弹性高性能计算E-HPC

最佳实践

最佳实践

概述

概述

本部分文档将通过实际操作案例，介绍如何通过E-HPC控制台完成实际的业务操作，案例涉及基本的性能评估（BenchMark）以及不同行业领域的具体时间，包括：

案例类型	主要内容	文档链接
计算性能评估	介绍如何基于HPL进行系统浮点性能评估	链接
内存性能评估	介绍如何基于STREAM工具进行内存带宽性能测试评估	链接
通信性能评估	介绍如何对不同消息粒度下节点间MPI通信进行性能评估	链接
LAMMPS	基于LAMMPS的分子动力学典型算例3d Lennard-Jones melt，包含作业创建、提交、可视化全流程	链接
GROMACS	基于GROMACS的分子动力学算例，通过GPU加速计算，包含作业创建、提交、性能监控、可视化全流程	链接
OpenFOAM	基于OpenFOAM的计算流体力学仿真算例	链接
WRF	基于WRF的气象预报典型算例	链接
TensorFlow	基于TensorFlow的机器学习环境部署、计算流程。使用HPC调度器提交作业	链接

HPC BenchMark

HPL

简介

HPL（the High-Performance Linpack Benchmark）是国际上最流行的用于测试高性能计算机系统浮点性能的benchmark。通过对高性能计算机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，评价高性能计算机的浮点性能。浮点计算峰值是指计算机每秒钟能完成的浮点计算最大次数。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能达到的每秒钟能完成浮点计算最大次数，它主要是由CPU的主频决定的。

```
理论浮点峰值 = CPU主频 × CPU每个时钟周期执行浮点运算的次数 × 系统中CPU数
```

准备工作

> 若您尚未拥有E-HPC集群，请先创建E-HPC集群

运行以下示例需要在创建集群时或者软件管理界面上选择安装linpack软件包和intel-mpi通信库。

☑	linpack	2018
☑	intel-mpi	2018

输入参数说明

输入文件HPL.dat包含了HPL的运行参数，下图是在单台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)
```

测试过程中需要根据节点硬件配置而做出调整的运行参数主要有：

- 第5、6行：代表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；但与此同时，矩阵规模N的增加会导致内存消耗量的增加，一旦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 x N x 8 = 系统总内存 x 80%（其中总内存换算以字节为单位）。

- 第7、8行：代表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，但还是有一些规律可循：NB不能太大或太小，一般在384以下；NB × 8一定是Cache line的倍数等。例如，L2 cache为1024K, NB就设置为192。另外，NB大小的选择还跟通信方式、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。所以最好在小规模测试时选择3个左右性能不错的NB，再通过大规模测试检验这些选择。
- 第10～12行：代表二维处理器网格（P × Q）。P × Q = 系统CPU数 = 进程数。一般来说一个进程对于一个CPU可以得到最佳性能。对于Intel Xeon来说，关闭超线程可以提高HPL性能。P≤Q；一般来说，P的值尽量取得小一点，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 = 2n，即P最好选择2的幂。HPL中，L分解的列向通信采用二元交换法（Binary Exchange），当列向处理器个数P为2的幂时，性能最优。例如，当系统进程数为4的时候，P × Q选择为1 × 4的效果要比选择2 × 2好一些。 在集群测试中，P × Q = 系统CPU总核数。

运行HPL测试

- E-HPC控制台创建HPL.dat输入文件

返回E-HPC管理控制台，点选左侧栏的**“作业”**标签，进入作业管理界面。依次选择**“创建作业”** -> **“新建文件”** -> **“使用文件模板”** -> **“HPL.dat”** ，根据节点硬件配置调整HPL输入参数，得到HPL输入文件如下。

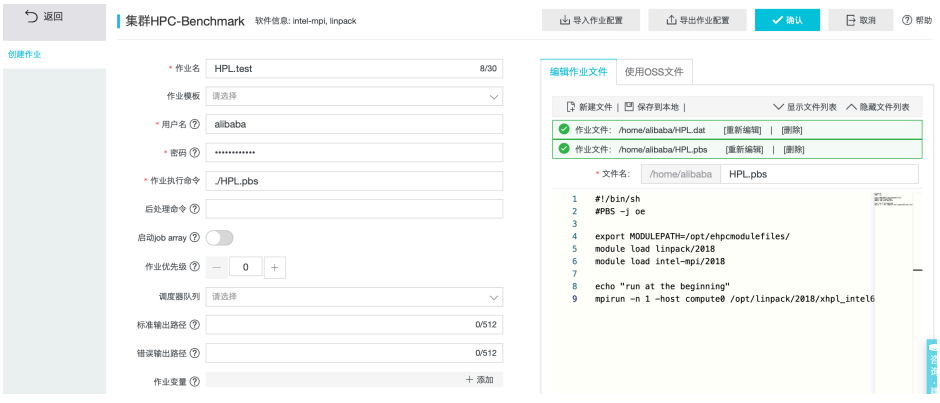
- E-HPC控制台创建HPL.pbs作业脚本

在作业管理界面中，依次选择**“创建作业”** -> **“新建文件”** -> **“使用文件模板”** -> **“pbs demo”** ，对pbs demo脚本进行修改，得到HPL作业脚本HPL.pbs如下。

```
#!/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 <node> /opt/linpack/2018/xhpl_intel64_static > hpl-ouput #测试单节点的浮点性能
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
```

- E-HPC控制台提交HPL测试作业

确定下图左侧作业基本参数后，点击右上角**“确认”**提交作业。作业个性化配置、作业导入、作业导出以及作业状态查看，请参见作业管理。



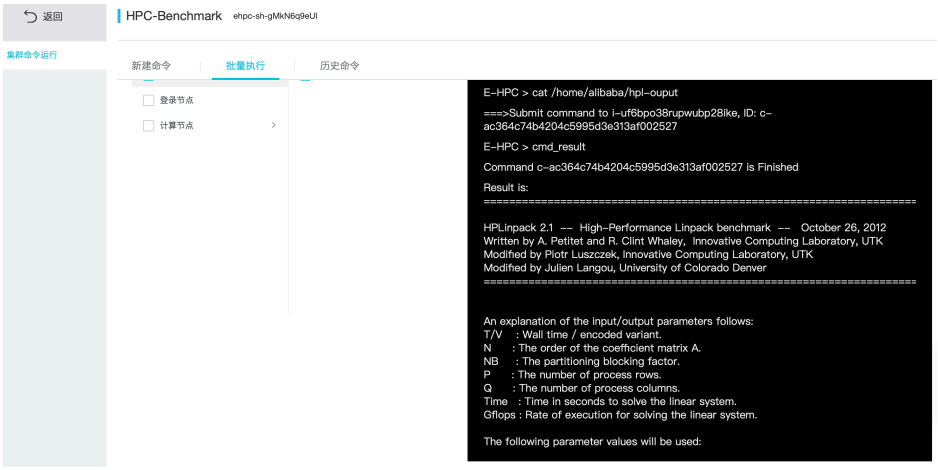
- E-HPC控制台查询作业状态

点击作业列表中HPL作业右侧的 **“详情”** 按钮，查看作业详细信息。

- E-HPC控制台查看结果文件

返回E-HPC管理控制台，点选集群右侧“**更多**”选项，选择“**执行命令**”，进入集群命令运行界面。

在集群命令运行界面点击“**批量执行**”，选择集群登录/管控节点执行命令，查看HPL作业结果文件。



从结果文件中获取测得的HPL浮点运算效率数据，格式如下。

```
T/V                N    NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WC00C2R2 143360 384 1 1 XXXX XXXXXXX
```

Intel MPI Benchmarks

简介

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。

准备工作

若您尚未拥有E-HPC集群，请先创建E-HPC集群

- 运行以下示例需要在创建集群时或者软件管理界面上选择安装intel-mpi-benchmarks软件包和intel-mpi通信库

☑	intel-mpi-benchmarks	2019
☑	intel-mpi	2018

IMB测试方法说明

```
$ /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> #非root用户下执行

$ /opt/intel/impi/2018.3.222/bin64/mpirun -genv I_MPI_DEBUG 5 -np 2 -ppn 1 -host <node0>,<node1> /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

$ /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


$ /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

############关键参数说明##############
-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 指定至少运行的进程数
-msglog 指定消息片粒度范围
```

运行IMB测试

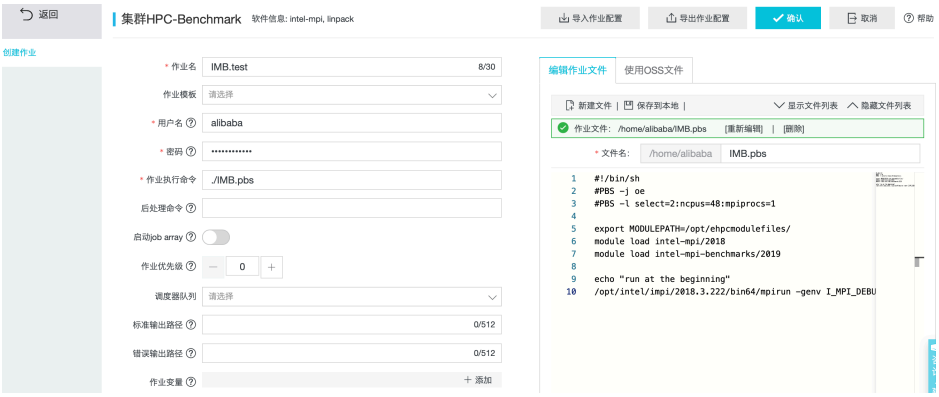
- E-HPC控制台创建IMB.pbs作业脚本

在作业管理界面中，依次选择“创建作业”->“新建文件”->“使用文件模板”->“pbs demo”，对pbs demo脚本进行修改，得到IMB作业脚本IMB.pbs如下。

```
#!/bin/sh
#PBS -j oe
#PBS -l select=2:ncpus=<N>:mpiprocs=1 #N为节点CPU核数，实际测试中根据节点配置进行设置

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 compute0,compute1 /opt/intel-mpi-benchmarks/2019/IMB-MPI1 pingpong > IMB-pingpong
```

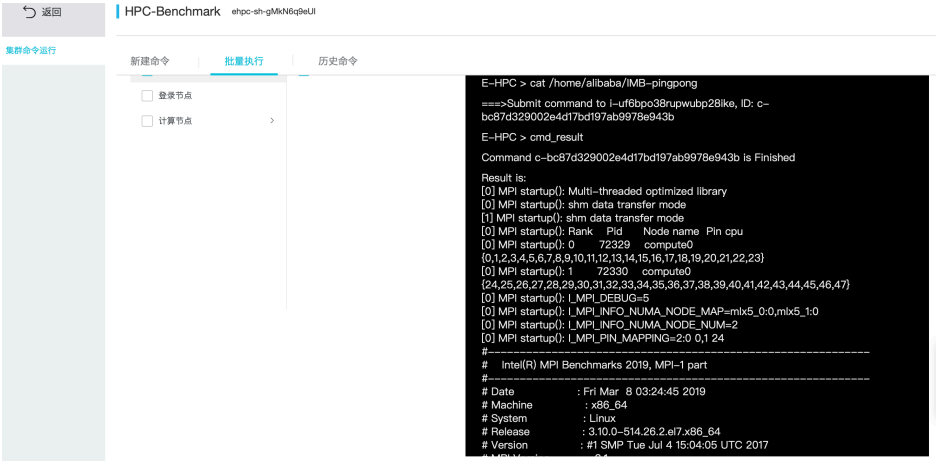
- E-HPC控制台提交IMB测试作业

确定下图左侧作业基本参数后，点击右上角“确认”提交作业。作业个性化配置、作业导入、作业导出以及作业状态查看，请参见作业管理。



- E-HPC控制台查看结果文件

从E-HPC管理控制台，点选集群右侧“更多”选项，选择“执行命令”，进入集群命令运行界面。点击“批量执行”，选择集群登录/管控节点执行命令，查看IMB作业结果文件。



STREAM

简介

STREAM测试是内存测试中业界公认的内存带宽性能测试基准工具，是衡量服务器内存性能指标的通用工具。STREAM具有良好的空间局部性，是对 TLB 友好、Cache友好的一款测试，支持Copy、Scale、Add、Triad四种操作。

准备工作

若您尚未拥有E-HPC集群，请先创建E-HPC集群

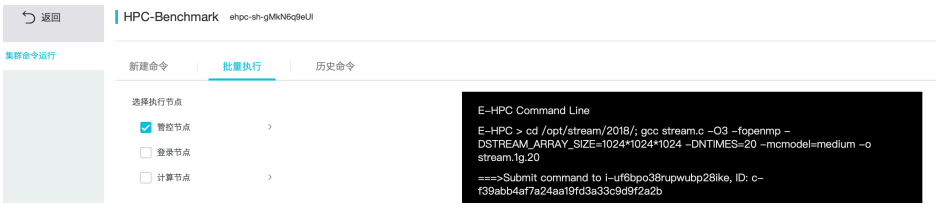
- 运行以下示例需要在创建集群时或者软件管理界面上选择安装STREAM软件包

☑	stream	2018

运行STREAM测试

- E-HPC控制台编译STREAM

为了避免数据Cache重用对测试结果准确度产生较大影响，需确保STREAM开辟的数组大小远大于L3 Cache的容量且小于内存的容量。因此在实际测试中要根据测试节点配置对STREAM进行重新编译。由E-HPC管理控制台进入集群命令运行界面，登录节点执行如下操作。




```
$ cd /opt/stream/2018/; gcc stream.c -O3 -fopenmp -DSTREAM_ARRAY_SIZE=1024*1024*1024 -DNTIMES=20 -mcmodel=medium -o stream.1g.20  #-DSTREAM_ARRAY_SIZE用于指定STREAM一次搬运的数据量，-DTIMES用于指定迭代次数
```

- E-HPC控制台创建STREAM.pbs作业脚本

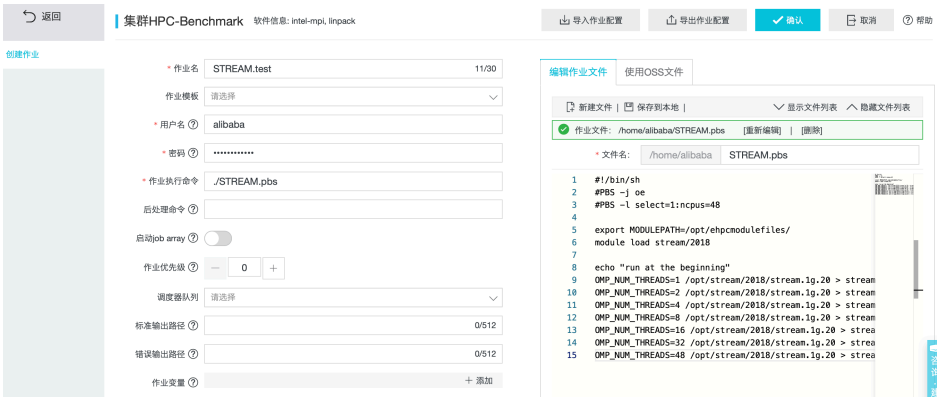
在作业管理界面中，依次选择**“创建作业”** -> **“新建文件”** -> **“使用文件模板”** -> **“pbs demo”** ，对pbs

demo脚本进行修改，得到STREAM作业脚本STREAM.pbs如下。

```
#!/bin/sh
#PBS -j oe
#PBS -l select=1:ncpus=<N> #N为节点CPU核数，实际测试中根据节点配置进行设置

export MODULEPATH=/opt/ehpcmodulefiles/
module load stream/2018

echo "run at the beginning"
OMP_NUM_THREADS=1 /opt/stream/stream.1g.20 > stream-1-thread.log
OMP_NUM_THREADS=2 /opt/stream/stream.1g.20 > stream-2-thread.log
OMP_NUM_THREADS=4 /opt/stream/stream.1g.20 > stream-4-thread.log
OMP_NUM_THREADS=8 /opt/stream/stream.1g.20 > stream-8-thread.log
...
OMP_NUM_THREADS=<N> /opt/stream/stream.1g.20 > stream-<N>-thread.log
```

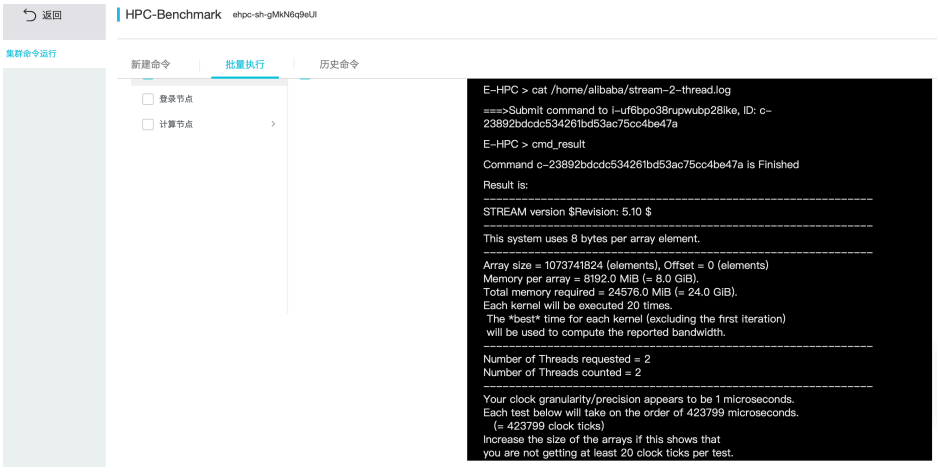
- E-HPC控制台提交STREAM测试作业

确定下图左侧作业基本参数后，点击右上角**“确认”**提交作业。作业个性化配置、作业导入、作业导出以及作业状态查看，请参见作业管理。



- E-HPC控制台查看结果文件

从E-HPC管理控制台，点选集群右侧**“更多”**选项，选择**“执行命令”**，进入集群命令运行界面。点击**“批量执行”**，选择集群登录/管控节点执行命令，查看STREAM作业结果文件。



LAMMPS

官网

http://lammps.sandia.gov/ 。

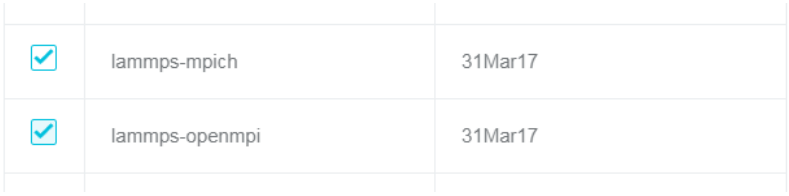
简介

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。

算例 1 “3d Lennard-Jones melt”

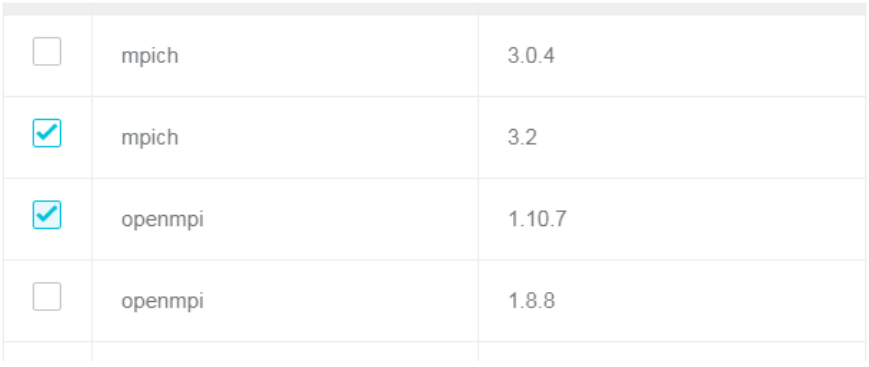
准备工作

运行以下示例需要在创建集群时选择安装LAMMPS相关软件包。



☑	lammps-mpich	31Mar17
☑	lammps-openmpi	31Mar17

同时还需选择所依赖的相关MPI库



☐	mpich	3.0.4
☑	mpich	3.2
☑	openmpi	1.10.7
☐	openmpi	1.8.8

以及可视化结果查询服务中用到的VMD

操作步骤

1.. 进入EHPC控制台作业界面，点击右上角创建作业。

2.. 在创建作业页面左侧添加用户信息，和作业执行命令：

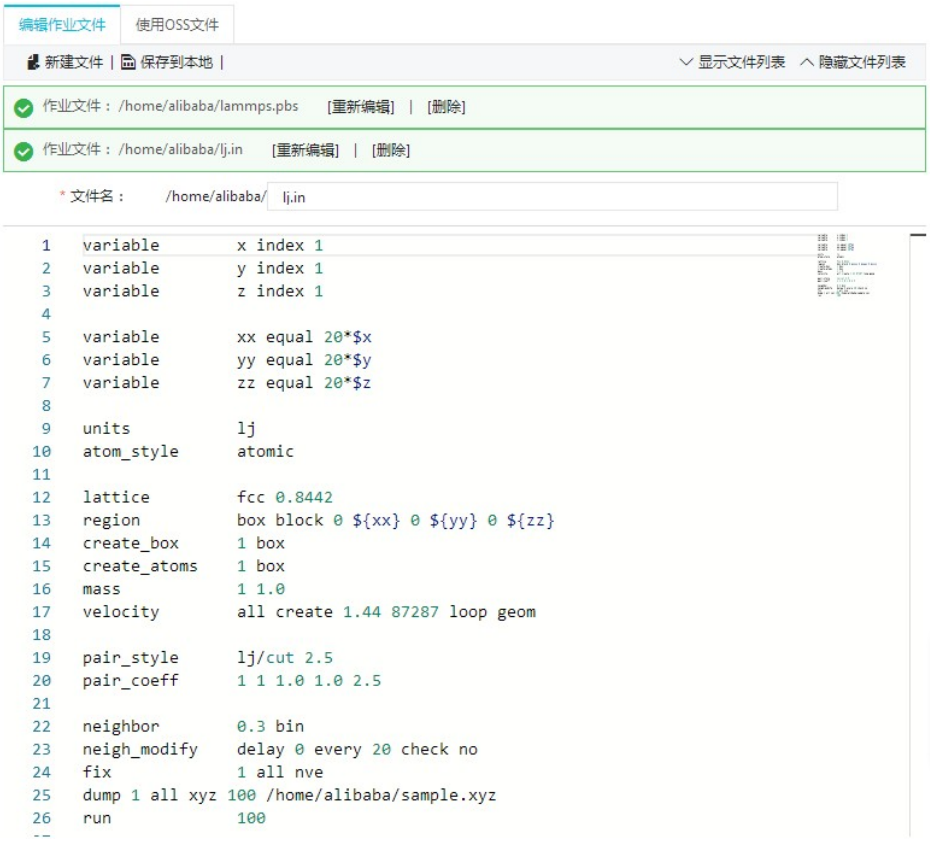
```
./lammps.pbs
```

3.. 点击页面右侧编辑作业文件按钮，新建作业文件lammps.pbs脚本和3d Lennard-Jones melt算例文件lj.in。

- lammps.pbs脚本:

```
#!/bin/sh
#PBS -l select=2:ncpus=1:mpiprocs=1
#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
```

- lj.in算例文件:



4.. 提交作业，等待作业运行完成。

5.. ‘VNC+VMD’ 方式查看作业运行结果内容：

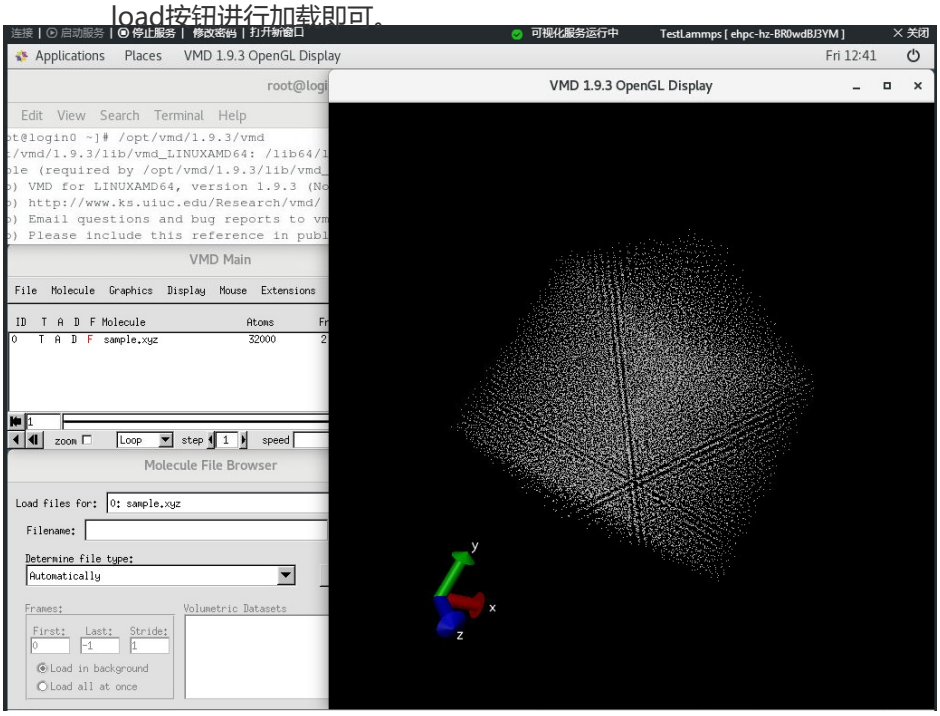
- 如果没有安装VMD，进入集群界面，安装VMD。

- 打开远程可视化VNC服务（创建集群时必须打开可视化服务按钮）。
- 在可视化服务窗口，右键打开open Terminal，在root用户下输入：

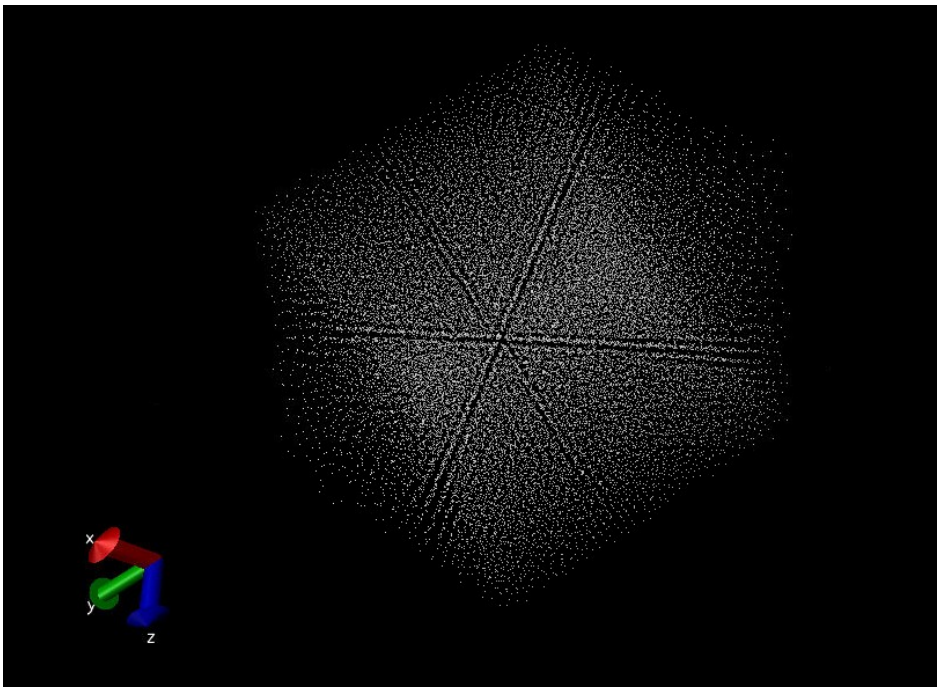
```
/opt/vmd/1.9.3/vmd
```

- 在VMD Main页面，点击file按钮新建一个Molecule，其中filename文件为集群用户目录（如：/home/alibaba/sample.xyz）下的sample.xyz文件，file type类型为automatically,最后直接点击

load按钮进行加载即可。



6.. 3d Lennard-Jones melt算例结果可视化展示：



GROMACS

GROMACS 简介

GROMACS（GROningen MAchine for Chemical Simulations）是一款通用软件包，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。由于GROMACS在计算典型的主流模拟应用如非键合相互作用非常高效，许多研究人员将其用于非生物系统如聚合物的研究。

GROMACS支持从现代分子动力学实现中预期的所有常见算法，可以采用GPU卡来加速核心计算过程。其代码由世界各地的开发人员维护。详情可参见官网www.gromacs.org 。

准备工作

若您尚未拥有E-HPC集群，请先创建E-HPC集群

安装软件包

运行以下示例需要在创建集群时或者软件管理界面上选择安装GROMACS相关软件包。

- 使用GROMACS的GPU加速版本需要安装如下软件包

☑	gromacs-gpu	2018.1
☑	openmpi	3.0.0
☑	cuda-toolkit	9.0

注：若需运行gromacs-gpu加速版本，在创建集群时**必须使用GPU系列机型**作为计算节点，否则集群无法按照以下指引运行。

创建用户

进入E-HPC管理控制台，点选左侧栏的**“用户”**标签，进行用户创建。本案例中，我们创建一个名为gmx.test的sudo用户。

输入算例介绍

算例1：水中的溶菌酶（Lysozyme in Water）

本算例为用户设置一个蛋白质(lysozyme)加上离子在水盒子里的模拟过程。

官方教程链接：http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials/lysozyme/index.html

非官方中文翻译链接：http://jerkwin.github.io/GMX/GMXtut-1/

下载地址：http://public-ehs.oss-cn-hangzhou.aliyuncs.com/packages/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。

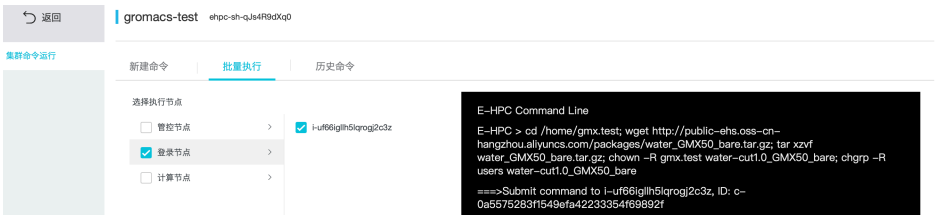
下载地址：http://public-ehs.oss-cn-hangzhou.aliyuncs.com/packages/water_GMX50_bare.tar.gz

运行GROMACS的GPU加速版本

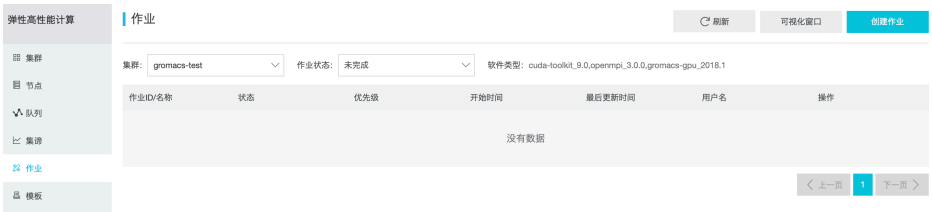
算例下载与解压

- 进入E-HPC管理控制台，点选集群右侧**“更多”**选项，选择**“执行命令”**，进入集群命令运行界面。

- 在集群命令运行界面点击**“批量执行”**，选择集群登录节点执行下图所示的算例下载、解压、修改权限操作。



- 返回E-HPC管理控制台，点选左侧栏的**“作业”**标签，进入作业管理界面。



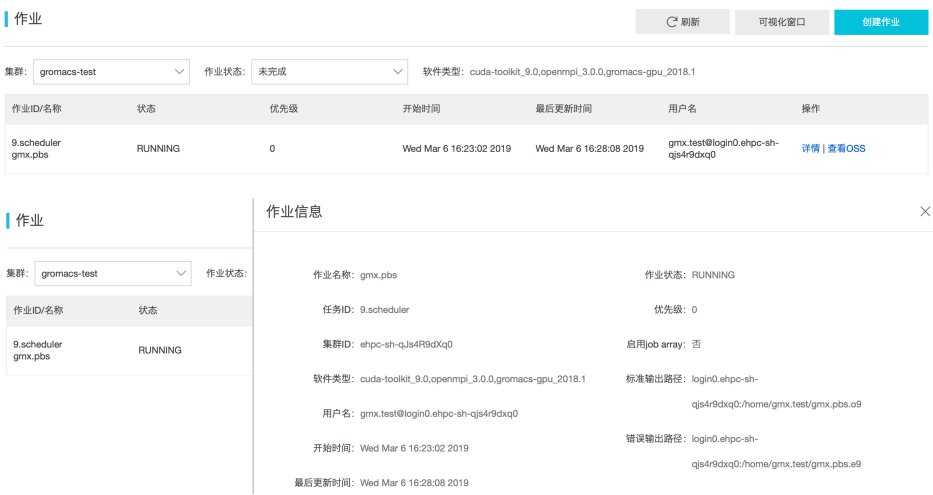
- 依次选择”创建作业”->“新建文件”->“使用文件模板”->“pbs demo”，对pbs demo脚本进行编辑，得到运行GROMACS-GPU版本作业的pbs脚本如下所示。

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

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

cd /home/gmx.test/water-cut1.0_GMX50_bare/1536
/opt/gromacs-gpu/2018.1/bin/gmx_mpi grompp -f pme.mdp -c conf.gro -p topol.top -o topol_pme.tpr #前处理过程，生成tpr格式输入文件
mpirun -np 1 -host compute9 /opt/gromacs-gpu/2018.1/bin/gmx_mpi mdrun -ntomp 8 -nsteps 400000 -pin on -nb gpu -s topol_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
```

注： 本例中，作业在名为gmx.test的用户下提交，在一个包含8个CPU核和1块P100 GPU卡的计算节点compute9上运行。在实际使用场景中用户可根据集群配置情况做出适当修改。

- 设置下图左侧作业基本参数后，点击**确认**提交作业。作业个性化配置、作业导入、作业导出以及作业状态查看，请参见**作业管理**。

- 点击作业列表右侧的 **“详情”** 按钮，查看作业详细信息。



GROMACS作业性能监测

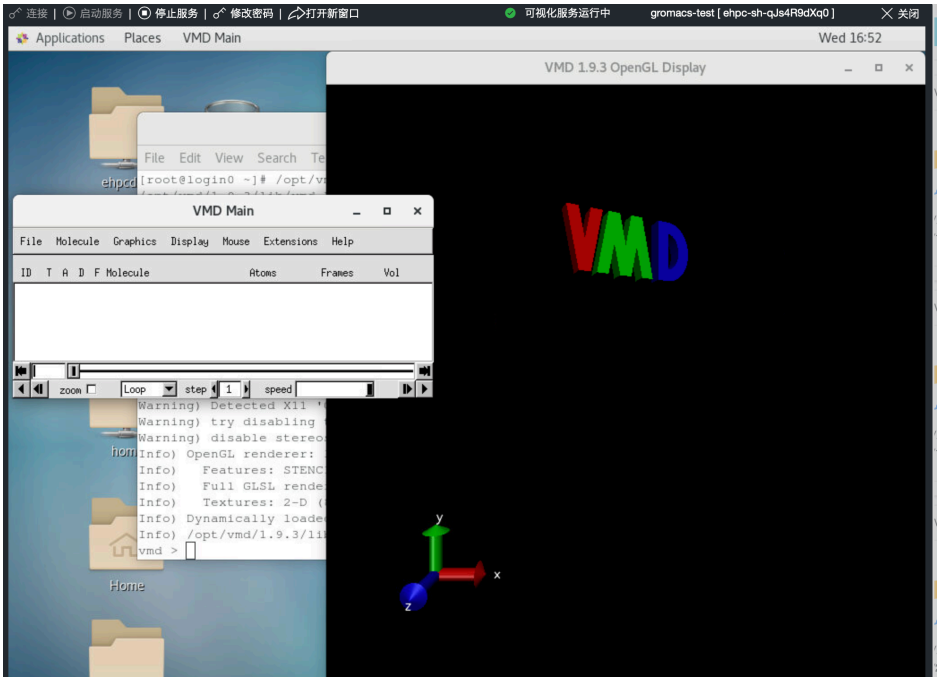
- 返回E-HPC管理控制台，点击左侧栏的**“集谛”**标签，进入集谛性能监测界面。在**“节点性能”**面板上查看各项硬件性能指标，实时监测节点硬件资源的利用情况以及随时间的变化趋势。如下图所示，GROMACS作业的GPU利用率维持在60%以上。

- 点击**“进程性能”**面板，查看当前CPU利用率前五的进程信息。由于本案例中的GROMACS作业仅使用一个进程，每个进程开启八个线程，因此图中“gmx_mpi”进程始终占据第一位，且CPU占用率远超其它四个进程之和。

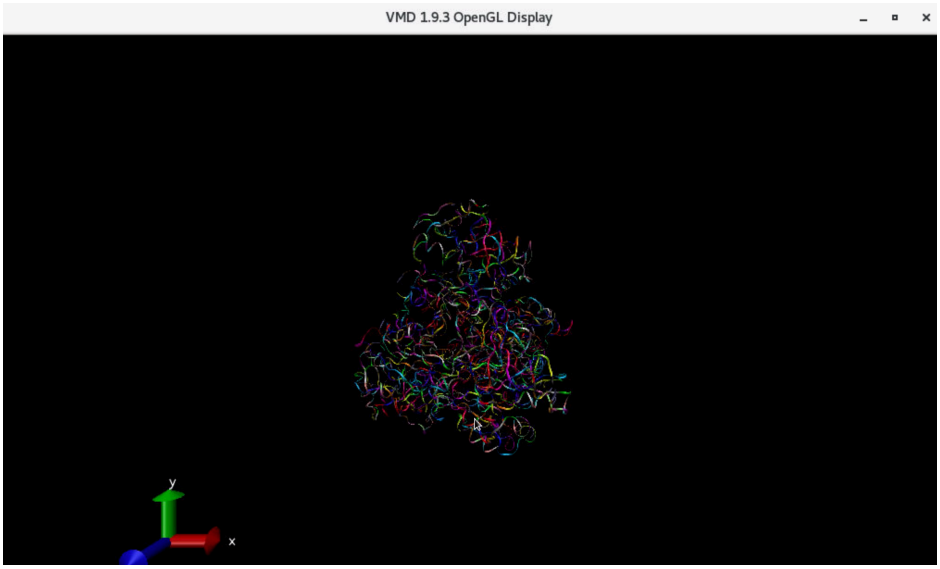
- 点击上图中**“剖析进程5833”**，设置剖析时长和采样频率，启动对GROMACS作业的实时性能剖析，获取热点函数火焰图如下。从图中可以查看GROMACS作业中各函数的耗时占比和调用栈关系。

GROMACS计算结果可视化

- 在软件管理界面安装MD可视化工具VMD，使用远程可视化功能打开远程可视化桌面。Terminal运行/opt/vmd/1.9.3/vmd，打开VMD软件。



- 加载分子结构文件和轨迹文件，查看模拟效果。



OpenFOAM

OpenFOAM

官网

http://www.openfoam.com/

简介

OpenFOAM（英文 Open Source Field Operation and Manipulation 的缩写，意为开源的场运算和处理软件）是对连续介质力学问题进行数值计算的C++自由软件工具包，其代码遵守GNU通用公共许可证。它可进行数据预处理、后处理和自定义求解器，常用于计算流体力学(CFD)领域。

算例1 “Motorbike”

准备工作

运行以下示例需要在创建集群时h选择安装OpenFOAM相关软件包

☑	openfoam-openmpi	5.0

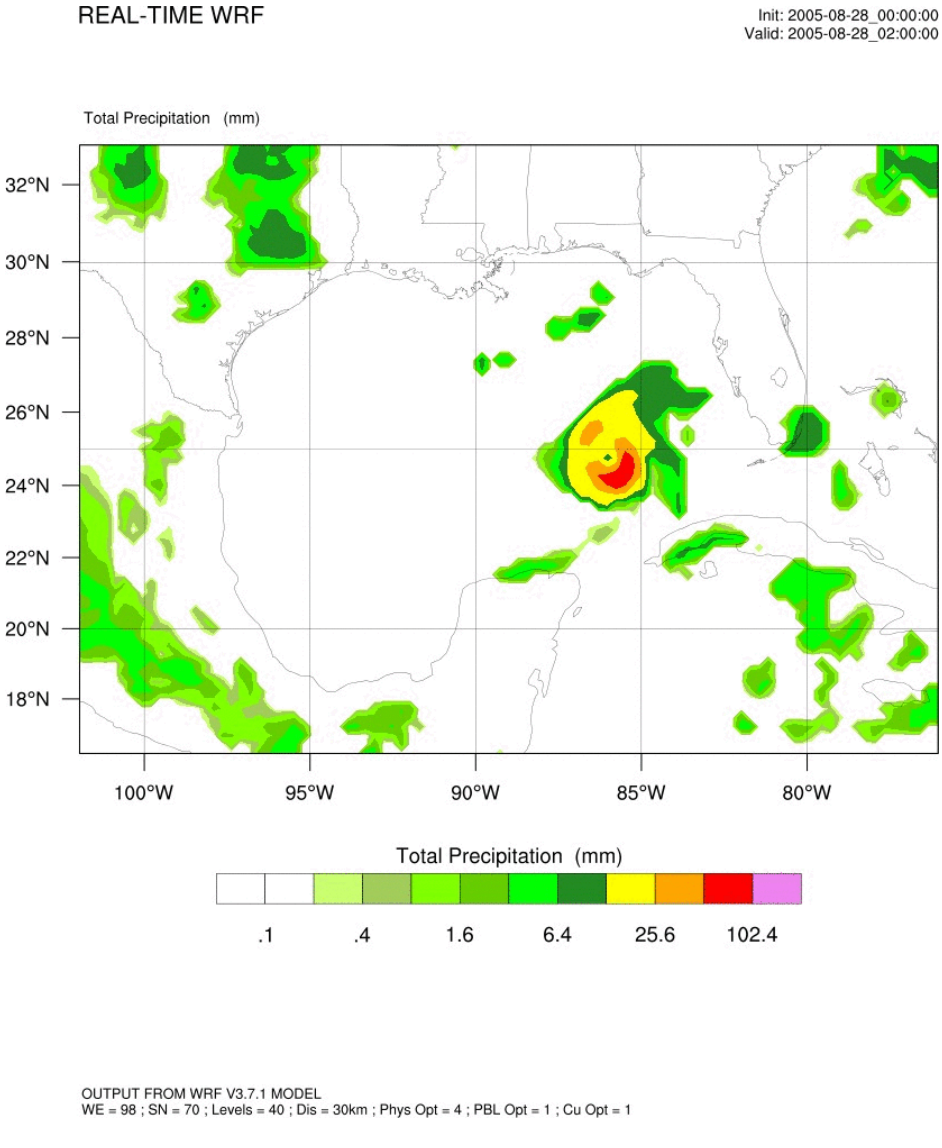
操作步骤

以下为OpenFOAM-5.0的示例

```
cd /opt/OpenFOAM/OpenFOAM-5.0
source etc/bashrc
cd tutorials/incompressible/simpleFoam/motorBike/
./Allrun
```

WRF

WRF



官网

http://www2.mmm.ucar.edu/wrf/users/

简介

WRF（Weather Research and Forecasting）是新一代中尺度预报模式，被广泛应用的开源气象模拟软件。

准备工作

运行以下示例需要在创建集群时选择安装WRF相关软件包。

算例1

下载链接：http://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog_V3.html

运行步骤：

step 1. 运行module avail 查看是否安装WRF软件:

```
$ export MODULEPATH=/opt/ehpcmodulefiles/
$ module avail
------------------------------ /opt/ehpcmodulefiles -------------------------------------
wrf-mpich/3.8.1 wrf-openmpi/3.8.1 mpich/3.2.2 openmpi/1.10.7
```

step 2. module load 加载WRF:

```
$ module load wrf-mpich/3.8.1 mpich
$ echo $WPSHOME $WRFHOME
/opt/WRF_WPS-mpich-3.8.1/WPS /opt/WRF_WPS-mpich-3.8.1/WRFV3
```

Step 3. 将安装的WPS和WRF软件拷贝到工作目录，并将算例拷贝到相应目录位置。

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

Step 4. 进入WPSCOPYHOME软件目录，并执行：

```
$ srun --mpi=pmi2 –n 1 ./geogrid.exe
$ ./link_grib.csh 下载的气象数据
$ ln –sf ungrib/Variable_Tables/Vtable.GFS Vtable # 选择气象数据相应的Vtable，如GFS
$ srun --mpi=pmi2 –n 1 ./ungrib.exe
$ srun --mpi=pmi2 –n 1 ./metgrid.exe
```

Step 5. 进入WRFCOPYHOME软件的run目录，并执行：

```
$ ln –sf $WPSCOPYHOME/met_em* . #连接WPS的处理结果
$ srun --mpi=pmi2 –n 1 ./real.exe
$ srun --mpi=pmi2 –n 4 ./wrf.exe
```

TensorFlow

简介

E-HPC不仅支持工业/科研行业的高性能计算作业，还可以支持机器学习类作业，本文档介绍在E-HPC上部署和执行TensorFlow作业的基本流程。

本案例使用的Perseus是阿里云提供的一种统一支持Tensorflow、Caffe、MXNET、PyToch的分布式训练的深度优化框架，目的是为了机器学习提速，提升训练效率。部署Perseus的同时会自动部署TensoFlow框架。

本案例测试程序为tensorflow benchmarks，E-HPC集群创建完成后，存放在/root/perseus-tf-vm-demo目录中。

创建E-HPC集群并适配Persues环境

目前，Persues已经集成在E-HPC产品中，若在E-HPC适配Persues运行环境，需要在E-HPC创建过程完成以下几步：

1）创建集群时，在【硬件配置】中选择【计算节点】 时，选择带有NVIDIA P100 GPU的实例，如下图所示:

2）在【软件配置】下，【镜像类型】选择 镜像市场，【操作系统】选择 阿里ai云加速镜像Perseus v0.9.3r3

示例程序测试

待集群启动后，可以通过以下几个步骤进行示例的测试：

1）拷贝测试程序：perseus-tf-vm-demo 示例程序存放在镜像的/root目录下。运行时，可以将perseus-tf-vm-demo从/root目录中拷贝到自己普通用户的家目录下（可以用root用户登录执行），并改为普通用户的属主、属组。

```
$ cd /root
$ cp -r ./perseus-tf-vm-demo /home/username/
$ cd /home/username
$ chown -R username:users ./perseus-tf-vm-demo
```

2）编写PBS作业脚本：普通用户模式登录管控节点，在perseus-tf-vm-demo文件下有两个文件：benchmarks和launch-example.sh。可以创建以下test.pbs作业脚本启动测试程序。

```
$ cat test.pbs
#! /bin/bash
#PBS -N Perseus
#PBS -l nodes=x:ppn=y
#PBS -o perseus_pbs.log
#PBS -j oe
cd $PBS_O_WORKDIR
nodefile=`cat $PBS_NODEFILE|uniq -d |awk -F "." '{print $1}'`
sh launch-example.sh x z $nodefile
```

其中，x 为申请计算节点数量，y为每计算节点cpu核数，z 为每节点gpu卡数量

3）提交作业：通过qsub提交PBS作业，此时作业由调度系统调度执行。

```
$ qsub  test.pbs
```

运行结果及分析

1）当计算节点数量为1时，每节点gpu卡为1，运行后的结果可以作为基准。图表示在tensorflow环境但节点下每秒钟处理292.57张图片。

```
----------------------------------------------------------------
total images/sec: 292.57
----------------------------------------------------------------
```

当计算节点数量为1，每节点gpu卡为2，即单机多卡计算时：

```
----------------------------------------------------------------
total images/sec: 568.44
----------------------------------------------------------------
500     images/sec: 284.3 +/- 0.1 (jitter = 0.8)        7.965
----------------------------------------------------------------
total images/sec: 568.45
----------------------------------------------------------------
```

当计算节点数量为2，每节点gpu卡为1，即多机分布式计算时：

```
----------------------------------------------------------------
total images/sec: 544.36
----------------------------------------------------------------
500     images/sec: 272.2 +/- 0.3 (jitter = 4.5)        7.910
----------------------------------------------------------------
total images/sec: 544.36
----------------------------------------------------------------
```

2）结果分析

```
多卡并行效率 = (total imgages/sec) / 基准 / gpu总数量 / 节点数量
= ( ( 568.45+568.44 ) / 2 ) / (292.57) / 2 / 1
= 0.9714
```

当Perseus 框架下的benchmarks程序运行在同一节上不同gpu配置的情况下，以单节点1gpu卡配置运行的结果为基准，通过计算其并行效率，可以分析出相对于基准，单计算节点2gpu卡配置的计算性能损耗。

```
多机并行效率 = ( 多节点 total images / sec ) / 基准 / 节点数量
= ( ( 544.36+544.36 ) / 2 ) / ( 292.57 ) / 2
= 0.9303
```

当Perseus 框架下的benchmarks程序运行在不同节点数量上每节点相同gpu配置的情况下，以单节点1gpu卡配置运行的结果为基准，通过计算其并行效率，可以分析出相对于基准，多计算节点1gpu卡配置的计算性能损耗。

创建和使用SCC集群

SCC（超级计算集群）简介

SCC概述

超级计算集群（Super Computing Cluster，SCC）使用高速RDMA网络互联的CPU以及GPU等异构加速设备，面向高性能计算、人工智能/机器学习、科学/工程计算、数据分析、音视频处理等应用，提供极致计算性能和并行效率的计算集群服务。

SCC实例类型

类型	CPU	Memory	网络	存储	适用场景
ecs.scch5.16xlarge	64核 Skylake Xeon Gold 6149 3.1GHz	192GB	50 Gbps RDMA	高效云盘（容量可选）+ SSD云盘（容量可选）	CPU主频高，单核计算能力强，适用于多数计算密集型应用场景
ecs.sccg5.24xlarge	96核 Skylake Xeon Platinum 8163 2.5GHz	384GB	50 Gbps RDMA	高效云盘（容量可选）+ SSD云盘（容量可选）	CPU核数多，内存容量大，适用于内存需求较高、扩展性好的科学计算场景以及高并发的批

					处理场景

使用SCC实例创建E-HPC集群

创建过程

- 目前配备有SCC实例的可用区主要有：华东1可用区H、华东2可用区B、华北1可用区C、华北3可用区A。考虑到库存的变化，用户在创建集群之前可以通过**ECS管理控制台**查看SCC实例在不同可用区的分布情况。
- 从**E-HPC管理控制台**进入**集群创建**页面，在**计算节点**下划栏中勾选SCC实例。

注意：上图中SCC实例的CPU核数是按照vCPU数目来显示的，而实际交付的SCC实例为超线程关闭（HT off）状态，即scch5.16xlarge和sccg5.24xlarge的CPU核数分别为32物理核和48物理核。

- 后续创建过程请参考**E-HPC集群创建与配置**

硬件信息

相比于普通ECS实例，SCC实例的核心硬件升级之一在于配备了50Gbps的RoCE(RDMA over Converged Ethernet)网络，故网络信息与普通ECS实例相比有明显差异。

网络硬件信息

- 相比于普通ECS实例，SCC实例同时拥有10Gbps VPC网络和50Gbps RoCE网络的网口，因此在会**ECS管理控制台**上会同时显示两个IP地址。

- 正常的SCC实例会显示如下网口信息，其中bond0为RoCE网口，eth0为VPC网口。

```
[root@compute12 ~]# ifconfig
bond0: flags=5187<UP,BROADCAST,RUNNING,MASTER,MULTICAST>  mtu 1500
        inet 200.0.107.2  netmask 255.255.255.252  broadcast 200.0.107.3
        ether 50:6b:4b:47:5a:fc  txqueuelen 1000  (Ethernet)
        RX packets 88  bytes 11464 (11.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 82  bytes 10784 (10.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.9  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 00:16:3e:00:79:4c  txqueuelen 1000  (Ethernet)
        RX packets 343833  bytes 465375807 (443.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 82040  bytes 7008227 (6.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

网络连通性验证

- 同一个E-HPC集群下的SCC实例间的VPC网络IP和RoCE网络IP均可以相互ping通
- 同一个E-HPC集群下的SCC实例间可以通过VPC网络IP和RoCE网络IP进行ssh登陆

RoCE网络性能测试

测试RoCE网络的峰值带宽与延迟

- 带宽测试样例

```
##读带宽测试
ib_read_bw -a -q 20 --report_gbits ##服务端compute0执行
ib_read_bw -a -q 20 --report_gbits compute0 ##用户端compute1执行

##写带宽测试
ib_write_bw -a -q 20 --report_gbits ##服务端compute0执行
ib_write_bw -a -q 20 --report_gbits compute0 ##用户端compute1执行
```

- 延迟测试样例

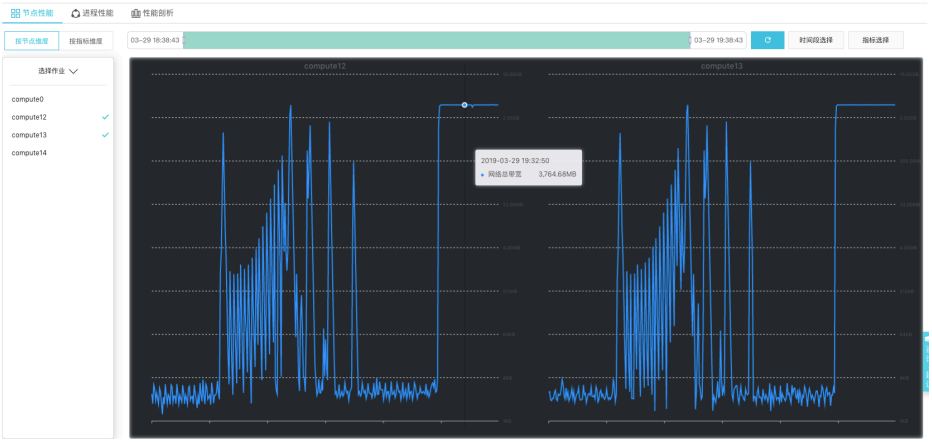
```
##读延迟测试
ib_read_lat -a ##服务端compute0执行
ib_read_lat -F -a compute0 ##用户端compute1执行

##写延迟测试
ib_write_lat -a ##服务端compute0执行
ib_write_lat -F -a compute0 ##用户端compute1执行
```

监测RoCE网络的实际带宽利用情况

- 在SCC实例root用户下执行rdma_monitor -s实时获取RoCE网络信息

```
------
2019-03-29 20:11:56 CST
tx_rate: 40.776Gbps
rx_rate: 46.853Mbps
tx_pause: 0
rx_pause: 0
tx_pause_duration: 0
rx_pause_duration: 0
np_cnp_sent: 0
rp_cnp_handled: 1931
num_of_qp: 23
np_ecn_marked: 0
rp_cnp_ignored: 0
out_of_buffer: 0
out_of_seq: 0
packet_seq_err: 0
cpu_usage: 0.09%
free_mem: 189864828 kB
```

- 使用E-HPC性能监控与分析引擎**集谛**来监测各SCC实例RoCE网络带宽随时间的变化情况。



在SCC集群上编译和运行MPI程序

由于SCC实例同时支持50Gbps RoCE网络和10Gbps VPC网络，用户在执行跨节点MPI程序时可能会遇到节点间数据流量默认走VPC网口的情况，这里我们推荐用户在SCC集群上使用IntelMPI来编译和运行跨节点MPI程序。

编译跨节点MPI程序

安装IntelMPI

E-HPC集成了IntelMPI 2018版本，用户只需在E-HPC控制台**集群创建**或**软件管理功能**界面中勾选IntelMPI 2018进行安装即可。

☑	intel-mpi	2018

配置MPI环境变量

- 方法一：使用E-HPC集成的Module管理工具

```
$ module avail
---------------------------------- /opt/ehpcmodulefiles ----------------------------------
intel-mpi/2018
$ module load intel-mpi/2018
$ which mpicc
/opt/intel/impi/2018.3.222/bin64/mpicc
```

- 方法二：执行IntelMPI自带的环境变量配置脚本

```
$ source /opt/intel/compilers_and_libraries/linux/bin/compilervars.sh intel64
$ which mpicc
/opt/intel/impi/2018.3.222/bin64/mpicc
```

设置MPI编译参数

完成MPI环境变量配置后，需要在软件Makefile或预编译脚本中指定MPI编译器的相对/绝对路径，然后执行编译过程。

```
-DCMAKE_C_COMPILER=mpicc
-DCMAKE_CXX_COMPILER=mpicxx
```

运行跨节点MPI程序

- 对于在E-HPC软件环境中采用IntelMPI编译的软件，提交任务时无需额外指定网口参数，便可以直接通过RoCE网络进行跨节点数据通信。

```
#!/bin/sh
#PBS -j oe
#PBS -l select=<节点数>:ncpus=<每节点核数>:mpiprocs=<每个节点进程数>

module load intel-mpi/2018
mpirun <软件执行命令>
```

- 对于在用户本地环境编译的软件或预编译的商用软件，可以在提交MPI任务时指定RoCE网卡信息来避免可能出现的数据流量不走RoCE网络或网卡设备not found等问题。

```
#!/bin/sh
#PBS -j oe
#PBS -l select=<节点数>:ncpus=<每节点核数>:mpiprocs=<每个节点进程数>

export I_MPI_FABRICS=shm:dapl
module load intel-mpi/2018
mpirun -genv I_MPI_DAPL_PROVIDER ofa-v2-mlx5_bond_0 <软件执行命令>
```

- 用户可以使用集谛性能监测功能对SCC实例的CPU利用率、访存带宽、RoCE网络带宽等性能数据进行

实时监测。


节点性能
进程性能
性能剖析
按节点维度
按指标维度
04-04 01:20:22
04-04 02:20:22
时间段选择
指标选择
选择作业
compute0
compute1
compute2
compute1
compute2
2019-04-04 01:45:00
CPU使用率 99.92%
内存使用率 2.98%
total bandwidth 29,090.12MB
网络总带宽 1,257.79MB