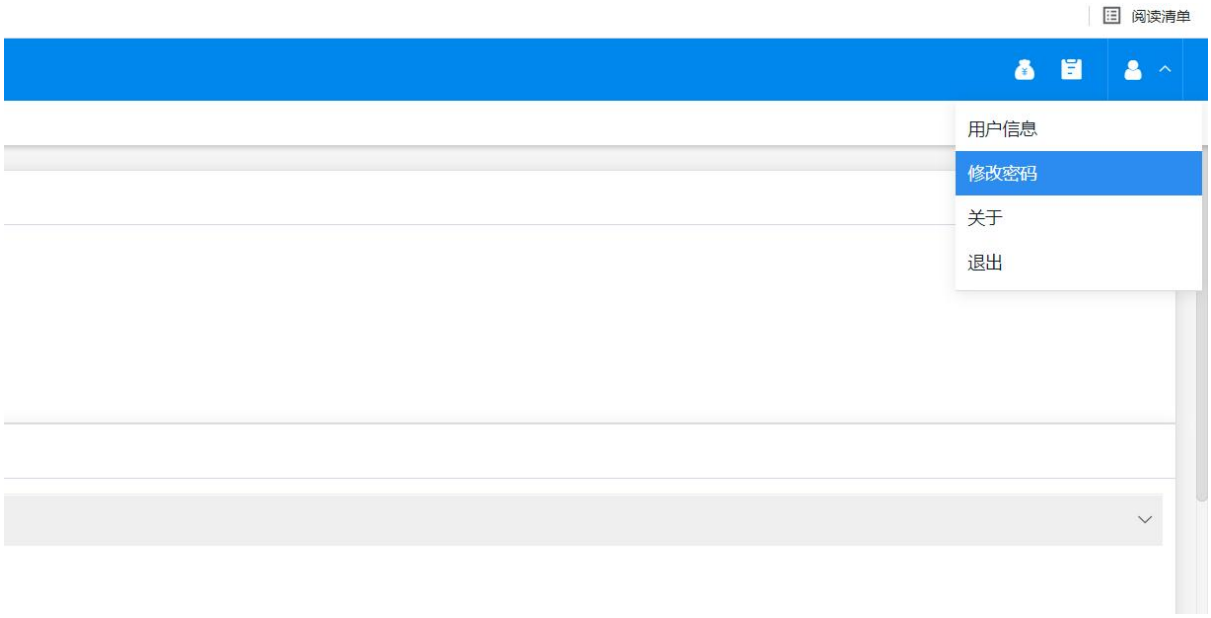
1、客户端与集群连通（网络）

slurm 集群登录节点为 10.27.3.2，端口为 22

2、密码修改

slurm 集群登录的初始化账号密码默认与一期 pbs 集群的账号密码一致，但是两个集群密码各自独立，用户可通过以下方法修改自己账号在 slurm 集群的密码。

浏览器登录 https://10.27.3.2/ued/login.html 地址，输入个人初始化的账号及密码，登录进去 web 后在右上方，点击修改密码



根据提示输入原有密码以及修改新的密码，点击提交（注意密码复杂度，防止被盗用）。修改完密码后可以用新的密码通过 shell 工具重新登录集群

3、数据上传及 slurm 作业脚本编写

用户通过文件传输工具将自己的数据及代码上传至集群，然后编写 slurm 作业脚本，参考下面的 slurm 作业模板

CPU 计算作业的 slurm 脚本

```
#!/bin/bash
#SBATCH --job-name=cpu-test          ##作业名称
#SBATCH --partition=cpu        ##作业申请的分区名称
#SBATCH --nodes=2                    ##作业申请的节点数
```

```
#SBATCH --ntasks-per-node=8          ##作业在每个节点申请的 CPU 核心数
#SBATCH --error=%j.err
#SBATCH --output=%j.out

CURDIR=`pwd`
rm -rf $CURDIR/nodelist.$SLURM_JOB_ID
NODES=`scontrol show hostnames $SLURM_JOB_NODELIST`
for i in $NODES
do
echo "$i:$SLURM_NTASKS_PER_NODE" >> $CURDIR/nodelist.$SLURM_JOB_ID
done
echo $SLURM_NPROCS

echo "process will start at : "
date
echo "++++++++++++++++++++++++++++++++++++++++"

export PATH=/home/software/software_path          加载软件的环境变量

Program excute Command                    程序执行的命令

echo "++++++++++++++++++++++++++++++++++++++++"
echo "processs will sleep 30s"
sleep 30
echo "process end at : "
date
rm -rf $CURDIR/nodelist.$SLURM_JOB_ID
```

GPU 计算作业的 slurm 脚本

```
#!/bin/bash
#SBATCH --job-name=gpu-test              ##作业名称
#SBATCH --partition=gpu                  ##作业申请的分区名称
#SBATCH --nodes=1                        ##作业申请的节点数
#SBATCH --ntasks-per-node=8              ##作业在每个节点申请的 CPU 核心数
#SBATCH --gres=gpu:1                     ##作业申请的 GPU 总数
#SBATCH --error=%j.err
#SBATCH --output=%j.out

CURDIR=`pwd`
rm -rf $CURDIR/nodelist.$SLURM_JOB_ID
NODES=`scontrol show hostnames $SLURM_JOB_NODELIST`
```

```
for i in $NODES
do
echo "$i:$SLURM_NTASKS_PER_NODE" >> $CURDIR/nodelist.$SLURM_JOB_ID
done
echo $SLURM_NPROCS
echo $SLURM_GPUS

echo "process will start at : "
date
echo "++++++++++++++++++++++++++++++++++++++++"

export PATH=/home/software/software_path          加载软件的环境变量

Program excute Command                            程序执行的命令

echo "++++++++++++++++++++++++++++++++++++++++"
echo "processs will sleep 30s"
sleep 30
echo "process end at : "
date
rm -rf $CURDIR/nodelist.$SLURM_JOB_ID
```

更多适合的脚本后续会放到/home/sourcecode/slurm-samples 路径下

4、作业提交、查看、删除

提交作业

sbatch 作业脚本名称

```
[inspur@mu01general-sample]$ sbatch slurm.sh
```

查看作业运行状态

squeue

```
[inspur@mu01general-sample]$ squeue
             JOBID  PARTITION           NAME           USER  ST                TIME      NODES
NODELIST(REASON)
               503 cpuPartit       bash     inspur    R          5:55          1 node009
               499 cpuPartit       bash     inspur    R         12:55          1 node009
```

查看作业详细信息

scontrol show job 作业 ID

[inspur@mu01~]$ scontrol show job 503

删除作业

scancel
[inspur@mu01general-sample]$ scancel 505

5、交互式提交作业

交互式提交作业有两种方式，第一种是通过 srun 命令提交交互式作业

第二种是通过 salloc 命令提交交互式作业

srun 命令提交交互式作业

srun 可以交互式提交运行并行作业，提交后，作业等待运行，等运行完毕后，才返回终端

[inspur@mu01~]$ srun -N 1 -n 1 -p cpuPartition hostname
node014
[inspur@mu01~]$

salloc 命令提交交互式作业

salloc 将获取作业需要的资源，当命令结束后需要手动执行 exit 命令释放所分配的资源

[inspur@mu01~]$ salloc -N 1 -n 1 -p cpuPartition
salloc: Granted job allocation 499
salloc: Waiting for resource configuration
salloc: Nodes node009 are ready for job
[inspur@mu01~]$ mpirun -np 8 vasp_std
[inspur@mu01~]$exit
exit
salloc: Relinquishing job allocation 499
[inspur@mu01~]$

三、slurm 与 pbs 命令对比

功能	pbs	slurm
作业名称	#PBS -N test	#SBATCH --job-name=test
指定队列/分区	#PBS -q cpu	#SBATCH --partition= cpu
指定作业使用节	#PBS -l nodes=1	#SBATCH --nodes=1

点数量		
指定作业在每个节点的 CPU 核心	#PBS -l ppn=1	#SBATCH --ntasks-per-node=8
指定具体某个节点	#PBS -l nodes=cu001	# SBATCH –nodelist=node001
指定 GPU 卡数	N/A	#SBATCH --gres=gpu:1
提交作业	qsub	sbatch
查看作业	qstat	squeue
删除作业	qdel	scancel
交互式提交作业	qsub -I 自动切换节点	salloc 无需切换节点
		srun 自动提交到远端节点