魔方-2 高性能 计算平 台应用 环境

1. 简介

魔方-2 高性能计算平台是基于集群概念设计的大型计算机系统，由 416 台双路刀片式服务器组成，其整体计算能力理论峰值为 399.36T flops（1Tflops 即为每秒 10^{12} 浮点计算）。魔方-2 系统每台服务器包含 2 颗英特尔志强 E5-2680 v3 处理器，每颗处理器包含 12 个处理器核心，计算主频 2.50GHz，全系统合计 9986 颗处理器核心。全系统包含三套内部互联网络，一套线速互联的 Infiniband 网络，一套千兆管理网络和一套 IPMI 网络。2015 年 9 月在上海超级计算中心完成安装后投入试运行。本文主要介绍在魔方 -2 高性能计算平台上部署的应用软件和机器的使用方法及环境。

2. 硬件环境

计算节点的硬件配置如下：

- 两路十二核 Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz Haswell 处理器（每节点合计 24 核）
- 每节点合计 128GB 内存
- Mellanox InfiniBand 光纤网络

魔方-2 系统的存储分为两种：每个计算节点配备的本地硬盘和由存储节点建立的高速并行文件系统。其中本地硬盘用于计算节点操作系统使用，用户的所有操作都应该在帐号所对应$HOME（该$HOME 所在的位置为高速并行文件系统）下进行，用户登录时，会自动被引导到自己帐号的$HOME 下面。鉴于存储空间有限和数据安全的考虑，请用户务必做到及时下载计算结果文件并清理空间。

3. 软件环境

3.1. 操作系统

魔方-2 系统的计算节点和前端接入节点的操作系统均为 CentOS release 6.6 (Final)，提供了一个标准的 64 位 Linux 操作环境（内核：2.6.32-504.e16.x86_64），用户需要事先适当熟悉命令行方式的基本 Linux 操作，特别是文件目录操作，并应该会熟练使用一种编辑器(vi 或者 emacs 等)。

3.2. 作业调度系统

对于大规模超级计算机系统，为了有效利用众多处理器核心所提供的计算能力，必须有一个统一的作业管理系统，统一地跟用户交互，接收提交的各类计算任务，统一地分配计算资源，将各种各样的用户作业具体指派到节点上执行。对用户来说不需要关心计算具体是在哪里进行的，系统会自动按照优化原则调度，这不仅方便了用户的使用，而且提高了整个系统的利用率。作业管理系统是整个超级计算机最重要的软件环境之一，魔方-2 系统使用 PBS 作业管理系统提交和管理计算作业任务。

3.3. 编译器和并行实现

魔方-2 系统支持 OpenMP 和 MPI 两种并行方式，前者为共享内存方式，仅能在一个计算节点内并行，最大线程数不能超过处理器核心数，在魔方-2 系统上不能超过 24；后者则是分布式内存并行，计算作业可以在一个或者若干个节点上进行，最大进程数仅受用户帐号所能使用的上限限制。

共享内存的 OpenMP 并行方式通常由编译器来支持，目前在魔方-2 系统上的 GNU 编译器和 Intel 的编译器软件均支持该标准，可使用相应的编译参数编译使用。

分布式消息传递的 MPI 并行方式，其实是一个设计规范的标准，提供了大量用于消息传递和管理的函数，支持从 C/C++和 Fortran 语言编写的程序中调用，也可以绑定到其它一些编程语言。MPI 只是一个标准，遵从这一标准可以有很多不同的软件实现，但具体的应用程序应该不加修改就可以重新编译运行，这也是标准化带来的优点。目前在魔方-2 系统上主要使用支持 InfiniBand 网络的 OpenMPI 实现。

在高级编程语言支持方面，主要可以使用 GNU 编译器和 Intel 编译器，相应的编译

器安装运行目录已经加入用户的环境变量 PATH 中了。在 Linux 操作系统下一般用 make 工具来组织管理源代码编译，而不是直接调用这些编译命令。

查找编译命令所在的路径可以使用 which 命令，例如”which mpicc”将返回 mpicc 命令所在的具体路径。确认编译器的版本请在编译命令后使用-v 或者-V 参数，例如”gcc -v”、”icc -V”，MPI 编译器的详细命令行调用则可以用”mpicc -show”获得。

3.4. 数学库

实际的开放源码程序往往要调用大量的数学函数进行各种计算，经过长时间的积累，已经有一些比较成熟的标准化的数学库，其中最常见的诸如线性代数方面的 BLAS、LAPACK、ScaLAPACK 和快速傅立叶变换 FFT 等等。在魔方-2 系统上使用 intel 编译器，可以使用相关的 MKL 数学库。

3.5. 应用软件情况

目前，在魔方-2 系统上已经测试或部署了常见的科学计算应用软件。主要测试过的软件包括

abinit、blast、cp2k、cpmd、nwchem、vasp、amber、espresso、gromacs、siesta、WIEN2k、gaussian09、lammps、namd、yamboo、MaterialsStudio 等，其中部分商业软件由用户协助测试完成。

计算节点的编号由 4 个字符组成，第一个字符为字母 a，后三位为一定范围内的数字，即 a[1-7][10-69]，其中[1-7]表示该数字的变化范围为 1 到 7，[10-69]表示该数字的变化范围为 10 到 69。其中节点 a110 仅作为编译节点。

魔方-2 系统上的作业调度系统设置了两个队列：

- score：可提交任意并行度的作业，主要用于串行作业（np=1）、小规模的并行作业（np<24），也可以用于并行度非 24 倍数的其他规模作业。
- snode：只能提交 24 的整数倍并行度的作业，作业最终提交运行时会独占所申请分配的整个计算节点。

4. 上机操作

魔方-2 系统内部有着复杂的网络系统来实现大规模集群群系统的各类功能，为了安全起见，只有 telnet 和 ftp 接入节点能够通过外部公网直接访问，即便如此也需要网络

防火墙作为额外的安全措施，将这些接入节点跟外部有风险的网络环境相对隔离开来。

使用魔方-2，必须登录系统，通过作业调度系统进行作业提交、管理、监控、删除等操作。所有作业提交均通过提交作业脚本的方式来进行。魔方-2 系统分配有各自独立 telnet 登录节点和 FTP 文件传输节点，这些节点分配有独立的对外 IP 地址，禁止在登录节点运行任何大规模程序和编译任何程序，只可以进行简单的文本操作。用户可以到编译节点编译程序，运行小规模的测试。

运行作业基本步骤如下：

(1) 模型准备——用户准备模型数据文件和作业脚本文件。

(2) 模型上传——通过 FTP 工具将模型数据文件和脚本文件上传至 FTP server。

(3) 作业提交——利用 Telnet 登陆魔方超级计算机，用 dos2unix 命令处理上传的文本文件后，用作业提交命令提交脚本文件进行计算。

(4) 作业监控——通过 Telnet 方式登录魔方机，采用作业管理命令监控作业的执行情况。

(5) 结果下载——计算完成后，通过 ftp 工具从 FTP Server 下载结果文件。

下面分别介绍上机操作的相关内容。

4.1. 登录和传输文件

魔方-2 系统目前仍然采用 telnet 方式登录主机以及 ftp 方式传输文件。登陆之前依然需要首先登陆 VPN。具体来说需要通过 https://vpn.ssc.net.cn 接入，使用 VPN 用户名和密码登陆后，选择主机应用下的“开始 VPN 客户端”后就可建立连接。建立连接后才可以使用终端操作（telnet 192.168.235.10）和文件传输（ftp 192.168.235.30），这两个操作均需要主机用户名和密码的认证。

魔方-2 系统也同时支持 IPV6 的接入[2001:DA8:8019:235::30]（telnet 和 ftp）。

4.2. 编译

用户用 telnet 登录成功后，首先进入的是登录节点。用户可以在登录节点查看目录、编辑文件、查看作业、查看资源使用情况等。但是用户不允许在登录节点运行计算程序或前后处理程序，也不允许进行程序编译。用户可以从登录节点转移到编译节点进行程序编译，编译节点为 all0，用户可以使用命令 ssh all0 登录编译节点 all0。

5. 作业提交

系统利用 PBS 进行资源和作业管理， 所有需要运行的作业无论是用于程序调试还是业务计算均必须通过 qsub 命令提交，提交后可以利用作业调度系统的相关命令查询作业状态等。为了利用 qsub 提交作业，用户需针对此作业创建提交脚本，在脚本里面设定需要运行的作业参数等。在此分别给出串行和并行的简单脚本，用户可以修改此脚本以适用于自己的作业。脚本文件是一个常规文本文件，具有执行权限，可以直接使用 vi 编辑器编写，也可异地编写上传至用户作业工作目录，但要注意 dos2unix 转换一下。脚本文件名无特殊规定，起一个有意义的名字即可。编辑完成脚本文件后，将脚本赋予可执行权限，然后提交。例如对一个名称为 test.job.pbs 的作业脚本文件，编辑完成后，需要执行命令 chmod 755 test.job.pbs 然后使用命令 qsub test.job.pbs 来提交。

注意：严禁使用任何前台或者后台方式直接由用户运行程序，所有的计算都必须作为任务提交到 PBS 系统然后统一调度执行。

5.1. 串行作业

对于串行程序，用户可编写命名为 test.job.pbs 的串行作业提交脚本，其内容如下：

```
#!/bin/sh
#PBS -N test.job
#PBS -l nodes=1:ppn=1
#PBS -q score
cd $PBS_O_WORKDIR
/path/to/test/job/binary/file
```

提交作业的所需信息需在作业提交脚本文件中利用#PBS 设置。上述脚本利用 qsub 命令提交后，表示进入提交作业的工作目录后，提交到 score 队列，其作业名为 test.job，默认的标准输出和错误输出将分别存在此目录下的 test.job.o[作业号]和 test.job.e[作业号]文件中。上述脚本中以#PBS 开头的几行中：

其中-N 参数后设置的是这个作业的名字 test.job；

其中-q 参数后设置作业使用的队列名 score；

其中-l 参数后的设置申请了计算资源分配情况，nodes=1 表示申请 1 个计算节点，ppn=1 表示申请计算节点内的 1 个计算进程。

作业脚本编写完成后，可以按照下面命令提交作业：

```
$qsub testjob.pbs
```

提交成功后将会看到类似如下的输出：

```
2580.mgmt34
```

其中 2580.mgmt34 表示的是作业号，其中数字 2580 表示的是作业号，mgmt34 表示的是作业管理系统的主机名，之后可以使用作业号来查询作业及对此作业的其他一些操作。

5.2. 并行作业

对于并行作业，用户需要例如命名为 testjob.pbs 的并行作业提交脚本，其内容如下：

```
#!/bin/sh
#PBS -N testjob
#PBS -l nodes=2:ppn=24
#PBS -q snode
cd $PBS_O_WORKDIR
NP=`wc -l<$PBS_NODEFILE`
mpirun -np $NP -machinefile $PBS_NODEFILE  \
/path/to/test/job/binary/file
```

与串行作业类似，提交作业的所需信息同样利用#PBS 设置：

其中-q 参数后设置作业使用的队列名 snode；

其中-l 参数后的设置申请了计算资源分配情况，nodes=2 表示申请 2 个计算节点，ppn=24 表示申请计算节点内的全部 24 个计算进程。

对于提交到 snode 队列的并行作业，我们要求并行规模 np 为 24 或者 24 的倍数，也就是说，我们要求这里始终设置 ppn=24，同时按照并行规模对 24 的倍数来设置 nodes 数。

作业脚本编写完成后，提交方法与串行作业类似。

5.3. 并行规模非 24 倍数的作业

当并行规模大于 24 同时非 24 的倍数时，例如提交并行度 32 的作业设置如下的提交脚本：

```
#!/bin/sh
#PBS -N testjob
#PBS -l nodes=1:ppn=24+1:ppn=8
```

```
#PBS -q score
cd $PBS_O_WORKDIR
NP=`wc -l<$PBS_NODEFILE`
mpirun -np $NP -machinefile $PBS_NODEFILE  \
/path/to/test/job/binary/file
```

因为并行规模并非 24 的倍数，所以利用#PBS 设置：

其中-q 需要设置 score 队列；

其中-l 设置了 nodes=1:ppn=24+1:ppn=8，申请 1 个计算节点，该计算节点申请 24 个计算进程，另外申请 1 个计算节点，该计算节点申请 8 个计算进程。

与此类似，例如并行度 64 的作业需通过#PBS -l 相关的设置为：

```
#PBS -l nodes=2:ppn=24+1:ppn=16
```

对于并行规模小于 24 的并行作业，我们要求提交到 score 队列（-q score），同时希望始终设置 nodes=1。例如，并行度 16 的作业需通过#PBS -l 相关的设置为：

```
#PBS -l nodes=1:ppn=16
```

5.4. 大内存占用作业

对于魔方-2 高性能计算系统来说，我们认为每个进程占用内存大于 5GB 就属于大内存占用作业，这时如果按照原有的提交办法提交作业就会出现计算节点内存不够用的情况。

我们假设需要提交一个并行度 16 的作业，经预估该作业每个计算进程需要占用内存 15GB，因为每个计算节点物理内存 128GB，所以每个计算节点的物理内存最多只能分配 8 个计算进程，这样的作业提交脚本例子如下所示：

```
#!/bin/sh
#PBS -N bigmemjob
#PBS -l nodes=2:ppn=24
#PBS -q snode
REAL_NP_PER_NODE=8
#generate nodelist
rm -rf $PWD/nodelist 1>/dev/null 2>&1
for i in `cat $PBS_NODEFILE | sort | uniq`
do
```

```
        for j in `seq 1 $REAL_NP_PER_NODE`
        do
        echo $i >> $PWD/nodelist
        done
done
#nodelist done
cd $PBS_O_WORKDIR
NP=`cat $PWD/nodelist | wc -l`
mpirun -np $NP -machinefile $PWD/nodelist  \
/path/to/big/mem/job/binary/file
```

该提交脚本中 REAL_NP_PER_NODE 参数的设置需要事先经过内存占用情况估算出来，在本例中即为 8，因为总的并行度为 16，这样就需要两台计算节点完成计算，故#PBS -l 参数后的设置的申请了计算资源情况中，nodes=2 表示申请 2 个计算节点。另外，这里请注意，提交到 snode 队列的作业，我们要求始终设置 ppn=24。

6. 作业管理

下面列出常用的作业管理命令，如果需要更详细的资料可以参考作业调度系统相关手册。

qdel：取消作业。

checkjob：显示作业状态、资源需求、环境、限制、信任、历史、已分配资源和资源利用等。（仅登陆节点可用）。

pbsnodes：显示节点信息。

qstat：显示队列、服务节点和作业的信息。

qsub：提交作业。

对于习惯使用上海超级计算中心之前高性能计算平台的用户，下面列出了原有的一些常用作业管理命令的替代办法。

bsub：qsub 替代。

bkill：qdel 替代。

bjobs\busers\bqueues：在魔方-2 系统上仍然可以使用这些命令，同时 qstat 命令

也可以实现相关的一些查询功能。

bpeek：在 PBS 作业调度系统中，标准输出和标准错误输出都会输出到文件中，如果需要实现原 bpeek -f 的即时更新输出功能，可以使用例如 tail -f 输出文件的方法，默认的标准输出和错误输出将分别存在工作目录下的[作业名].o[作业号]和[作业名].e[作业号]文件中，也可以在提交文件中使用#PBS -j oe 合并标准输出和标准错误输出到同一个文件。