ALIBABA CLOUD

阿里云

弹性高性能计算
最佳实践

文档版本：20220713

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

1.最佳实践概览 05
2.使用LAMMPS软件进行工业仿真 07
3.使用Intel oneAPI编译运行LAMMPS 12
4.使用GROMACS进行分子动力学模拟 16
5.使用WRF软件进行气象模拟计算 23
6.使用BWA、GATK、Samtools软件进行基因测序 29
7.使用AutoDock Vina软件进行药物筛选 33
8.EDA上云最佳实践 38
9.使用OpenFOAM软件进行流体力学仿真计算 46
10.使用CP2K软件进行分子动力学研究 48
11.自动伸缩最佳实践 51
12.创建以CPFS为共享存储的E-HPC集群 54
13.配置E-HPC集群与Windows AD域用户账号互通 59
14.使用E-HPC集群调度器插件 68
14.1. E-HPC集群调度器插件 68
14.2. 构建调度器插件 70
14.3. 创建带有插件的集群 71
14.4. 附录：调度器插件的常用OpenAPI说明 73
15.测试E-HPC性能 76
15.1. 使用HPL测试集群浮点性能 76
15.2. 使用Stream测试集群内存带宽性能 79
15.3. 使用IMB软件和MPI通信库测试集群通信性能 82
15.4. 测试SCC集群性能 85

1.最佳实践概览

本文汇总典型场景中基于E-HPC集群完成计算任务的最佳实践。

案例类型	主要内容
使用HPL测试E-HPC浮点性能	HPL（The High-Performance Linpack Benchmark）是测试高性能计算集群系统浮点性能的基准。HPL通过对高性能计算集群采用高斯消元法求解一元N次稠密线性代数方程组的测试，评价高性能计算集群的浮点计算能力。
使用STREAM软件测试E-HPC内存带宽性能	STREAM软件是内存带宽性能测试的基准工具，也是衡量服务器内存性能指标的通用工具。STREAM软件支持复制（Copy）、尺度变换（Scale）、矢量求和（Add）、复合矢量求和（Triad）四种运算方式测试内存带宽的性能。
使用IMB软件和MPI通信库测试E-HPC通信性能	IMB（Intel MPI Benchmarks）用于评估HPC集群在不同消息粒度下节点间点对点、全局通信的效率。MPI（Message Passing Interface）是支持多编程语言的并行计算通信库，具有高性能、大规模性、可移植性、可扩展性等特点。
测试SCC集群性能	超级计算集群SCC（Super Computing Cluster）实例针对E-HPC多机并行计算需求，提供了低延时RDMA（Remote Direct Memory Access）网络互联。SCC实例无虚拟化损耗，同时提供VPC网络隔离能力，您可直接访问硬件资源。
使用LAMMPS软件进行高性能计算	LAMMPS（Large-scale Atomic/Molecular Massively Parallel Simulator）是一款经典分子动力学软件。 LAMMPS包含的势函数可用于固体材料（金属、半导体）、软物质（生物大分子，聚合物）、粗粒化或介观尺度模型体系。
使用GROMACS软件进行高性能计算	GROMACS（GROningen MAchine for Chemical Simulations）是一款通用软件，用于对具有数百万颗粒子的系统进行基于牛顿运动方程的分子动力学模拟。GROMACS主要用于生物化学分子，如蛋白质、脂质等具有多种复杂键合相互作用的核酸分析。
使用OpenFOAM软件进行流体力学仿真计算计算	OpenFOAM（Open Source Field Operation and Manipulation）是对连续介质力学问题进行数值计算的软件。可进行数据预处理、后处理和自定义求解器，常用于计算流体力学领域。
使用WRF软件进行气象模拟计算	WRF（Weather Research and Forecasting）采用新一代中尺度天气预报模式，是气象行业广泛应用的开源气象模拟软件。它为研究大气过程提供了大量选项，并可以在多种计算平台运行。

案例类型	主要内容
使用TensorFlow软件进行高性能计算	TensorFlow是一个为深度神经网络开发的功能强大的开源软件库，被广泛应用于机器学习算法的编程实现。
创建以CPFS为共享存储的E-HPC集群	CPFS（Cloud Paralleled File System）是一种高性能并行文件存储系统，专为AI训练和E-HPC等高性能计算场景打造，最大支持数十GB/s吞吐能力。以CPFS为共享存储的E-HPC集群适合动画渲染、生命科学、气象预报、能源勘探等需要超高吞吐的应用场景。
自动伸缩最佳实践	当您需要每天不定时提交作业，使用E-HPC集群几个小时进行大规模计算， 然后释放节点。您可以针对不同的作业类型，配置不同的伸缩策略。
使用BWA、GATK、Samtools软件进行基因测序	在进行基因测序时，您可以使用BWA构建索引及比对记录，再使用Samtools对比对记录进行排序，然后使用GATK去除重复序列、重新校正碱基质量值、变异检查。

2.使用LAMMPS软件进行工业仿真

本文介绍如何使用弹性高性能计算E-HPC运行LAMMPS开源仿真软件，以3d Lennard-Jones melt模型进行工业仿真，并通过可视化的方式查看仿真结果。

背景信息

LAMMPS（Large-scale Atomic/Molecular Massively Parallel Simulator）是一款经典分子动力学软件。LAMMPS包含的势函数可用于固体材料（金属、半导体）、软物质（生物大分子，聚合物）、粗粒化或介观尺度模型体系。

E-HPC作为高性能且弹性的计算集群，可为复杂工程和力学结构提供辅助分析，通过大量数据仿真模拟优化产品结构和性能，目前在工业仿真领域的应用越来越多。

准备工作

1. 登录弹性高性能计算控制台。
2. 创建一个名为LAMMPS的集群。

   具体操作，请参见使用向导创建集群。您需要注意以下配置参数：

   - **部署方式**：选择**标准**。本示例计算量不大，登录节点1个，管控节点2个，计算节点2个，实际使用中可以依据作业计算量弹性增加计算节点的数量。
   - **计算节点**：选择vCPU≥32的节点，例如：ecs.c7.8xlarge。
   - **调度器**：选择**pbs**。
   - **VNC**：打开VNC开关，打开后可以自动部署远程可视化窗口。
3. 安装软件。

   i. 在**集群**页面，找到目标球集群，单击**软件管理**后的图标。

   ii. 在**可安装软件**页签下，选中lammps-openmpi 31Mar17、openmpi 1.10.7、ehpc-app-server_1.1.4和vmd 1.9.3软件复选框。

   iii. 在页面右上角，单击**安装**。

   iv. 在弹出的**安装软件**对话框中，单击**确定**。

   v. 在**已安装软件**页签下，可查看安装的软件。
4. 创建一个名为testuser的普通用户。

   具体操作，请参见创建用户。

操作步骤

1. 下载并解压算例。

   i. 在**集群**页面，找到目标集群，单击**远程连接**。

   ii. 在**远程连接**页面，输入testuser的用户名、密码和端口，单击**ssh连接**。

iii. 执行如下命令下载并解压算例。

```
cd /home/testuser
wget https://code.aliyun.com/best-practice/21/raw/master/LammpsTest.zip
unzip LammpsTest.zip
```

解压后的主要作业文件包括：

- lj.in算例文件
- 作业执行脚本lammps.pbs

2. 编辑作业文件。

i. 在**集群**页面，在对应集群信息的右上角，单击**作业**。

ii. 在**作业**页面左上角，单击**作业文件编辑**。

iii. 单击**浏览集群文件**，并在弹出的**请输入集群用户名和密码**对话框中，输入testuser的用户名和密码后，单击**确定**。

iv. 在/home/testuser/LammpsTest/lammps.pbs作业脚本文件上，右键单击**打开**，您可以修改lammps.pbs作业脚本文件并保存。

作业文件配置如下：

```
#!/bin/sh
#PBS -l select=1:ncpus=32:mpiprocs=32
#本示例使用1个计算节点的32 vCPU，使用32个MPI任务进行高性能计算。实际测试中根据节点配置设置CPU个数。算力要求vCPU≥32。

#PBS -j oe

exportMODULEPATH=/opt/ehpcmodulefiles/     #module命令依赖的环境变量
module load lammps-openmpi/31Mar17
module load openmpi/1.10.7

echo"run at the beginning"
mpirun lmp -in./LammpsTest/lj.in       #需根据实际修改lj.in文件的路径
```

3. 提交作业。

i. 在**作业**页面，单击**提交作业**。

ii. 根据界面提示，配置作业参数。

配置作业名和作业执行命令，其他参数保持默认。

- **作业名**：请输入作业名lammps。
- **作业执行命令**：请输入作业执行命令./LammpsTest/lammps.pbs。

iii. 在**作业**页面右上角，单击**提交作业**。

iv. 在弹出的对话框中，输入testuser用户名和密码后，单击**确定**。

v. 在**作业列表**中查看作业完成情况。

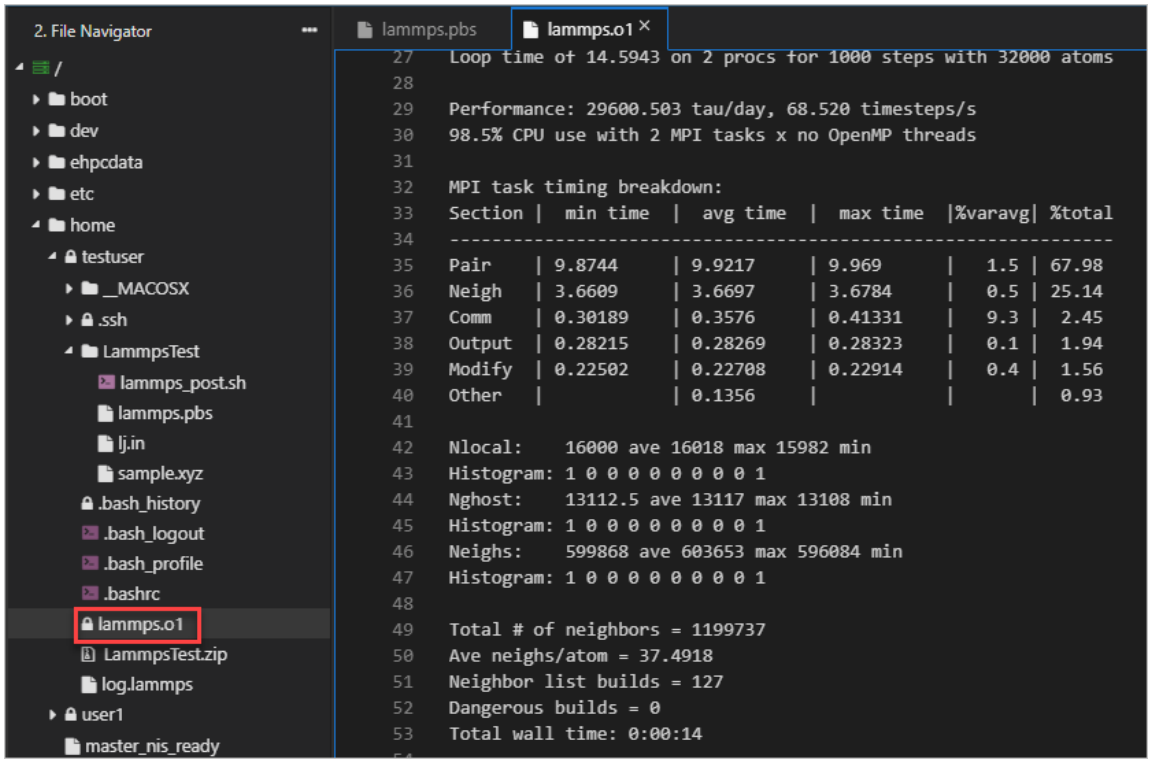
当状态为**FINISHED**时，表示作业已运行完成。

查看作业结果

1. 登录弹性高性能计算控制台。
2. 在**集群**页面，找到目标集群，单击**远程连接**。
3. 在**远程连接**页面，输入testuser的用户名、密码和端口，单击**ssh连接**。
4. 执行如下命令，查看LAMMPS作业结果文件。

```
[lammps@login0~]$ls
lammps.o1  LammpsTest  LammpsTest.zip  log.lammps  __MACOSX
[lammps@login0~]$cat lammps.o1
-------
-------
MPI task timing breakdown:
Section |  min time  |  avg time  |  max time  |%varavg| %total
---------------------------------------------------------------
Pair    | 9.8744     | 9.9217     | 9.969      |   1.5 | 67.98
Neigh   | 3.6609     | 3.6697     | 3.6784     |   0.5 | 25.14
Comm    | 0.30189    | 0.3576     | 0.41331    |   9.3 |  2.45
Output  | 0.28215    | 0.28269    | 0.28323    |   0.1 |  1.94
Modify  | 0.22502    | 0.22708    | 0.22914    |   0.4 |  1.56
Other   |            | 0.1356     |            |       |  0.93

Nlocal:    16000 ave 16018 max 15982 min
Histogram: 1 0 0 0 0 0 0 0 0 1
Nghost:    13112.5 ave 13117 max 13108 min
Histogram: 1 0 0 0 0 0 0 0 0 1
Neighs:    599868 ave 603653 max 596084 min
Histogram: 1 0 0 0 0 0 0 0 0 1

Total # of neighbors = 1199737
Ave neighs/atom = 37.4918
Neighbor list builds = 127
Dangerous builds = 0
Total wall time: 0:00:14
```

如果您不指定作业标准输出路径，则默认按照调度器行为生成输出文件。默认作业结果文件输出/home/<用户名>/目录下，本示例中的作业结果文件在/home/testuser/lammps.o1下。

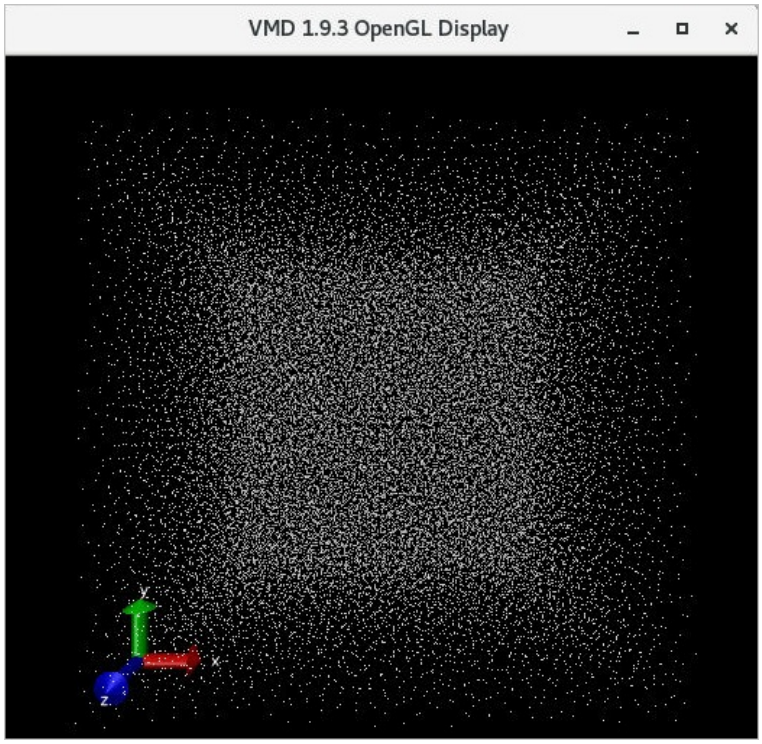


5. 使用VNC远程可视化查看作业结果。
   i. 在左侧导航栏，单击**集群**。
   ii. 在**集群**页面，找到目标集群，单击**更多 > VNC**。
   iii. 使用VNC远程连接可视化服务。

   具体操作，请参见连接可视化服务。
   iv. 在可视化桌面，右键单击**Open Terminal**。
   v. 在Terminal窗口运行 `/opt/vmd/1.9.3/vmd` ，打开VMD软件。
   vi. 在VMD Main对话框中，选择**File > New Molecule...**。
   vii. 单击**Browse...**选择结果文件sample.xyz。

> **说明**
>
> sample.xyz文件的路径为/home/testuser/lammpsTest/sample.xyz。

viii. 单击Load，可在VMD 1.9.3 OpenGL Display窗口查看可视化结果。



3.使用Intel oneAPI编译运行LAMMPS

E-HPC集群集成了Intel oneAPI工具包，该工具包结合HPC软件使用，可以加快构建跨架构应用程序。本文以LAMMPS软件为例，为您演示如何在E-HPC集群下使用Intel oneAPI编译并运行LAMMPS。

背景信息

Intel oneAPI是一种开放的、标准的统一编程模型。Intel oneAPI工具包为Intel CPU和FPGA等异构平台提供构建部署应用程序和解决方案所需的工具，包括用于高性能异构计算的优化编译器、库、框架和分析工具，可以简化编程，帮助开发者提高生产力。

LAMMPS是一个经典的分子动力学模拟代码，用于模拟液态、固态或气态的粒子集合。在模拟原子、分子计算中并行效率高，广泛应用于材料、物理、化学等模拟场景。

使用Intel oneAPI编译运行LAMMPS，可以加快构建应用程序，提升应用性能。

准备工作

1. 创建E-HPC集群。具体操作，请参见使用向导创建集群。

   配置集群时，相关参数如下：

   - 硬件参数：部署方式为标准，包含2个管控节点，1个计算节点和1个登录节点，均采用ecs.c7.large实例规格，配置为2 vCPU，4 GiB内存，Ice Lake处理器，2.7 GHz。
   - 软件配置：镜像选择CentOS 7.6公共镜像，调度器选择pbs。

2. 创建集群用户。具体操作，请参见创建用户。

   集群用户用于登录集群，进行编译软件、提交作业等操作，配置用户权限时，权限组请选择**sudo权限组**。

3. 安装oneAPI工具包。具体操作，请参见安装软件。

   需安装的软件如下：

   - intel-oneapi-mkl，版本为2022.1.2。
   - intel-oneapi-mpi，版本为2022.1.2。
   - intel-oneapi-hpckit，版本为2022.1.2。

步骤一：编译LAMMPS

1. 登录E-HPC集群。

   登录时，请使用具有sudo权限的用户。具体操作，请参见登录集群。

2. 执行以下命令，下载最新的LAMMPS源码。

   i. 从GitHub下载LAMMPS源码。

   ```
   git clone -b release https://github.com/lammps/lammps.git mylammps
   ```

ii. 查看下载的LAMMPS源码文件。

```
ls -al
```

预期返回：

```
...
drwxr-xr-x 15 test users 4096 May 31 16:39 mylammps
...
```

3. 执行以下命令，加载oneAPI模块。

i. 将环境变量写入 `$HOME/.bashrc` 。

```
vim $HOME/.bashrc
```

添加以下内容：

```
source /opt/intel-oneapi-mpi/oneapi/setvars.sh --force
source /opt/intel-oneapi-mkl/oneapi/setvars.sh --force
source /opt/intel-hpckit/oneapi/setvars.sh --force
```

ii. 更新 `$HOME/.bashrc` 。

```
source $HOME/.bashrc
```

4. 执行以下命令，编译LAMMPS。

i. 使用2个进程进行编译。

```
cd /$HOME/mylammps/src
make -j 2 intel_cpu_intelmpi
```

ii. 查看当前文件路径下生成的LAMMPS可执行文件。

```
ll lmp_intel_cpu_intelmpi
```

预期返回：

```
-rwxr-xr-x 1 test users 9041824 May 31 16:48 lmp_intel_cpu_intelmpi
```

5. 执行以下命令，将LAMMPS可执行文件配置为共享命令。

```
mkdir -p $HOME/bin
mv /$HOME/mylammps/src/lmp_intel_cpu_intelmpi  /$HOME/bin
```

步骤二：运行LAMMPS

1. 切换到lmp_intel_cpu_intelmpi所在bin目录。

```
cd /$HOME/bin
```

2. 执行以下命令创建算例文件，算例文件命名为in.intel.lj。

```
vim in.intel.lj
```

内容示例如下：

```
# 3d Lennard-Jones melt

variable        x index 1
variable        y index 1
variable        z index 1

variable        xx equal 20*$x
variable        yy equal 20*$y
variable        zz equal 20*$z

units           lj
atom_style      atomic

lattice         fcc 0.8442
region          box block 0 ${xx} 0 ${yy} 0 ${zz}
create_box      1 box
create_atoms    1 box
mass            1 1.0

velocity        all create 1.44 87287 loop geom

pair_style      lj/cut 2.5
pair_coeff      1 1 1.0 1.0 2.5

neighbor        0.3 bin
neigh_modify    delay 0 every 20 check no

fix             1 all nve
dump 1 all xyz 100 sample.xyz
run             10000
```

3. 执行以下命令编写测试脚本，脚本命名为test.pbs。

```
vim test.pbs
```

脚本内容如下：

```
#!/bin/bash
#PBS -N testLmp              #设置作业名称
#PBS -l nodes=2:ppn=2        #向调度器申请2个计算节点，每个计算节点使用两个进程运行该作业
export I_MPI_HYDRA_BOOTSTRAP=ssh
cd $PBS_O_WORKDIR
mpirun ./lmp_intel_cpu_intelmpi -in ./in.intel.lj
```

4. 执行以下命令，提交作业。

```
qsub test.pbs
```

预期返回如下，表示生成的作业ID为0.scheduler。

```
0.scheduler
```

查看结果

1. 执行以下命令，查看作业状态。

```
qstat -x 0.scheduler
```

预期返回如下，当返回信息中 `S` 为 `F` 时，表示作业已经运行结束。

```
Job id            Name             User             Time Use S Queue
----------------  ---------------- ----------------  -------- - -----
0.scheduler       test.pbs         test              00:00:00 F workq
```

2. 执行以下命令，查看日志。

```
cat log.lammps
```

预期返回：

```
...
Per MPI rank memory allocation (min/avg/max) = 11.75 | 11.75 | 11.75 Mbytes
   Step          Temp          E_pair         E_mol          TotEng         Press
         0   1.44          -6.7733681      0             -4.6134356     -5.0197073
     10000   0.69579461    -5.6648333      0             -4.621174       0.7601771
Loop time of 108.622 on 4 procs for 10000 steps with 32000 atoms

Performance: 39770.920 tau/day, 92.062 timesteps/s
97.0% CPU use with 2 MPI tasks x 2 OpenMP threads

MPI task timing breakdown:
Section |  min time  |  avg time  |  max time  |%varavg| %total
---------------------------------------------------------------
Pair    | 85.42      | 85.632     | 85.844     |   2.3 | 78.83
Neigh   | 13.523     | 13.564     | 13.604     |   1.1 | 12.49
Comm    | 4.4182     | 4.5452     | 4.6722     |   6.0 |  4.18
Output  | 2.1572     | 2.1683     | 2.1793     |   0.7 |  2.00
Modify  | 2.1047     | 2.1398     | 2.175      |   2.4 |  1.97
Other   |            | 0.5734     |            |       |  0.53

Nlocal:          16000 ave       16007 max       15993 min
Histogram: 1 0 0 0 0 0 0 0 0 1
Nghost:          13030 ave       13047 max       13013 min
Histogram: 1 0 0 0 0 0 0 0 0 1
Neighs:         600054 ave      604542 max      595567 min
Histogram: 1 0 0 0 0 0 0 0 0 1

Total # of neighbors = 1200109
Ave neighs/atom = 37.503406
Neighbor list builds = 500
Dangerous builds not checked
Total wall time: 0:01:48
```

4.使用GROMACS进行分子动力学模拟

本文以GROMACS软件为例介绍如何在E-HPC上进行高性能计算。

背景信息

GROMACS（GROningen MAchine for Chemical Simulations）是一款通用软件，用于对具有数百万颗粒子的系统进行基于牛顿运动方程的分子动力学模拟。

GROMACS主要用于生物化学分子，如蛋白质、脂质等具有多种复杂键合相互作用的核酸分析。GROMACS计算典型的模拟应用，如高效地计算非键合相互作用，许多研究人员用其研究非生物系统的聚合物。

GROMACS支持分子动力学的常见算法，可以采用GPU来加速核心计算过程。更多信息，请参见GROMACS官网。

相关算例

- 算例1：水中的溶菌酶

  本算例为一个蛋白质加上离子在水盒子里的模拟过程。更多信息，请参见官方教程和非官方中文教程。

  算例下载地址：https://public-ehpc-package.oss-cn-hangzhou.aliyuncs.com/Lysozyme.tar.gz

- 算例2：水分子运动

  本算例为模拟大量水分子在给定空间、温度内的运动过程。

  算例下载地址：https://public-ehpc-package.oss-cn-hangzhou.aliyuncs.com/water_GMX50_bare.tar.gz

准备工作

1. 登录弹性高性能计算控制台。
2. 创建E-HPC集群。

   具体操作，请参见创建集群。请注意以下配置参数：

   - **计算节点**：选择GPU机型，如ecs.gn5-c8g1.2xlarge。
   - **VNC**：打开VNC开关，打开后可以自动部署远程可视化窗口。
3. 安装软件。

   i. 在**集群**页面，找到目标集群，单击**软件管理**后的图标。

   ii. 在**可安装软件**页签下，选中gromacs-gpu 2018.1、openmpi 3.0.0、cuda-toolkit 9.0、vmd 1.9.3软件复选框。

   iii. 在页面右上角，单击**安装**。

   iv. 在弹出的**安装软件**对话框中，单击**确定**。

   v. 在**已安装软件**页签下，可查看安装的软件。
4. 创建一个名为gmx.test的sudo用户。

   具体操作，请参见创建用户。

操作步骤

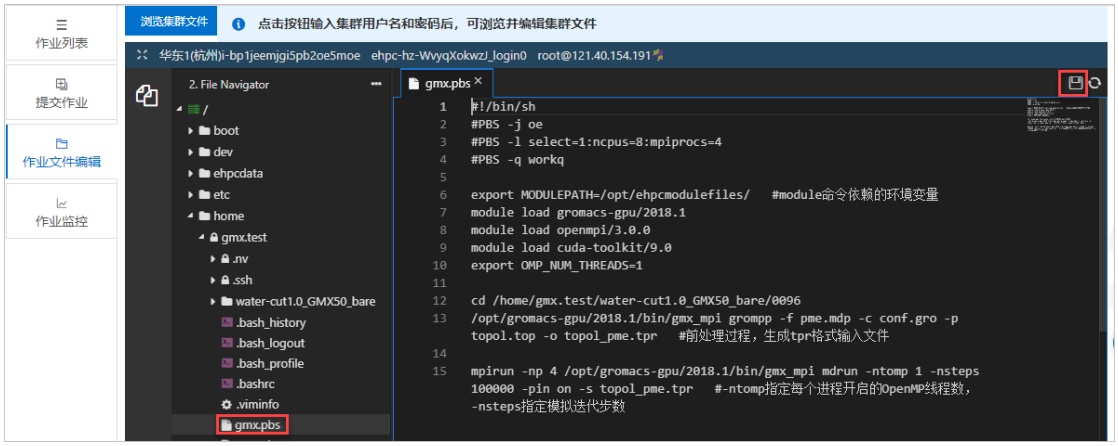
1. 下载并解压算例（本示例使用算例2的相关文件）。
   i. 在**集群**页面，找到目标集群，单击**远程连接**。
   ii. 在**远程连接**页面，输入gmx.test的用户名、密码和端口，单击**ssh连接**。
   iii. 执行如下命令，下载并解压算例。

```
cd /home/gmx.test
wget https://public-ehpc-package.oss-cn-hangzhou.aliyuncs.com/water_GMX50_bare.tar.gz
tar xzvf water_GMX50_bare.tar.gz
chown -R gmx.test water-cut1.0_GMX50_bare
chgrp -R users water-cut1.0_GMX50_bare
```

2. 在**集群**页面，在对应集群信息的右上角，单击**作业**。

3. 编辑作业文件。
   i. 在**作业**页面左上角，单击**作业文件编辑**。
   ii. 单击**浏览集群文件**，并在弹出的**请输入集群用户名和密码**对话框中，输入gmx.test的用户名和密码后，单击**确定**。
   iii. 在/home/gmx.test/water-cut1.0_GMX50_bare路径下右键单击**新建文件**，输入标题后单击**确定**。例如：gmx.pbs。

iv. 在gmx.pbs文件上，右键单击**打开**，将以下代码拷贝到gmx.pbs文件中并保存。



作业文件配置如下：

> **说明**
>
> 本示例使用名为gmx.test的用户提交作业，在一个包含8个CPU核和1块P100 GPU卡的计算节点compute9上运行。在实际使用场景中您可根据集群配置情况做出适当修改。

```
#!/bin/sh
#PBS -j oe
#PBS -l select=1:ncpus=8:mpiprocs=4
#PBS -q workq

#module命令依赖的环境变量
export MODULEPATH=/opt/ehpcmodulefiles/
module load gromacs-gpu/2018.1
module load openmpi/3.0.0
module load cuda-toolkit/9.0
export OMP_NUM_THREADS=1

cd /home/gmx.test/water-cut1.0_GMX50_bare/0096

#前处理过程，生成tpr格式输入文件
/opt/gromacs-gpu/2018.1/bin/gmx_mpi grompp -f pme.mdp -c conf.gro -p topol.top -o t
opol_pme.tpr

#-ntomp指定每个进程开启的OpenMP线程数，-nsteps指定模拟迭代步数
mpirun -np 4 /opt/gromacs-gpu/2018.1/bin/gmx_mpi mdrun -ntomp 1 -eps nst100000 -pin
on -s topol_pme.tpr
```

4. 提交作业。

   i. 在**作业**页面，单击**提交作业**。

   ii. 根据界面提示，配置作业参数。

   配置作业名和作业执行命令，其他参数保持默认。

   - **作业名**：请输入作业名gmx.test。
   - **作业执行命令**：请输入作业执行命令./gmx.pbs，表示执行gmx.pbs文件。

iii. 在**作业**页面右上角，单击**提交作业**。

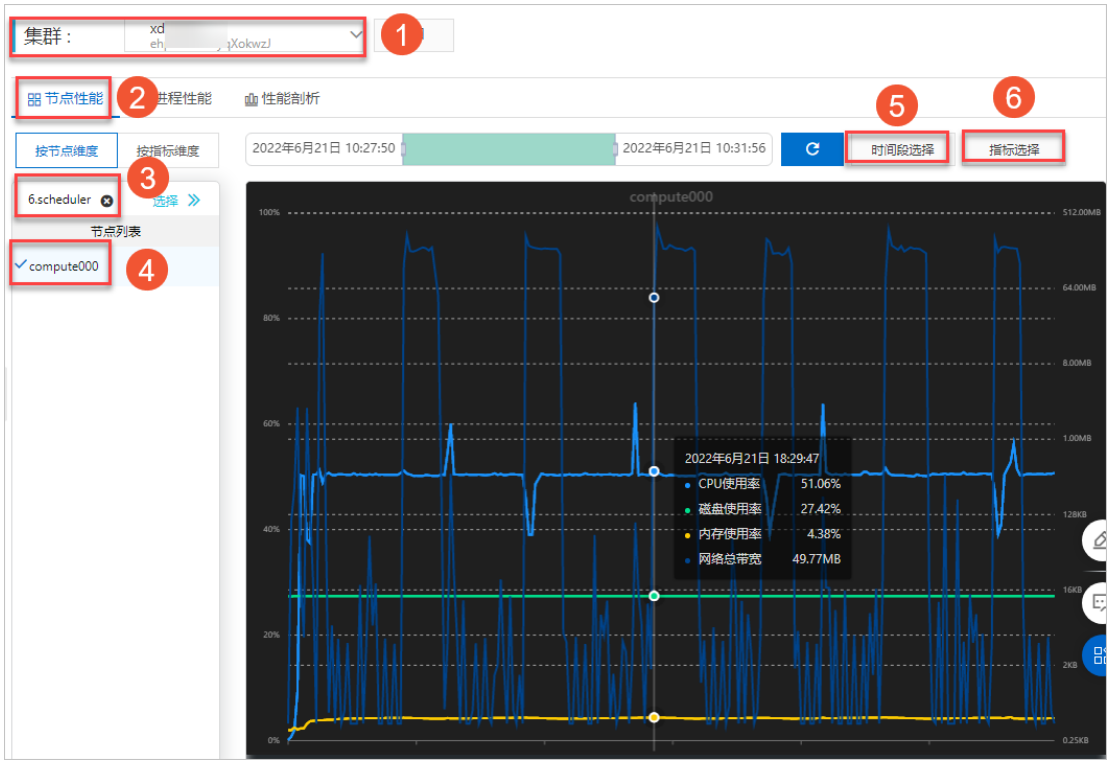
iv. 在弹出的对话框中，输入gmx.test用户名和密码后，单击**确定**。

查看作业计算性能和结果

1. 在左侧导航栏，选择**作业与性能管理 > 作业**。
2. 单击目标作业列表右侧**详情**，可以查看作业详细信息。

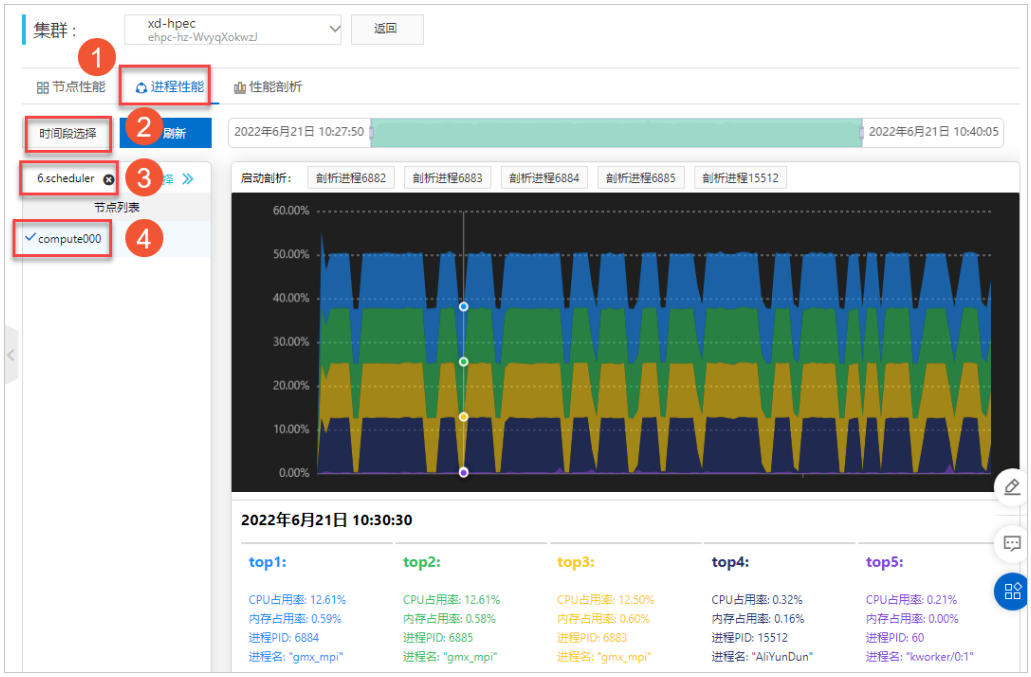
> **说明**
>
> 作业运行需要一定的时间，请您耐心等待。

3. 查看本次作业计算性能。
   i. 在左侧导航栏，选择**作业与性能管理 > E-HPC优化器**。
   ii. 按照下图选择目标集群，然后在**节点性能**页签下，依次选择作业、节点、时间段和指标，查看指标性能。

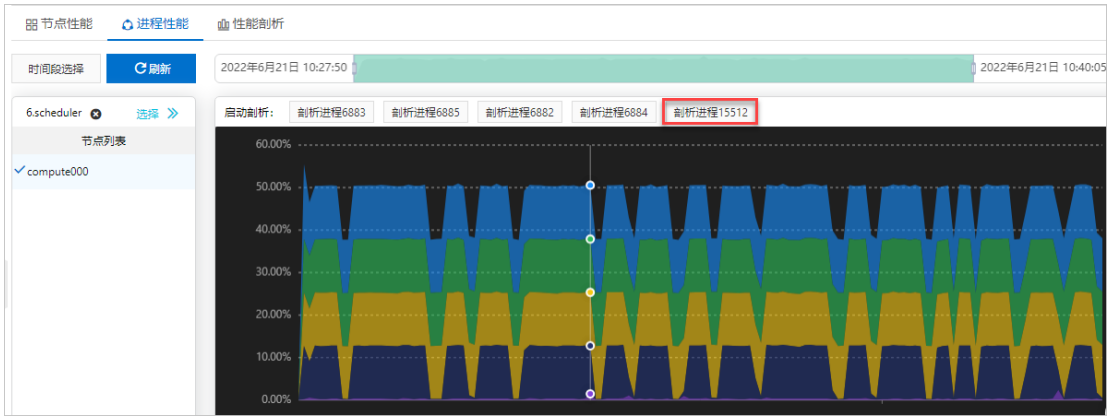


4. 单击**进程性能**页签，选择时间段、作业和节点，查看当前CPU使用率Top5的进程信息。

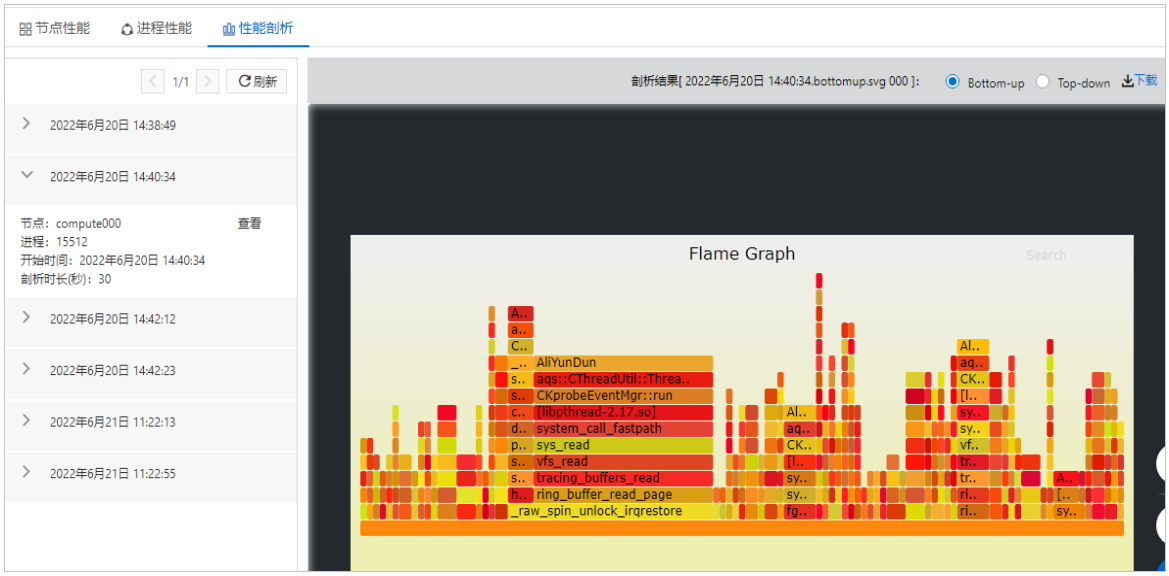
时间段支持选择5分钟、10分钟、15分钟、1小时、4小时、12小时和1天，您可根据需求选择对应的时间段，默认选择10分钟。



5. 单击进程图，然后单击剖析进程中您想剖析的进程，在弹出的对话框中设置剖析时长和采样频率，启动对GROMACS作业的实时性能剖析，获取热点函数的剖析图。



6. 单击**性能剖析**页签，查看剖析结果。

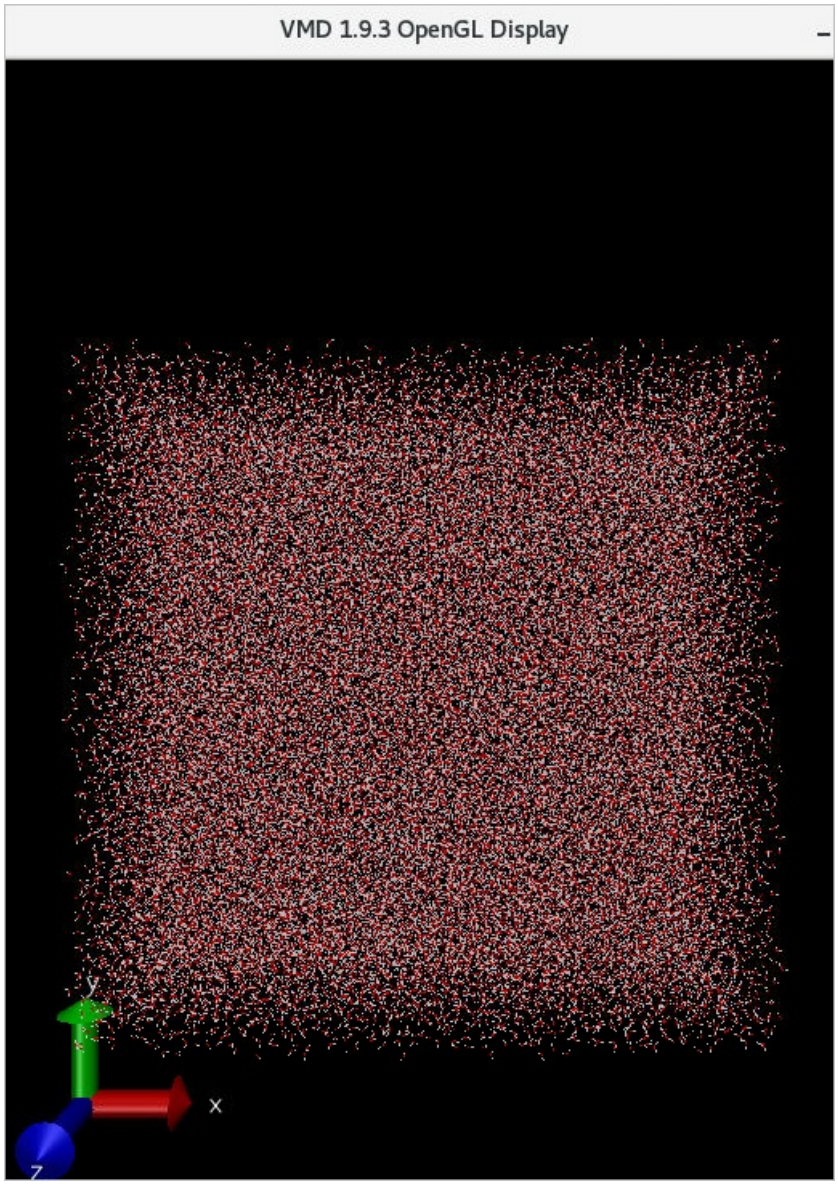


7. 使用VNC远程可视化查看作业结果。
   i. 在左侧导航栏，单击**集群**。
   ii. 在**集群**页面，找到目标集群，单击**更多** > **VNC**。
   iii. 使用VNC远程连接可视化服务。

   具体操作，请参见连接可视化服务。

   iv. 在Terminal窗口运行 `/opt/vmd/1.9.3/vmd` ，打开VMD软件。
   v. 在VMD Main对话框中，选择**File** > **Read Molecule**，选择/home/gmx.test/water-cut1.0_GMX50_bare/0096路径下的conf.gro文件后并导入。
   vi. 在conf.gro文件上右键选择**File** > **Read Molecule**，选择/home/gmx.test/water-cut1.0_GMX50_bare/0096路径下的traj.trr文件后并导入。
   vii. 在VMD Main对话框中，单击**Graphics**。

viii. 在Graphicl Representations对话框中，可以选择Coloring Method、Drawing Method等，修改模拟样式。

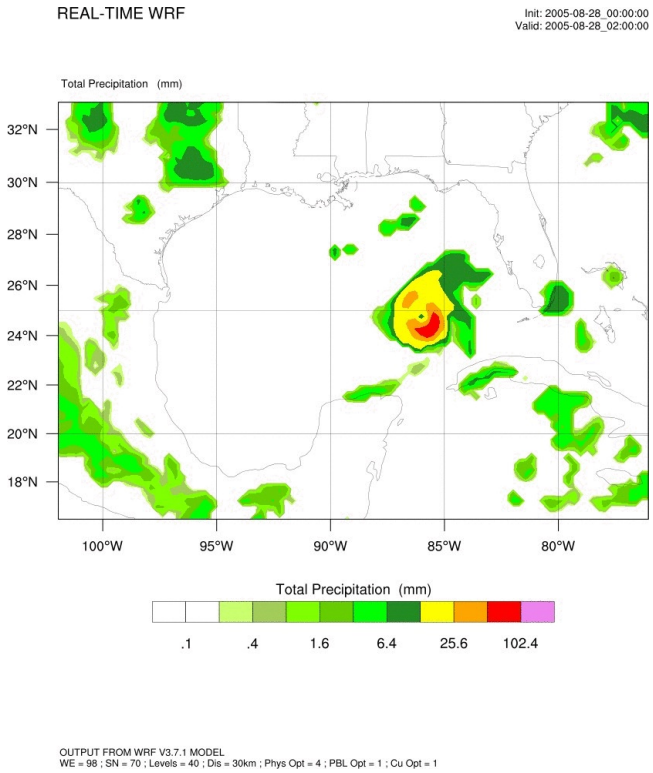


5.使用WRF软件进行气象模拟计算

本文介绍如何使用E-HPC集群运行WRF软件进行气象模拟计算。

背景信息

WRF（Weather Research and Forecasting）采用新一代中尺度天气预报模式，是气象行业广泛应用的开源气象模拟软件。它为研究大气过程提供了大量选项，并可以在多种计算平台运行。更多信息，请参见WRF官网。



准备工作

1. 创建E-HPC集群。具体操作，请参见使用向导创建集群。

   配置集群时，软硬件参数配置如下：

参数	说明
硬件参数	部署方式为标准，包含2个管控节点，1个计算节点和1个登录节点，均采用ecs.c7.large实例规格，配置为2 vCPU，4 GiB内存，Ice Lake处理器，2.7 GHz。
软件参数	镜像选择CentOS 7.6公共镜像，调度器选择slurm，打开VNC开关。

2. 创建集群用户。具体操作，请参见创建用户。

   集群用户用于登录集群，进行编译软件、提交作业等操作，配置用户权限时，权限组请选择**sudo权限组**。

3. 安装OpenFOAM软件。具体操作，请参见安装软件。

   需安装的软件如下：

   - wrf-mpich，版本为3.8.1。
   - wrf-openmpi，版本为3.8.1。
   - mpich，版本为3.2。
   - openmpi，版本为1.10.7。

步骤一：准备算例文件

测试前您需要准备好namelist.wps文件。namelist是WPS（WRF Preprocessing System，WRF预处理系统）中的一个共享文件，该文件按照各个程序（geogrid.exe、ungrib.exe、metgrid.exe）所需要参数的不同分成三个部分（&geogrid、&ungrib、&metgrid）及一个共享部分（&share），分别定义了WPS模块所需要的各种参数。如下示例为推荐配置，未提及的参数保持默认即可。详细的namelist.wps文件参数及说明，请参见namelist.wps。

> **说明**
>
> 本示例中，namelist.wps放置在 `/home/wrftest/WPS` 目录下。

```
#共享部分
&share
wrf_core = 'ARW',    #wrf_core：选择WRF dynamical core，有'ARW'和'NMM'两个选项，默认值为'ARW'。
start_date = '2005-08-28_00:00:00',    #start_date：模拟开始时间
end_date = '2005-08-29_00:00:00',      #end_date：模拟结束时间
interval_seconds = 21600,
max_dom = 1,            #max_dom：模拟网格数（粗网格+嵌套网格），本示例中包含一个粗网格
io_form_geogrid = 2,    #io_form_geogrid: geogrid程序输出格式


#geogrid部分
#确定区域范围、嵌套关系、模型投影
&geogrid
parent_id = 1,
parent_grid_ratio = 1,
i_parent_start = 1,
j_parent_start = 1,

#确定网格在东西方向、南北方向的尺度（区域的矢量场的栅格数），本示例中为98*70个网格点
e_we = 98,
e_sn = 70,
geog_data_res = 'default',

#定义区域的栅格尺寸，本示例中网格分辨率为30km
dx = 30000,
dy = 30000,

#定义投影方式，关于投影方式说明可以参考WRF官网
map_proj = 'mercator'

#定义区域的中心经纬度坐标
ref_lat = 25.00,
ref_lon = -89.00,

#投影的三个参数值，随投影方式不同设定不同
truelat1 = 0.0,
truelat2 = 0.0,
stand_lon = -89.0,
geog_data_path = '地表数据存储路径'


#ungrib部分
&ungrib
out_format = 'WPS',  #out_format：ungrib生成的可供metgrid读取的文件格式，有'WPS'、'SI'、'MM5'三种格式，默认值：'WPS'
prefix = 'FILE'      #prefix：ungrib生成的中间文件路径和文件前缀名


#metgrid部分
&metgrid
fg_name = 'FILE',       #fg_name：ungrib程序生成的文件
io_form_metgrid = 2,    ##io_form_metgrid：metgrid生成的文件格式，支持三种格式1（binary，后缀名.int）、2（net CDF，后缀名.nc）、3（Grib1，后缀名.gr1），默认值：2
```

步骤二：运行geogrid.exe

geogrid.exe用于确定模拟区域，并把静态地形数据插值到网格点。

1. 登录E-HPC集群。

   登录时，请使用具有sudo权限的用户。具体操作，请参见登录集群。

2. 安装NCL软件。具体安装步骤，请参见NCL官网。

3. 查看集群是否已安装WRF的相关软件。

   ```
   export MODULEPATH=/opt/ehpcmodulefiles/
   module avail
   ```

4. 加载WRF软件环境。

   ```
   module load wrf-mpich/3.8.1 mpich/3.2
   echo $WPSHOME $WRFHOME
   ```

5. 将安装的WPS和WRF软件拷贝到工作目录。

   ```
   cp -r $WPSHOME $WPSCOPYHOME
   cp -r $WRFHOME $WRFCOPYHOME
   ```

   > **说明**
   >
   > 请将 `$WPSCOPYHOME` 和 `$WRFCOPYHOME` 修改为实际的工作目录，如本示例中的 `/home/wrftest/WPS` 。

6. 下载并解压地表数据。

   > **说明**
   >
   > 本示例中地表数据使用geog_complete.tar.gz，您也可以根据需要下载其他地表数据。更多信息，请参见WRF官网。

   ```
   cd /home/wrftest/WPS
   wget https://www2.mmm.ucar.edu/wrf/src/wps_files/geog_complete.tar.gz
   tar -zxvf geog_complete.tar.gz
   ```

7. 链接到GEOGRID.TBL文件。

   GEOGRID.TBL文件定义了geogrid.exe需要插值到网格点上的各静态地理数据集参数。

   ```
   ln -s geogrid/GEOGRID.TBL GEOGRID.TBL
   ```

8. 将静态地形数据插值到网格点。

   ```
   ./geogrid.exe
   ```

   运行geogrid.exe成功后，会在WPS目录下生成geo_em.d0N.nc地形文件，预期返回结果如下：

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!  Successful completion of geogrid.        !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

步骤三：运行ungrib.exe

ungrib.exe用于从GRIB格式的气象数据中提取所需要的气象要素场。

1. 下载并解压Katrina气象数据。

   > **说明**
   >
   > 本示例中气象数据为Katrina.tar.gz，请下载Katrina.tar.gz。您也可以根据需要下载其他气象数据，更多信息，请参见气象数据。

   ```
   wget http://www2.mmm.ucar.edu/wrf/TUTORIAL_DATA/Katrina.tar.gz
   tar -zxvf Katrina.tar.gz
   ```

2. 将气象数据文件链接到WPS目录下。

   ```
   ./link_grib.csh /home/wrftest/wrfdata/Katrina/avn
   ```

3. 选择气象数据相应的Vtable。

   本示例使用的Vtable为Vtable.GFS，您可以根据需要使用其他的Vtable。

   ```
   ln -sf ungrib/Variable_Tables/Vtable.GFS Vtable
   ```

4. 提取所需要的气象要素场。

   ```
   ./ungrib.exe
   ```

   运行ungrib.exe成功后，会在WPS目录下生成FILE:YYYY-MM-DD_hh*文件，预期返回结果如下：

   ```
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   !  Successful completion of ungrib.      !
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   ```

步骤四：运行metgrid.exe

metgrid.exe用于将ungrib.exe提取出的气象场数据水平插值到由geogrid.exe确定的网格点上。

1. 链接到GEOGRID.TBL文件。

   GEOGRID.TBL文件定义了metgrid.exe如何将气象数据水平插值到网格点上。

   ```
   ln -s metgrid/METGRID.TBL.ARW METGRID.TBL
   ```

2. 将气象场数据水平插值到由geogrid确定的网格点上。

   ```
   ./metgrid.exe
   ```

   运行metgrid.exe成功后，会在WPS目录下生成met_em.d0N.yyyy-mm-dd_hh:mm:ss.nc文件，预期返回结果如下：

   ```
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   !  Successful completion of metgrid.     !
   !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
   ```

步骤五：运行wrf.exe

wrf.exe用于输出天气预测数据。运行wrf.exe前，请先定义好namelist.input文件。namelist.input中&time_control、&domains部分的相关参数需要与namelist.wps文件中参数保持一致。详细的namelist.input文件参数及说明，请参见namelist.input。

> **说明**
>
> 本示例中，namelist.input放置在 `/home/wrftest/WRFV3/run` 目录下。

1. 进入WRFV3软件目录。

   ```
   cd /home/wrftest/WRFV3
   ```

2. 连接WPS的处理结果。

   ```
   ln -s /home/wrftest/WRFV3/run/met_em*
   ```

3. 初始化模拟数据。

   ```
   ./real.exe
   ```

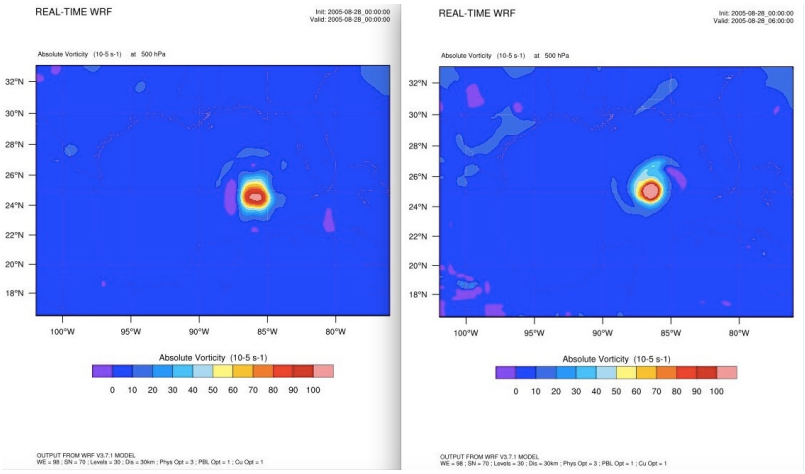
   运行real.exe成功后，会在 `/home/wrftest/WRFV3/run` 目录下生成wrfinput_d0N、wrfbyd_dxx。

4. 输出天气预测数据。

   ```
   mpirun -np 10 /home/wrftest/WRFV3/run/wrf.exe
   ```

   运行wrf.exe成功后，会在WRF目录下生成wrfout_d0N_[date]文件。

   使用NCL图像化WRF运行结果如下图所示：



6.使用BWA、GATK、Samtools软件进行基因测序

本文介绍如何使用E-HPC集群运行BWA、GATK、Samtools软件进行基因测序计算。

背景信息

生命科学领域内基因测序技术的飞速发展，人类发现的基因序列以指数级增长，对于如此数量庞大的基因进行同源性搜寻、比对、变异检查等，往往伴随着巨大的数据处理和并行计算。高性能计算可以提供强大的算力支持，使用多种调度器提高并发效率，使用GPU进行计算加速等。

本文以经典及普及的二代全基因组测序WGS（Whole Genome Sequencing）流程为例，结合二代测序软件GATK，介绍人类全基因组测序的通用流程。在实际生信分析中，需要结合不同的业务需求进行相应的调整，如增加变异检测质控及过滤等，您可以结合实际场景进行调整。

在进行基因测序时，您可以使用BWA构建索引及比对记录，再使用Samtools对比对记录进行排序，然后使用GATK去除重复序列、重新校正碱基质量值、变异检查。

- BWA（Burrows-Wheeler-Alignment Tool）是一款将DNA序列映射到参考基因组上的软件，例如比对人类基因组。
- GATK（The Genome Analysis Toolkit）是一款二代重测序数据分析软件，是基因分析的工具集。主要用于去除重复序列、重新校正碱基质量值、变异检查等。
- Samtools是用于处理sam和bam格式的工具软件，能够查看二进制文件、转换文件格式、对文件排序及合并，可以结合sam格式文件中的flag、tag等信息，对比对结果进行统计汇总。

准备工作

1. 创建E-HPC集群。具体操作，请参见使用向导创建集群。

   配置集群时，软硬件参数配置如下：

参数	说明
硬件参数	部署方式为精简，包含1个管控节点和1个计算节点，规格如下： ○ 管控节点：采用ecs.c7.large实例规格，该规格配置为2 vCPU，4 GiB内存。 ○ 计算节点：采用ecs.ebmc5s.24xlarge实例规格，该规格配置为96 vCPU、192 GiB内存。
软件参数	镜像选择CentOS 7.6公共镜像，调度器选择pbs，打开VNC开关。

2. 创建集群用户。具体操作，请参见创建用户。

   集群用户用于登录集群，进行编译软件、提交作业等操作，配置用户权限时，权限组请选择**sudo权限组**。

操作步骤

1. 登录E-HPC集群。

   登录时，请使用具有sudo权限的用户。具体操作，请参见登录集群。

2. 安装BWA、GATK、Samtools软件。

   关于如何安装BWA、GATK、Samtools，请参见BWA、GATK、Samtools。

3. 下载并解压人类参考基因组、人类基因测序样本1、人类基因测序样本2。

```
wget https://public-ehpc-package.oss-cn-hangzhou.aliyuncs.com/lifescience/b37_human_g1k_v37.fasta
wget https://public-ehpc-package.oss-cn-hangzhou.aliyuncs.com/lifescience/gatk-examples_example1_NA20318_ERR250971_1.filt.fastq.gz
gunzip gatk-examples_example1_NA20318_ERR250971_1.filt.fastq.gz
wget https://public-ehpc-package.oss-cn-hangzhou.aliyuncs.com/lifescience/gatk-examples_example1_NA20318_ERR250971_2.filt.fastq.gz
gunzip gatk-examples_example1_NA20318_ERR250971_2.filt.fastq.gz
```

4. 构建索引。

   i. 构建bwa比对所需的参考基因组的index数据库。

```
bwa index b37_human_g1k_v37.fasta
```

   ii. 创建fasta序列格式索引。

```
samtools faidx b37_human_g1k_v37.fasta
```

5. 比对reads。

   将样本测序数据reads与人类参考基因组进行比对，并将输出文件转化为bam格式，可有效节省磁盘空间。

```
time bwa mem -t 52 \ -R '@RG\tID:ehpc\tPL:illumina\tLB:library\tSM:b37' b37_human_g1k_v37.fasta \
NA20318_ERR250971_1.filt.fastq NA20318_ERR250971_2.filt.fastq \
| samtools view -S -b - > ERR250971.bam
```

6. 将比对记录按照顺序从小到大进行排序。

```
time samtools sort -@ 52 -O bam -o ERR250971.sorted.bam ERR250971.bam
```

7. 去除重复序列。

   i. 去除DNA序列PCR扩增产生的重复reads序列。

```
time gatk MarkDuplicates \
-I ERR250971.sorted.bam -O ERR250971.markdup.bam \
-M ERR250971.sorted.markdup_metrics.txt
```

   ii. 构建markdup测序bam文件索引。

```
samtools index ERR250971.markdup.bam
```

8. 重新校正碱基质量值。

i. 利用已有的snp及indels数据库，建立相关模型，构建重校准表。

```
time gatk BaseRecalibrator \
-R b37_human_g1k_v37.fasta \
-I ERR250971.markdup.bam  \
--known-sites b37_1000G_phase1.indels.b37.vcf.gz \
--known-sites b37_Mills_and_1000G_gold_standard.indels.b37.vcf.gz \
--known-sites b37_dbsnp_138.b37.vcf.gz -O ERR250971.BQSR.table
```

ii. 对原始碱基质量值进行调整。

```
time gatk ApplyBQSR \
--bqsr-recal-file ERR250971.BQSR.table \
-R b37_human_g1k_v37.fasta \
-I ERR250971.markdup.bam \
-O ERR250971.BQSR.bam
```

iii. 构建BQSR测序bam文件索引。

```
time samtools index ERR250971.BQSR.bam
```

9. 变异检查。

i. 生成参考基因组dict文件。

```
time gatk CreateSequenceDictionary  \
-R b37_human_g1k_v37.fasta \
-O b37_human_g1k_v37.dict
```

ii. 输出变异检测结果vcf文件。

```
time gatk HaplotypeCaller \
-R b37_human_g1k_v37.fasta  \
-I ERR250971.BQSR.bam \
-O ERR250971.HC.vcf.gz
```

测序耗时说明

本文使用的计算节点为ecs.ebmc5s.24xlarge，仅为演示全基因组测序流程，下表展示了基因测序各流程使用的时间，并非最优性能。

> **说明**
>
> 该测序流程可结合实际场景作进一步的优化，以提升性能。例如，在染色体切割时结合调度器提高并发率，通过GATK分布式并发测序提高测序效率，结合FPGA加速测序等。如您有业务需求，请提交工单。

流程	功能	运行时间（min）
bwa index	索引	50.60
bwa mem	比对	21.98

流程	功能	运行时间（min）
sort	排序	1.62
MarkDuplicates	去重	21.60
BaseRecalibrator	构建校准表	38.47
ApplyBQSR	重校准	17.22
CreateSequenceDictionary	创建字典	0.18
HaplotypeCaller	变异检查	175.93

7.使用AutoDock Vina软件进行药物筛选

本文以AutoDock Vina软件为例，介绍如何在E-HPC上进行高性能计算实现虚拟药物筛选。

背景信息

分子对接（Molecular docking）是虚拟药物筛选中的关键环节。本文通过模拟小分子配体和生物大分子受体相互作用的过程，预测配体与受体的结合模式和亲和力，模拟实现对药物的筛选。目前商业应用较广泛的Specs、Enamine和ChemDiv化合物库，均可提供大量配体模拟计算配体和给定受体的相互作用。由于不同配体之间没有依赖，因此可以大规模并行处理。本文同样适用于其它大批量、高并发处理需求的生物、医药等场景。

AutoDock Vina作为一款开源的分子对接软件，具有速度快、算法准确等优点，特别适用于搭建基于分子对接的虚拟筛选，它基于MGLTools工具包进行使用。MGLTools包括AutoDock Tools（ADT）和Python Molecular Viewer（PMV）。ADT用来为Vina生成输入文件，PMV用来查看结果。更多信息，请参见AutoDock Vina和MGLTools。

准备工作

1. 创建E-HPC集群。具体操作，请参见使用向导创建集群。

   配置集群时，软硬件参数配置如下：

参数	说明
硬件参数	部署方式为标准，包含2个管控节点，1个计算节点和1个登录节点。 节点均采用ecs.c7.large实例规格，配置为2 vCPU，4 GiB内存，Ice Lake处理器，2.7 GHz。
软件配置	镜像选择CentOS 7.6公共镜像，调度器选择pbs。

2. 创建集群用户，本实践中以ehpcuser为例。具体操作，请参见创建用户。

   集群用户用于登录集群，进行编译软件、提交作业等操作，配置用户权限时，权限组请选择**普通用户组**。

3. 为计算节点绑定EIP。

   > **说明**
   >
   > 计算节点在下载安装AutoDock Tools软件时需要使用公网地址。配置完成后，您可以按需为其解绑EIP。

4. 安装AutoDock Vina软件。

   i. 登录E-HPC控制台。

   ii. 在**集群**页面，在右侧单击**远程连接**，以root用户登录集群。

iii. 执行如下命令下载并安装Vina软件。

```
cd /opt
wget https://vina.scripps.edu/wp-content/uploads/sites/55/2020/12/autodock_vina_1_1_2_linux_x86.tgz    #下载vina
tar xzvf autodock_vina_1_1_2_linux_x86.tgz     #解压
./autodock_vina_1_1_2_linux_x86/bin/vina --help    #查看安装结果

export PATH=$PATH:/opt/autodock_vina_1_1_2_linux_x86/bin

source ~/.bashrc
```

5. 安装AutoDock Tools软件。

i. 登录E-HPC控制台。

ii. 在**集群**页面，在右侧单击**远程连接**，以root用户登录集群。

iii. 执行如下命令下载并安装AutoDock Tools软件。

```
cd /opt
wget https://vina.scripps.edu/wp-content/uploads/sites/55/2020/12/autodock_vina_1_1_2_linux_x86.tgz    #下载vina
tar xzvf autodock_vina_1_1_2_linux_x86.tgz     #解压
./autodock_vina_1_1_2_linux_x86/bin/vina --help    #查看安装结果

export PATH=$PATH:/opt/autodock_vina_1_1_2_linux_x86/bin

source ~/.bashrc
```

操作步骤

1. 登录E-HPC控制台。
2. 安装git及下载作业文件。

i. 在**集群**页面，在右侧单击**远程连接**，以ehpcuser用户登录集群。

ii. 执行如下命令，下载本文用到的操作命令和代码。

```
git clone https://code.aliyun.com/best-practice/022.git
```

iii. 执行如下命令检查作业文件。

```
cd 022
ls
```

当显示如下时，说明已下载完成。

```
[ehpcuser@login0 ~]$ git clone https://code.aliyun.com/best-practice/022.git
Cloning into '022'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 6 (delta 0)
Unpacking objects: 100% (6/6), done.
[ehpcuser@login0 ~]$ cd 022
[ehpcuser@login0 022]$ ls
README.md  vina-ehpcarrayjob.tar.gz
[ehpcuser@login0 022]$
```

3. 提交作业并查看作业运行结果。
   i. 登录E-HPC客户端。

      具体操作，请参见登录客户端。
   ii. 在左侧导航栏，选择**应用中心**。
   iii. 单击Vina应用。
   iv. 在弹出的应用信息面板，配置作业参数。

参数类型	参数	描述
基础参数	作业名称	自定义设置，如vina001。
	作业队列	运行该作业的队列，如workq。
	CPU核数	单个节点的CPU核数，如2。
	节点数	运行该作业所需的计算节点数。
	输出日志	作业运行日志的输出路径。
应用参数	图形界面参数	选择VNC。
	受体（刚性）	/home/ehpcuser/vina-ehpcarrayjob/1fkn_rgd.pdbqt
	配体	/home/ehpcuser/vina-ehpcarrayjob/test/ligand_1.pdbqt
	配置文件	/home/ehpcuser/vina-ehpcarrayjob/conf.txt
	输出文件目录	/home/ehpcuser/
	是否需要GUI	是

   v. 单击**提交**。
4.
5. 提交作业。
   i. 使用ehpcuser用户远程登录集群。

ii. 切换到022目录并解压作业文件。

```
cd 022
tar xzvf vina-ehpcarrayjob.tar.gz
```

iii. 切换到vina-ehpcarrayjob目录执行 `qsub` 命令提交作业。

```
cd /home/ehpcuser/022/vina-ehpcarrayjob
qsub qjob.sh
```

iv. 通过 `qstat -t` 命令查看作业执行情况。

回显如下所示。

```
[ehpcuser@login0 022]$ cd /home/ehpcuser/022/vina-ehpcarrayjob
[ehpcuser@login0 vina-ehpcarrayjob]$ qsub qjob.sh
4[].scheduler
[ehpcuser@login0 vina-ehpcarrayjob]$ qstat -t
Job id            Name             User             Time Use S Queue
----------------  ---------------- ----------------  -------- - -----
4[].scheduler     arrt             ehpcuser                 0 B workq
4[1].scheduler    arrt             ehpcuser          00:00:00 X workq
4[2].scheduler    arrt             ehpcuser          00:00:00 X workq
4[3].scheduler    arrt             ehpcuser          00:00:00 X workq
4[4].scheduler    arrt             ehpcuser          00:00:00 X workq
4[5].scheduler    arrt             ehpcuser          00:00:00 X workq
4[6].scheduler    arrt             ehpcuser          00:00:00 X workq
4[7].scheduler    arrt             ehpcuser          00:00:00 X workq
4[8].scheduler    arrt             ehpcuser          00:00:00 X workq
4[9].scheduler    arrt             ehpcuser          00:00:00 X workq
4[10].scheduler   arrt             ehpcuser          00:00:00 X workq
4[11].scheduler   arrt             ehpcuser          00:00:00 X workq
4[12].scheduler   arrt             ehpcuser          00:00:00 X workq
4[13].scheduler   arrt             ehpcuser          00:00:00 X workq
4[14].scheduler   arrt             ehpcuser          00:00:00 X workq
4[15].scheduler   arrt             ehpcuser          00:00:00 X workq
4[16].scheduler   arrt             ehpcuser          00:00:00 X workq
4[17].scheduler   arrt             ehpcuser          00:00:00 X workq
4[18].scheduler   arrt             ehpcuser          00:00:00 X workq
4[19].scheduler   arrt             ehpcuser          00:00:00 X workq
4[20].scheduler   arrt             ehpcuser          00:00:00 X workq
4[21].scheduler   arrt             ehpcuser          00:00:00 X workq
4[22].scheduler   arrt             ehpcuser          00:00:00 X workq
4[23].scheduler   arrt             ehpcuser          00:00:00 X workq
4[24].scheduler   arrt             ehpcuser          00:00:00 X workq
4[25].scheduler   arrt             ehpcuser          00:00:00 R workq
4[26].scheduler   arrt             ehpcuser          00:00:00 R workq
4[27].scheduler   arrt             ehpcuser          00:00:00 R workq
4[28].scheduler   arrt             ehpcuser          00:00:00 R workq
4[29].scheduler   arrt             ehpcuser                 0 Q workq
4[30].scheduler   arrt             ehpcuser                 0 Q workq
[ehpcuser@login0 vina-ehpcarrayjob]$
```

6. 可视化显示作业结果。

i. 在E-HPC控制台左侧导航栏选择**集群**。

ii. 在集群列表中选择对应集群，在页面右侧单击**更多 > VNC**。

> 说明
>
> - 如果未安装VNC可视化服务，请先进行安装。具体操作，请参见使用向导创建集群。
> - 首次登录VNC需要启用可视化服务。具体操作，请参见使用VNC远程可视化。

iii. 连接VNC成功后，使用VMD打开分析结果。

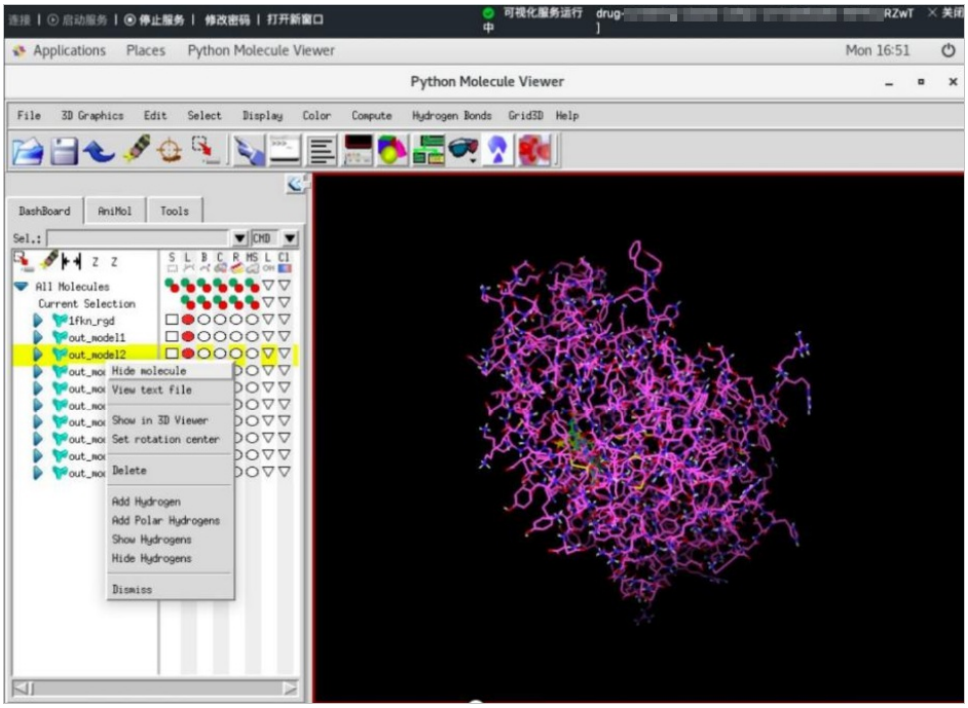
a. 在可视化桌面，单击右键选择**Open Terminal**。

b. 输入**/opt/vmd/1.9.3/vmd**。

c. 在VMD Main页面，单击**File > Read Molecule**，选择/home/ehpcuser/vina-ehpcarrayjob路径下的1fkn.pdbqt文件。

d. 单击**File > Read Molecule**，选择/home/ehpcuser/vina-ehpcarrayjob路径下的ligand_1/out.pdbqt文件。下

图显示了ligend1这个配体和受体1fkn的多种匹配模式。



8.EDA上云最佳实践

本文介绍如何利用弹性高性能计算E-HPC解决EDA工具在IC设计过程中遇到的问题。

背景信息

IC（Integrated Circuit）设计依赖于IT（Information Technology）技术的支撑和服务，既包括EDA（Electronics Design Automation）工具的使用，也包括计算、存储、网络等基础设施。伴随芯片规模增长、设计复杂度提升、工艺尺寸缩小以及EDA工具持续优化的机器学习技术和敏捷方法学的变革，传统IT面临着算力暴涨与传统IT矛盾加剧、IT基础设施管理难度大、成本居高不下等问题，愈发难以满足IC设计的需求。

阿里云所提供的云上架构和功能全面适配和满足EDA业务和技术需求，具备长期支持的迭代能力。计算和存储资源规格性能优异，且在弹性扩容、资源灵活调度以及资源成本方面体现出显著的优势：

- 一站式的高性能计算服务：E-HPC支持方便快捷地部署及配置集群环境、EDA软件及其他工具软件，支持LSF调度器，交互体验友好。
- 强大的计算性能：符合EDA行业需求特点的裸金属服务器机型，不仅具备虚拟机的弹性，还保有物理机的性能无损、完整特性、高隔离性和高安全性。确保租户真正独占资源，充分满足客户对性能、稳定性以及数据安全和监管合规的业务诉求。
- 优异的存储表现：阿里云的并行文件系统CPFS具有吞吐量高、网络延迟小、读写性能强的特点，尤其在超大规模小文件读写方面表现出强大的性能。
- 数据安全保障：除裸金属服务器高隔离性的安全保障外，CPFS支持数据加密。
- 弹性和高性价比：计算和存储等资源具备弹性扩容、按需使用、按量付费的良好特性，新资源投入生产的速度较快，有助于提升整体研发效率，最终缩短新产品面市时间。
- 资源供给和保障能力突出：阿里云拥有强大的弹性调度能力和强大的供应链体系，能够为您提供充足的资源保障。

准备工作

本实践资源规划如下：

资源类型	产品	配置项
网络资源	专有网络VPC	• 状态：新购 • 地域：华东2（上海） • 名称：vpc-eda • 网段：192.168.0.0/16
	虚拟交换机	• 状态：新购 • 可用区：华东2可用区B • 名称：vswitch-eda • 网段：192.168.0.0/24

资源类型	产品	配置项
	弹性公网EIP	• 状态：新购 • 类型：按量付费 • 名称：EIP • 带宽：50 Mbit/s
弹性计算资源	云服务器ECS	• 可用区：华东2可用区B • 名称：image • 实例规格：ecs.c6.xlarge • 镜像：Centos 7.6 64位 • 系统盘：ESSD 40 GiB
	弹性高性能计算E-HPC	• 部署方式：精简 • 计算节点：ecs.ebmc6.26xlarge，1个 • 登录节点：ecs.c7.4xlarge，1个 • 系统盘：Cloud_ESSD 40 GiB • 镜像：自定义镜像 • 文件类型：`CPFS` • 调度器：用户自行安装
存储资源	文件存储CPFS	• 支付类型：按量付费 • 存储类型：200 MB/s/TiB基线 • 容量：48,000 GiB

> **说明**
>
> 本实践所用资源仅用于案例演示，实际使用过程中请根据需要进行合理规划。

操作流程

EDA上云实践的操作流程如下：

步骤一：搭建基础环境

步骤二：安装CPFS客户端

步骤三：创建自定义镜像

步骤四：部署E-HPC

步骤五：安装LSF插件（可选）

步骤一：搭建基础环境

1. 登录云速搭CADT控制台。
2. 在页面左上方的菜单栏，选择**新建** > **官方模板库新建**。
3. 在页面右上方搜索框中搜索**EDA上云**模板，单击**基于应用新建**。

> **说明**
>
> 本实践使用的模板是根据准备工作中所列的资源项配置的。实际操作过程中，您可以根据需要，双击应用架构中的资源类型图标进行修改。

4. 在创建完成的应用架构中双击  ，填写ECS的登录密码并确认。

> **说明**
>
> E-HPC部署开关默认为**关闭**状态，需在制作好自定义镜像后开启部署开关并配置自定义镜像。

5. 配置完成后，单击**保存**，在弹出的对话框中输入**应用名称**，单击**确认**。
6. 单击**部署应用**。

> **说明**
>
> 如果弹出**属性校验报错**或**校验失败**的提示，请根据页面提示信息进行修正，完成后需要重新单击**保存**和**部署应用**。

7. 校验和计价完成后，在**确认订单**页面，勾选**《云速搭服务条款》**并单击**下一步：支付并创建**。

   资源部署状态页面下方显示**部署成功**，说明基础环境搭建成功。
8. 返回应用架构页面，双击ECS实例image，记录私网IP地址（private_ip）以备后用。
9. 确认CPFS挂载点状态并记录初始密码。
   i. 在应用架构中双击CPFS实例，单击**前往控制台**，进入NAS控制台。
   ii. 在左侧导航栏选择**文件系统** > **文件系统列表**。
   iii. 待CPFS文件系统挂载点状态为**可用**后，记录客户端管理节点的初始密码。

> **说明**
>
> CPFS文件系统由CADT创建，大约需要20分钟来完成挂载点的自动创建。

步骤二：安装CPFS客户端

1. 设置CPFS管理节点访问规则。
   i. 登录ECS控制台。
   ii. 在左侧导航栏选择**网络与安全** > **安全组**。

iii. 选择地域为**华东2（上海）**。

iv. 在**安全组**页面找到CPFS安全组，在**操作**列单击**配置规则**。

v. 手动添加一条**入方向**的访问规则，对Workbench IP地址段100.104.0.0/16开放22号端口的访问。

vi. 单击**保存**。

2. 远程登录CPFS客户端管理节点`qr-001`。

i. 登录ECS控制台。

ii. 在左侧导航栏选择**实例与镜像** > **实例**。

iii. 选择地域为**华东2（上海）**。

iv. 在**实例**页面选择CPFS管理节点ECS实例（实例名称以`qr-001`结尾），在**操作**列单击**远程连接**。

v. 选择**Workbench远程连接**，单击**立即登录**。

vi. 输入已经获取的CPFS管理节点的`root`账号密码，单击**确定**。

3. 配置CPFS管理节点对客户端节点的免密钥登录。

i. 确认CPFS管理节点的/etc/ssh/ssh_config文件中的如下配置。

```
# This system is following system-wide crypto policy.
# To modify the system-wide ssh configuration, create a  *.conf  file under
#  /etc/ssh/ssh_config.d/  which will be automatically included below
Include /etc/ssh/ssh_config.d/*.conf
StrictHostKeyChecking no
```

ii. 执行以下命令，将公钥信息拷贝至制作自定义镜像的ECS实例。

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.168.0.198
```

> **说明**
>
> 该步骤实现从CPFS管理节点到目标ECS（image）的免密登录。命令中的IP为前面步骤中已获取的ECS实例ecs-image的私网IP地址。

iii. 执行以下命令，测试免密登录。

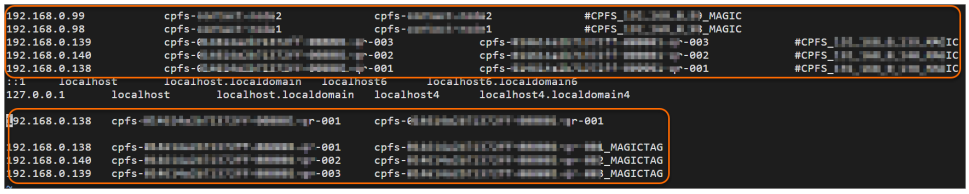
```
ssh root@192.168.0.96

Welcome to Alibaba Cloud Elastic Compute Service !
```

iv. 测试成功后执行`exit`命令，返回CPFS管理节点。

4. 记录CPFS管理节点的Quorum和Contact内容。

i. 执行 `vim /etc/hosts` 命令。

ii. 获取除localhost记录以外的全部其他记录，以备后用。



5. 远程登录用于制作镜像的ECS实例。

i. 登录ECS控制台，选择地域为**华东2（上海）**。

ii. 在**实例**页面选择ECS实例（实例名称以 `image` 结尾），单击**远程连接**。

iii. 选择**Workbench远程连接**，单击**立即登录**。

iv. 输入在CADT中创建ECS时的 `root` 账号密码，单击**确定**。

6. 在用于制作镜像的ECS上安装CPFS客户端。

i. 在/etc/hosts路径中添加CPFS管理节点的Quorum和Contact内容。

ii. 执行以下命令，下载并解压RPM包。

```
mkdir /tmp/rpms
cd /tmp/rpms
wget https://gpfs-rpms.oss-cn-beijing.aliyuncs.com/CPFS2.2-CentOS.tar.gz
tar xvfz CPFS2.2-CentOS.tar.gz
```

iii. 执行以下命令，安装CPFS客户端的依赖软件。

```
yum install -y cpp gcc gcc-c++ binutils ksh elfutils elfutils-devel rpm-build
```

iv. 执行以下命令，安装CPFS客户端。

```
cd CentOS/CentOS7/
yum install -y gpfs.adv-*.x86_64.rpm gpfs.base-*.x86_64.rpm gpfs.docs-
*.noarch.rpm gpfs.gpl-*.noarch.rpm gpfs.gskit-*.x86_64.rpm
gpfs.gss.pmsensors-*.x86_64.rpm gpfs.license.dm-*.x86_64.rpm
gpfs.msg.en_US-*.noarch.rpm
```

v. 执行以下命令，构建系统。

```
/usr/lpp/mmfs/bin/mmbuildgpl
```

> **说明**
>
> 当返回Building GPL module completed successfully...信息时，说明系统已构建成功。

步骤三：创建自定义镜像

1. 登录ECS控制台，选择地域为**华东2（上海）**。

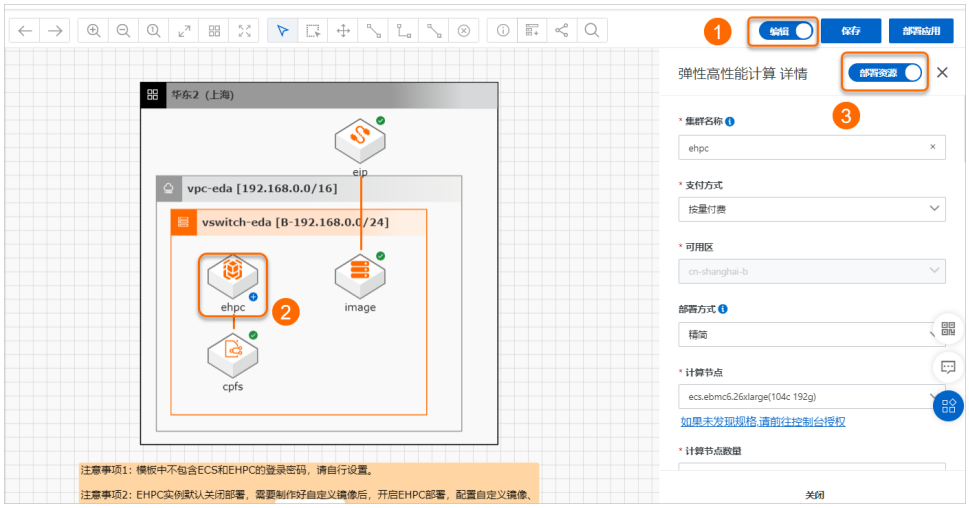
2. 在实例页面选择ECS实例（实例名称以 `image` 结尾），在**操作**列中，单击**更多 > 云盘和镜像 > 创建自定义镜像**。
3. 输入自定义镜像名称（ehpc-image）和自定义镜像描述，单击**确认**。
4. 进入控制台镜像页面，查看镜像创建进度。

   进度为100%时，说明已创建成功。

> **说明**
>
> 本例的自定义镜像创建耗时约10分钟左右。

步骤四：部署E-HPC

1. 登录云速搭CADT控制台。
2. 单击**应用 > 我的应用**，找到已部署的EDA上云应用，单击**编辑架构图**。
3. 切换到**编辑**模式，双击**ehpc**图标后打开**部署资源**开关。



4. 在E-HPC配置项中，**镜像类型**选择**自定义镜像**，**镜像**选择已创建的ehpc-image，并输入E-HPC登录密码。
5. 单击**保存 > 部署应用**，根据页面提示完成部署。
6. 返回应用架构页面，双击ehpc图标，在**资源清单**中单击**前往控制台**。
7. 查看E-HPC集群运行状态是否正常。

> **说明**
>
> 部署过程大约需要15分钟。

8. 查看E-HPC集群架构。
   i. 单击**资源清单**。

ii. 在E-HPC集群后面的**操作**列选择**查看详情**。

架构探查任务完成后，自动弹出集群架构图。

> **说明**
>
> 您可以在集群架构图页面右侧查看资源列表，也可导出架构图和资源清单。

步骤五：安装LSF插件（可选）

EDA业务场景下，由于LSF调度器需要付费购买License，本实践中E-HPC未集成该调度器。您可根据提供的插件模版及配置文件，自定义调度器并以插件的形式在E-HPC控制台创建集群，从而提供对应的节点管理、作业管理及自动伸缩等能力。

1. 执行如下命令构建插件目录结构。

```
mkdir -p /plugin/LSF/10.1.0
```

2. 下载插件模版及配置文件。

```
cd /plugin
wget https://public-ehpc-package.oss-cn-
hangzhou.aliyuncs.com/plugintemplate/ehpc_custom.conf
wget -P /plugin/LSF/10.1.0 https://public-ehpc-package.oss-cn-
hangzhou.aliyuncs.com/plugintemplate/plugin_template.tar.gz
```

3. 根据实际需要编辑配置文件。

根据调度器插件结构及接入模式配置调度器信息；根据需求及功能实现将支持的功能设置为 `True` ，不支持的功能设置为 `False` 。

```
vim ehpc_custom.conf
```

4. 解压插件模板。

```
cd /plugin/LSF/10.1.0
tar xvfz /plugin/LSF/10.1.0/plugin_template.tar.gz
```

5. 根据插件模版自定义调度器功能实现。

例如下图红框部分为节点调度服务检测的功能实现，系统根据当前不同的节点类型自定义返回状态。

- 对于计算节点和登录节点角色，在调度服务检测实现中直接返回 `True` 表示检测通过。

- 对于管理节点角色来说，需要检测LSF服务是否在节点上正常运行来返回最终的检测结果。

```
def sched_service_check():
        """Scheduling service check on the node, maybe should differentiate each node role(master/slave/compute)

        Return:
            success(boolean): True if service is ready,
                              False if service is unactive
        """
        roles = conf['Role'].split(',')

        if 'ResourceManager' in roles or 'ResourceManagerBackup' in roles:
                return __service_check__("lsf")

        if 'ComputeNode' in roles:
                return True

        if 'LoginNode' in roles:
                return True

        return True
```

9.使用OpenFOAM软件进行流体力学仿真计算

本文介绍如何使用E-HPC集群运行OpenFOAM进行流体力学仿真计算。

背景信息

OpenFOAM（Open Source Field Operation and Manipulation）是对连续介质力学问题进行数值计算的软件。可进行数据预处理、后处理和自定义求解器，常用于计算流体力学领域。更多信息，请参见OpenFOAM官网。

本文利用OpenFOAM中的simpleFoam求解器计算摩托车外流场，算例路径：$FOAM_TUTORIALS/incompressible/simpleFoam/motorBike/。

准备工作

1. 创建E-HPC集群。具体操作，请参见使用向导创建集群。

   配置集群时，软硬件参数配置如下：

参数	说明
硬件参数	部署方式为标准，包含2个管控节点，1个计算节点和1个登录节点，均采用ecs.c7.large实例规格，配置为2 vCPU，4 GiB内存，Ice Lake处理器，2.7 GHz。
软件参数	镜像选择CentOS 7.6公共镜像，调度器选择pbs。

2. 创建集群用户。具体操作，请参见创建用户。

   集群用户用于登录集群，进行编译软件、提交作业等操作，配置用户权限时，权限组请选择**sudo权限组**。

3. 安装OpenFOAM软件。具体操作，请参见安装软件。

   需安装的软件如下：

   - openfoam-openmpi，版本为5.0。
   - openmpi，版本为1.10.7。

操作步骤

1. 登录E-HPC集群。

   登录时，请使用具有sudo权限的用户。具体操作，请参见登录集群。

2. 执行以下命令，提交作业。

   i. 设置环境变量。

   ```
   export MODULEPATH=/opt/ehpcmodulefiles/
   module load openfoam-openmpi/5.0
   module load openmpi/1.10.7
   ```

ii. 准备算例文件。

```
mkdir /home/foamtest/motorBike
cp -r /opt/OpenFOAM/OpenFOAM-5.0/tutorials/incompressible/simpleFoam/motorBike/* /h
ome/foamtest/motorBike
```

> **说明**
>
> 本文使用OpenFOAM中的simpleFoam求解器计算摩托车外流场作为示例，算例路径为 `$FOAM_TUTORIALS/incompressible/simpleFoam/motorBike/` 。

iii. 执行算例。

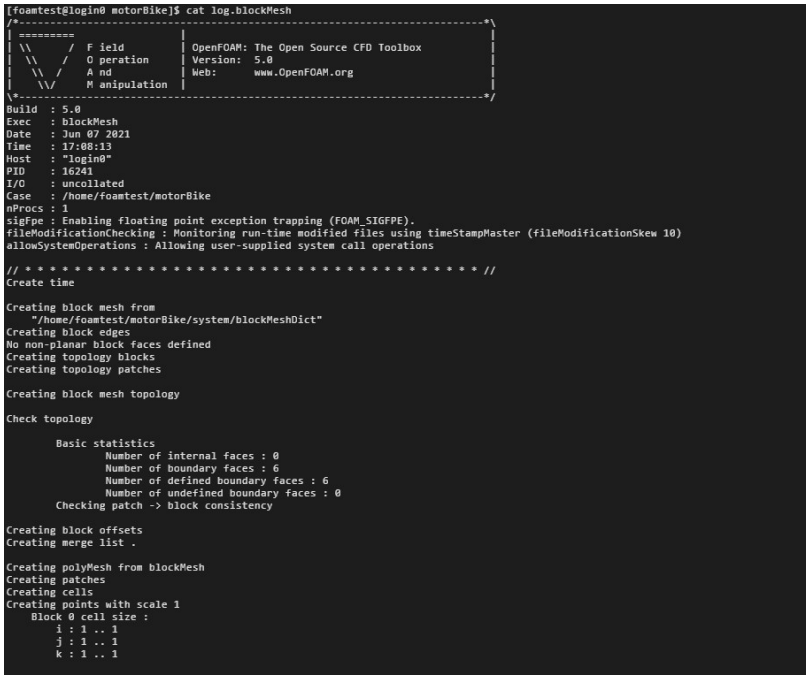
```
cd /home/foamtest/motorBike
source /opt/OpenFOAM/OpenFOAM-5.0/etc/bashrc
./Allrun
```

查看结果

1. 执行以下命令，查看作业结果文件。

```
cat log.blockMesh
```

预期返回如下：



10.使用CP2K软件进行分子动力学研究

本文以CP2K软件为例，介绍如何在E-HPC上进行分子动力学研究。

背景信息

CP2K是一款功能强大的分子动力学模拟软件，主要用于研究计算固态、液体、分子和生物体系的性质。CP2K基于密度泛函理论(DFT)，为不同的建模方法（例如使用混合高斯的DFT和使用平面波方法的GPW和GAPW）提供了通用框架 。更多信息，请参见CP2K介绍。

本实践中，E-HPC结合对象存储OSS、云速搭CADT、云服务器ECS、GPU云服务器及文件存储NAS等产品，在阿里云上构建低成本的分子模拟MDaaS（Molecular Dynamics as a Service）超算集群，通过运行CP2K这款开源仿真软件来实现分子运动及动力模拟。

准备工作

- 登录阿里云控制台，开通相关产品（E-HPC、NAS、ECS、OSS、CADT、RAM）的服务。
- 请根据需要进行网络部署及资源规划。本实践规划内容如下：
  - 地域选择华北 2（北京）。
  - 专有网络（VPC）的CIDR为192.168.0.0/16，虚拟交换机（vSwitch）使用可用区H 192.168.0.0/24。
  - E-HPC创建为标准集群（登录节点1个，管控节点2个，计算节点2个）。

操作流程

使用CP2K软件进行分子动力学研究的操作流程如下：

步骤一：搭建基础环境

步骤二：创建用户

步骤三：提交作业

步骤四：配置弹性伸缩

步骤一：搭建基础环境

1. 登录云速搭CADT控制台。
2. 在页面左上方的菜单栏，选择**新建 > 官方模板库新建**。
3. 在页面右上方搜索**EHPC分子动力学最佳实践**模板，单击**基于应用新建**。
4. 在创建完成的应用架构中双击  ，**镜像类型**选择**自定义镜像**，**镜像**选择**cp2k-20210910**。
5. 配置完成后，单击**保存**，在弹出的对话框中输入**应用名称**，单击**确认**。
6. 单击**部署应用**。

> **说明**
>
> 如果弹出**属性校验报错**或**校验失败**的提示，请根据页面提示信息进行修正，完成后需要重新单击**保存**和**部署应用**。

7. 校验和计价完成后，在**确认订单**页面，勾选**《云速搭服务条款》**并单击**下一步：支付并创建**。

   资源部署状态页面显示**部署成功**，说明基础环境搭建成功。

步骤二：创建用户

1. 登录云速搭CADT控制台。
2. 打开**资源清单**，单击**ehpc**。
3. 在E-HPC控制台左侧导航栏，选择**资源管理 > 用户**。
4. 在**集群用户管理**页签单击**新增用户**。
5. 设置用户名、用户组及密码后，单击**确定**。

> **说明**
>
> 此例中创建的用户名为 `user1` ，用户组为 `sudo用户组` 。

步骤三：提交作业

1. 登录E-HPC控制台。
2. 在集群列表中选择对应集群，在页面右侧单击**远程连接**。
3. 使用 `root` 用户登录集群。
4. 执行如下命令关联文件并设置权限。

```
ln -sf /root/.local/ /home/user1/
```

5. 切换为 `user1` 用户，在 `/home/user1` 目录下，拷贝作业文件。

```
cp -r /root/user1/* .
```

6. 执行 `dpgen` 命令提交作业。

```
cd dzh/CH4
dpgen run param.json machine.json
```

> **说明**
>
> - param.json文件指定了运行时的一些参数和目录等信息，需要根据实际情况进行修改。
> - machine.json文件指定了计算节点的一些配置参数和相关目录等信息，需要根据实际情况修改。
> - 作业产生的日志文件存放在 `/home/user1/dpgen_work` 目录中。

7. 返回E-HPC控制台，选择**作业与性能管理 > 作业**，查看作业完成状态。
8. 单击**E-HPC优化器**，在**性能大盘**查看系统性能指标及状态。

步骤四：配置弹性伸缩

1. 登录E-HPC控制台。
2. 在左侧导航栏选择**弹性** > **自动伸缩**。
3. 按照以下信息进行全局配置。
   - 开关设置：同时勾选启动扩容和启动缩容
   - 缩容时间（分钟）：6
   - 镜像类型：自定义镜像，cp2k-20210910
4. 进行队列配置。
   i. 单击右侧的**编辑**。
   ii. **队列节点数**设置为0~100。
   iii. 单击**配置清单**右侧的**增加**。
   iv. 按照以下信息进行新建配置，完成后单击**确认**。
      - 可用区：华北2可用区H
      - 交换机ID：vsw-2**********khyl
      - 实例类型：ecs.gn6**********grge
      - 抢占式策略：系统自动出价，最高按量付费价格
   v. 在**队列配置**页面，单击**确认**。
5. 在**自动伸缩**页面，单击**确认**。
6. 输入校验码，单击**确定**。

> **说明**
>
> 弹性自动伸缩功能设置完成后，系统会根据作业的情况自动进行节点扩容和缩容。您可以在E-HPC控制台的**节点与队列**页面查看计算节点的扩容和缩容状态。

11.自动伸缩最佳实践

本文以使用LAMMPS软件进行高性能计算介绍如何配置自动伸缩策略。

背景信息

当您需要每天不定时提交作业，使用E-HPC集群几个小时进行大规模计算， 然后释放节点，您可以针对不同的作业类型，配置不同的伸缩策略。配置伸缩策略后，系统可以根据实时负载自动增加或减少计算节点。可以帮您合理利用资源，减少使用成本。

操作步骤

1. 登录弹性高性能计算控制台。
2. 创建一个名为AutoScaling的集群。

   具体操作，请参见创建集群。请注意以下配置参数：

   - **可用区**：选择**华东1可用区I**
   - **计算节点**：选择**ecs.c6.large**
   - **交换机**：选择**vsw-bp122ezf0hvk1xnhj****
   - **调度器**：选择**pbs**
   - **其他软件**：选中lammps-mpich 31Mar17、mpich 3.2、openmpi 1.10.7
   - **是否新建队列**：新建名为low的队列

3. 创建一个名为AutoScaling的普通用户。具体操作，请参见创建用户。
4. 配置自动伸缩策略。
   i. 在左侧导航栏，选择**弹性 > 自动伸缩**。
   ii. 在**队列配置**区域，选择low队列，单击**编辑**。
   iii. 单击**配置清单**右侧的**增加**，配置如下参数，单击**确认**。
   - **可用区**：选择**华东1可用区I**
   - **交换机ID**：选择**vsw-bp122ezf0hvk1xnhj****
   - **实例类型**：选择**ecs.c6.large**
   - **抢占式策略**：选择**不使用抢占式实例**

iv. 在**队列配置**面板中，配置如下参数，单击**确认**。

a. **启动扩容**：打开**启动扩容**开关

b. **启动缩容**：打开**启动缩容**开关

c. **队列节点数**：2~10

d. **主机名前缀**：输入computelow

e. **镜像类型**：选择**公共镜像**

f. **镜像ID**：选择**CentOS_7.6_64**

v. 在自动伸缩页面，单击右上角的**确认**。

启动自动伸缩之后，如果集群未运行作业，会尝试保持最小节点数（2个节点）。由于已经存在一个计算节点，因此会在low队列中扩容一个计算节点。

5. 创建作业脚本并提交作业。

i. 在左侧导航栏，选择**作业与性能管理 > 作业**。

ii. 在**集群**列表中，选择AutoScaling集群，单击**创建作业**。

iii. 在**创建作业**页面，选择**编辑作业文件 > 新建文件 > 使用文件模板 > pbs demo**。

iv. 在编辑作业文件页面，配置lj.in文件和AutoScaling.pbs，单击**确认**提交作业。

lj.in算例参数及说明，请参见使用LAMMPS软件进行高性能计算。AutoScaling.pbs配置如下所示：

```
#!/bin/sh
#PBS -l select=3:ncpus=1:mpiprocs=1  #该脚本在3个计算节点上运行，每个节点使用1 vCPU，1个MPI进程。
#PBS -j oe

export MODULEPATH=/opt/ehpcmodulefiles/
module load lammps-openmpi/31Mar17
module load openmpi/1.10.7

echo "run at the beginning"
mpirun lmp -in ./lj.in
```

6. 查看自动伸缩结果。

i. 在左侧导航栏，选择**资源管理 > 节点与队列**。

ii. 在**集群**列表中选择AutoScaling集群，在**节点类型**列表中选择**计算节点**。

iii. 在**节点列表**区域中，在**队列**列表中选择low队列。

队列中节点数量随作业运行有如下变化：

- 当集群有作业运行时，可以看到节点列表中自动扩容了一个计算节点在安装中，几分钟后，计算节点状态变为**运行中**，此时作业开始在当前low队列的3个节点上运行。
- 当集群没有新的作业运行时，6分钟之后，low队列中空闲的计算节点会被释放掉，但会保持该队列最小节点数的计算节点（2个）。

7. 查看作业运行详情。

i. 在左侧导航栏，选择**作业与性能管理 > 作业**。

ii. 在集群列表中选择AutoScaling集群，作业状态选择**已完成**。

iii. 单击目标作业右侧的**详情**，查看作业运行详情。

8. 查看扩容和缩容操作日志。

i. 在左侧导航栏，选择**运维与监控 > 操作日志**。

ii. 在集群列表中选择AutoScaling集群，可以看到扩容和缩容的操作日志记录。

12.创建以CPFS为共享存储的E-HPC集群

本文介绍如何创建以CPFS（Cloud Paralleled File System）为共享存储的高性能计算集群。为您提供一个高IOPS、高吞吐、低时延的计算集群。

背景信息

CPFS是一种高性能并行文件存储系统，专为AI训练和E-HPC等高性能计算场景打造，最大支持数十GB/s吞吐能力。CPFS的数据存储在集群中的多个数据节点，并可由多个客户端同时访问，从而能够为大型E-HPC提供高IOPS、高吞吐、低时延的数据存储服务。更多信息，请参见什么是文件存储CPFS。

以CPFS为共享存储的E-HPC集群适合动画渲染、生命科学、气象预报、能源勘探等需要超高吞吐的应用场景。

使用限制

以CPFS为共享存储的E-HPC集群存在以下使用限制：

- 一个CPFS文件系统只能供一个E-HPC集群使用。
- CPFS客户端仅支持CentOS（7.2、7.3、7.4、7.6）操作系统。
- 扩容集群时，只能选择创建集群时的自定义镜像对集群进行扩容，否则会出现集群异常。

前提条件

已创建CPFS文件系统、添加挂载点，并获取CPFS管理节点。创建CPFS文件系统和添加挂载点操作，请参见管理文件系统和管理挂载点。

获取CPFS管理节点步骤如下：

1. 登录NAS控制台。
2. 在左侧导航栏，选择**文件系统 > 文件系统列表**。
3. 在**文件系统列表**页，单击已创建的CPFS文件系统ID。
4. 在文件系统详情页，单击**挂载使用**。

5. 登录如上图所示的第一台ECS实例，执行`mmlsmgr`命令，获取CPFS管理节点。

- 默认情况下，返回信息为 `cpfs-<cpfs-id>-qr-001` ，该节点则为CPFS管理节点。
- 如果返回信息为 `cpfs-<cpfs-id>-qr-002` ，登录 `cpfs-<cpfs-id>-qr-002` 节点，执行 `reboot` 命令，等待1分钟左右，再次执行 `mmlsmgr` 命令，若返回信息为 `cpfs-<cpfs-id>-qr-001` ，表示管理节点已经完成切换，可被E-HPC集群使用。

步骤一：创建自定义镜像

1. 创建一台ECS实例。

   具体操作，请参见使用向导创建实例。您需要注意以下配置参数：

   - **vCPU和内存**：选择至少包含2 vCPU、4 GiB内存的ECS实例，确保CPFS客户端软件正常运行。
   - **镜像**：选择CPFS客户端支持的CentOS（7.2、7.3、7.4、7.6）操作系统。
   - **网络**：必须与CPFS文件系统的专有网络和交换机保持一致。
   - **安全组**：必须与CPFS文件系统的安全组保持一致。
2. 登录ECS实例。

   具体操作，请参见通过密码或密钥认证登录Linux实例。
3. 安装客户端和依赖包。

   i. 运行以下命令，下载并解压RPM（Red Hat Package Manager）包。

   ```
   mkdir /tmp/rpms
   cd /tmp/rpms
   wget https://gpfs-rpms.oss-cn-beijing.aliyuncs.com/CPFS2.2-CentOS.tar.gz
   tar xvfz CPFS2.2-CentOS.tar.gz
   ```

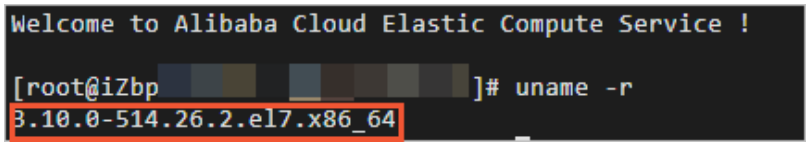
   ii. 运行以下命令，安装CPFS客户端的依赖软件。

   ```
   yum install -y cpp gcc gcc-c++ binutils ksh elfutils elfutils-devel rpm-build
   ```

   iii. 如果ECS实例为CentOS 7.2、7.3、7.4操作系统时，需安装对应系统版本的kernel-devel。

   a. 执行 `uname -r` 查看kernel版本。

   如返回的kernel如下所示：

   b. 在CentOS官网下载kernel版本对应kernel-devel的rpm包。

   ```
   wget https://buildlogs.centos.org/c7.1611.u/kernel/20170704132018/3.10.0-514.26
   .2.el7.x86_64/kernel-devel-3.10.0-514.26.2.el7.x86_64.rpm
   ```

   c. 安装kernel-devel。

   ```
   yum install kernel-devel-3.10.0-514.26.2.el7.x86_64.rpm -y
   ```

iv. 运行以下命令，安装CPFS客户端。

```
cd /tmp/rpms/CentOS/CentOS7
yum install -y gpfs.adv-*.x86_64.rpm gpfs.base-*.x86_64.rpm gpfs.docs-*.noarch.rpm
gpfs.gpl-*.noarch.rpm gpfs.gskit-*.x86_64.rpm gpfs.gss.pmsensors-*.x86_64.rpm gpfs.
license.dm-*.x86_64.rpm gpfs.msg.en_US-*.noarch.rpm
```

4. 运行以下命令，构建系统。

```
/usr/lpp/mmfs/bin/mmbuildgpl
```

当返回如下信息时，说明系统已构建。若无返回信息，请再次执行该命令。

```
------------------------------------------------------------
mmbuildgpl: Building GPL module completed successfully at Mon Aug 30 17:39:26 CST 2021.
------------------------------------------------------------
```

5. 在ECS实例的/etc/hosts文件中增加CPFS管理节点的Quorum和Contact内容。

i. 登录CPFS管理节点，获取/etc/hosts文件中的相关内容。

> **说明**
>
> 关于如何获取CPFS管理节点操作，请参见前提条件。

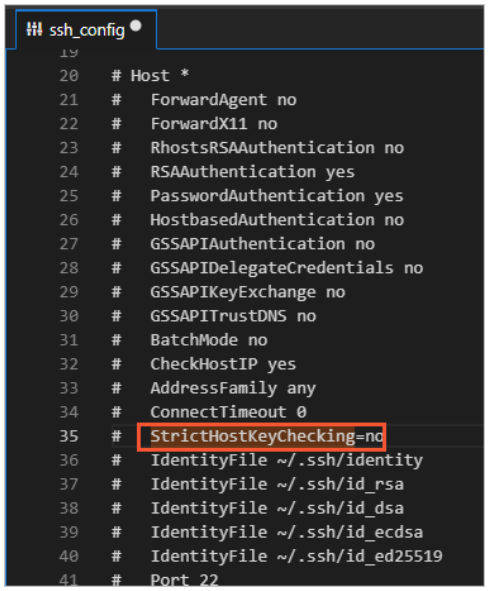
ii. 将获取的内容添加到ECS实例的/etc/hosts文件中。

```
172.**.**.87 cpfs-contact-node1 #CPFS_172_**_**_87_MAGIC

172.**.**.88 cpfs-contact-node2 #CPFS_172_**_**_88_MAGIC

172.**.**.89 cpfs-contact-node3 #CPFS_172_**_**_89_MAGIC
172.**.**.90 cpfs-0****a6-000001-qr-001 #CPFS_172_**_**_90_MAGIC

172.**.**.91 cpfs-0****a6-000001-qr-002 #CPFS_172_**_**_91_MAGIC
172.**.**.92 cpfs-0****a6-000001-qr-003 #CPFS_172_**_**_92_MAGIC
```

6. 在CPFS管理节点，获取CPFS客户端节点的免密钥登录文件。

i. 修改CPFS管理节点的/etc/ssh/ssh_config文件中的如下配置。

```
StrictHostKeyChecking=no
```



ii. 运行以下命令，将公钥信息拷贝至制作自定义镜像的ECS实例。

```
ssh-copy-id -i ~/.ssh/id_rsa.pub root@192.**.**.169
```

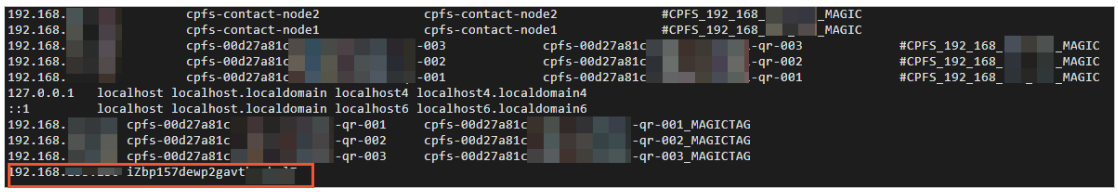
7. 使用ECS实例创建自定义镜像。

   具体操作，请参见使用实例创建自定义镜像。

8. 检查创建的自定义镜像是否可用。

> **说明**
>
> 此步骤中出现的192.168.XX.XX为ECS实例的IP，请替换成实际的IP。

i. 登录CPFS管理节点，在/etc/hosts中加入ECS实例的IP和HostName。



ii. 执行如下命令挂载CPFS文件系统。

```
mmaddnode -N 192.168.XX.XX
mmchlicense client --accept -N  192.168.XX.XX
mmchnode -N 192.168.XX.XX --perfmon
mmstartup -N 192.168.XX.XX
```

iii. 登录ECS实例，执行 `df -h` 检查是否挂载上CPFS。

当返回如下信息时，说明ECS实例已挂载上CPFS。

```
[root@iZbp157dewp2gavthszbylZ ~]# df -h
Filesystem               Size  Used Avail Use% Mounted on
/dev/vda1                 40G  2.9G   35G   8% /
devtmpfs                 1.8G     0  1.8G   0% /dev
tmpfs                    1.8G     0  1.8G   0% /dev/shm
tmpfs                    1.8G  464K  1.8G   1% /run
tmpfs                    1.8G     0  1.8G   0% /sys/fs/cgroup
tmpfs                    365M     0  365M   0% /run/user/0
00d27a81c      -000001  3.6T  432M  3.6T   1% /cpfs/00d27a81c      -000001
```

iv. 登录CPFS管理节点，执行如下命令卸载CPFS文件系统。

```
mmshutdown -N 192.168.XX.XX
mmdelnode -N 192.168.XX.XX
```

v. 删除在/etc/hosts中添加的ECS实例的IP和HostName。

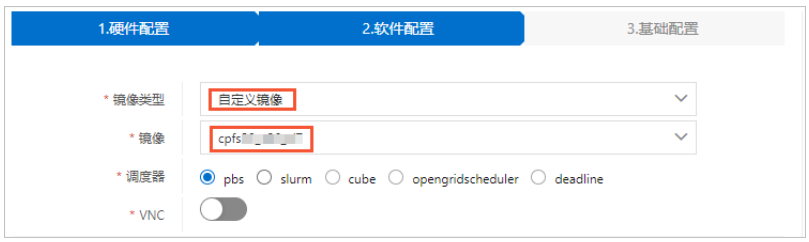
步骤二：创建集群

1. 登录弹性高性能计算控制台。
2. 在顶部菜单栏左上角处，选择地域。
3. 在左侧导航栏，选择**集群**。
4. 在**集群**页面，单击**创建集群**。
5. 在**创建集群**页面，完成填写集群配置信息。

   更多信息，请参见创建集群。您需要注意以下配置参数：

   - **文件系统类型**：选择**CPFS**。
   - **文件系统ID**和**挂载点**：选择您已经创建的CPFS文件系统ID和挂载点。

   - **镜像类型**：选择**自定义镜像**。
   - **镜像**：选择您制作的自定义镜像，创建集群时会自动安装CPFS客户端和对应的登录文件。



13.配置E-HPC集群与Windows AD域用户账号互通

您可以将线下AD域用户体系与云上E-HPC的集群LDAP用户账号建立互通，使用一套用户体系来减少用户账号的管理和维护成本。本文为您介绍配置域账户类型为LDAP的集群与Windows AD域建立互通的具体操作，从而实现AD域账号密码同步至集群的效果。

背景信息

本教程适用于首次配置账号互通，以及当Windows AD域的用户体系发生变化（如新增或删除用户）的场景。

请确保集群账户管理节点与Windows AD域机器网络互通。

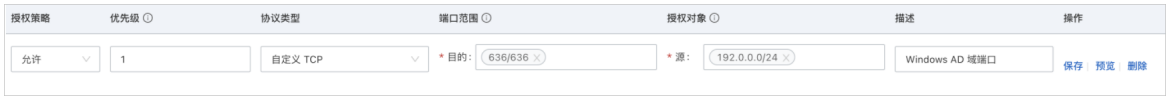
安装AD域并开启SSL服务

如果您已经在Windows机器上完成了AD域安装并开启了SSL服务，可以跳过本节，直接执行配置Windows AD域与LDAP用户账号互通。

1. 如果您的Windows AD域机器为云服务器ECS，需要先在网络安全组添加入方向636端口。具体操作，请参见添加安全组规则。

   请在**访问规则**的**入方向**页签下，手动添加636端口，配置信息如下：

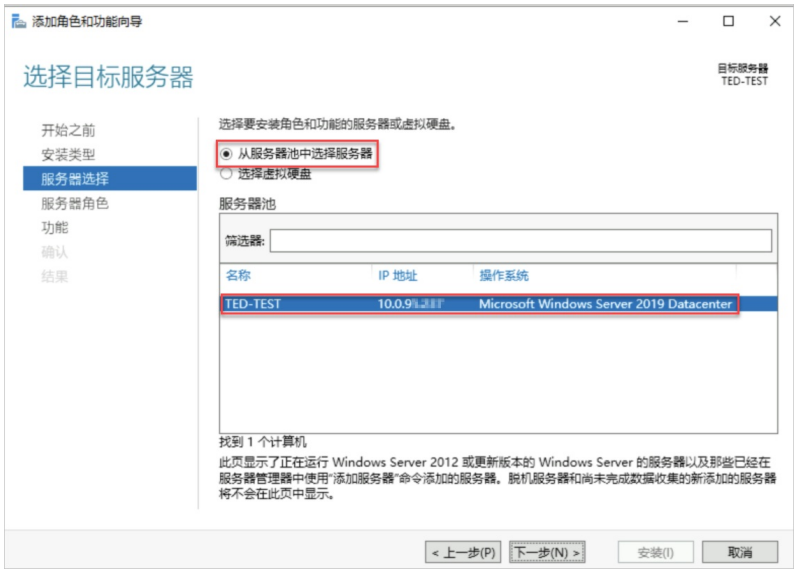
配置项	参数值
授权策略	允许
优先级	1
协议类型	自定义 TCP
端口范围	636/636
授权对象	ECS实例所处交换机的IPv4网段，例如192.0.0.0/24。
描述	端口描述信息，例如Windows AD域端口。



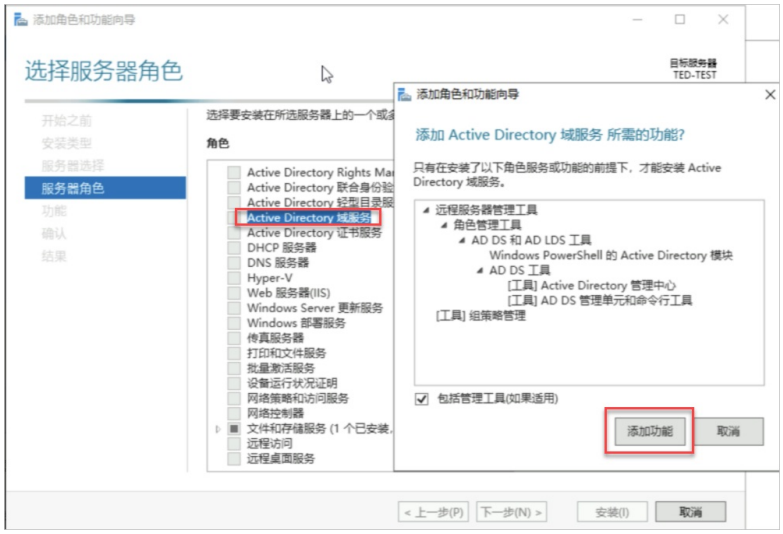
2. 在Windows桌面左下角单击，然后在Windows Server区域，单击**服务器管理器**。

> **说明**
>
> 用于安装AD域的本地机器或云服务器的操作系统，仅支持Windows Server 2012/2016/2019。

3. 在**服务器管理器**对话框，单击**添加角色和功能**。
4. 在**添加角色和功能向导**对话框的**开始之前**页面，单击**下一步(N)**。
5. 在**选择安装类型**页面，选中**基于角色或基于功能的安装**，然后单击**下一步(N)**。
6. 在**选择目标服务器**页面，选中**从服务器池中选择服务器**，然后选中服务器池中的本地服务器，单击**下一步(N)**。

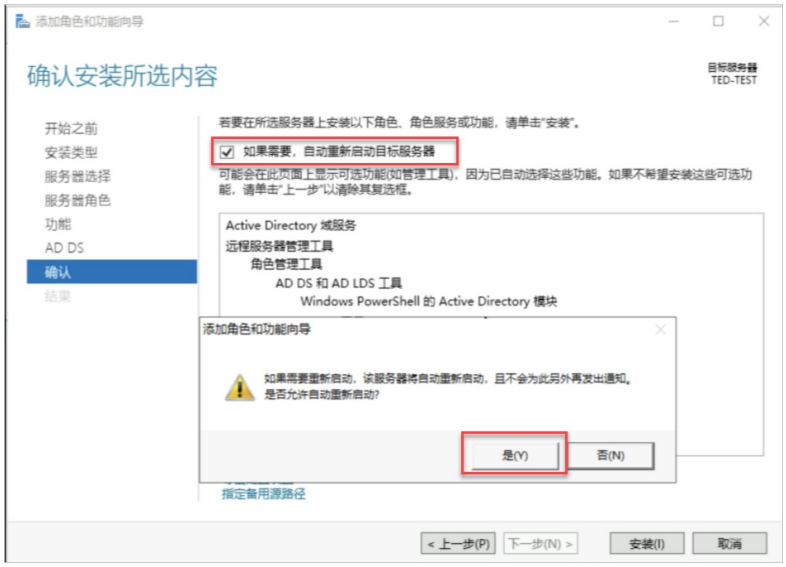


7. 安装Active Directory域服务角色。
   i. 在**选择服务器角色**页面，选中**Active Directory 域服务**，在弹出的**添加Active Directory域服务所需的功能**对话框，单击**添加功能**，然后单击**下一步(N)**。

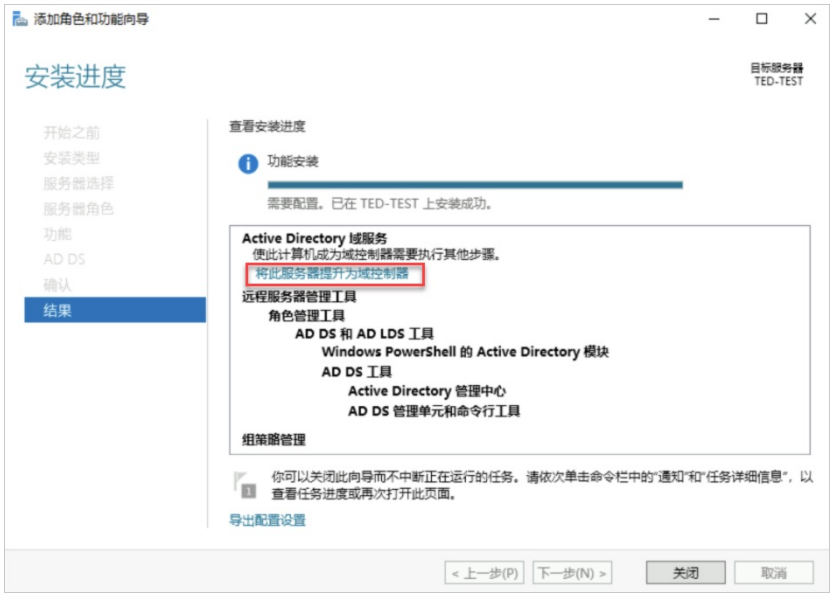


   ii. 在**选择功能**页面，保持默认选项，单击**下一步(N)**。
   iii. 在**Active Directory域服务**页面，单击**下一步(N)**。

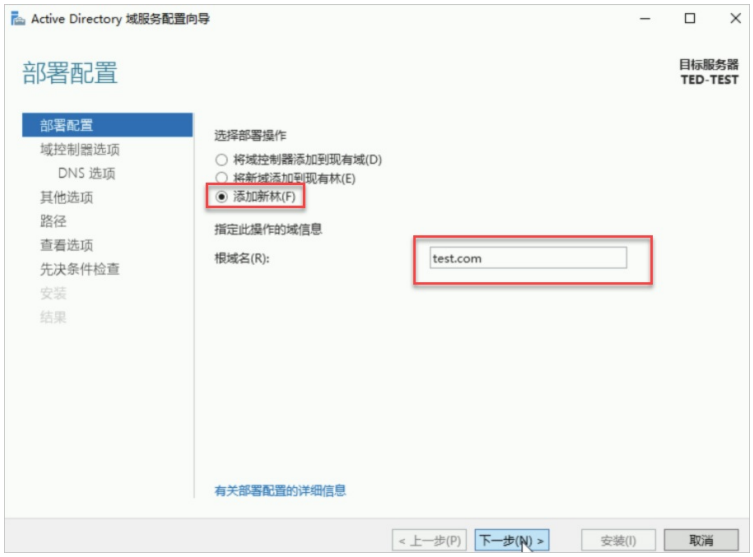
iv. 在**确认安装所选内容**页面选中**如果需要，自动重新启动目标服务器**，单击**安装(I)**。然后在弹出的对话框单击**是(Y)**。



8. Active Directory域服务角色安装完成后，单击**将此服务器提升为域控制器**，在弹出的**Active Directory 域服务配置向导**对话框完成相关配置，部署Active Directory域服务。



i. 在**Active Directory 域服务配置向导**对话框的**部署配置**页面，选择**添加新林(F)**，输入**根域名(R)**，然后单击**下一步(N)**。



ii. 在**域控制器选项**页面，设置**域密码**，然后单击**下一步(N)**。

iii. 在**DNS选项**页面，保持默认配置，单击**下一步(N)**。

iv. 在**其它选项**页面，保持默认配置，单击**下一步(N)**。

v. 在**路径**页面，保持默认配置，单击**下一步(N)**。

vi. 在**查看选项**页面，保持默认配置，单击**下一步(N)**。

vii. 在**先决条件检查**页面，等待检查通过，单击**安装(I)**。

> **说明**
>
> 如果不通过，请根据提示信息排查原因。

启动安装后，系统自动部署，完成后服务器将自动重启。

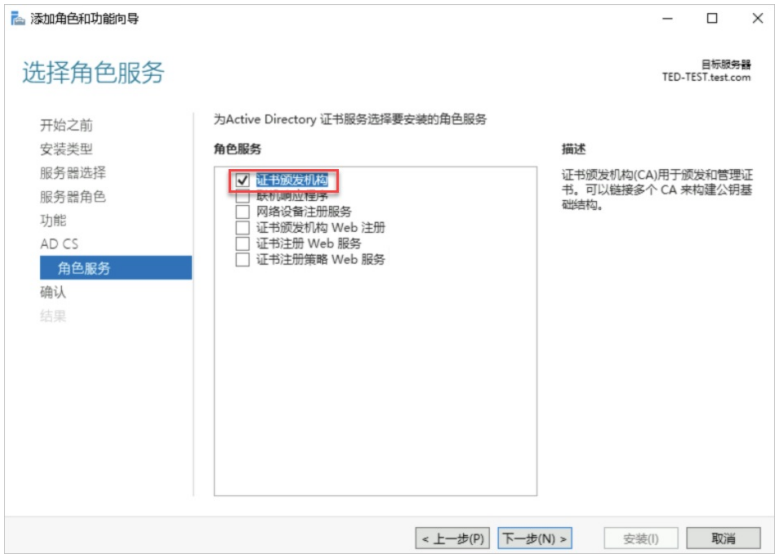
9. AD域控制器部署完成后，重新打开**服务器管理器**，安装Active Directory 证书服务。

i. 在**选择服务器角色**页面，选中**Active Directory 证书服务**，在弹出的对话框中，单击**添加功能**，然后单击**下一步(N)**。

ii. 在**选择功能**页面，保持默认选项，单击**下一步(N)**。

iii. 在**Active Directory 证书服务**页面，保持默认选项，单击**下一步(N)**。

iv. 在**选择角色服务**页面的**角色服务**区域，选中**证书颁发机构**，单击**下一步(N)**。



v. 在**确认安装所选内容**页面，选中**如果需要，自动重新启动目标服务器**，在弹出的对话框单击**是(Y)**，然后单击**安装(I)**。

10. Active Directory证书安装完成后，单击**配置目标服务器上的Active Directory证书服务**，在弹出的**AD CS配置**对话框完成相关配置。

i. 在**凭据**页面，保持默认选项，单击**下一步(N)**。

ii. 在**角色服务**页面，选中**证书颁发机构**，单击**下一步(N)**。

iii. 在**设置类型**页面，选中**企业 CA(E)**，然后单击**下一步(N)**。

iv. 在**CA类型**页面，选中**根 CA(R)**，然后单击**下一步(N)**。

v. 在**私钥**页面，选中**创建新的私钥(R)**，然后单击**下一步(N)**。

vi. 在**CA 的加密**页面，保持默认选项，单击**下一步(N)**。

vii. 在**CA 名称**页面，保持默认选项，单击**下一步(N)**。

viii. 在**有效期**页面，保持默认选项，单击**下一步(N)**。

ix. 在**CA 数据库**页面，保持默认选项，单击**下一步(N)**。

x. 在**确认**页面，单击**配置**，配置成功后，单击**关闭**。

重启服务器，在证书颁发机构中即可查看给域控制器颁发的证书。

11. 测试Windows AD域的LDAP协议是否生效。

i. 右键单击开始菜单，选择**运行(R)**。

ii. 在**运行**对话框中输入 `ldp.exe` ，然后在打开的**Ldp**窗口的导航栏，选择**连接(C) > 连接(C)**。

iii. 在**连接**对话框，输入服务器名称以及端口号389，然后单击**确定(O)**。

连接后显示类似如下信息，表示连接正常。

iv. 再次打开Ldp窗口，并在导航栏选择**连接(C) > 连接(C)**。

v. 在**连接**对话框，输入服务器名称以及端口号636，同时选中**SSL(L)**，然后单击**确定(O)**。

连接后显示类似如下信息，表示连接正常。

配置Windows AD域与LDAP用户账号互通

您需要提前准备Windows AD域机器的以下信息：

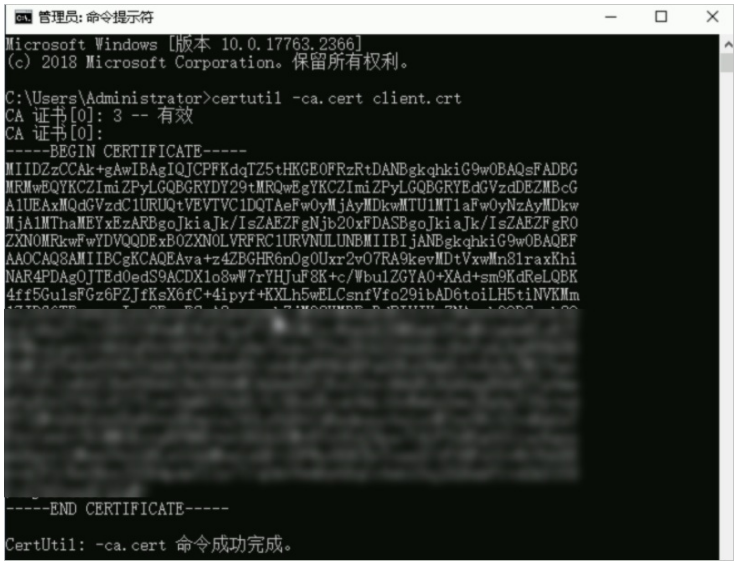
- IP地址：xxx.xxx.xxx.xxx，例如172.16.0.47。
- 主机名：xxxx.xxxx.xxxx，例如TED-ECS-WIN001.ted.com。
- Windows AD域密码。

具体配置操作如下：

1. 导出证书。
   i. 登录已安装了Windows AD域的机器。

ii. 打开命令行，输入 `certutil -ca.cert client.crt` 命令，生成证书。

> **说明**
>
> 证书文件位于C:\Users\Administrator目录下。



iii. 将生成的证书client.crt从AD服务器下载并上传至集群账户管理节点某个目录下，例如/root/。

> **说明**
>
> 您可以通过WinSCP工具进行文件传输。

2. 创建域账户类型为LDAP的集群。具体操作，请参见使用向导创建集群。

   您需要在创建时，将**域账号服务**设置为**ldap**，然后填入本地集群域名，该域名与Windows AD域的域名称相同。例如Windows主机名为TED-ECS-WIN001.ted.com，则本地集群域名为ted。

3. 创建完成后，登录集群。具体操作，请参见登录集群。
4. 执行如下命令，登录账户管理节点。

```
ssh account
```

5. 执行如下命令，配置Windows AD域与集群互通。

```
/usr/local/ehpc/bin/ehpcutil_py account connected --ad_hostname xxxx.xxxx.xxxx --ad_ip
xxx.xxx.xxx.xxx --ad_passwd xxxxxx
```

   部分参数说明如下：

   - `--ad_hostname`：Windows AD域主机名。
   - `--ad_ip`：Windows AD域IP。

- `--ad_passwd`：Windows AD域密码。

示例命令如下：

```
/usr/local/ehpc/bin/ehpcutil_py account connected --ad_hostname TED-ECS-WIN001.ted.com --ad_ip 172.16.0.47 --ad_passwd Alihpc123
```

6. 执行如下命令，在集群中导入从Windows AD域机器导出的证书。

```
/usr/local/ehpc/bin/ehpcutil_py account importcert --filename /root/client.crt --ad_passwd xxxxxx
```

部分参数说明如下：

`--filename`：证书存放位置。

`--ad_passwd`：Windows AD域密码。

示例命令如下：

```
/usr/local/ehpc/bin/ehpcutil_py account importcert --filename /root/client.crt --ad_passwd Alihpc123
```

7. 执行如下命令，实现账户同步。

```
/usr/local/ehpc/bin/ehpcutil_py account syncad
```

14.使用E-HPC集群调度器插件

14.1. E-HPC集群调度器插件

E-HPC提供了调度器插件作为平台的外扩组件，在E-HPC现有调度器类型或版本不满足当前业务时，您可以通过该插件构建自定义调度器并接入E-HPC平台的能力。本文为您介绍E-HPC集群调度器插件的概念及组成。

什么是调度器插件

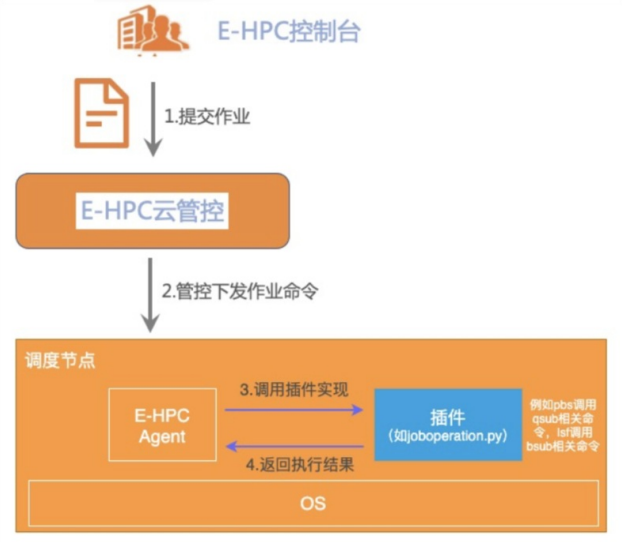
E-HPC作为一款PaaS平台，集成了常用的开源调度器来提供平台级服务。当您的业务需要迁移到云上时，往往需要将云下的调度器集成至云上，但因HPC行业调度器众多，且不同调度器有多种定制版本，会出现E-HPC内置调度器无法满足的情况。

因此，E-HPC提供了调度器插件作为平台的外扩组件，您可以使用E-HPC提供的调度器插件接入自定义调度器，避免受到调度器类型或版本的限制。例如，在EDA业务场景下，通常情况使用的调度器为商用调度器，但E-HPC平台无法提供商用License供您安装，此时，您可以自行安装调度器并通过调度器插件接入E-HPC平台的能力。

调度器插件为您提供了插件模版及配置文件，并将功能定义进行模块化分拆，您可以根据自身业务需求及调度器特征进行任意方式的自定义实现。在构建出自定义调度器插件之后，即可在E-HPC控制台创建带有插件的集群，无缝衔接至E-HPC以提供对应的节点管理、作业管理、自动伸缩等能力。

插件使用流程

以在E-HPC控制台提交作业为例，为您展示调度器插件在集群操作中的具体作用，插件示例流程图及说明如下：



1. 登录E-HPC控制台，选择指定集群并提交作业。
2. E-HPC云管控接收到控制台的请求，向指定集群下发作业命令。
3. 集群调度器节点识别插件类型，下载调度器插件到本地路径，并解析插件配置文件JobSubmit功能项是否打开。若JobSubmit=false，则返回插件功能不支持的错误，反之则调用具体插件功能并实现。
4. 调用插件中的作业提交实现代码，例如pbs会调用qsub相关命令、lsf会调用bsub相关命令。作业提交执行完成，则返回执行结果。

调度器插件组成

调度器插件主要分为以下两个组成部分，目录结构如下图所示：

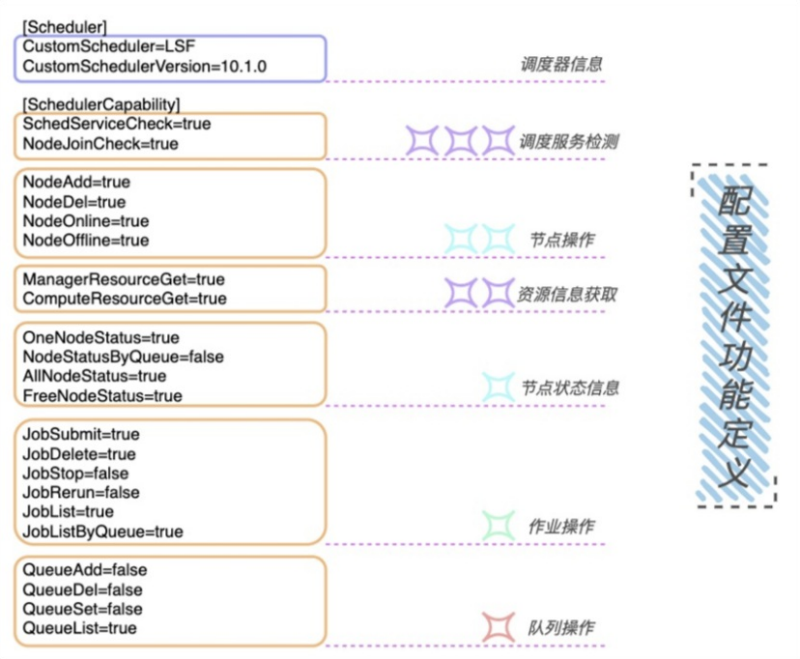
```
[root@manager plugin]# tree -A -C
.
├── ehpc_custom.conf
└── LSF
    └── 10.1.0
        ├── pluginjoboperation.py
        ├── pluginnodeoperation.py
        ├── pluginnodestatus.py
        ├── pluginqueueoperation.py
        ├── pluginresourceget.py
        └── pluginschedulercheck.py
```

目录结构说明如下：

1. ehpc_custom.conf：插件配置文件，记录插件包含的调度器信息及调度器可接入的功能项。更多信息，请参见调度器插件配置文件。
2. *.py：根据调度器模版，对自定义调度器进行具体功能实现的脚本文件。该文件需要位于/<调度器名>/<调度器版本号>的二级目录下，例如/LSF/10.1.0。

调度器插件配置文件

插件配置文件定义了调度器信息及调度器可接入功能项，详细功能定义如下所示：



其中，[Scheduler]表示调度器信息，包含了调度器名称及版本号。[SchedulerCapability]表示调度器可接入功能项，包含了众多可接入E-HPC的调度器功能，等号左侧表示调度器功能名称，等号右侧表示是否打开该功能。功能项具体说明如下：

- 调度服务检测（三星）：通过检测节点调度服务设置集群在控制台展示的节点状态。
- 节点操作（两星）：通过节点操作可以在控制台实现手动扩容或缩容。
- 资源信息获取（两星）：通过节点资源信息获取到正确资源在控制台展示。
- 节点状态信息（一星）：通过获取不同节点状态在控制台实现自动伸缩能力。
- 作业操作（一星）：通过作业操作在控制台实现提交作业、查询作业等能力。
- 队列操作（一星）：通过队列操作在控制台实现增加队列、查询队列等能力。

> **说明**
>
> 星级越高，代表该功能越基础。并且，调度服务检测作为最基础的功能，只有当该功能项设置为true时，其他功能项可用。

14.2. 构建调度器插件

您需要先构建自定义调度器插件后，才可以在E-HPC控制台创建带有插件的集群。本文以LSF插件为例，为您介绍构建调度器插件的具体操作。

操作步骤

1. 在本地机器上创建插件目录结构。

   调度器插件目录结构的更多信息，请参见调度器插件组成。

   ```
   mkdir /plugin
   mkdir /plugin/LSF
   mkdir /plugin/LSF/10.1.0
   ```

2. 使用公网下载插件配置文件及功能模版。

   i. 下载调度器插件配置文件。

   ```
   cd /plugin
   wget https://public-ehpc-package.oss-cn-hangzhou.aliyuncs.com/plugintemplate/ehpc_custom.conf
   ```

   ii. 下载调度器插件功能模版。

   ```
   wget -P /plugin/LSF/10.1.0 https://public-ehpc-package.oss-cn-hangzhou.aliyuncs.com/plugintemplate/plugin_template.tar.gz
   ```

3. 编辑配置文件。

   根据自身需求及功能实现修改调度器配置文件，将需要支持的功能项设置为true，无需支持的功能项设置为false。配置文件的更多信息，请参见调度器插件配置文件。

   假设，您需要在插件中实现节点操作中的添加节点和删除节点功能，以及队列操作的队列枚举功能，那么只需在配置文件中将NodeAdd与NodeDel设置为true，将NodeOnline与NodeOffline设置为false，并且将QueueList设置为true，将QueueAdd、QueueDel和QueueSet设置为false即可。配置示例如下图所示：

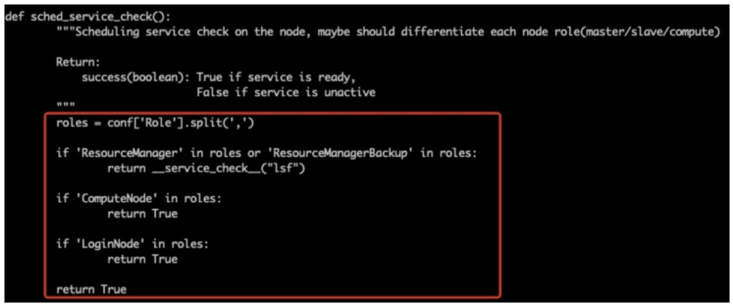
   ```
   [SchedulerCapability]
   NodeAdd=true
   NodeDel=true
   NodeOnline=false
   NodeOffline=false
   QueueAdd=false
   QueueDel=false
   QueueSet=false
   QueueList=true
   ```

4. 编辑完成后，执行如下命令解压插件模版。

   ```
   cd LSF/10.1.0
   tar xvfz /plugin/LSF/10.1.0/plugin_template.tar.gz
   ```

5. 依据插件模版实现自定义调度器功能。

   假设，您需要实现节点调度服务检测的功能，那么需在解压后的模版文件中找到对应服务检测功能的模版文件pluginschedulercheck.py，并在该文件中找到sched_service_check函数，然后进行调度器功能实现。如下图所示：



```
def sched_service_check():
        """Scheduling service check on the node, maybe should differentiate each node role(master/slave/compute)

        Return:
            success(boolean): True if service is ready,
                              False if service is unactive
        """
        roles = conf['Role'].split(',')

        if 'ResourceManager' in roles or 'ResourceManagerBackup' in roles:
                return __service_check__("lsf")

        if 'ComputeNode' in roles:
                return True

        if 'LoginNode' in roles:
                return True

        return True
```

   其中，红框部分为此示例的功能实现。您需要根据不同的当前节点类型自定义返回。本示例对于计算节点和登录节点角色，在调度服务检测实现中返回true以表示检测通过，而对于管理节点角色，需要先检测lsf服务是否在节点上正常运行，然后再返回最终的检测结果。

6. 构建调度器插件。

   实现了具体调度器功能后，在主目录执行 `tar` 命令进行压缩，即可形成最终的调度器插件。

```
cd /plugin
tar cvfz lsf_plugin.tgz *
```

14.3. 创建带有插件的集群

当您构建完自定义调度器插件后，即可参考本文操作，选择OpenAPI或控制台方式创建带有插件的集群。

插件接入模式

插件的接入模式主要分为以下两种：

- Image模式：

  将插件解压至自定义镜像指定路径下，后续可以直接使用该自定义镜像创建带有插件的E-HPC集群。创建自定义镜像的具体操作，请参见使用实例创建自定义镜像。

  > **说明**
  >
  > 插件在自定义镜像中的存放路径由pluginLocalPath指定。

- OSS模式：

  将打包好的插件上传到指定OSS Bucket，后续在控制台创建集群时会自动下载插件到本地，OSS上传文件的具体操作，请参见上传文件。

  > **说明**
  >
  > 创建集群时，插件所在的OSS路径由pluginOssPath指定，插件下载到本地的路径由pluginLocalPath指定。

使用OpenAPI创建带有插件的集群

通过CreateCluster接口创建带有插件的E-HPC集群，关于CreateCluster接口的更多信息，请参见CreateCluster。不同接入模式需要设置的参数如下：

- Image接入模式
  - 设置请求参数SchedulerType为custom。
  - 设置请求参数Plugin为'{ 'pluginMod': 'image', 'pluginLocalPath': '<local-path>' }'。

    其中，参数说明如下：
    - pluginMod：插件模式。设置为image模式。
    - pluginLocalPath：插件存放的本地路径。建议设置为非共享存储目录。
- OSS接入模式
  - 设置请求参数SchedulerType为custom。
  - 设置请求参数Plugin为'{ 'pluginMod': 'oss', 'pluginLocalPath': '<local-path>','pluginOssPath': '<oss-path>'}'。

    其中，参数说明如下：
    - pluginMod：插件模式。设置为OSS模式。
    - pluginLocalPath：插件存放的本地路径。建议设置为共享存储目录。
    - pluginOssPath：插件放置在OSS上的远程路径。

> **说明**
>
> 如果您的域账号服务也需要通过插件进行功能定制，请您在设置以上参数时，将AccountType参数设置为custom。关于构建域账号插件的具体操作，您可以提交工单获取支持。

使用控制台创建带有插件的集群

创建不同接入模式集群的操作如下：

- Image接入模式
  i. 登录弹性高性能计算控制台。
  ii. 在顶部菜单栏左上角处，选择地域。
  iii. 在左侧导航栏，单击**集群**。
  iv. 在**集群**页面右上角，单击**创建集群**。

  根据自身业务场景进行设置，在**软件配置**阶段，需要注意的配置项如下：
  - **调度器**：选择**不安装**。
  - **域账户服务**：选择nis或ldap。
  - **插件配置**：将模式设置为**镜像模式**。
  - **本地存放位置**：填写插件存放的本地路径。

  创建完成后，在E-HPC集群页面，查看新创建的集群状态。若新创建的集群所有节点都处于**运行中**状态，则集群已创建完成。
- OSS接入模式
  i. 登录弹性高性能计算控制台。

ii. 在顶部菜单栏左上角处，选择地域。

iii. 在左侧导航栏，单击**集群**。

iv. 在**集群**页面右上角，单击**创建集群**。

根据自身业务场景进行设置，在**软件配置**阶段，需要注意的配置项如下：

- **调度器**：选择**不安装**。
- **域账户服务**：选择nis或ldap。
- **插件配置**：将模式设置为**OSS模式**。
- **本地存放位置**：填写插件存放的本地路径。
- **OSS插件位置**：填写插件放置在OSS上的远程路径。

创建完成后，在E-HPC集群页面，查看新创建的集群状态。若新创建的集群所有节点都处于**运行中**状态，则集群已创建完成。

> 说明
>
> 如果您的域账号服务也需要通过插件进行功能定制，您需要将**域账户服务**设置为**不安装**。关于构建域账号插件的具体操作，您可以提交工单获取支持。

14.4. 附录：调度器插件的常用OpenAPI说明

本文将为您介绍常用的OpenAPI会调用到插件功能中的具体功能项，帮助您了解集群调度器插件功能与常用OpenAPI之间的关联关系，从而进行自身业务的具体实现。

背景信息

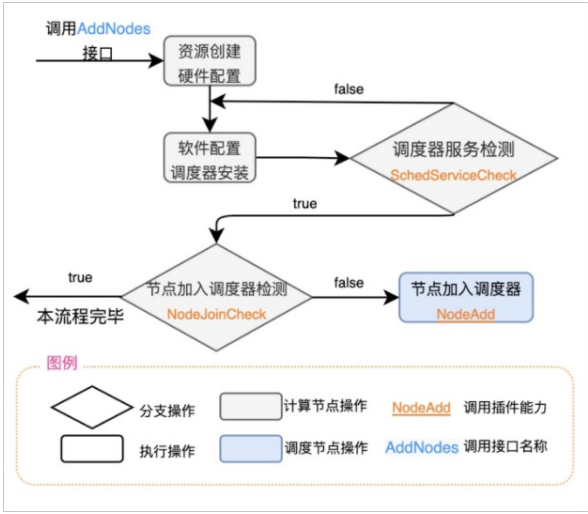
本文示例仅针对纯调度器插件场景，如果您的业务还需要同时配置域账号插件，请添加域账号相关功能项。

节点扩容（AddNodes）

关联说明

调用节点扩容接口AddNodes，将会涉及SchedServiceCheck、NodeJoinCheck和NodeAdd三种调度器插件功能，因此需要在配置文件中将这三个功能项设置为true，并且实现对应的插件函数功能。

流程说明



流程说明如下：

1. 调用**AddNodes**接口后，云管控开始生产计算节点资源，即启动硬件配置、软件配置等操作。
2. 软件配置阶段，在安装调度器过程中，系统会定时地调用服务检测功能（SchedServiceCheck），检测调度器是否安装成功。若不成功则继续等待软件安装，若成功则继续执行下一步。
3. 执行节点加入调度器检测（NodeJoinCheck）。在此定时检测中，若检测节点未加入调度器，则调度器节点开始执行节点加入调度器（NodeAdd）的操作，反之则表示计算节点已经加入调度器，节点状态为运行中。

插件功能说明

> **说明**
>
> 以下仅说明需要您自行实现的插件功能。

- SchedServiceCheck
  - 功能：检测调度服务是否正常，需要根据节点角色（调度节点或计算节点）分别实现。
  - 具体实现：
    - 源文件：pluginschedulercheck.py。
    - 实现函数：sched_service_check()。
  - 实现举例：例如pbs调度器通过 `systemctl status pbs` 命令检测节点自身pbs服务是否正常运行。
- NodeJoinCheck
  - 功能：检测计算节点是否已经加入到调度器中。
  - 具体实现：
    - 源文件：pluginschedulercheck.py。
    - 实现函数：node_join_check()。
  - 实现举例：例如pbs调度器通过 `pbsnodes` 命令检测自身是否已经在调度器中。
- NodeAdd

- 功能：在调度节点上将该节点加入到调度器中。
- 具体实现：
  - 源文件：pluginnodeoperation.py。
  - 实现函数：node_add()。
- 实现举例：例如pbs调度器通过 `qmgr` 命令将该节点加入到调度器中。

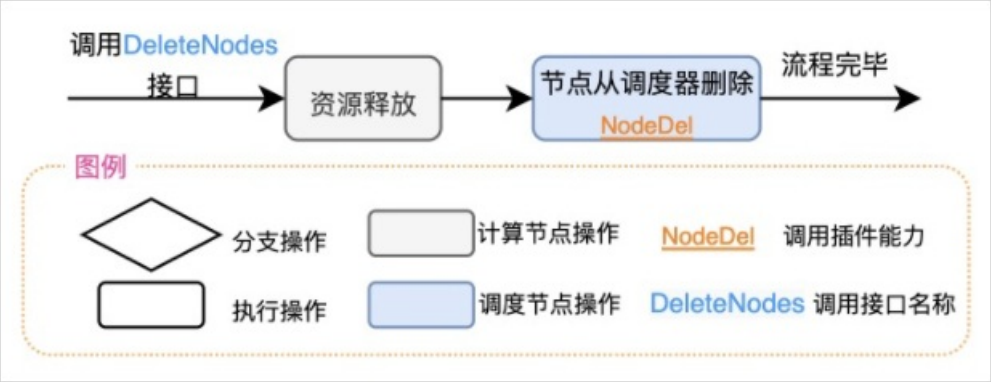
节点缩容（DeleteNodes）

关联说明

调用节点缩容接口DeleteNodes，将会涉及NodeDel调度器插件功能，因此需要在配置文件中将该功能项设置为true，并且实现对应的插件函数功能。

流程说明

调用DeleteNodes接口，计算节点将开始进行资源释放，此时调度器节点开始执行节点从调度器删除（NodeDel）的操作。流程图如下：



插件功能说明

> 说明
> 以下仅说明需要您自行实现的插件功能。

NodeDel

- 功能：在调度节点上将该节点从调度器中删除。
- 具体实现：
  - 源文件：pluginnodeoperation.py。
  - 实现函数：functionnode_del()。
- 实现举例：例如pbs调度器通过 `qmgr` 命令将该节点从调度器中删除。

15.测试E-HPC性能

15.1. 使用HPL测试集群浮点性能

本文介绍如何使用HPL测试E-HPC集群的浮点性能。

背景信息

HPL（The High-Performance Linpack Benchmark）是测试高性能计算集群系统浮点性能的基准。HPL通过对高性能计算集群采用高斯消元法求解一元N次稠密线性代数方程组的测试，评价高性能计算集群的浮点计算能力。

浮点计算峰值是指计算机每秒可以完成的浮点计算次数，包括理论浮点峰值和实测浮点峰值。理论浮点峰值是该计算机理论上每秒可以完成的浮点计算次数，主要由CPU的主频决定。理论浮点峰值 = CPU主频×CPU核数×CPU每周期执行浮点运算的次数。本文将为您介绍如何利用HPL测试实测浮点峰值。

准备工作

1. 创建一个E-HPC集群。具体操作，请参见使用向导创建集群。

   配置集群时，软硬件参数配置如下：

参数	说明
硬件参数	部署方式为精简，包含1个管控节点和1个计算节点，规格如下： ○ 管控节点：采用ecs.c6.large实例规格，该规格配置为2 vCPU，4 GiB内存。 ○ 计算节点：采用ecs.scch5.16xlarge实例规格，该规格配置为64 vCPU，32个物理内核，192 GiB内存。
软件参数	镜像选择CentOS 7.6公共镜像，调度器选择pbs。

2. 创建一个集群用户。具体操作，请参见创建用户。

   集群用户用于登录集群，进行编译软件、提交作业等操作，配置用户权限时，权限组请选择**sudo权限组**。

3. 安装软件。具体操作，请参见安装软件。

   需安装的软件如下：

   - linpack，版本为2018。
   - intel-mpi，版本为2018。

步骤一：准备算例文件

测试前您需要在本地准备好算例文件HPL.dat，文件包含了HPL运行的参数。如下示例是在单台scch5实例上运行HPL的推荐配置。

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
1            # of problems sizes (N)
143360 256000 1000         Ns
1            # of NBs
384 192 256      NBs
1            PMAP process mapping (0=Row-,1=Column-major)
1            # of process grids (P x Q)
1 2          Ps
1 2          Qs
16.0         threshold
1            # of panel fact
2 1 0        PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
2            NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
1 0 2        RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
0            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
0            DEPTHs (>=0)
0            SWAP (0=bin-exch,1=long,2=mix)
1            swapping threshold
1            L1 in (0=transposed,1=no-transposed) form
1            U  in (0=transposed,1=no-transposed) form
0            Equilibration (0=no,1=yes)
8            memory alignment in double (> 0)
```

测试过程中您可以根据节点的硬件配置，调整HPL.dat文件中相关参数，参数的说明如下所示。

- 第5~6行内容。

  ```
  1            # of problems sizes (N)，N表示求解的矩阵数量与规模
  143360 256000 1000         Ns
  ```

  N表示求解的矩阵数量与规模。矩阵规模N越大，有效计算所占的比例也越大，系统浮点处理性能也就越高。但矩阵规模越大会导致内存消耗量越多，如果系统实际内存空间不足，使用缓存、性能会大幅度降低。矩阵占用系统总内存的80%左右为最佳，即N×N×8=系统总内存×80%（其中总内存的单位为字节）。

- 第7~8行内容。

  ```
  1            # of NBs，NB表示求解矩阵过程中矩阵分块的大小
  384 192 256      NBs
  ```

  求解矩阵过程中矩阵分块的大小。分块大小对性能有很大的影响，NB的选择和软硬件许多因素密切相关。NB值的选择主要是通过实际测试得出最优值，一般遵循以下规律：

  - NB不能太大或太小，一般小于384。
  - NB×8一定是缓存行的倍数。

- NB的大小和通信方式、矩阵规模、网络、处理器速度等有关系。

一般通过单节点或单CPU测试可以得到几个较好的NB值，但当系统规模增加、问题规模变大，有些NB取值所得性能会下降。因此建议在小规模测试时选择3个性能不错的NB值，再通过大规模测试检验这些选择。

- 第10~12行内容。

```
1            # of process grids (P x Q)，P表示水平方向处理器个数，Q表示垂直方向处理器个数
1 2          Ps
1 2          Qs
```

P表示水平方向处理器个数，Q表示垂直方向处理器个数。P×Q表示二维处理器网格。P×Q=系统CPU数=进程数。一般情况下一个进程对应一个CPU，可以得到最佳性能。对于Intel® Xeon®，关闭超线程可以提高HPL性能。P和Q的取值一般遵循以下规律：

- P≤Q，一般情况下P的取值小于Q，因为列向通信量（通信次数和通信数据量）要远大于横向通信。
- P建议选择2的幂。HPL中水平方向通信采用二元交换法（Binary Exchange），当水平方向处理器个数P为2的幂时性能最优。

步骤二：提交作业

1. 登录弹性高性能计算控制台。
2. 在左侧导航栏，单击**作业**。
3. 在集群列表中选择目标集群。
4. 上传算例文件。
   i. 在**作业**页面，单击**作业文件编辑**，然后单击**浏览集群文件**。
   ii. 在弹出的对话框中输入集群用户名和密码，单击**确定**。
   iii. 打开要上传算例文件的路径，右键文件夹，选择**上传文件**。

   建议上传到集群用户的共享home目录下，例如集群用户名为hpltest，则上传到`/home/hpltest`目录下。
   iv. 选择步骤一准备好的算例文件HPL.dat，单击**打开**。
5. 创建作业脚本文件。
   i. 在算例文件所在路径，右键文件夹，单击**新建文件**，然后输入文件名（例如hpl.pbs），单击**确定**。

ii. 打开新建的hpl.pbs脚本文件，拷贝以下脚本内容后保存。

脚本内容示例如下：

> **说明**
>
> 本示例测试单节点的实测浮点峰值。如果您想测试多个节点的实测浮点峰值，可以修改如下配置文件。

```
#!/bin/sh
#PBS -j oe
export MODULEPATH=/opt/ehpcmodulefiles/
module load linpack/2018
module load intel-mpi/2018
echo "run at the beginning"
mpirun -n 1 -host <node0> /opt/linpack/2018/xhpl_intel64_static > hpl-ouput    #测试单节点的浮点性能，<node0>为运行作业的节点名称
#mpirun -n <N> -ppn 1 -host <node0>,...,<nodeN> /opt/linpack/2018/xhpl_intel64_static > hpl-ouput    #测试多节点的浮点性能
```

6. 提交作业。

i. 在**作业**页面，单击**提交作业**。

ii. 配置作业相关参数。

作业执行命令请填写作业脚本文件所在路径，例如/home/hpltest/hpl.pbs。其他参数保持默认即可。

iii. 单击右上角的**提交作业**。

iv. 在弹出的对话框中输入集群用户名和密码，单击**确定**。

查看结果

作业运行完成后，您可以查看作业结果。

1. 在**集群**页面，找到目标集群，单击**远程连接**。
2. 在**远程连接**页面，输入集群用户名和密码，单击**ssh连接**。
3. 执行以下命令，查看作业结果。

```
cat  /home/hpltest/hpl-ouput
```

本次测试结果如下图所示。

```
================================================================================
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC00C2R2      143360   384     1     1             913.61              2.15001e+03
HPL_pdgesv() start time Tue Mar 23 10:21:03 2021

HPL_pdgesv() end time   Tue Mar 23 10:36:17 2021

--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0041543 ...... PASSED
================================================================================
```

15.2. 使用Stream测试集群内存带宽性能

本文以Stream软件为例，介绍如何测试E-HPC集群内存和带宽性能。

背景信息

Stream软件是内存带宽性能测试的基准工具，也是衡量服务器内存性能指标的通用工具。Stream软件具有良好的空间局部性，是对转换检测缓冲区TLB（Translation Lookaside Buffer）友好、缓存友好的一款软件。STREAM软件支持复制（Copy）、尺度变换（Scale）、矢量求和（Add）、复合矢量求和（Triad）四种运算方式测试内存带宽的性能。

准备工作

1. 创建一个E-HPC集群。具体操作，请参见使用向导创建集群。

   配置集群时，计算节点请选择vCPU核数≥4的规格，例如ecs.c7.xlarge。

2. 创建一个集群用户。具体操作，请参见创建用户。

   集群用户用于登录集群，进行编译软件、提交作业等操作，配置用户权限时，权限组请选择**sudo权限组**。

3. 安装Stream软件。具体操作，请参见安装软件。

操作步骤

1. 登录弹性高性能计算控制台。
2. 重新编译Stream软件，指定相关参数。
   i. 在**集群**页面，找到目标集群，单击**远程连接**。
   ii. 输入root用户名和密码，单击**ssh连接**。
   iii. 执行以下命令，重新编译tream软件。

   ```
   cd /opt/stream/2018/
   gcc stream.c -O3 -fopenmp -DSTREAM_ARRAY_SIZE=1024*1024*1024 -DNTIMES=20 -mcmodel=m
   edium -o stream.1g.20
   ```

   命令相关参数说明如下：

   - DSTREAM_ARRAY_SIZE：指定Stream软件一次搬运的数据量。
   - DTIMES：指定迭代次数。

3. 创建作业脚本文件。
   i. 在**作业**页面，单击**作业文件编辑**，然后单击**浏览集群文件**。
   ii. 在弹出的对话框中输入集群用户名和密码，单击**确定**。
   iii. 打开要创建脚本文件的路径，右键文件夹，单击**新建文件**，然后输入文件名（例如stream.pbs），单击**确定**。

   建议将脚本文件保存集群用户的共享home目录下，例如集群用户名为streamtest，则在`/home/streamtest`目录下创建脚本文件。

iv. 打开新建的stream.pbs脚本文件，拷贝以下脚本内容后保存。

脚本示例如下：

```
#!/bin/sh
#PBS -j oe
#PBS -l select=1:ncpus=4
#本示例使用集群1个计算节点的4 vCPU进行高性能计算。实际应用中请根据节点配置修改。
export MODULEPATH=/opt/ehpcmodulefiles/
module load stream/2018
echo "run at the beginning"
OMP_NUM_THREADS=1 /opt/stream/2018/stream.1g.20 > stream-1-thread.log
OMP_NUM_THREADS=2 /opt/stream/2018/stream.1g.20 > stream-2-thread.log
OMP_NUM_THREADS=3 /opt/stream/2018/stream.1g.20 > stream-3-thread.log
OMP_NUM_THREADS=4 /opt/stream/2018/stream.1g.20 > stream-4-thread.log

#OMP_NUM_THREADS=<N> /opt/stream/2018/stream.1g.20 > stream-<N>-thread.log
```

4. 提交作业。
   i. 在**作业**页面，单击**提交作业**。
   ii. 配置作业相关参数。

   作业执行命令请填写作业脚本文件所在路径，例如/home/streamtest/stream.pbs。其他参数保持默认即可。
   iii. 单击右上角的**提交作业**。
   iv. 在弹出的对话框中输入集群用户名和密码，单击**确定**。

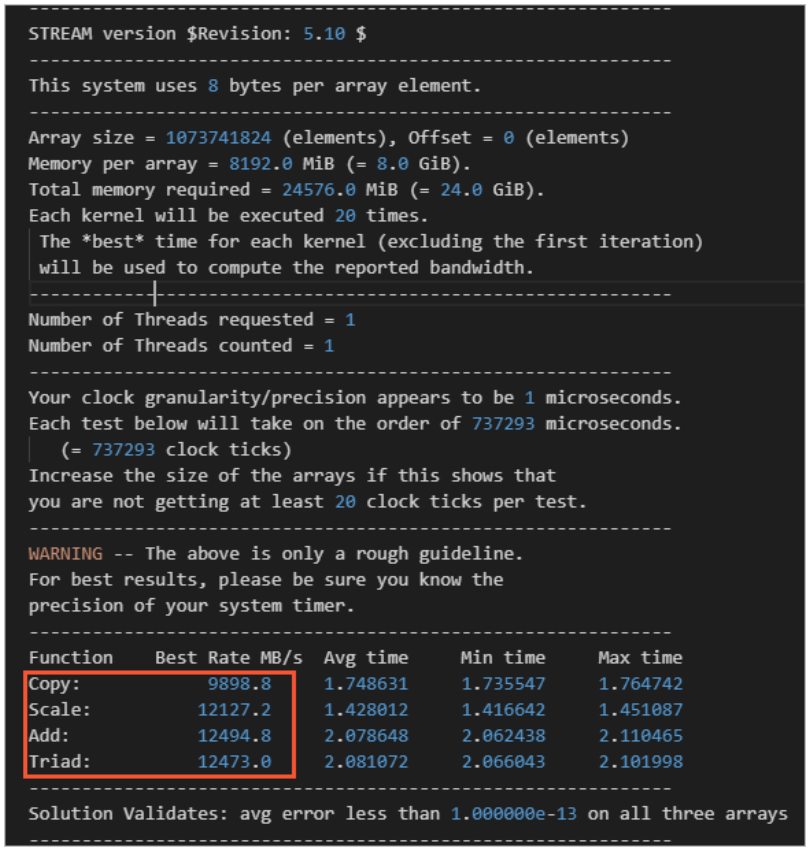
查看结果

作业运行完成后，您可以查看作业结果。

1. 在**集群**页面，找到目标集群，单击**远程连接**。
2. 在**远程连接**页面，输入集群用户名、登录密码和端口，单击**ssh连接**。
3. 执行以下命令，查看作业结果。

```
cat /home/streamtest/stream-2-thread.log
```

本次测试结果如下图所示。



15.3. 使用IMB软件和MPI通信库测试集群通信性能

本文以IMB软件和MPI通信库为例介绍如何测试E-HPC集群的通信性能。

背景信息

IMB（Intel MPI Benchmarks）用于评估HPC集群在不同消息粒度下节点间点对点、全局通信的效率。

MPI（Message Passing Interface）是支持多编程语言编程的并行计算通信库，具有高性能、大规模性、可移植性、可扩展性等特点。

准备工作

1. 创建一个E-HPC集群。具体操作，请参见使用向导创建集群。

   配置集群时，计算节点请选择vCPU核数≥8的规格，例如ecs.c7.2xlarge。

2. 创建一个集群用户。具体操作，请参见创建用户。

   集群用户用于登录集群，进行编译软件、提交作业等操作，配置用户权限时，权限组请选择**sudo权限组**。

3. 安装软件。具体操作，请参见安装软件。

   需安装的软件如下：

   - intel-mpi，版本为2018。
   - intel-mpi-benchmarks，版本为2019。

步骤一：准备算例文件

测试前您需要在本地准备好算例文件IMB.dat，文件中包含了IMB运行的参数。以下为算例文件的示例，您可以在测试过程中根据实际情况调整参数。

```
/opt/intel-mpi-benchmarks/2019/IMB-MPI1 -h    #查看IMB支持的通信模式及参数说明

cd /home/<user>/<work_dir>     #请将<user>修改成实际用户名，只能使用非root用户执行

/opt/intel/impi/2018.3.222/bin64/mpirun -genv I_MPI_DEBUG 5 -np 2 -ppn 1 -host <node0>,<nod
e1> /opt/intel-mpi-benchmarks/2019/IMB-MPI1 pingpong    #测试两节点间pingpong通信模式效率，获取
通信延迟和带宽，-genv I_MPI_DEBUG 打印mpi debug信息；-np 指定mpi总进程数
# -ppn 指定每个节点的进程数
# -host 指定任务节点列表
# -npmin 指定最少运行的进程数
# -msglog 指定消息片粒度范围

/opt/intel/impi/2018.3.222/bin64/mpirun -genv I_MPI_DEBUG 5 -np <N*2> -ppn 2 -host <node0>,
...,<nodeN> /opt/intel-mpi-benchmarks/2019/IMB-MPI1 -npmin 2 -msglog 19:21 allreduce
#测试N节点间allreduce通信模式效率，每个节点开启两个进程，获取不同消息粒度下的通信时间，请将<node0>等参
数修改成实际节点名称
/opt/intel/impi/2018.3.222/bin64/mpirun -genv I_MPI_DEBUG 5 -np <N> -ppn 1 -host <node0>,..
.,<nodeN> /opt/intel-mpi-benchmarks/2019/IMB-MPI1 -npmin 1 -msglog 15:17 alltoall
#测试N节点间alltoall通信模式效率，每个节点开启一个进程，获取不同消息粒度下的通信时间，请将<node0>等参
数修改成实际节点名称
```

步骤二：提交作业

1. 登录弹性高性能计算控制台。
2. 在左侧导航栏，单击**作业**。
3. 在集群列表中选择目标集群。
4. 上传算例文件。
   i. 在**作业**页面，单击**作业文件编辑**，然后单击**浏览集群文件**。
   ii. 在弹出的对话框中输入集群用户名和密码，单击**确定**。
   iii. 打开要上传算例文件的路径，右键文件夹，选择**上传文件**。

   建议上传到集群用户的共享home目录下，例如集群用户名为mpitest，则上传到 `/home/mpitest` 目录下。
   iv. 选择步骤一准备好的算例文件IMB.dat，单击**打开**。
5. 创建作业脚本文件。
   i. 在算例文件所在路径，右键文件夹，单击**新建文件**，然后输入文件名（例如IMB.pbs），单击**确定**。

ii. 打开新建的IMB.pbs脚本文件，拷贝以下脚本内容后保存。

脚本内容示例如下：

```
#!/bin/sh
#PBS -j oe
#PBS -l select=2:ncpus=8:mpiprocs=1  #本示例使用2个计算节点，每个节点使用8 vCPU，使用一个MPI任务进行高性能计算。实际测试中根据节点配置设置CPU个数。

export MODULEPATH=/opt/ehpcmodulefiles/
module load intel-mpi/2018
module load intel-mpi-benchmarks/2019

echo "run at the beginning"
/opt/intel/impi/2018.3.222/bin64/mpirun -genv I_MPI_DEBUG 5 -np 2 -ppn 1 -host compute000,compute001 /opt/intel-mpi-benchmarks/2019/IMB-MPI1 pingpong > IMB-pingpong
#本示例使用2个计算节点：compute000，compute001，请修改为您的集群节点名称。
```

6. 提交作业。
   i. 在**作业**页面，单击**提交作业**。
   ii. 配置作业相关参数。

   作业执行命令请填写作业脚本文件所在路径，例如/home/mpitest/IMB.pbs。其他参数保持默认即可。
   iii. 单击右上角的**提交作业**。
   iv. 在弹出的对话框中输入集群用户名和密码，单击**确定**。

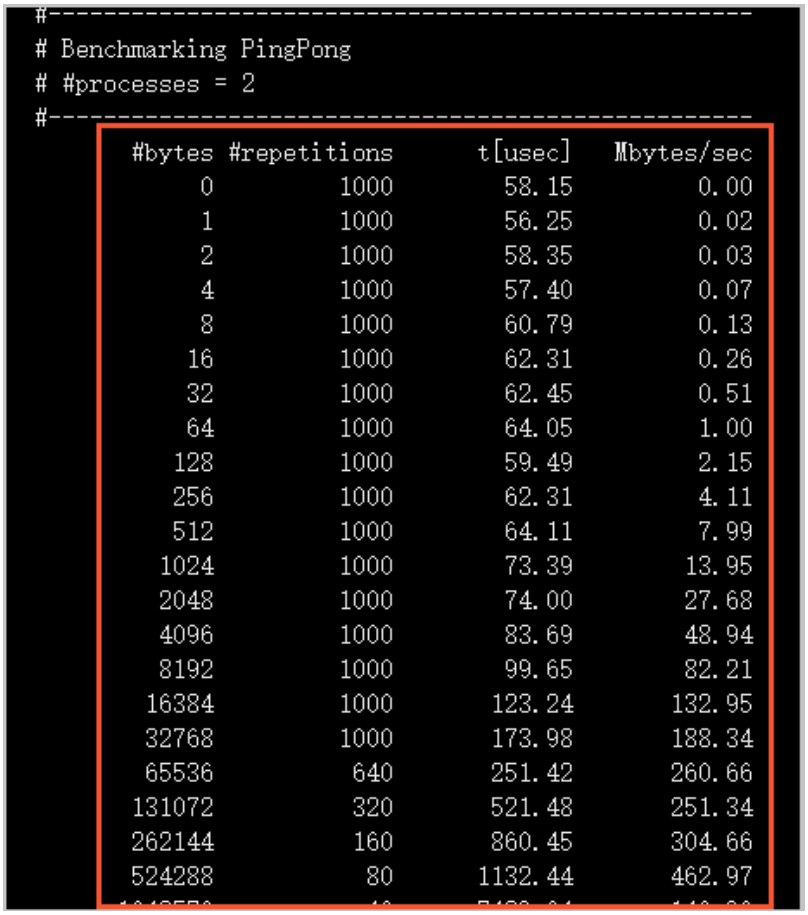
查看结果

作业运行完成后，您可以查看作业结果。

1. 在**集群**页面，找到目标集群，单击**远程连接**。
2. 在**远程连接**页面，输入集群用户名、登录密码和端口，单击**ssh连接**。
3. 执行以下命令，查看作业结果。

```
cat /home/mpitest/IMB-pingpong
```

本次测试结果如下图所示。



```
#------------------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#------------------------------------------------------------
       #bytes #repetitions      t[usec]   Mbytes/sec
            0         1000        58.15         0.00
            1         1000        56.25         0.02
            2         1000        58.35         0.03
            4         1000        57.40         0.07
            8         1000        60.79         0.13
           16         1000        62.31         0.26
           32         1000        62.45         0.51
           64         1000        64.05         1.00
          128         1000        59.49         2.15
          256         1000        62.31         4.11
          512         1000        64.11         7.99
         1024         1000        73.39        13.95
         2048         1000        74.00        27.68
         4096         1000        83.69        48.94
         8192         1000        99.65        82.21
        16384         1000       123.24       132.95
        32768         1000       173.98       188.34
        65536          640       251.42       260.66
       131072          320       521.48       251.34
       262144          160       860.45       304.66
       524288           80      1132.44       462.97
```

15.4. 测试SCC集群性能

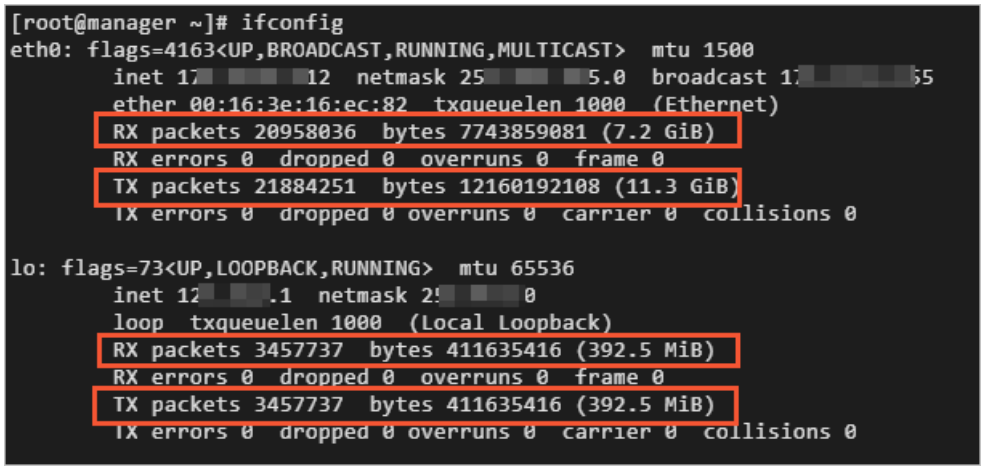
超级计算集群SCC具有无虚拟化损耗、高带宽低延迟网络的优点，可以保证高性能计算和人工智能、机器学习等应用的高度并行需求。本文为您介绍如何创建SCC集群，并测试SCC集群的相关性能。

背景信息

超级计算集群SCC（Super Computing Cluster）在弹性裸金属服务器基础上，加入高速RDMA（Remote Direct Memory Access）互联支持，大幅提升网络性能，提高大规模集群加速比。因此SCC在提供高带宽、低延迟优质网络的同时，还具备弹性裸金属服务器的所有优点。更多信息，请参见超级计算集群概述。

针对E-HPC多机并行计算需求，SCC可以提供低延时RDMA网络互联，同时提供VPC网络隔离能力；SCC实例无虚拟化损耗，您可以直接访问硬件资源。因此，SCC适合仿真制造、生命科学、机器学习、大规模分子动力学和气象预报等应用场景。

SCC实例与普通ECS实例相比，配备了高带宽低延迟的RDMA网络，所以网络通信能力与普通ECS实例相比有明显差异。正常的SCC实例会显示如下网口信息，其中eth0为RDMA网口，lo为VPC网口。



E-HPC支持ecs.scchfg6.20xlarge、ecs.scch5s.16xlarge等SCC规格，更多信息，请参见产品规格。

创建SCC集群

1. 登录弹性高性能计算控制台。
2. 创建一个E-HPC集群。具体操作，请参见使用向导创建集群。

   配置集群时，计算节点请选择ecs.scch5s.16xlarge实例规格。
3. 创建一个集群用户。具体操作，请参见创建用户。

   集群用户用于登录集群，进行编译软件、提交作业等操作，配置用户权限时，权限组请选择**sudo权限组**。
4. 安装软件。具体操作，请参见安装软件。

   需安装的软件如下：
   - linpack，版本为2018。
   - intel-mpi，版本为2018。

测试SCC集群的网络性能

1. 在左侧导航栏，单击**集群**。
2. 在**集群**页面，找到目标集群，单击**远程连接**。
3. 在**远程连接**页面，输入集群用户名和密码，单击**ssh连接**。
4. 通过以下测试样例，测试RDMA网络的峰值带宽。

   按照以下步骤，测试读带宽的峰值：

   i. 登录compute000节点，运行以下命令。

   ```
   ib_read_bw -a -q 20 --report_gbits    ##服务端compute000执行
   ```

ii. 登录compute001节点，运行以下命令。

```
ib_read_bw -a -q 20 --report_gbits compute000  ##用户端compute001执行
```

```
---------------------------------------------------------------------------------------
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 2          20000          0.059465           0.059168             3.697995
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 4          20000           0.21               0.20                6.262095
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 8          20000           0.41               0.40                6.199123
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 16         20000           0.82               0.79                6.194389
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 32         20000           1.66               1.66                6.486258
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 64         20000           3.27               3.17                6.188440
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 128        20000           6.67               6.65                6.495019
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 256        20000           13.31              13.30               6.494630
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 512        20000           16.97              11.61               2.833492
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 1024       20000           19.52              16.84               2.055595
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 2048       20000           20.66              17.97               1.097047
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 4096       20000           20.99              19.00               0.579779
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 8192       20000           20.97              20.73               0.316380
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 16384      20000           21.70              21.70               0.165520
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 32768      20000           22.26              22.26               0.084900
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 65536      20000           22.48              22.48               0.042886
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 131072     20000           22.65              22.65               0.021603
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 262144     20000           22.73              22.73               0.010839
```

按照以下步骤，测试写带宽的峰值：

i. 登录compute000节点，运行以下命令。

```
ib_write_bw -a -q 20 --report_gbits  ##服务端compute000执行
```

ii. 登录compute001节点，运行以下命令。

```
ib_write_bw -a -q 20 --report_gbits compute000  ##用户端compute001执行
```

```
 #bytes     #iterations    BW peak[Gb/sec]    BW average[Gb/sec]   MsgRate[Mpps]
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 2          100000          0.070035           0.068161             4.260051
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 4          100000          0.22               0.22                6.846936
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 8          100000          0.45               0.44                6.900898
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 16         100000          0.89               0.87                6.813752
Conflicting CPU frequency values detected: 1200.000000 != 1500.000000. CPU Frequency is not max.
 32         100000          1.80               1.76                6.860073
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 64         100000          3.59               3.52                6.870596
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 128        100000          7.17               7.04                6.870991
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 256        100000          14.20              13.90               6.788701
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 512        100000          19.99              18.17               4.435906
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 1024       100000          22.65              21.30               2.600382
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 2048       100000          25.22              22.83               1.393473
Conflicting CPU frequency values detected: 1200.000000 != 1300.000000. CPU Frequency is not max.
 4096       100000          30.45              26.29               0.802179
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 8192       100000          36.70              30.62               0.467298
Conflicting CPU frequency values detected: 1200.000000 != 1600.000000. CPU Frequency is not max.
 16384      100000          37.24              33.80               0.257846
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 32768      100000          37.54              35.81               0.136607
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 65536      100000          37.73              36.92               0.070417
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 131072     100000          37.74              37.44               0.035702
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 262144     100000          37.70              37.66               0.017955
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
 524288     100000          37.82              37.82               0.009016
Conflicting CPU frequency values detected: 1200.000000 != 3101.000000. CPU Frequency is not max.
```

5. 通过以下测试样例，测试RDMA网络的延迟。

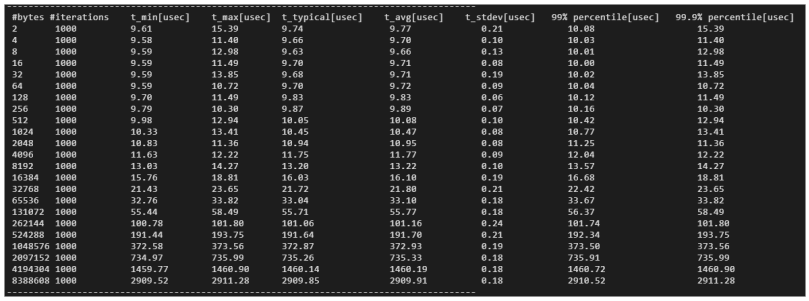
   按照以下步骤，测试RDMA网络的读延迟：

   i. 登录compute000节点，运行以下命令。

   ```
   ib_read_lat -a    ##服务端compute000执行
   ```

   ii. 登录compute001节点，运行以下命令。

   ```
   ib_read_lat -F -a compute000  ##用户端compute001执行
   ```



#bytes	#iterations	t_min[usec]	t_max[usec]	t_typical[usec]	t_avg[usec]	t_stdev[usec]	99% percentile[usec]	99.9% percentile[usec]
2	1000	9.61	15.39	9.74	9.77	0.21	10.08	15.39
4	1000	9.58	11.40	9.66	9.70	0.10	10.03	11.40
8	1000	9.59	12.98	9.63	9.66	0.13	10.01	12.98
16	1000	9.59	11.49	9.70	9.71	0.08	10.00	11.49
32	1000	9.59	13.85	9.68	9.71	0.19	10.02	13.85
64	1000	9.59	10.72	9.70	9.72	0.09	10.04	10.72
128	1000	9.70	11.49	9.83	9.83	0.06	10.12	11.49
256	1000	9.79	10.30	9.87	9.89	0.07	10.16	10.30
512	1000	9.98	12.94	10.05	10.08	0.10	10.42	12.94
1024	1000	10.33	13.41	10.45	10.47	0.08	10.77	13.41
2048	1000	10.83	11.36	10.94	10.95	0.08	11.25	11.36
4096	1000	11.63	12.22	11.75	11.77	0.09	12.04	12.22
8192	1000	13.03	14.27	13.20	13.22	0.10	13.57	14.27
16384	1000	15.76	18.81	16.03	16.10	0.19	16.68	18.81
32768	1000	21.43	23.65	21.72	21.80	0.21	22.42	23.65
65536	1000	32.76	33.82	33.04	33.10	0.18	33.67	33.82
131072	1000	55.44	58.49	55.71	55.77	0.18	56.37	58.49
262144	1000	100.78	101.80	101.06	101.16	0.24	101.74	101.80
524288	1000	191.44	193.75	191.64	191.70	0.21	192.34	193.75
1048576	1000	372.58	373.56	372.87	372.93	0.19	373.50	373.56
2097152	1000	734.97	735.99	735.26	735.33	0.18	735.91	735.99
4194304	1000	1459.77	1460.98	1460.14	1460.19	0.18	1460.72	1460.90
8388608	1000	2909.52	2911.28	2909.85	2909.91	0.18	2910.52	2911.28

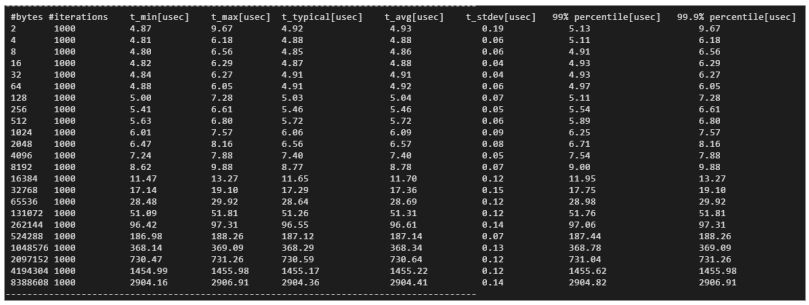
   按照以下步骤，测试RDMA网络的写延迟：

   i. 登录compute000节点，运行以下命令。

   ```
   ib_write_lat -a  ##服务端compute000执行
   ```

   ii. 登录compute001节点，运行以下命令。

   ```
   ib_write_lat -F -a compute000  ##用户端compute001执行
   ```



#bytes	#iterations	t_min[usec]	t_max[usec]	t_typical[usec]	t_avg[usec]	t_stdev[usec]	99% percentile[usec]	99.9% percentile[usec]
2	1000	4.87	9.67	4.92	4.93	0.19	5.13	9.67
4	1000	4.81	6.18	4.88	4.88	0.06	5.11	6.18
8	1000	4.80	6.56	4.85	4.86	0.06	4.91	6.56
16	1000	4.82	6.29	4.87	4.88	0.04	4.93	6.29
32	1000	4.84	6.27	4.91	4.91	0.04	4.93	6.27
64	1000	4.88	6.05	4.91	4.92	0.06	4.97	6.05
128	1000	5.00	7.28	5.03	5.04	0.07	5.11	7.28
256	1000	5.41	6.61	5.46	5.46	0.05	5.54	6.61
512	1000	5.63	6.80	5.72	5.72	0.06	5.89	6.80
1024	1000	6.01	7.57	6.06	6.09	0.09	6.25	7.57
2048	1000	6.47	8.16	6.56	6.57	0.08	6.71	8.16
4096	1000	7.24	7.88	7.40	7.40	0.05	7.54	7.88
8192	1000	8.62	9.88	8.77	8.78	0.07	9.00	9.88
16384	1000	11.47	13.27	11.65	11.70	0.12	11.95	13.27
32768	1000	17.14	19.10	17.29	17.36	0.15	17.75	19.10
65536	1000	28.48	29.92	28.64	28.69	0.12	28.98	29.92
131072	1000	51.09	51.81	51.26	51.31	0.12	51.76	51.81
262144	1000	96.42	97.31	96.55	96.61	0.14	97.06	97.31
524288	1000	186.98	188.26	187.12	187.14	0.07	187.44	188.26
1048576	1000	368.14	369.09	368.29	368.34	0.13	368.78	369.09
2097152	1000	730.47	731.26	730.59	730.64	0.12	731.04	731.26
4194304	1000	1454.99	1455.98	1455.17	1455.22	0.12	1455.62	1455.98
8388608	1000	2904.16	2906.91	2904.36	2904.41	0.14	2904.82	2906.91

监测RDMA网络的实际带宽利用情况

1. 在左侧导航栏，单击**集群**。
2. 在**集群**页面，找到目标集群，单击**远程连接**。
3. 在**远程连接**页面，输入root用户名和密码，单击**ssh连接**。
4. 运行以下命令监测RDMA网络的实际带宽利用情况。

   ```
   rdma_monitor -s
   ```

```
>_ 1. root@compute000:~ ×
2021-03-12 17:51:33 CST
tx_rate: 0.000bps (0.000bps/0.000bps)
rx_rate: 0.000bps (0.000bps/0.000bps)
tx_pause: 0 (0/0)
rx_pause: 0 (0/0)
tx_pause_duration: 0 (0/0)
rx_pause_duration: 0 (0/0)
np_cnp_sent: 0
rp_cnp_handled: 0
num_of_qp: 3
np_ecn_marked: 0
rp_cnp_ignored: 0
out_of_buffer: 0
out_of_seq: 0
packet_seq_err: 0
cpu_usage: 0.46%
free_mem: 188936936 kB
```

查看SCC集群节点的性能

1. 在左侧导航栏，选择**作业与性能管理 > E-HPC优化器**。
2. 在**性能大盘**页面，选择需要查看的集群，在操作列单击**节点**。
3. 在**节点性能**页签，选择对应的节点和指标即可查看集群节点的相关性能。