2021 年 PAC 大赛

平台简明使用手册

单位：北京并行科技股份有限公司
编写人：刘东

目录

一 环境介绍

1.1 硬件环境介绍

目前使用阿里云资源，有 1 个登录节点和 16 个计算节点。
计算节点硬件配置：
cpu：Intel(R) Xeon(R) Platinum 8369B CPU @ 2.90GHz 单节点 64 核
内存：256G
存储：本地磁盘 1000GB；
文件系统：共享存储 20T；分别给/opt 和/home 目录；

1.2 软件环境介绍

1.2.1 操作系统

管理节点操作系统版本：centos7.6
计算节点操作系统版本：centos7.6

1.2.2 目录划分

/opt 下是安装的共享软件目录，里面有 oneapi、slurm、munge、paramon、paratune

/home 用户的家目录；每个账户都有对应的一个家目录；账户只可以在自己家目录下安装软件，进行增加删除管理；

1.2.3 编译软件

oneapi 安装了基础模块和 HPC 模块；提供：icc、icpc 编译器、MKL 库和 ifort；

1.2.4 调度软件

slurm 提交作业需要通过 slurm 调度来分配任务；

1.2.5 性能分析软件

paramon、paratune 监控和性能分析软件；

二 集群使用

2.1 使用须知

1、集群每个账户只能提交 5 个作业（一个运行中的作业，4 个排队中的作业）；

2、用户修改自己账户密码：在账户下输入 passwd ，输入旧密码-回车，再输入新密码-回车，再确认新密码-回车（密码需要满足密码的的复杂度；）

3、请每个用户定期清理自己家目录下的文件；以免造成集群存储使用过高造成文件无法写入等问题；

2.2 集群登陆方式

通过 xshell 登录集群；使用通知到的账户和密码登录集群；
登录 IP：120.79.3.153 登录端口:22

2.3 文件上传下载

下载软件可以直接通过 wget 来下载到集群；
下载集群文件到 PC 机的话可以通过 XSHELL 的 xftp 来实现；
上传文件的话也可以通过 XSHELL 的 xftp 来实现；
*XSHELL 有商业版本和试用版本，请参赛队伍按照自己习惯准备工具

2.4 作业介绍

每个账户只有 1 个作业运行和 4 个作业排队（最多提交 5 个作业）只有运行完一个作业，才可以提交作业；

2.4.1 作业提交

通过 slurm 来提交作业
例如:slurm 提交作业脚本

```
#!/bin/bash
#SBATCH -J test            //指定提交作业名
#SBATCH -p compute    //指定提交队列
#SBATCH -N 1               //指定提交节点数
#SBATCH -n 32              //指定提交核数，一个节点 64 核
#SBATCH --error=slurm-%j.out //指定错误输出文件
#SBATCH --output=slurm-%j.out //指定正确输出文件
cd $SLURM_SUBMIT_DIR // cd 到提交任务目录
srun -N 1 -n 1 hostname >hostlist //  获取分配的节点列表， -N 指定几个节点，-n 指定每个节点一个进程即可
```

source /opt/intel/onrapi/setvars.sh
mpirun -np 32 -machinefile ./hostlist ./test

2.4.2 作业查看

squeue -j 396 （查看作业号是 396 的作业信息）
squeue -u root （查看用户 root 的作业信息）
squeue -p compute （查看提交到 compute 队列的作业信息）

2.4.3 作业删除

scancel 343 （取消作业号是 343 的作业）
scancel -n test （取消作业名是 test 的作业）

2.4.4 作业输出

查看提交作业目录下的 slurm-作业号.out 的文件内容；
提交作业时显示的作业错误信息；

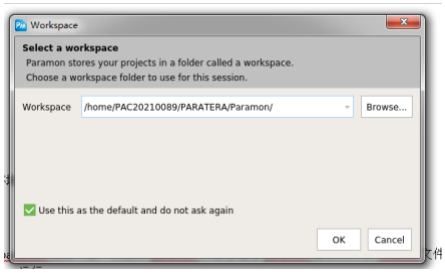
三 特征分析

3.1 paramon 使用

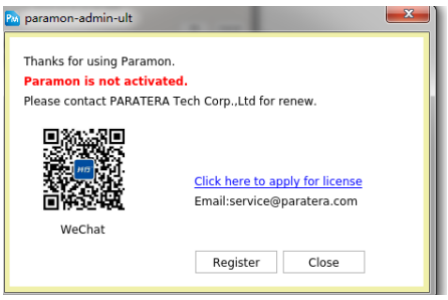
parmon 和 paratune 也可以在自己 PC 端使用；下载客户端和用户使用说明地址：https://www.paratera.com/pages/otherService/downloadSoftware.html

1、运行 paramon
命令：paramon-admin-ult

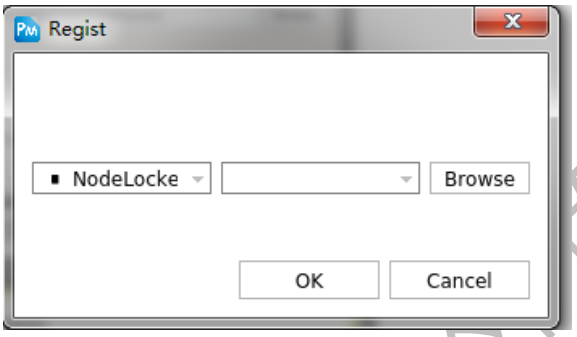
2、打开 paramon 界面，需要指定一个工作目录；（一般默认自己家目录下就可以）



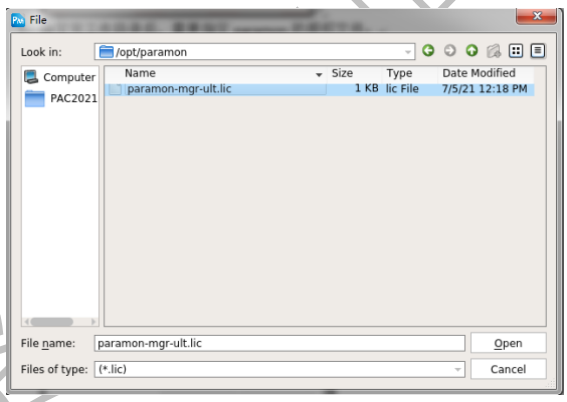
3、指定完工作目录后，需要指定 paramon 的授权文件；



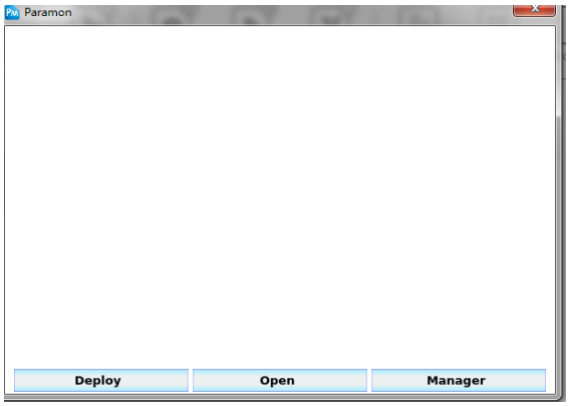
4、选择 Register 选项，进入选择授权文件界面



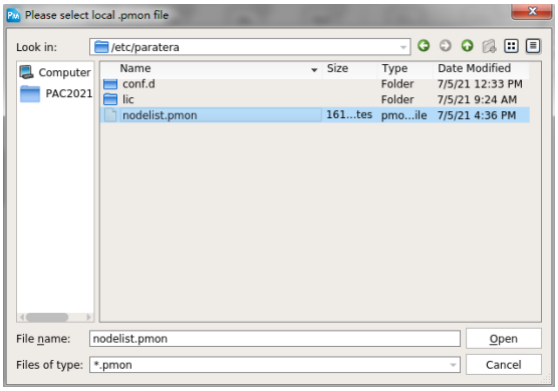
5、选择 Browse 选项，会打开文件目录，选择/opt/paramon/paramon-admin-ult.lic 文件，点击 open;在点击 OK；



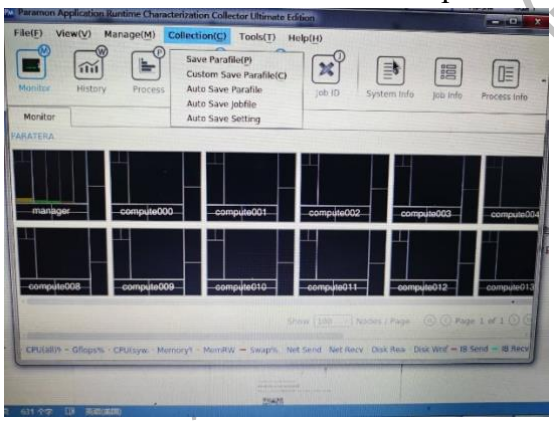
6、选择 open



7、选择/etc/paratera/nodelist.pmon 文件，点击 open，就会打开 paramon 功能界面；

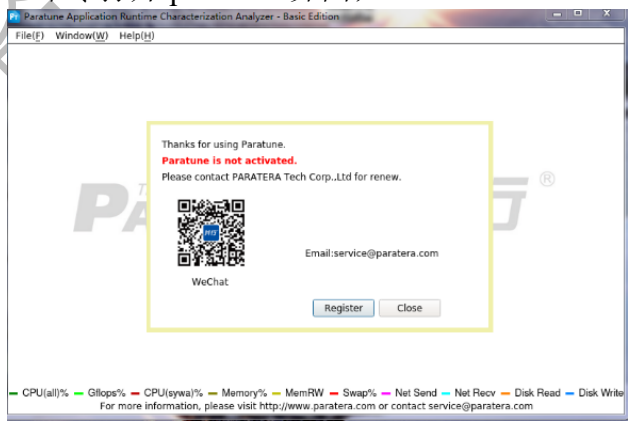


8、提取以.para 结尾的文件，打开 paratune 软件需要使用；提取.para 文件参考 paramon 用户使用手册；在和功能区选择 collection，在选择 cave parafile 或者 custom cave parafile 选项；运行一段时间后在 stop 就可以了；

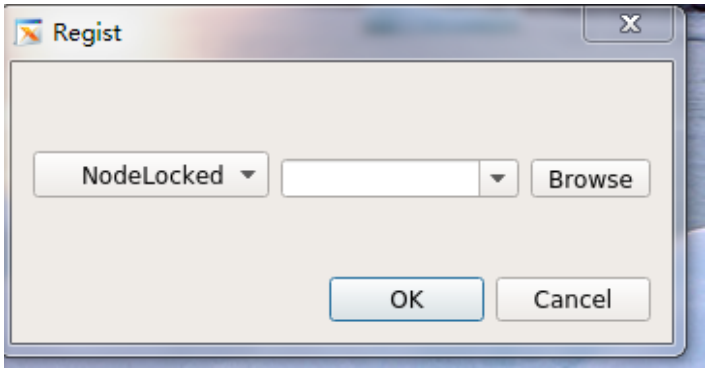


3.2 paratune 使用

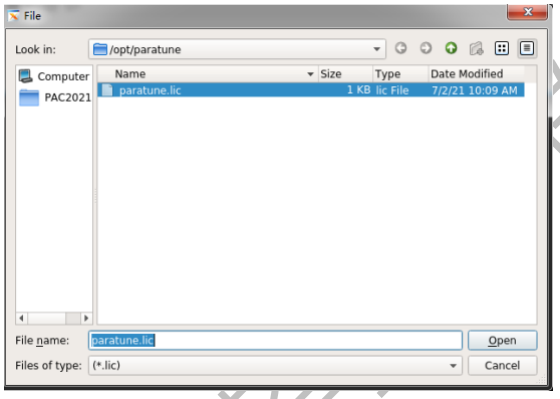
1、执行 paratune 命令打开 paratune 界面；



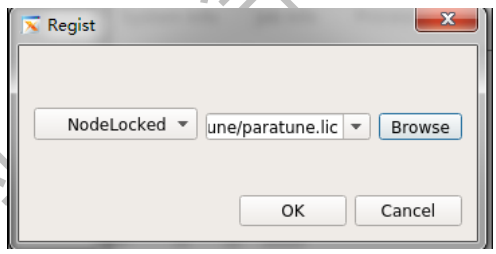
2、选择 Register 选项；选择 Browse；



3、选择/opt/paratune/下的 paratune.lic 授权文件，点击 open 就可以了；

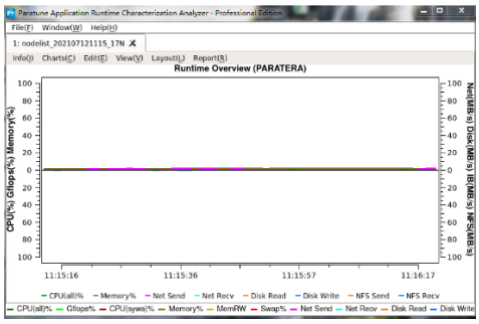


4、然后点击 OK 即可；



5、然后在功能区点击 file，点击 open

6、就会出现然你选择.para 结尾的文件；选中文件打开即可；



四 slurm 简要使用说明

使用 Slurm 作业管理系统，当前 debug 作业队列设置为节点可以共享，但作业独占 CPU core/GPU 资源。多个用户可以提交作业到同一个节点上，但是节点上 CPU core/GPU 资源只能被单一作业占有使用。作业管理系统常用命令如下：

4.1 sinfo 查看系统资源

sinfo 得到的结果是当前账号可使用的队列资源信息，如下图所示：

其中，

第一列 PARTITION 是队列名，默认能使用的队列名为 debug。

第二列 AVAIL 是队列可用情况，如果显示 up 则是可用状态；如果是 inact 则是不可用状态。第三列 TIMELIMIT 是作业运行时间限制，默认是 infinite 没有限制。

第四列 NODES 是节点数。

第五列 STATE 是节点状态，idle 是空闲节点，alloc 是已被占用节点，comp 是正在释放资源的节点，其他状态的节点都不可用，mix 是该节点有作业在运行或有程序占用 cpu 导致的。

第六列 NODELIST 是节点列表。

sinfo 的常用命令选项：

sinfo -n gm26　指定显示节点 gm26 的使用情况

sinfo -p debug　指定显示队列 debug 情况

其他选项可以通过 sinfo --help 查询

4.2 squeue 查看作业状态

squeue 得到的结果是当前账号正在运行作业的状态，如果 squeue 没有作业信息，说明作业已退出。

其中，

第一列 JOBID 是作业号，作业号是唯一的。

第二列 PARTITION 是作业运行使用的队列名。第三列 NAME 是作业名。

第四列 USER 是超算账号名。

第五列 ST 是作业状态，R 表示正常运行，PD 表示在排队，CG 表示正在退出，S 是管理员暂时挂起，只有 R 状态会计费。

第六列 TIME 是作业运行时间。

第七列 NODES 是作业使用的节点数。

第八列 NODELIST(REASON)对于运行作业（R 状态）显示作业使用的节点列表；对于排队作业（PD 状态），显示排队的原因。

squeue 的常用命令选项：

squeue -j 396　查看作业号为 396 的作业信息

squeue -u hutengteng　查看集群账号为 hutengteng 的作业信息

squeue -p debug 查看提交到 debug 队列的作业信息

squeue -w gm26 查看使用到 cn123 节点的作业信息

其他选项可通过 squeue --help 命令查看

4.3 srun 交互式提交作业

srun [options] program 命令属于交互式提交作业，有屏幕输出，但容易受网络波动影响，断网或关闭窗口会导致作业中断。

srun 命令示例：

srun -p debug -w gk[11-15] -N 2 -n 12 -t 20 --gres=gpu:2 my_job.sh

交互式提交 my_job.sh 程序。如果不关心节点和时间限制，可简写为 srun -p debug my_job.sh 其中，

-p debug 指定提交作业到 debug 队列；

-w gk[11-15] 指定使用节点 gk[11-15]；

-N 2 指定使用 2 个节点；

-n 12 指定运行的任务数为 12， 默认情况下一个 CPU 核一个任务

-t 20 指定作业运行时间限制为 20 分钟。

--gres 申请其他通用资源，例如 GPU、MIC 卡等。 参数的值为: “gpu:2”表明需要申请 2 块 GPU 显卡资源。

注意：

提交作业时不显式指定显卡资源，默认情况下不能使用显卡进行计算。即忽略 --gres 参数，作业只能使用 CPU 计算资源。

srun 的一些常用命令选项：

-N 3　指定节点数为 3

-n 12　指定任务数为 12， 默认一个 CPU 核一个任务

-p debug　指定提交作业到 debug 队列

-w gk[11-15]　指定提交作业到 gk11/gk12/gk13/gk14/gk15 节点

-x gm[11-12]　排除 gm11、gm12 节点

-o out.log　指定标准输出到 out.log 文件

-e err.log　指定重定向错误输出到 err.log 文件

-J JOBNAME　指定作业名为 JOBNAME

-t 20　限制运行 20 分钟

--gres=gpu:2　为作业分配 2 块 GPU 显卡资源 （最大值为 8）

srun 的其他选项可通过 srun --help 查看。

4.4 sbatch 后台提交作业

sbatch 一般情况下与 srun 一起提交作业到后台运行，需要将 srun 写到脚本中，再用 sbatch 提交脚本。这种方式不受本地网络波动影响，提交作业后可以关闭本地电脑。sbatch 命令没有屏幕输出，默认输出日志为提交目录下的 slurm-xxx.out 文件，可以使用 tail -f slurm-xxx.out 实时查看日志，其中 xxx 为作业号。

sbatch 命令示例 1（4 个进程提交 A.exe 程序）： 编写脚本 job1.sh，内容如下：

```
#!/bin/bash
srun -n 4　A.exe
```

然后在命令行执行 sbatch -p debug job1.sh 提交作业。脚本中的#!/bin/bash 是 bash 脚本的固定格式。从脚本的形式可以看出，提交脚本是一个 shell 脚本，因此常用的 shell 脚本语法都可以使用。作业开始运行后，在提交目录会生成一个 slurm-xxx.out 日志文件，其中 xxx 表示作业号。

sbatch 命令示例 2（指定 2 个节点，4 个任务，每个任务 12 个 cpu 核提交 A.exe 程序，限制运行 60 分钟）：编写脚本 job2.sh，内容如下：

```
#!/bin/bash
#SBATCH -N 2
#SBATCH -n 4
#SBATCH -c 12
#SBATCH -t 60
#SBATCH –gres=gpu:4
```

然后在命令行执行 sbatch -p debug job2.sh 就可以提交作业。其中 #SBATCH 注释行是 slurm 定义的作业执行方式说明，一些需要通过命令行指定的设置可以通过这些说明写在脚本里，避免了每次提交作业写很长的命令行。

sbatch 命令示例 3（一次提交多任务） 编写脚本 job3.sh，内容如下：

```
 #!/bin/bash
srun -n 8 A.exe &
srun -n 8 B.exe &
srun -n 8 C.exe &
wait
```

然后在命令行执行 sbatch -N 1 -p debug job3.sh，这里是单节点同时提交 3 个任务，每个任务使用 8 个进程。这里“wait” 需要 3 个任务全部执行完毕，作业才会退出。

sbatch 的一些常用命令选项基本与 srun 的相同，具体可以通过 sbatch --help 查看。

注意：

与 srun 一样，若作业需要使用 GPU 显卡资源， 需要在 sbatch 脚本中或在 sbatch 的命令行的参数中使用--gres=gpu:n (n 为显卡数量，不能超过 8)

4.5 salloc 分配模式作业提交

salloc 命令用于申请节点资源，一般用法如下：

1、执行 salloc -p debug；

2、执行 squeue 查看分配到的节点资源，比如分配到 gj16；

3、执行 ssh gj16 登陆到所分配的节点；

4、登陆节点后可以执行需要的提交命令或程序；

5、作业结束后，执行 scancel JOBID 释放分配模式作业的节点资源。

4.6 scancel 取消已提交的作业

scancel 可以取消正在运行或排队的作业。

scancel 的一些常用命令示例：

命令示例 功能

scancel 343 取消作业号为 343 的作业

scancel -n test 取消作业名为 test 的作业

scancel -p debug取消提交到 debug 队列的作业

scancel -t PENDING取消正在排队的作业

scancel -w cn100 取消运行在 cn100 节点上的作业

scancel 的其他参数选项，可通过 scancel --help 查看

4.7 scontrol 查看正在运行的作业信息

scontrol 命令可以查看正在运行的作业详情，比如提交目录、提交脚本、使用核数情况等，对已退出的作业无效。

scontrol 的常用示例：

scontrol show job 345

查看作业号为 345 的作业详情。

scontrol 的其他参数选项，可通过 scontrol --help 查看。

4.8 sacct 查看历史作业信息

sacct 命令可以查看历史作业的起止时间、结束状态、作业号、作业名、使用的节点数、节点列表、运行时间等。

sacct 的常用命令示例：

sacct -u hutengteng -S 2017-09-01 \

-E now --

field=jobid,partition,jobname,user,nnodes,nodelist,start,end,elapsed,state

其中，-u hutengteng 是指查看 hutengteng 用户的历史作业，-S 是开始查询时间，-E 是查询结束时间，

--format 定义了输出的格式：

- jobid 是指作业号
- partition 是指提交队列
- user 是指超算账号名，
- nnodes 是节点数，
- nodelist 是节点列表，
- start 是开始运行时间，
- end 是作业退出时间，
- elapsed 是运行时间，
- state 是作业结束状态。

sacct 的其他参数选项可通过 sacct --help 查看； 其他输出格式可通过 sacct --helpformat 查看。

4.9 Slurm 常用环境变量

SLURM_NPROCS 要加载的进程数
SLURM_TASKS_PER_NODE 每节点要加载的任务数
SLURM_JOB_ID 作业的 JobID
SLURM_SUBMIT_DIR 提交作业时的工作目录
SLURM_JOB_NODELIST 作业分配的节点列表
SLURM_JOB_CPUS_PER_NODE 每个节点上分配给作业的 CPU 数
SLURM_JOB_NUM_NODES 作业分配的节点数
HOSTNAME 对于批处理作业，此变量被设置为批处理脚本所执行节点的节点名

4.10 提交作业模板

```
#!/bin/bash
#SBATCH -J test //指定提交作业名
#SBATCH -p COMPUTE //指定提交队列
#SBATCH -N 1 //指定提交节点数
#SBATCH -n 32 //指定提交核数，一个节点 64 核
#SBATCH --error=slurm-%j.out //指定错误输出文件
#SBATCH --output=slurm-%j.out //指定正确输出文件
cd $SLURM_SUBMIT_DIR // cd 到提交任务目录
srun -N 1 -n 1 hostname >hostlist // 获取分配的节点列表， -N 指定几个节点，-n 指定每个节点一个进程即可
```

```
source /opt/intel/onrapi/setvars.sh
mpirun -np 32 -machinefile ./hostlist ./test
```

五 paratune 用户使用说明

https://www.paratera.com/pages/otherService/downloadProducts.html 该网址可以下载用户手册

六 paramon 用户使用说明

https://www.paratera.com/pages/otherService/downloadProducts.html 该网址可以下载用户使用手册