HPC Cloud 解决方案 用户指南 01

文档版本 01

发布日期 2022-05-31

华为技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 简介

1.1 HPC 简介

什么是 HPC

高性能计算（High-performance computing，HPC）是一个计算机集群系统，它通过各种互联技术将多个计算机系统连接在一起，利用所有被连接系统的综合计算能力来处理大型计算问题，所以又通常被称为高性能计算集群。

HPC 的业务特点

科学研究、气象预报、仿真实验、生物制药、基因测序、图像处理等行业都涉及高性能计算集群来解决大型计算问题，管理节点对计算任务进行分解，交给不同的计算节点完成计算。

各种业务场景下，因数据处理量、计算任务关联关系等不同，对计算能力、存储效率、网络带宽及时延要求有各自侧重。

HPC 的应用场景

HPC提供了超高浮点计算能力解决方案，可用于解决计算密集型、海量数据处理等业务的计算需求，如科学研究、气象预报、计算模拟、军事研究、CAD/CAE、生物制药、基因测序、图像处理等，缩短需要的大量计算时间，提高计算精度。

1.2 HPC 和公有云

公有云上部署 HPC 的优势

传统的HPC使用中存在如下问题：

- 投资成本高，扩容部署复杂，重复利用已有投资十分困难。
- 应用复杂，资源预测困难，灵活性差，亟待提升效率。
- 效率低下导致决策缓慢，失去市场、以及开发研究成果的良机。
- 应用计算量快速膨胀，对性能要求越来越高。

公有云上应用HPC场景，能充分利用云服务的优势。使用公有云进行高性能计算具有以下优势：

- 降低TCO

  可以按需租用，成本低，降低中小客户使用HPC的门槛。
- 提高效率

  按需发放，快速部署与扩容，加速产品上市时间和缩短科研周期。
- 使用灵活
  - 在镜像模板中预制MPI库、编译库及优化配置，加快环境部署。
  - 企业分支、科研组织机构等跨全球地理位置进行及时协同工作，提高效率。
  - 可以利用公有云的跨地域能力，共享计算资源，海量数据，并能实现云端大数据分析。
- 优化性能
  - 性能比普通云服务器提高30%。
  - 通过虚拟化优化（SR-IOV、PCI直通）等，各类测试报告显示：大规模云化HPC性能损耗不大。

HPC 与云服务的关系

表 1-1 所需云服务

云服务	作用
弹性云服务器（ECS）	用于在公有云平台上创建高性能计算服务器。
虚拟私有云（VPC）	HPC场景下所涉及的云服务器，都位于同一个VPC中，并且需要使用VPC中的子网和安全组的相关网络安全隔离。
镜像服务（IMS）	● 在创建高性能计算的云服务器时，需要使用符合要求的镜像文件。 ● 在制作私有镜像时，需要将已有的高性能计算云服务器创建为私有镜像，从而创建集群使用。
云硬盘（EVS）	HPC场景下使用的弹性云服务器，均绑定了云硬盘。
裸金属服务器（BMS）	为用户提供专属的物理服务器，提供卓越的计算性能，满足核心应用对高性能及稳定性的需求，结合了传统托管服务器的稳定性与云中资源高度弹性的优势。
对象存储服务（OBS）	是一种基于对象的海量存储服务，为用户提供海量、低成本、高可靠、高安全的数据存储能力。
弹性文件服务（SFS）	为用户的弹性云服务器提供一个完全托管的共享文件存储，符合标准文件协议（NFS），能够弹性伸缩至PB规模，具备可扩展的性能，为海量数据、高带宽型应用提供有力支持。

云服务	作用
数据快递服务（DES）	是一种海量数据传输服务，它使用物理存储介质（USB或eSATA接口）向华为公有云传输大量数据。解决了海量数据在互联网上传输的难题（高昂网络带宽成本、较长传输时间等）。
云专线（DC）	用于搭建企业自有数据中心到华为公有云的高速、稳定、安全的专属连接通道，充分利用公有云服务优势的同时，继续使用现有的IT设施，实现灵活一体，可伸缩的混合云计算环境。
云监控服务（CES）	为用户提供一个针对弹性云服务器、带宽等资源的立体化监控平台。给用户提供实时监控告警、通知以及个性化报表视图，精准掌握产品资源状态。
云桌面（Workspace）	是一种由华为公有云提供的虚拟Windows桌面与应用的服务，用户可随时随地接入云桌面办公。云桌面服务提供专业的办公应用，帮助用户打造更精简、更安全、更低维护成本、更高服务效率的IT办公系统。

1.3 IPoIB 功能简介

什么是 IPoIB

IPoIB（Internet Protocol over InfiniBand），指利用物理IB网络（包括服务器上的IB卡、IB连接线、IB交换机等）通过IP协议进行连接，并进行数据传输。

它提供了基于RDMA之上的IP网络模拟层，允许应用无修改的运行在InfiniBand网络上。但是，IPoIB性能比RDMA通信方式性能要低，大多数应用都会采用RDMA方式获取高带宽低延时的收益，少数的关键应用会采用IPoIB方式通信。

说明

无修改：指运行在IP协议上的应用不需要作任何修改，即可适配IB网络运行。

IPoIB 的通信模式有哪些

IPoIB设备能够配置为datagram和connected两种模式，前者提供不可靠的、无连接的链路，后者提供可靠的、有连接的链路。

- 在datagram模式下，queue pair不允许报文大小超过IB链路层的MTU值，由于IPoIB头还包含了4字节，因此IPoIB的MTU值要小于IB链路层的MTU值。
- 在connected模式下，queue pair允许发送比IB链路层更大的报文，理论上可以发送大小65535长度的报文。 connected模式具有更好的性能，但是会消耗系统更多的内存。多数系统更关注性能，因此大多数场景下IB网口配置为connected模式。

当前版本的网卡驱动不支持connected模式。

说明

由于网卡驱动性能原因，系统关闭了connected模式，因此，当前版本网卡驱动不支持配置connected模式。

IPoIB 的 IP 地址分配方式有哪些

有两种IP地址分配方式：静态配置和DHCP动态配置。

- 静态配置：

  IPoIB设备有20个字节的硬件地址，前4个字节是queue pair number，中间8个字节是子网前缀，最后8个字节是guid。

  IPoIB设备的硬件地址只能通过ip命令查询，ifconfig查询不到完整地址。静态IP地址配置举例如下：

  图 1-1 静态 IP 地址配置

```
$ more ib0
DEVICE=mlx4_ib0
TYPE=InfiniBand
ONBOOT=yes
HWADDR=80:00:00:4c:fe:80:00:00:00:00:00:00:f4:52:14:03:00:7b:c
b:a1
BOOTPROTO=none
IPADDR=172.31.0.254
PREFIX=24
NETWORK=172.31.0.0
BROADCAST=172.31.0.255
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
MTU=65520
CONNECTED_MODE=yes
NAME=mlx4_ib0
```

- DHCP动态配置：

  标准的DHCP帧格式包括了硬件类型（ htype ）、硬件地址长度（ hlen ）、 硬件地址（ chaddr ）等字段。由于MAC地址字段的长度不能容纳IPoIB的硬件地址，因此定义client-identifier字段来标识client端dhcp会话。该client-identifier用于IP地址和client关联，DHCP Server根据该标识来区分客户端分配IP地址。

  HPC解决方案当前采用的是DHCP方式的IPoIB方案，实现IP地址自动化发放和配置。

IPoIB 的约束和限制

- 支持管理一个IB网卡。
- 继承BMS、H2型、HL1型、HI3型云服务器对IB网卡的使用约束与限制，使用IPoIB特性的弹性云服务器不支持迁移。
- 继承H2型、HL1型、HI3型云服务器对IB网卡的使用约束与限制，使用的IB网络不支持安全组、Qos、三层及以上网络功能。
- 受限于网卡驱动限制，使用IPoIB特性的弹性云服务器不支持anti arp-spoofing、dhcp-spoofing。

1.4 配额调整

什么是配额?

为防止资源滥用，平台限定了各服务资源的配额，对用户的资源数量和容量做了限制。如您最多可以创建多少台弹性云服务器、多少块云硬盘。

如果当前资源配额限制无法满足使用需要，您可以申请扩大配额。

怎样查看我的配额?

1. 登录管理控制台。
2. 单击管理控制台左上角的 ，选择区域和项目。
3. 在页面右上角，选择“资源 > 我的配额”。
   系统进入“服务配额”页面。

   图 1-2 我的配额

4. 您可以在“服务配额”页面，查看各项资源的总配额及使用情况。
   如果当前配额不能满足业务要求，请参考后续操作，申请扩大配额。

如何申请扩大配额?

1. 登录管理控制台。
2. 在页面右上角，选择“资源 > 我的配额”。
   系统进入“服务配额”页面。

图 1-3 我的配额

3. 单击“申请扩大配额”。
4. 在“新建工单”页面，根据您的需求，填写相关参数。
   其中，“问题描述”项请填写需要调整的内容和申请原因。
5. 填写完毕后，勾选协议并单击“提交”。

2 弹性云服务器场景典型应用

2.1 创建支持 IB 网卡的弹性云服务器

操作场景

您可以在几分钟之内快速获得基于公有云平台的弹性云服务器设施，并且这些设施是弹性的，可以根据需求伸缩。该任务指导用户如何创建支持IB网卡的弹性云服务器，包括管理控制台方式和基于HTTPS请求的API（Application programming interface）方式。

控制台方式

1. 登录管理控制台。
2. 选择“计算 > 弹性云服务器”。
   进入弹性云服务器信息页面。
3. 单击“创建弹性云服务器”，开始创建弹性云服务器。
4. 根据表2-1，填写待创建弹性云服务器的基本信息。

表 2-1 参数说明

参数	解释	取值样例
区域	如果所在区域不正确，请单击页面左上角的 进行切换。	亚太-新加坡
可用区	指在同一地域下，电力、网络隔离的物理区域，可用分区之内内网互通，不同可用分区之间物理隔离。 • 如果您需要提高应用的高可用性，建议您将云服务器创建在不同的可用分区。 • 如果您需要较低的网络时延，建议您将弹性云服务器创建在相同的可用区。	az-01
规格	选择H2或HI3型弹性云服务器。	h2.4xlarge.8

参数	解释	取值样例
专属主机	不涉及该参数。 HPC为单主机单个弹性云服务器场景，用户无需指定专属主机。	-
镜像	● 公共镜像 常见的标准操作系统镜像，所有用户可见，包括操作系统以及预装的公共应用。请根据您的实际情况自助配置应用环境或相关软件。 选择“公共镜像”，并展开下拉框，选择所需的公共镜像。 ● 私有镜像 用户基于弹性云服务器创建的个人镜像，仅用户自己可见。包含操作系统、预装的公共应用以及用户的私有应用。选择私有镜像创建弹性云服务器，可以节省您重复配置弹性云服务器的时间。 选择“私有镜像”，并展开下拉框，选择所需的私有镜像。您可以选择使用加密镜像，更多关于加密镜像的信息，请参见《镜像服务用户指南》。 ● 共享镜像 用户将接受公有云其他用户共享的私有镜像，作为自己的镜像进行使用。 选择“共享镜像”，并展开下拉框，选择所需的共享镜像。 ● 市场镜像 提供预装操作系统、应用环境和各类软件的优质第三方镜像。无需配置，可一键部署，满足建站、应用开发、可视化管理等个性化需求。 选择“市场镜像”，并单击“选择镜像”，选择所需的市场镜像。	公共镜像
许可证类型	可选参数，在公有云平台上使用操作系统或软件的许可证类型。 如果您选择的镜像为免费的，则系统不会展示该参数。如果您选择的镜像为计费镜像（SUSE、Oracle Linux、RedHat），此时系统会展示该参数。 ● 使用平台许可证 使用公有云平台提供的许可证，申请许可证需要支付一定的费用。 ● 使用自带许可证（BYOL） 使用用户已有操作系统的许可证，无需重新申请。	Bring your own license (BYOL)

参数	解释	取值样例
磁盘	也称云硬盘，包括系统盘和数据盘。 • 系统盘 如果镜像未加密，则系统盘也不加密，并在界面上显示“Unencrypted”。如果您选择加密镜像，系统盘会自动加密，具体请参见**• 加密涉及的参数（可选配置）**。 • 数据盘 您可以为弹性云服务器添加多块数据盘，并设置每块数据盘的设备类型，以及共享、加密功能。 – SCSI：勾选后，数据盘的设备类型为SCSI。它支持SCSI指令透传，允许云服务器操作系统直接访问底层存储介质。除了简单的SCSI读写命令，SCSI类型的云硬盘还支持更高级的SCSI命令。 **说明** 如果不勾选，默认创建VBD类型的云硬盘，该类型云硬盘只支持简单的SCSI读写命令。 – 共享盘：勾选后，数据盘为共享云硬盘。该共享盘可以同时挂载给多台云服务器使用。 – 加密：勾选后，数据盘加密，具体请参见**• 加密涉及的参数（可选配置）**。 • 加密涉及的参数（可选配置） 为了使用加密特性，需单击“Create Xrole”授权EVS访问KMS。如果您有授权资格，则可直接授权，如果权限不足，需先联系拥有Security Administrator权限的用户授权，然后再重新操作。 – Encrypted：表示云硬盘已加密。 – Create Xrole：用于授权EVS访问KMS获取KMS密钥。授权成功后，无需再次授权。 – 密钥名称：该加密云硬盘使用的密钥名称，默认为evs/default。 – Xrole名称:EVSAccessKMS：表示已授权EVS获取KMS密钥，用于加解密云硬盘。 – 密钥ID：该加密数据盘使用的密钥的ID。 更多关于云硬盘类型、设备类型、共享云硬盘、加密等信息，请参见《云硬盘用户指南》。	系统盘：超高I/O，40GB

5. 设置网络，包括“虚拟私有云”、“安全组”、“网卡”、“弹性IP”等信息。

第一次使用公有云服务时，系统将自动为您创建一个虚拟私有云，包括安全组、网卡。

表 2-2 参数说明

参数	解释	取值样例
虚拟私有云	弹性云服务器网络使用虚拟私有云（VPC）提供的网络，包括子网、安全组等。 您可以选择使用已有的虚拟私有云网络，或者单击“查看虚拟私有云”创建新的虚拟私有云。 **说明** 对于HPC集群中的弹性云服务器，需要属于同一VPC、同一子网内。	-
安全组	安全组用来实现安全组内和安全组间弹性云服务器的访问控制，加强弹性云服务器的安全保护。用户可以在安全组中定义各种访问规则，当弹性云服务器加入该安全组后，即受到这些访问规则的保护。 创建弹性云服务器时，可支持选择多个安全组（建议不超过5个）。此时，弹性云服务器的访问规则遵循几个安全组规则的并集。 **说明** 弹性云服务器初始化需要确保安全组出方向规则满足如下要求： ● 协议：TCP ● 端口范围：80 ● 远端地址：169.254.0.0/16 如果您使用的是默认安全组出方向规则，则已经包括了如上要求，可以正常初始化。默认安全组出方向规则为： ● 协议：ANY ● 端口范围：ANY ● 远端地址：0.0.0.0/16	-
网卡	包括主网卡和扩展网卡。 您可以添加多张扩展网卡，并指定网卡（包括主网卡）的IP地址。	-
弹性公网IP	弹性公网IP是指将公网IP地址和路由网络中关联的弹性云服务器绑定，以实现虚拟私有云内的弹性云服务器通过固定的公网IP地址对外提供访问服务。 必须绑定弹性公网IP，您可以根据实际情况进行选择： ● 现在购买：自动为每台弹性云服务器分配独享带宽的弹性IP，带宽值可以由您设定。 ● 使用已有：为弹性云服务器分配已有弹性IP。使用已有弹性IP时，不能批量创建弹性云服务器。	现在购买

6. 设置“登录方式”。

   “密钥对”方式创建的弹性云服务器安全性更高，建议选择“密钥对”方式。如果您习惯使用“密码”方式，请增强密码的复杂度，如表2-3所示，保证密码符合要求，防止恶意攻击。

   - 密钥对

     指使用密钥对作为弹性云服务器的鉴权方式。您可以选择使用已有的密钥，或者单击“查看密钥对”创建新的密钥。

     说明

     如果选择使用已有的密钥，请确保您已在本地获取该文件，否则，将影响您正常登录弹性云服务器。

   - 密码

     指使用设置root用户（Linux）和Administrator用户（Windows）的初始密码作为弹性云服务器的鉴权方式，如果选择此方式，您可以通过用户名密码方式登录弹性云服务器。

     Linux操作系统时为root用户的初始密码，Windows操作系统时为Administrator用户的初始密码。

     表 2-3 密码设置规则

参数	规则	样例
密码	• 密码长度范围为8到26位。 • 密码至少包含以下4种字符中的3种： – 大写字母 – 小写字母 – 数字 – Windows操作系统云服务器特殊字符：包括“$”、“!”、“@”、“%”、“-”、“ ”、“_”、“=”、“+”、“[”、“]”、“:”、“.”、“/”、“,”和“?” – Linux操作系统云服务器特殊字符：包括“!”、“@”、“%”、“-”、“ ”、“_”、“=”、“+”、“[”、“]”、“:”、“.”、“/”、“^”、“,”、“{”、“}”和“?” • 密码不能包含用户名或用户名的逆序。 • Windows操作系统的云服务器，不能包含用户名中超过两个连续字符的部分。	YNbUwp!dUc9MClnv **说明** 样例密码随机生成，请勿复制使用样例。

说明

系统不会定期自动修改弹性云服务器密码。为安全起见，建议您定期修改密码。

7. 高级配置

   可选配置，如需使用“高级配置”中的功能，请单击“现在配置”。否则，请单击“暂不配置”。

   - 文件注入

     可选配置，主要用于创建弹性云服务器时向弹性云服务器注入脚本文件或其他文件。配置文件注入后，系统在创建弹性云服务器时自动将文件注入到指定目录下。

   - 用户数据注入

     可选配置，主要用于创建弹性云服务器时向弹性云服务器注入用户数据。配置用户数据注入后，弹性云服务器首次启动时会自行注入数据信息。

   - 云服务器组

     可选配置，云服务器组内的弹性云服务器将遵循反亲和策略，尽量分散地创建在不同主机上。

     说明

     如果您使用SCSI类型的共享云硬盘作为数据盘，此时，为支持SCSI锁命令，建议您设置待创建弹性云服务器的云服务器组。

   - 标签

     对弹性云服务器的标识。

     可选配置，给弹性云服务器添加标签，方便识别和管理您拥有的弹性云服务器资源。

8. 设置“云服务器名称”。

   名称可自定义，但需符合命名规则：只能由中文字符、英文字母、数字及“_”、“-”、“.”组成。

   如果同时创建多台弹性云服务器，系统会自动按序增加后缀。

9. 设置您创建弹性云服务器的数量。

   设置完成后，您可通过单击“价格计算器”，查询当前配置的费用。

10. 单击“立即申请”。
11. 在确认规格页面，您可以查看规格详情并提交申请。

    如果您确认规格无误，单击“提交申请”。

    弹性云服务器创建成功后，您可以在弹性云服务器信息页面看到您新创建的弹性云服务器。

12. （可选）如果您创建弹性云服务器时添加了数据盘，待弹性云服务器创建完成后，需要初始化数据盘。

    操作方法请参考《云硬盘用户指南》中的“初始化数据盘”。

API 方式

以创建H2型弹性云服务器为例：

1. 获取Token。

   - URI

     POST /v3/auth/tokens

   - 请求样例

     ```
     curl -i -k -H 'Accept:application/json;charset=utf8' -H 'Content-Type:application/json' -d '
     {"auth": {"identity": {"methods": ["password"],"password": {"user": {"name":
     ```

```
"$OS_USERNAME","password": "$OS_PASSWORD","domain": {"name”:”
$OS_USER_DOMAIN_NAME”"}}}},"scope": {"project": {"name": "eu-de"}}}}' -X POST https://
iam.eu-de.otc.t-systems.com/v3/auth/tokens
```

- 响应样例

图 2-1 获取 Token 响应样例

2. 创建VPC。
   - URI

     POST /v1/{$tenant_id}/ vpcs
   - 请求样例

```
curl -i -k -H 'Accept:application/json;charset=utf8' -H 'Content-Type:application/json' -H "X-Auth-
Token:$TOKEN " -d '
{
    "vpc": {
            "name": "vpc-test",
            "cidr": "192.168.0.0/16"
    }
}' -X POST https://iam.eu-de.otc.t-systems.com:443/v1/{$tenant_id}/vpcs
```

   - 响应样例

     VPC-id: 97701dc4-bfd3-4021-8b89-044486c8b317

图 2-2 创建 VPC 响应样例

```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 24 Apr 2016 11:44:34 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Accept-Ranges: bytes
Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept
X-Frame-Options: SAMEORIGIN
Cache-Control: no-cache
X-Content-Type-Options: nosniff
X-Download-Options: noopen
X-XSS-Protection: 1; mode=block
Strict-Transport-Security: max-age=31536000; includeSubdomains;
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
X-Download-Options: noopen
X-XSS-Protection: 1; mode=block;

{"vpc":{"id":"97701dc4-bfd3-4021-8b89-044486c8b317","name":"vpc-test","cidr":"19
2.168.0.0/16","status":"CREATING"}}root@server8d3f4359-8b47-4779-abe0-94c199030b
bd:~/openstack#
```

3. 创建子网。
   - URI

     POST /v1/{$tenant_id}/subnets

- 请求样例

```
curl-i-k-H'Accept: application/json;charset=utf8'-H'Content-Type: application/json'-H"X-Auth-Token:$TOKEN "-d'{
    "subnet": {
        "name": "subnet_test",
        "cidr": "192.168.30.0/24",
        "gateway_ip": "192.168.30.1",
        "dhcp_enable": "true",
        "primary_dns": "114.114.114.114",
        "secondary_dns": "114.114.115.115",
        "availability_zone": "eu-de-01",
        "vpc_id": "97701dc4-bfd3-4021-8b89-044486c8b317"
    }
}'-XPOSThttps: //iam.eu-de.otc.t-systems.com: 443/v1/{
    $tenant_id
}/subnets
```

- 响应样例

  Subnet-id: 6712fc43-a196-4973-8b5e-5e4763f6449b

  图 2-3 创建子网响应样例

```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 24 Apr 2016 11:49:08 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Accept-Ranges: bytes
Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept
X-Frame-Options: SAMEORIGIN
Cache-Control: no-cache
X-Content-Type-Options: nosniff
X-Download-Options: noopen
X-XSS-Protection: 1; mode=block
Strict-Transport-Security: max-age=31536000; includeSubdomains;
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
X-Download-Options: noopen
X-XSS-Protection: 1; mode=block;

{"subnet":{"id":"6712fc43-a196-4973-8b5e-5e4763f6449b","name":"subnet_test","cid
r":"192.168.30.0/24","status":"BUILD","vpc_id":"97701dc4-bfd3-4021-8b89-044486c8
b317","gateway_ip":"192.168.30.1","dhcp_enable":true,"primary_dns":"114.114.114.
114","secondary_dns":"114.114.115.115","availability_zone":"eu-de-01"}}root@serv
er8d3f4359-8b47-4779-abe0-94c199030bbd:~/openstack#
```

4. 创建EIP。

- URI

  POST /v1/{$tenant_id}/publicips

- 请求样例

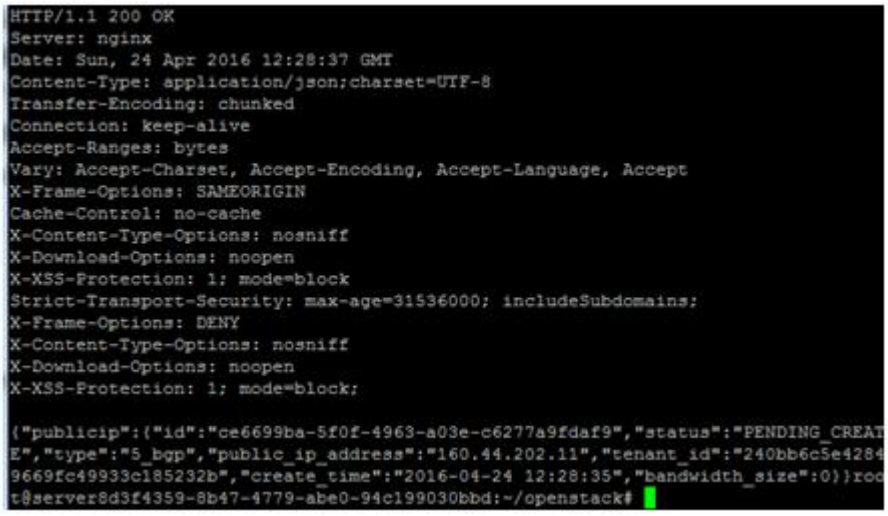
```
curl -i -k -H 'Accept:application/json;charset=utf8' -H 'Content-Type:application/json' -H 'X-Auth-Token:$TOKEN ' -d '{"publicip":{"type":"5_bgp"},"bandwidth":{"name":"apiTest","size":111,"share_type":"PER","charge_mode":"traffic"}}' -X POST https://iam.eu-de.otc.t-systems.com:443/v1/{$tenant_id}/publicips
```

- 响应样例

  EIP:160.44.202.11

  EIP ID: ce6699ba-5f0f-4963-a03e-c6277a9fdaf9

图 2-4 创建 EIP 响应样例

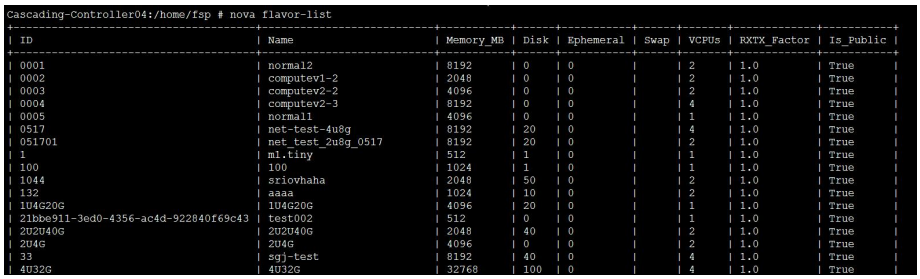


5. 查看规格列表。
   - Client方式

     执行以下命令，查看规格列表。

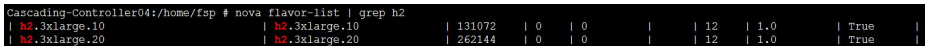
     nova flavor-list

     图 2-5 查看规格列表



     nova flavor-list | grep h2

     图 2-6 查看 H2 型规格列表



   - Curl命令方式
     - URI

       GET /v2/{$tenant_id}/flavors/detail
     - 请求样例

       curl -g -i -X GET https://iam.eu-de.otc.t-systems.com:443/v2/{$tenant_id}/flavors/detail -H "User-Agent: python-novaclient" -H "Accept: application/json" -H "X-Auth-Token: $TOKEN"
     - 响应样例

       Flavor id Example: h2.3xlarge.10

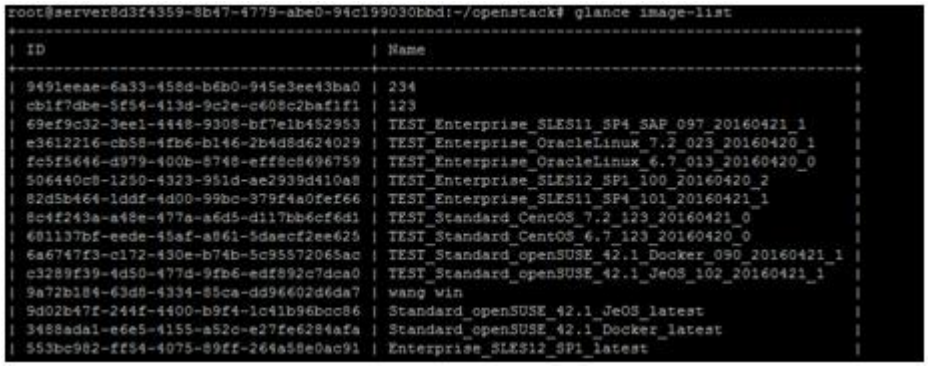
图 2-7 查看规格列表响应样例

6. 查看镜像列表。
   - Client方式

     执行以下命令，查看镜像列表。

     glance image-list

     图 2-8 查看镜像列表



ID	Name
9491eeae-6a33-458d-b6b0-945e3ee43ba0	234
cb1f7dbe-5f54-413d-9c2e-c608c2baf1f1	123
69ef9c32-3ee1-4448-9308-bf7e1b452953	TEST_Enterprise_SLES11_SP4_SAP_097_20160421_1
e3612216-cb58-4fb6-b146-2b4d8d624029	TEST_Enterprise_OracleLinux_7.2_023_20160420_1
fc5f5646-d979-400b-8748-eff8c8696759	TEST_Enterprise_OracleLinux_6.7_013_20160420_0
506440c8-1250-4323-951d-ae2939d410a8	TEST_Enterprise_SLES12_SP1_100_20160420_2
82d5b464-1ddf-4d00-99bc-379f4a0fef66	TEST_Enterprise_SLES11_SP4_101_20160421_1
8c4f243a-a48e-477a-a6d5-d117bb6cf6d1	TEST_Standard_CentOS_7.2_123_20160421_0
681137bf-eede-45af-a861-5daecf2ee625	TEST_Standard_CentOS_6.7_123_20160420_0
6a6747f3-c172-430e-b74b-5c95572065ac	TEST_Standard_openSUSE_42.1_Docker_090_20160421_1
e3289f39-4d50-477d-9fb6-edf892c7dca0	TEST_Standard_openSUSE_42.1_JeOS_102_20160421_1
9a72b186-63d8-4334-85ca-dd96602d6da7	wang win
9d02b47f-244f-4400-b9f4-1c41b96bcc86	Standard_openSUSE_42.1_JeOS_latest
3488ada1-e6e5-4155-a52c-e27fe6284afa	Standard_openSUSE_42.1_Docker_latest
553bc982-ff54-4075-89ff-264a58e0ac91	Enterprise_SLES12_SP1_latest

   - Curl命令方式
     - URI

       GET /v2/{$tenant_id}/images/detail
     - 请求样例

       curl -g -i -X GET https://iam.eu-de.otc.t-systems.com:443/v2/{$tenant_id}/images/detail -H "User-Agent: python-novaclient" -H "Accept: application/json" -H "X-Auth-Token:$TOKEN"
     - 响应样例

       Image id Example: 7474de73-9618-4c6a-afaa-df60df57c9b9

       图 2-9 查看镜像列表响应样例

7. 创建弹性云服务器。

- URI

  POST /v2/{project_id}/servers

- 请求样例

```
curl -i -k -H 'Accept:application/json;charset=utf8' -H 'Content-Type:application/json' -H 'X-Auth-Token:$TOKEN' -d '{"server": {"availability_zone": "eu-de-01","adminPass": "Test@123","name": "h2_vm","flavorRef": "h2.3xlarge.10","networks": [{"uuid":"6712fc43-a196-4973-8b5e-5e4763f6449b"}],"imageRef":"7474de73-9618-4c6a-afaa-df60df57c9b9"}}' -X POST https://46.29.103.37:443/v2/240bb6c5e42849669fc49933c185232b/servers
```

- 响应样例

```
{
   "server": {
      "security_groups": [
         {
            "name": "default"
         }
      ],
      "OS-DCF:diskConfig": " MANUAL",
      "id": "877a2cda-ba63-4e1e-b95f-e67e48b6129a",
      "links": [
         {
            "href": "https://46.29.103.37:443/v2/240bb6c5e42849669fc49933c185232b/servers/877a2cda-ba63-4e1e-b95f-e67e48b6129a",
            "rel": "self"
         },
         {
            "href": "http://46.29.103.37:443/240bb6c5e42849669fc49933c185232b/servers/877a2cda-ba63-4e1e-b95f-e67e48b6129a",
            "rel": "bookmark"
         }
      ],
      "adminPass": "******"
   }
}
```

8. 执行以下命令，查看弹性云服务器的网卡ID。

   nova interface-list {$VMID}

   系统回显类似如下：

   图 2-10 查看网卡 ID

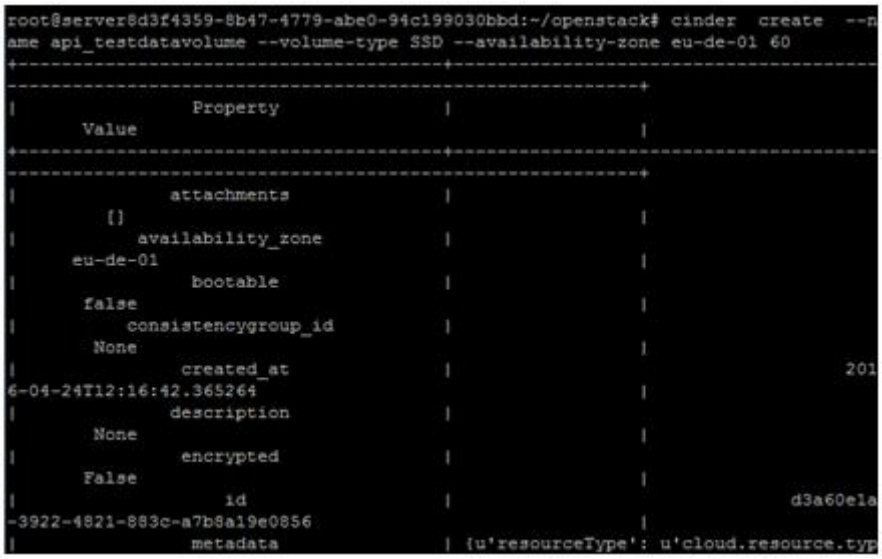
   则查看的网卡ID为“Vmid= eaf85b32-9912-4630-a9db-ab2d9b7c18b4”。

9. 执行以下命令，创建数据盘。

   cinder create --name datavolume --volume-type SATA --availability-zone eu-de-01 60

   系统回显类似如下：

图 2-11 创建数据盘



则创建的数据盘ID为“Datadiskid= d3a60e1a-3922-4821-883c-a7b8a19e0856”。

10. 执行以下命令，检查数据盘状态。

    cinder show {volumeId}

    如果数据盘状态为可用，则可以将其挂载至弹性云服务器上。

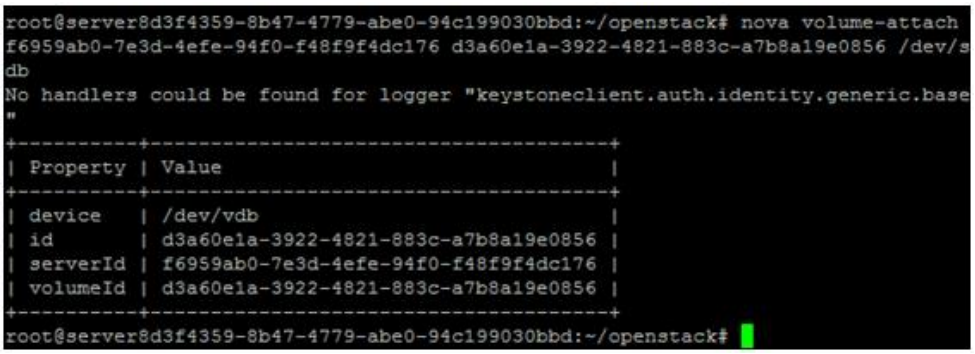
11. 执行以下命令，挂载可用的数据盘至弹性云服务器。

    nova volume-attach {serverId} {volumeId} device_name

    示例：

    nova volume-attach f6959ab0-7e3d-4efe-94f0-f48f9f4dc176 d3a60e1a-3922-4821-883c-a7b8a19e0856 /dev/sdb

图 2-12 挂载数据盘



12. 绑定弹性公网IP。
    - URI

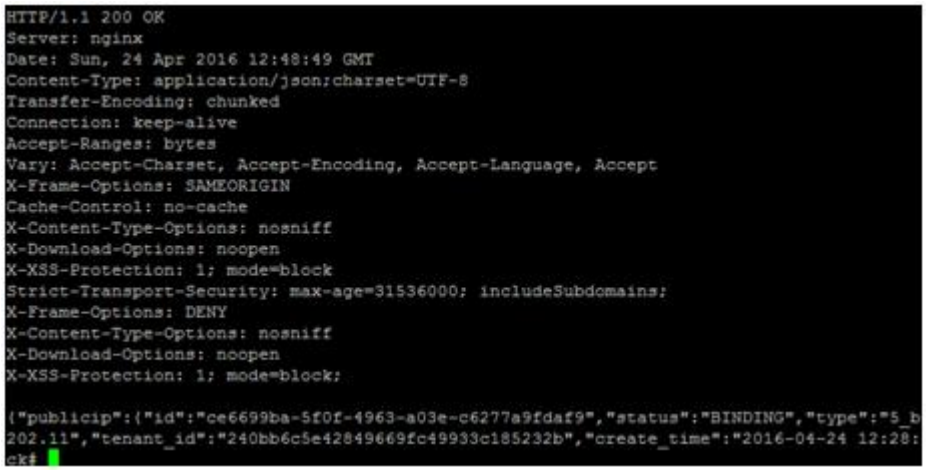
      PUT /v1/{$tenant_id}/publicips/{EIPid}

    - 请求样例

      ```
      curl -i -k -H 'Accept:application/json;charset=utf8' -H 'Content-Type:application/json' -H 'X-Auth-Token:$TOKEN' -d '{"publicip":{"port_id":"eaf85b32-9912-4630-a9db-ab2d9b7c18b4"}}' -X PUT https://46.29.103.37:443/v1/{$tenant_id}/publicips/ce6699ba-5f0f-4963-a03e-c6277a9fdaf9
      ```

- 响应样例

图 2-13 绑定弹性公网 IP 响应样例



2.2 配置单个 ECS 免密登录

操作场景

该任务指导用户在单个弹性云服务器内执行相关配置，使其可以免密登录。

背景信息

$：表示在普通用户下，执行相关操作。

#：表示在管理员用户下，执行相关操作。

普通用户切换至管理员用户，请使用命令**sudo su**。

前提条件

已成功创建弹性云服务器，并绑定了弹性IP进行登录。

操作步骤

1. 使用“PuTTY”，采用密钥对方式登录集群中任意一台ECS。
2. 执行以下命令，防止系统超时退出。
   # TMOUT=0
3. 将ECS对应的密钥文件（.pem文件，假设为*.pem）拷贝至.ssh目录下，命名为“id_rsa”。
   $ cd ~/.ssh
   $ mv *.pem id_rsa
4. 执行以下命令，给密钥文件配置权限。
   $ sudo chmod 600 id_rsa
5. 执行以下命令，查询主机名。
   # hostname

6. 执行以下命令，添加本主机私网IP地址和主机名。
   # vi /etc/hosts
   示例：
   192.168.0.1 ecs-ff-0001
7. 执行以下命令，ssh方式登录本节点，验证是否可以不输入密码登录ECS。
   假设本主机的主机名为hostname1，则命令行如下：
   $ ssh localhost
   $ ssh *hostname1*

2.3 安装和使用 MPI

2.3.1 弹性云服务器场景支持使用的 MPI

HPC当前支持的MPI包括：

- 驱动自带的OpenMPI
- 社区OpenMPI
- Spectrum MPI
- Intel MPI
- Platform MPI

以下小节的内容详细介绍了MPI的安装与使用，您可以根据需要选择合适的MPI进行安装。

2.3.2 IB 驱动自带的 OpenMPI

操作场景

本节指导用户安装和使用IB驱动自带的OpenMPI（以版本3.0.0rc6为例）。

前提条件

已配置弹性云服务器免密登录。

操作步骤

步骤1 检查是否已安装IB驱动。

1. 使用“PuTTY”，采用密钥对方式登录弹性云服务器。
2. 执行以下命令，切换为root 用户。
   $ sudo su
3. 执行以下命令，防止系统超时退出。
   # TMOUT=0
4. 执行以下命令，查询是否已安装IB驱动。
   # rpm -qa | grep mlnx-ofa
   # ls /usr/mpi/gcc/openmpi-3.0.0rc6/bin/mpirun

图 2-14 已安装 IB 驱动

```
[root@dyna-0002 ~]# ls  /usr/mpi/gcc/openmpi-3.0.0rc6/bin/mpirun
/usr/mpi/gcc/openmpi-3.0.0rc6/bin/mpirun
[root@dyna-0002 ~]#
[root@dyna-0002 ~]# rpm -qa|grep mlnx-ofa
mlnx-ofa_kernel-devel-4.2-OFED.4.2.1.2.0.1.gf8de107.rhel7u3.x86_64
kmod-mlnx-ofa_kernel-4.2-OFED.4.2.1.2.0.1.gf8de107.rhel7u3.x86_64
mlnx-ofa_kernel-4.2-OFED.4.2.1.2.0.1.gf8de107.rhel7u3.x86_64
```

- 如果上述两条命令均有如图2-14所示的返回值，则已安装IB驱动，执行**步骤3**。
- 如果返回值与图2-14不同，表示弹性云服务器未安装IB驱动，执行**步骤2**。

步骤2 下载并安装对应的IB驱动。

在Mellanox官网**https://network.nvidia.com/products/infiniband-drivers/linux/mlnx_ofed/**，选择相应版本的InfiniBand网卡驱动下载，并根据Mellanox提供的操作指导进行安装。

以操作系统CentOS 7.3为例，推荐安装4.2.1版本的安装包，下载安装包“MLNX_OFED_LINUX-4.2-1.2.0.0-rhel7.3-x86_64.tgz”，并执行以下命令进行安装：

 # yum install tk tcl

 # tar -xvf MLNX_OFED_LINUX-4.2-1.2.0.0-rhel7.3-x86_64.tgz

 # cd MLNX_OFED_LINUX-4.2-1.2.0.0-rhel7.3-x86_64/

 # ./mlnxofedinstall

步骤3 配置环境变量。

1. 执行以下命令，使用vim编辑“~/.bashrc”文件，添加如下配置内容。
   export PATH=$PATH:/usr/mpi/gcc/openmpi-3.0.0rc6/bin
   export LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-3.0.0rc6/lib64
2. 执行以下命令，导入配置的MPI环境变量。
   # source ~/.bashrc
3. 执行以下命令，查看MPI环境变量是否正常。
   # which mpirun

图 2-15 查看 MPI 环境变量

```
[root@dyna-0002 ~]# which mpirun
/usr/mpi/gcc/openmpi-3.0.0rc6/bin/mpirun
```

系统回显结果如图2-15所示，则环境配置正常。

步骤4 执行以下命令，在单个弹性云服务器上运行Intel MPI benchmark。

 # mpirun --allow-run-as-root -np 2 /usr/mpi/gcc/openmpi-3.0.0rc6/tests/imb/IMB-MPI1 PingPong

系统回显如下：

```
#---------------------------------------------------------------
#    Intel (R) MPI Benchmarks 4.1, MPI-1 part
#---------------------------------------------------------------
# Date                  : Mon Jul 16 10:11:14 2018
# Machine               : x86_64
# System                : Linux
# Release               : 3.10.0-514.10.2.el7.x86_64
# Version               : #1 SMP Fri Mar 3 00:04:05 UTC 2017
# MPI Version           : 3.1
# MPI Thread Environment:

# New default behavior from Version 3.2 on:

# the number of iterations per message size is cut down
# dynamically when a certain run time (per message size sample)
# is expected to be exceeded. Time limit is defined by variable
# "SECS_PER_SAMPLE" (=> IMB_settings.h)
# or through the flag => -time


# Calling sequence was:

# /usr/mpi/gcc/openmpi-3.0.0rc6/tests/imb/IMB-MPI1 PingPong

# Minimum message length in bytes:   0
# Maximum message length in bytes:   4194304
#
# MPI_Datatype                   :   MPI_BYTE
# MPI_Datatype for reductions    :   MPI_FLOAT
# MPI_Op                         :   MPI_SUM
#
#

# List of Benchmarks to run:

# PingPong

#---------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#---------------------------------------------------
#bytes #repetitions      t[usec]   Mbytes/sec
0         1000         0.24         0.00
1         1000         0.25         3.89
2         1000         0.23         8.17
4         1000         0.23        16.25
8         1000         0.23        32.48
16         1000         0.23        65.98
32         1000         0.26       115.35
64         1000         0.26       232.92
128         1000         0.38       320.59
256         1000         0.44       554.35
512         1000         0.54       902.98
1024         1000         0.64      1537.63
2048         1000         0.85      2298.79
4096         1000         1.28      3057.93
8192         1000         2.28      3426.14
16384         1000         1.41     11052.14
32768         1000         2.05     15218.39
65536          640         3.31     18882.34
131072          320         6.57     19036.27
262144          160        15.12     16535.96
524288           80        32.90     15195.74
1048576          40        64.62     15476.02
2097152          20       122.83     16282.06
4194304          10       242.95     16463.95
```



```
# All processes entering MPI_Finalize
```

----结束

2.3.3 社区 OpenMPI

操作场景

本节指导用户安装和使用社区OpenMPI（以3.1.1版本的OpenMPI为例）。

前提条件

已配置弹性云服务器免密登录。

操作步骤

步骤1 安装HPC-X工具套件。

1. 下载需要的HPC-X工具套件以及OpenMPI。

   使用社区OpenMPI时，需要同时使用Mellanox的HPC-X 套件，HPC-X 的下载需要参考弹性云服务器操作系统的版本以及IB驱动版本，例如，下载的HPC-X版本为：hpcx-v2.0.0-gcc-MLNX_OFED_LINUX-4.2-1.2.0.0-redhat7.3-x86_64.tbz。

2. 执行以下命令，解压HPC-X工具套件。

   # tar -xvf hpcx-v2.0.0-gcc-MLNX_OFED_LINUX-4.2-1.2.0.0-redhat7.3-x86_64.tbz

3. （可选）执行以下命令，修改HPC-X工具套件的目录。

   # mv hpcx-v2.0.0-gcc-MLNX_OFED_LINUX-4.2-1.2.0.0-redhat7.3-x86_64.tbz /opt/hpcx-v2.0.0

步骤2 安装OpenMPI。

1. 将下载的OpenMPI压缩包（以openmpi-3.1.1.tar.gz为例）拷贝至弹性云服务器内，并执行以下命令进行解压。

   # tar -xzvf openmpi-3.1.1.tar.gz

   # cd openmpi-3.1.1

2. 执行以下命令，安装需要的库文件。

   # yum install binutils-devel.x86_64 libibverbs-devel

3. 执行以下命令，编译安装OpenMPI。

   # ./autogen.pl

   # mkdir build

   # cd build

   # ../configure --prefix=/opt/openmpi-311 --with-mxm=/opt/hpcx-v2.0.0/mxm

   # make all install

图 2-16 安装 OpenMPI

```
make[3]: Nothing to be done for `install-exec-am'.
make[3]: Nothing to be done for `install-data-am'.
make[3]: Leaving directory `/root/openmpi-3.1.1/test'
make[2]: Leaving directory `/root/openmpi-3.1.1/test'
make[1]: Leaving directory `/root/openmpi-3.1.1/test'
make[1]: Entering directory `/root/openmpi-3.1.1'
make[2]: Entering directory `/root/openmpi-3.1.1'
make  install-exec-hook
make[3]: Entering directory `/root/openmpi-3.1.1'
make[3]: Leaving directory `/root/openmpi-3.1.1'
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/root/openmpi-3.1.1'
make[1]: Leaving directory `/root/openmpi-3.1.1'
```

系统回显的安装过程如图2-16所示，且退出后无报错，说明安装OpenMPI成功。

步骤3 配置MPI环境变量。

1. 在“~/.bashrc”添加如下环境变量：
   export PATH=$PATH:/opt/openmpi-311/bin
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/openmpi-311/lib
2. 执行以下命令，导入配置的MPI环境变量。
   # source ~/.bashrc
3. 执行以下命令，查看MPI环境变量是否正常。
   # which mpirun

图 2-17 查看社区 OpenMPI 环境变量

```
[root@dyna-0003 openmpi-3.1.1]# which mpirun
/opt/openmpi-311/bin/mpirun
```

系统回显结果如图2-17所示，则环境配置正常。

步骤4 执行以下命令，在单个弹性云服务器上运行Intel MPI benchmark。

$ mpirun --allow-run-as-root -np 2 /usr/mpi/gcc/openmpi-3.0.0rc6/tests/imb/IMB-MPI1 PingPong

系统回显如下：

```
#------------------------------------------------------------
#    Intel (R) MPI Benchmarks 4.1, MPI-1 part
#------------------------------------------------------------
# Date                  : Mon Jul 16 09:38:20 2018
# Machine               : x86_64
# System                : Linux
# Release               : 3.10.0-514.10.2.el7.x86_64
# Version               : #1 SMP Fri Mar 3 00:04:05 UTC 2017
# MPI Version           : 3.1
# MPI Thread Environment:

# New default behavior from Version 3.2 on:

# the number of iterations per message size is cut down
# dynamically when a certain run time (per message size sample)
# is expected to be exceeded. Time limit is defined by variable
# "SECS_PER_SAMPLE" (=> IMB_settings.h)
# or through the flag => -time
```



```
# Calling sequence was:

# /usr/mpi/gcc/openmpi-3.0.0rc6/tests/imb/IMB-MPI1 PingPong

# Minimum message length in bytes:   0
# Maximum message length in bytes:   4194304
#
# MPI_Datatype                   :   MPI_BYTE
# MPI_Datatype for reductions    :   MPI_FLOAT
# MPI_Op                         :   MPI_SUM
#
#

# List of Benchmarks to run:

# PingPong

#---------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#---------------------------------------------------
#bytes #repetitions      t[usec]   Mbytes/sec
0         1000         0.23         0.00
1         1000         0.23         4.06
2         1000         0.24         8.04
4         1000         0.24        16.19
8         1000         0.24        32.29
16        1000         0.24        64.06
32        1000         0.27       114.46
64        1000         0.27       229.02
128       1000         0.37       333.48
256       1000         0.46       535.83
512       1000         0.52       944.51
1024      1000         0.63      1556.77
2048      1000         0.83      2349.92
4096      1000         1.35      2896.07
8192      1000         2.29      3415.98
16384     1000         1.46     10727.65
32768     1000         2.08     15037.62
65536      640         3.53     17691.38
131072     320         6.52     19159.59
262144     160        15.62     16002.93
524288      80        31.37     15938.06
1048576     40        61.78     16185.93
2097152     20       124.04     16124.41
4194304     10       242.42     16500.33

# All processes entering MPI_Finalize
```

----结束

2.3.4 Spectrum MPI

操作场景

本节指导用户安装和使用IBM Spectrum MPI（以IBM Spectrum MPI v10.1为例）。

其中，IBM Spectrum MPI v10.1版本当前支持的操作系统列表如下：

- **IBM Spectrum MPI 10.1.0.1 Eval for x86_64 Linux**
  - Red Hat Enterprise Linux version 6.6及其之后的版本
  - Red Hat Enterprise Linux version 7.1及其之后的版本
  - SUSE Linux Enterprise Server version 11 SP4

  - SUSE Linux Enterprise Server version 12及其之后的版本
- **IBM Spectrum MPI 10.1.0.2 Eval for Power 8 Linux**
  - Red Hat Enterprise Linux version 7.3及其之后的版本

前提条件

已配置弹性云服务器免密登录。

操作步骤

步骤1 获取软件包。

1. 获取IBM Spectrum MPI软件包。

   下载地址：**https://www-01.ibm.com/marketing/iwm/iwm/web/preLogin.do?source=swerpsysz-lsf-3**

   获取的软件包一般为两个，包括license和软件两部分，例如，获取的IBM Spectrum MPI软件包为：

   smpi_lic_s-10.1Eval-rh7_Sep15.x86_64.rpm

   ibm_smpi-10.1.0.3eval_170901-rh7_Apr11.x86_64.rpm

2. 下载需要的HPC-X工具套件。

   IBM MPI在EDR SR-IOV场景下的运行需要HPC-X提供的MXM库的支持，HPC-X 的下载需要参考弹性云服务器操作系统的版本以及IB驱动版本，例如，下载的HPC-X版本为：hpcx-v2.0.0-gcc-MLNX_OFED_LINUX-4.2-1.2.0.0-redhat7.3-x86_64.tbz。

   下载地址：**https://developer.nvidia.com/networking/hpc-x**

步骤2 安装HPC-X工具套件。

1. 将**步骤1**中下载的HPC-X压缩包上传至运行MPI的弹性云服务器内。
2. 执行以下命令，解压HPC-X工具套件。

   $ tar xvf hpcx-v2.0.0-gcc-MLNX_OFED_LINUX-4.2-1.2.0.0-redhat7.3-x86_64.tbz
3. 执行以下命令，设置HPC-X环境变量。

   $ cd hpcx-v2.0.0-gcc-MLNX_OFED_LINUX-4.2-1.2.0.0-redhat7.3-x86_64

   $ export HPCX_HOME=$PWD

步骤3 安装IBM Spectrum MPI。

1. 将**步骤1**中下载的MPI软件包上传至运行MPI的弹性云服务器内。
2. 执行以下命令，切换至root用户。

   $ sudo su -
3. 执行以下命令，设置环境变量。
   - 如果选择自动接受IBM Spectrum MPI安装许可协议，执行以下命令：

     # export IBM_SPECTRUM_MPI_LICENSE_ACCEPT=yes
   - 如果选择手动接受IBM Spectrum MPI安装许可协议，执行以下命令：

     # export IBM_SPECTRUM_MPI_LICENSE_ACCEPT=no
4. 安装License部分。
   - 选择自动接受IBM Spectrum MPI安装许可协议的，执行以下命令：

 # rpm -ivh smpi_lic_s-10.1Eval-rh7_Sep15.x86_64.rpm

- 选择手动接受IBM Spectrum MPI安装许可协议的，执行以下命令：

  # rpm -ivh ibm_smpi_lic_s-10.1Eval-rh7_Sep15.x86_64.rpm

  图 2-18 手动接受 IBM Spectrum MPI 安装协议

```
[root@host-192-168-0-75 ~]# rpm -ivh ibm_smpi_lic_s-10.1Eval-rh7_Aug11.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:ibm_smpi_lic_s-10.1Eval-rh7_Aug11################################# [100%]
Running License acceptance script...
IBM Spectrum MPI  License acceptance script complete...return code:  0
```

  并根据界面提示，执行以下命令，运行脚本：

  # sh /opt/ibm/spectrum_mpi/lap_se/bin/accept_spectrum_mpi_license.sh

5. 执行以下命令，安装软件部分。

   # rpm -ivh ibm_smpi-10.1.0.3eval_170901-rh7_Apr11.x86_64.rpm

   图 2-19 安装软件

```
[root@host-192-168-0-75 ~]# rpm -ivh ibm_smpi-10.01.01.0Eval-rh7_Aug11.x86_64.rpm
Preparing...                          ################################# [100%]
Updating / installing...
   1:ibm_smpi-10.01.01.0Eval-rh7_Aug11################################# [100%]
```

步骤4 配置MPI环境变量。

1. 默认情况下，Spectrum MPI会安装至“/opt/ibm/spectrum_mpi”目录。该场景下需要设置如下环境变量：

   export MPI_ROOT=/opt/ibm/spectrum_mpi

   export LD_LIBRARY_PATH=$MPI_ROOT/lib:$LD_LIBRARY_PATH

   export PATH=$MPI_ROOT/bin:$PATH

   export MANPATH=$MPI_ROOT/share/man:$MANPATH

   unset MPI_REMSH

2. 执行以下命令，查看环境变量导入是否成功。

   # which mpirun

   图 2-20 查看环境变量

```
[root@host-192-168-0-75 ~]# which mpirun
/opt/ibm/spectrum_mpi/bin/mpirun
```

步骤5 执行以下命令，在单个弹性云服务器上通过Spectrum MPI运行可执行文件。

1. 执行以下命令，编辑文件。

   # cd

   # vi hello.c

   编辑内容如下：

```
#include<mpi.h>
#include<stdio.h>
int main(intargc, char**argv){
//Initialize the MPI environment
MPI_Init(NULL, NULL);
```



```
//Get the number of processes
int world_size;
MPI_Comm_size(MPI_COMM_WORLD, &world_size);
//Get the rank of the process
int world_rank;
MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
//Get the name of the processor
char processor_name[MPI_MAX_PROCESSOR_NAME];
int name_len;
MPI_Get_processor_name(processor_name, &name_len);
//Print off a hello world message
printf("Hello world from processor %s, rank %d"" out of %d processors\n",      processor_name,
world_rank, world_size);
//FinalizetheMPIenvironment.
MPI_Finalize();
}
```

2. 执行以下命令，生成可执行文件，假设在/home/linux目录下生成。

   # mpicc hello.c -o spe_hello

   说明

   不同版本的MPI运行的hello文件是不同的，因此，如果更换了MPI版本，需要重新执行命令**# mpicc hello.c -o spe_hello**进行编译，生成新的可执行文件。

3. 执行以下命令，在单个ECS上运行Spectrum MPI。

   # mpirun --allow-run-as-root -np 2 /root/spe_hello

   系统回显如图2-21所示。

图 2-21 运行结果

```
[root@host-192-168-0-75 ~]# mpirun --allow-run-as-root -np 2 spe_hello
Hello world from processor host-192-168-0-75, rank 0 out of 2 processors
Hello world from processor host-192-168-0-75, rank 1 out of 2 processors
```

----结束

2.3.5 Intel MPI

操作场景

本节指导用户在ECS上安装和使用Intel MPI应用（以版本l_mpi_2018.0.128为例）。

前提条件

已配置弹性云服务器免密登录。

操作步骤

步骤1 安装Intel MPI。

1. 下载Intel MPI。

   下载地址：**https://software.intel.com/en-us/intel-mpi-library**

2. 执行以下命令，解压并安装Intel MPI。

   以l_mpi_2018.0.128.tgz为例：

   # tar -xvf l_mpi_2018.0.128.tgz

   # cd l_mpi_2018.0.128/

   # ./install.sh

图 2-22 Intel MPI 安装成功

```
Thank you for installing Intel(R) MPI Library 2017 Update 3 for Linux*.

If you have not done so already, please register your product with Intel
Registration Center to create your support account and take full advantage of
your product purchase.

Your support account gives you access to free product updates and upgrades
as well as Priority Customer support at the Online Service Center
https://supporttickets.intel.com.

Click here https://software.intel.com/en-us/python-distribution
to download Intel(R) Distribution for Python*
This download will initiate separately. You can proceed with the installation
screen instructions.
```

步骤2 配置环境变量。

1. 普通用户下，在“~/.bashrc”中添加如下语句：
   export PATH=$PATH:/opt/intel/impi/2018.0.128/bin64
   export LD_LIBRARY_PATH=/opt/intel/impi/2018.0.128/lib64
2. 执行下列命令，导入环境变量。
   # source ~/.bashrc

步骤3 执行下列命令，查看是否导入成功。

 # which mpirun

图 2-23 环境变量导入成功

```
[root@host-192-168-0-75 ~]# which mpirun
/opt/intel/impi/2018.0.128/bin64/mpirun
```

回显结果如图2-23所示，表示环境变量导入成功。

步骤4 执行以下命令，在单个ECS上运行Intel MPI。

1. 执行以下命令，重新生成可执行文件。
   # cd
   # mpicc hello.c -o intel_hello
2. 执行以下命令，在单个ECS上运行Intel MPI。
   # mpirun -np 2 /root/intel_hello

图 2-24 在单个 ECS 上运行 Intel MPI

```
[root@host-192-168-0-75 ~]# mpirun -np 2 /root/intel_hello
Hello world from processor host-192-168-0-75, rank 1 out of 2 processors
Hello world from processor host-192-168-0-75, rank 0 out of 2 processors
```

----结束

2.3.6 Platform MPI

操作场景

本节指导用户在ECS上安装和使用Platform MPI应用（以版本platform_mpi-09.01.04.03r-ce为例）。

前提条件

已配置弹性云服务器免密登录。

操作步骤

步骤1 安装Platform MPI。

1. 执行以下命令，安装需要的库文件。
   # yum install glibc.i686 libgcc-4.8.5-11.el7.i686
2. 增加执行权限，例如安装包所在路径为/root。
   # cd /root && chmod +x platform_mpi- 09.01.04.03r-ce.bin
3. 执行以下命令，安装Platform MPI。
   # ./platform_mpi- 09.01.04.03r-ce.bin
   按照提示输入Enter或1（accept the agreement）直到安装完成，以下为安装成功界面。

   图 2-25 Platform MPI 安装成功

   ```
   Installation Complete
   ---------------------

   Congratulations. IBM_PlatformMPI has been successfully installed to:

      /opt/ibm/platform_mpi

   PRESS <ENTER> TO EXIT THE INSTALLER:
   ```

   默认安装路径为/opt/ibm/platform_mpi。

步骤2 配置MPI环境变量。

1. 执行以下命令，获取pkey。
   # cat /sys/class/infiniband/mlx5_0/ports/1/pkeys/* | grep -v 0000

   图 2-26 查询 pkey 值

   ```
   [root@host-192-168-0-75 ~]# cat /sys/class/infiniband/mlx5_0/ports/1/pkeys/* | grep -v 0000
   0x8c2b
   0x7fff
   ```

2. 普通用户下，在~/.bashrc中添加如下语句：
   export MPI_ROOT=/opt/ibm/platform_mpi
   export PATH=$MPI_ROOT/bin:$PATH
   export LD_LIBRARY_PATH=/opt/ibm/platform_mpi/lib/linux_amd64
   export MPI_IB_PKEY=*步骤2.1中获取的pkey*
   $source ~/.bashrc

   说明

   如果存在多个pkey，使用英文逗号隔开。

3. 执行以下命令，检查环境变量是否配置成功。
   # which mpirun

   图 2-27 Platform MPI 环境变量导入成功

   ```
   [root@host-192-168-0-75 ~]# which mpirun
   /opt/ibm/platform_mpi/bin/mpirun
   ```

步骤3 执行下列命令，在单个ECS上运行Platform MPI。

1. 执行以下命令，重新编译hello.c文件。

   # mpicc hello.c -o platform_hello

2. 执行以下命令，在单个ECS上运行Platform MPI。

   # mpirun -np 2 /root/platform_hello

   图 2-28 在单个 ECS 上运行 Platform MPI

```
[root@host-192-168-0-75 ~]# mpirun -np 2 platform_hello
Hello world from processor host-192-168-0-75, rank 1 out of 2 processors
Hello world from processor host-192-168-0-75, rank 0 out of 2 processors
```

----结束

2.4 制作私有镜像

操作场景

对于已完成HPC配置的弹性云服务器，您可以将其作为模板制作私有镜像，便于快速创建集群。该任务指导用户怎样将Linux弹性云服务器转化为私有镜像，包括管理控制台方式和基于HTTPS请求的API（Application programming interface）方式。

前提条件

- 已确保Linux云服务器网卡设置为DHCP的方式动态获取网络地址。
- 已清理Linux云服务器中的udev配置规则。
- 已安装并配置Cloud-init工具。
- 已卸载Linux云服务器中挂载的所有数据盘。

控制台方式

1. 登录管理控制台。
2. 选择“计算 > 弹性云服务器”。

   进入弹性云服务器信息页面。
3. 在弹性云服务器列表页，选择待制作镜像的云服务器，确认云服务器为“关机”状态。

   如果云服务器为“开机”状态，您可通过单击“操作”列下的“更多 > 关机”将云服务器关机。
4. 单击“操作”列下的“更多 > 制作镜像”，将弹性云服务器制作为私有镜像。
5. 根据界面提示，填写镜像的基本信息。
   - 源：云服务器
   - 弹性云服务器：保持系统默认值
   - 名称：用户自定义镜像名称。
6. 单击“立即申请”。

   系统将自动跳转至镜像服务页面，您可以在此页面查看新创建的私有镜像。

API 方式

- URI

  POST /v2/cloudimages/action

- 请求样例

```
POST /v2/cloudimages/action
{
    "name": "ims_test",
    "description": "云服务器制作镜像",
    "instance_id": "877a2cda-ba63-4e1e-b95f-e67e48b6129a"
}
```

- 响应样例

```
STATUS CODE 200
{
    "job_id": "8a12fc664fb4daa3014fb4e581380005"
}
```

2.5 创建应用集群

操作场景

您可以在几分钟之内，批量创建多台弹性云服务器。该任务指导用户使用制作的私有镜像创建应用集群，包括管理控制台方式和基于HTTPS请求的API（Application programming interface）方式。

管理控制台方式

1. 登录管理控制台。
2. 选择“计算 > 弹性云服务器”。

   进入弹性云服务器信息页面。
3. 单击“创建弹性云服务器”，开始创建应用集群。
4. 按照界面提示，填写弹性云服务器的参数配置，详细操作请参见**创建支持IB网卡的弹性云服务器**。其中，
   - 规格：需与转化的私有镜像的云服务器规格保持一致。
   - 镜像：选择“私有镜像”，然后选择**制作私有镜像**中制作的私有镜像。
   - 虚拟私有云：集群中的所有弹性云服务器需在同一VPC、同一子网内。
   - 弹性IP：选择暂不购买。

     说明

     一个集群绑定一个EIP即可。因此，可以在创建应用集群后，再绑定EIP。
   - 购买数量：待创建集群中弹性云服务器的数量。
5. 单击“立即购买”。
6. 在确认规格页面，您可以查看详情并提交申请。

   如果确认信息无误，单击“提交订单”。

   使用私有镜像创建的弹性云服务器创建成功后，您可以在弹性云服务器列表页查看详情，这些云服务器将被用作HPC集群。

API 方式

以创建H2型ECS集群为例：

- URI

  POST /v1/{$tenant_id}/cloudservers

- 请求样例

  假设需要批量创建4台弹性云服务器，则修改count值为4，请求样例如下：

```
curl -i -k -H 'Accept:application/json;charset=utf8' -H 'Content-Type:application/json' -H 'X-Auth-Token:$TOKEN' -d '{"server":{"availability_zone":"eu-de-01","name":"h2_cluster_vm","imageRef":"7474de73-9618-4c6a-afaa-df60df57c9b9","flavorRef":"h2.3xlarge.10","root_volume":{"volumetype":"SATA","size":40},"vpcid":"97701dc4-bfd3-4021-8b89-044486c8b317","nics":[{"subnet_id":"6712fc43-a196-4973-8b5e-5e4763f6449b"}],"personality":[],"count":4,"adminPass":"Test@123"}}' -X POST https://46.29.103.37:443/v1/240bb6c5e42849669fc49933c185232b/cloudserver
```

  说明

  如果集群中的每台弹性云服务器都需要绑定EIP，则需要创建多个EIP，并绑定在弹性云服务器上。具体操作请参见**创建支持IB网卡的弹性云服务器**的“API方式”。

2.6 配置 ECS 集群互相免密登录

操作场景

该任务指导用户在ECS集群上进行相关设置，并使其可以相互免密登录。仅支持在使用密钥登录弹性云服务器的情况下配置ECS集群互相免密登录。

背景信息

$：表示在普通用户下，执行相关操作。

#：表示在管理员用户下，执行相关操作。

普通用户切换至管理员用户，请使用命令**sudo su**。

前提条件

已成功创建ECS集群，并绑定了弹性IP进行登录。

操作步骤

1. 使用“PuTTY”，采用密钥对方式登录集群中任意一台ECS。
2. 执行以下命令，防止系统超时退出。

   # TMOUT=0
3. 执行以下命令，添加集群中所有主机的私网IP地址和主机名。

   # vi /etc/hosts

   添加的内容为集群中所有ECS的私网IP地址和主机名，例如：

   192.168.0.1 ecs-ff-0001

   192.168.0.2 ecs-ff-0002

   ..
4. 执行以下命令，ssh方式登录本节点，验证是否可以不输入密码登录ECS。其中hostname1为本主机名。

   $ ssh localhost

   $ ssh *hostname1*

5. 依次登录集群中其他ECS，重复执行步骤1～4。
6. 执行以下命令，验证参加测试的ECS之间是否可以免密码互相登录。

   假设集群中有2个弹性云服务器，另一个云服务器的主机名为hostname2，则命令行为：

   $ ssh *用户名@SERVER_IP*

   $ ssh *hostname2*

2.7 在 HPC 集群上运行 MPI 应用

2.7.1 在 HPC 集群上运行 IB 驱动自带的 OpenMPI

操作场景

该任务指导用户在已配置好的弹性云服务器上，运行IB驱动自带的MPI应用（3.0.0rc6版本）。

前提条件

- 已成功创建带IB网卡的弹性云服务器，并绑定了弹性IP进行登录。
- 已使用私有镜像创建多个弹性云服务器。

操作步骤

1. 使用“PuTTY”，采用密钥对方式登录弹性云服务器。

   登录用户为创建弹性云服务器时指定的用户名。
2. 执行以下命令，防止系统超时退出。

   # TMOUT=0
3. 执行以下命令，验证参加测试的弹性云服务器之间是否可以免密码互相登录。

   $ ssh *用户名@SERVER_IP*
4. 执行以下命令，关闭弹性云服务器的防火墙。

   # iptables -F

   # service firewalld stop
5. 执行以下命令，退出root权限。

   # exit
6. 执行以下命令，添加hostfile文件。

   # vi hostfile

   添加的内容为弹性云服务器的IP地址或者主机名（主机名需要在/etc/hosts中有对应IP地址信息），例如

   # cat hostfile

   192.168.0.1

   192.168.0.2

   ...
7. 执行以下命令，在集群中运行hostname命令。

 # mpirun --allow-run-as-root -np <hostfile_node_number> -pernode --hostfile hostfile hostname

图 2-29 在集群中运行 hostname 命令

```
[root@dyna-0003 ~]# mpirun --allow-run-as-root -np 2 -pernode --hostfile /root/hostfile hostname
dyna-0003
dyna-0002
```

8. 修改hostfile，运行MPI benchmark，运行时指定hostfile文件路径。

   以两个弹性云服务器为例：

   # mpirun --allow-run-as-root -np 2 -pernode --hostfile /root/hostfile /usr/mpi/gcc/openmpi-3.0.0rc6/tests/imb/IMB-MPI1 PingPong

   在两个节点的集群间运行Intel MPI benchmark，在RDMA网络下，最低时延应该在1.5 us（microseconds）以内：

```
#------------------------------------------------------------
#    Intel (R) MPI Benchmarks 4.1, MPI-1 part
#------------------------------------------------------------
# Date                  : Mon Jul 16 10:12:51 2018
# Machine               : x86_64
# System                : Linux
# Release               : 3.10.0-514.10.2.el7.x86_64
# Version               : #1 SMP Fri Mar 3 00:04:05 UTC 2017
# MPI Version           : 3.1
# MPI Thread Environment:

# New default behavior from Version 3.2 on:

# the number of iterations per message size is cut down
# dynamically when a certain run time (per message size sample)
# is expected to be exceeded. Time limit is defined by variable
# "SECS_PER_SAMPLE" (=> IMB_settings.h)
# or through the flag => -time

# Calling sequence was:

# /usr/mpi/gcc/openmpi-3.0.0rc6/tests/imb/IMB-MPI1 PingPong

# Minimum message length in bytes:   0
# Maximum message length in bytes:   4194304
#
# MPI_Datatype                 :   MPI_BYTE
# MPI_Datatype for reductions    :   MPI_FLOAT
# MPI_Op                       :   MPI_SUM
#
#

# List of Benchmarks to run:

# PingPong

#---------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#---------------------------------------------------
#bytes #repetitions      t[usec]   Mbytes/sec
0         1000         1.87         0.00
1         1000         1.93         0.49
2         1000         1.78         1.07
4         1000         1.79         2.13
8         1000         1.77         4.31
16        1000         1.78         8.57
32        1000         1.79        17.09
64        1000         1.85        33.02
128       1000         1.90        64.12
256       1000         2.40       101.58
```

```
512         1000         2.53       192.90
1024         1000         2.85       342.61
2048         1000         3.23       604.14
4096         1000         4.32       904.98
8192         1000         5.89      1325.65
16384         1000         8.48      1842.47
32768         1000        12.50      2500.57
65536          640        21.79      2867.89
131072          320        34.28      3646.50
262144          160        42.19      5925.52
524288           80        66.55      7513.14
1048576          40       114.95      8699.54
2097152          20       213.71      9358.48
4194304          10       402.59      9935.78


# All processes entering MPI_Finalize
```

9. 在Linux集群上部署您自己的MPI应用，并参考上述MPI运行方法，在Linux集群中运行MPI程序。

2.7.2 在 HPC 集群上运行社区 OpenMPI

操作场景

该任务指导用户在已配置好的弹性云服务器上，运行社区MPI应用（3.1.1版本）。

前提条件

- 已成功创建带IB网卡的弹性云服务器，并绑定了弹性IP进行登录。
- 已使用私有镜像创建多个弹性云服务器。

操作步骤

1. 使用“PuTTY”，采用密钥对方式登录弹性云服务器。
   登录用户为创建弹性云服务器时指定的用户名。
2. 执行以下命令，防止系统超时退出。
   # TMOUT=0
3. 执行以下命令，验证参加测试的弹性云服务器之间是否可以免密码互相登录。
   $ **ssh *用户名@SERVER_IP***
4. 执行以下命令，关闭弹性云服务器的防火墙。
   # iptables -F
   # service firewalld stop
5. 执行以下命令，给参与测试的弹性云服务器配置主机名。
   # hostnamectl set-hostname vm1
6. 执行以下命令，添加“/etc/hosts”文件。
   # vi /etc/hosts
   添加的内容为弹性云服务器的主机名及IP地址，例如
   #cat /etc/hosts
   192.168.1.3 vm1
   192.168.1.4 vm2
   ...

7. 执行以下命令，添加hostfile文件。

   # vi hostfile

   添加的内容为集群中参与测试的弹性云服务器的主机名。

   vm1

   vm2

   ...

8. 修改hostfile，运行MPI benchmark，运行时指定hostfile文件路径。

   以两个弹性云服务器为例：

   # mpirun --allow-run-as-root -np 2 --pernode -hostfile /root/hostfile /usr/mpi/gcc/openmpi-3.0.0rc6/tests/imb/IMB-MPI1 PingPong

   在两个节点的集群间运行Intel MPI benchmark，在RDMA网络下，最低时延应该在1.5 us（microseconds）以内：

```
#------------------------------------------------------------
#    Intel (R) MPI Benchmarks 4.1, MPI-1 part
#------------------------------------------------------------
# Date                  : Mon Jul 16 09:42:15 2018
# Machine               : x86_64
# System                : Linux
# Release               : 3.10.0-514.10.2.el7.x86_64
# Version               : #1 SMP Fri Mar 3 00:04:05 UTC 2017
# MPI Version           : 3.1
# MPI Thread Environment:

# New default behavior from Version 3.2 on:

# the number of iterations per message size is cut down
# dynamically when a certain run time (per message size sample)
# is expected to be exceeded. Time limit is defined by variable
# "SECS_PER_SAMPLE" (=> IMB_settings.h)
# or through the flag => -time

# Calling sequence was:

# /usr/mpi/gcc/openmpi-3.0.0rc6/tests/imb/IMB-MPI1 PingPong

# Minimum message length in bytes:   0
# Maximum message length in bytes:   4194304
#
# MPI_Datatype                  :   MPI_BYTE
# MPI_Datatype for reductions    :   MPI_FLOAT
# MPI_Op                        :   MPI_SUM
#
#

# List of Benchmarks to run:

# PingPong

#---------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#---------------------------------------------------
#bytes #repetitions      t[usec]   Mbytes/sec
0         1000         1.75         0.00
1         1000         1.75         0.55
2         1000         1.74         1.10
4         1000         1.74         2.19
8         1000         1.77         4.31
16         1000         1.79         8.54
32         1000         1.77        17.26
64         1000         1.85        33.02
128         1000         1.89        64.45
```

```
256         1000         2.39       102.29
512         1000         2.54       192.56
1024         1000         2.81       346.99
2048         1000         3.24       603.08
4096         1000         4.30       907.66
8192         1000         5.91      1321.23
16384         1000         8.61      1814.29
32768         1000        12.31      2537.83
65536          640        21.80      2867.15
131072          320        33.91      3686.23
262144          160        42.65      5861.95
524288           80        68.61      7287.12
1048576          40       120.06      8329.50
2097152          20       221.55      9027.12
4194304          10       424.35      9426.16


# All processes entering MPI_Finalize
```

9. 在Linux集群上部署您自己的MPI 应用，并参考上述MPI运行方法，在Linux 集群中运行MPI 程序。

2.7.3 在 HPC 集群上运行 Spectrum MPI

操作场景

该任务指导用户在已配置好的弹性云服务器上，运行Spectrum MPI应用（ IBM Spectrum MPI v10.1 ）。

前提条件

- 已成功创建带IB网卡的弹性云服务器，并绑定了弹性IP进行登录。
- 已使用私有镜像创建多个弹性云服务器。

操作步骤

1. 使用“PuTTY”，采用密钥对方式登录弹性云服务器。

   登录用户为创建弹性云服务器时指定的用户名。
2. 执行以下命令，防止系统超时退出。

   # TMOUT=0
3. 执行以下命令，验证参加测试的弹性云服务器之间是否可以免密码互相登录。

   $ **ssh *用户名@SERVER_IP***
4. 执行以下命令，关闭弹性云服务器的防火墙。

   # iptables -F

   # service firewalld stop
5. 执行以下命令，用“IP:Number”的形式作为MPI集群运行程序时的hostlist参数，在集群上通过Spectrum MPI运行可执行文件。其中，
   - IP代表集群中的弹性云服务器IP地址。
   - Number代表该弹性云服务器的任务数。

   假设集群中共有两个弹性云服务器，主机名分别是host-192-168-0-27和host-192-168-0-75，可执行程序的目录为/root/spe_hello，文件名为spe_hello，则命令行如下：

   # mpirun --allow-run-as-root -np 2 -hostlist host-192-168-0-27,host-192-168-0-75 /root/spe_hello

图 2-30 在集群上通过 Spectrum MPI 运行可执行文件

```
[root@host-192-168-0-75 ~]# mpirun --allow-run-as-root -np 2 -hostlist host-192-168-0-27,host-192-168-0-75
 /root/spe_hello
Hello world from processor host-192-168-0-75, rank 1 out of 2 processors
Hello world from processor host-192-168-0-27, rank 0 out of 2 processors
```

说明

hostfile文件在运行时需要指定路径，可执行文件hello路径需为绝对路径，集群中所有可执行文件在同一路径下。

2.7.4 在 HPC 集群上运行 Intel MPI

操作场景

该任务指导用户在ECS集群（以CentOS7.3为例）上运行Intel MPI应用（l_mpi_2018.0.128版本）。

前提条件

- 已成功创建带IB网卡的弹性云服务器，并绑定了弹性IP进行登录。
- 已使用私有镜像创建多个弹性云服务器。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台ECS。
2. 执行以下命令，关闭ECS防火墙。

   # systemctl stop firewalld.service
3. 执行以下命令，查看防火墙是否关闭成功。

   # systemctl status firewalld.service

   图 2-31 成功关闭防火墙

   ```
   [root@host-192-168-0-75 ~]# systemctl status firewalld.service
   ● firewalld.service - firewalld - dynamic firewall daemon
      Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
      Active: inactive (dead)
        Docs: man:firewalld(1)
   ```
4. 依次登录集群中所有ECS，重复执行**步骤1.1**～**步骤1.3**，关闭所有ECS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台ECS。
2. 执行以下命令，查看ECS的主机名。

   # hostname

   图 2-32 查看主机名

   ```
   [root@host-192-168-0-75 ~]# hostname
   host-192-168-0-75
   ```
3. 依次登录集群中所有ECS，重复执行**步骤2.1**～**步骤2.2**，获取所有ECS的主机名。
4. 登录集群中任意一台ECS。

5. 执行以下命令，添加hosts配置文件。

   # vi /etc/hosts

   添加的内容为集群中所有ECS的私网IP和主机名，例如：

   192.168.0.1 host-192-168-0-1

   192.168.0.2 host-192-168-0-2

   ...

6. 执行以下命令，添加hostfile文件。

   # vi hostfile

   添加集群中所有ECS的主机名，例如：

   host-192-168-0-1

   host-192-168-0-2

   ...

7. 依次登录集群中所有ECS，重复执行**步骤2.4**～**步骤2.6**。

步骤3 配置IB网卡的IP地址。

1. 对集群中所有的ECS，执行以下命令，为IB驱动配置IP地址。

   # ifconfig ib0 *192.168.23.34*/24

   # ifconfig ib0 *192.168.23.35*/24

   ...

   说明

   IP地址可随意指定，但需要在同一网段内。

2. 在ECS中使用以下命令，验证连通性。

   # ping *192.168.23.35*

步骤4 执行以下命令，在ECS集群运行Intel MPI，。

以两台ECS为例：

 # mpirun -perhost 2 -machinefile hostfile -np 12 /root/intel_hello

说明

hostfile文件在运行时需要指定路径，可执行文件hello路径需为绝对路径，集群中所有可执行文件在同一路径下。

图 2-33 集群上运行 Intel MPI 成功

```
[root@host-192-168-0-75 ~]# mpirun -perhost 2 -machinefile hostfile -np 12 /root/intel_hello
Hello world from processor host-192-168-0-75, rank 0 out of 12 processors
Hello world from processor host-192-168-0-75, rank 2 out of 12 processors
Hello world from processor host-192-168-0-75, rank 4 out of 12 processors
Hello world from processor host-192-168-0-75, rank 6 out of 12 processors
Hello world from processor host-192-168-0-75, rank 8 out of 12 processors
Hello world from processor host-192-168-0-75, rank 10 out of 12 processors
Hello world from processor host-192-168-0-27, rank 1 out of 12 processors
Hello world from processor host-192-168-0-27, rank 3 out of 12 processors
Hello world from processor host-192-168-0-27, rank 5 out of 12 processors
Hello world from processor host-192-168-0-27, rank 7 out of 12 processors
Hello world from processor host-192-168-0-27, rank 9 out of 12 processors
Hello world from processor host-192-168-0-27, rank 11 out of 12 processors
```

----结束

2.7.5 在 HPC 集群上运行 Platform MPI

操作场景

该任务指导用户在ECS集群（以CentOS7.3为例）上运行Platform MPI应用（以版本platform_mpi-09.01.04.03r-ce为例）。

前提条件

- 已成功创建带IB网卡的弹性云服务器，并绑定了弹性IP进行登录。
- 已使用私有镜像创建多个弹性云服务器。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台ECS。
2. 执行以下命令，关闭ECS防火墙。
   # systemctl stop firewalld.service
3. 执行以下命令，查看防火墙是否关闭成功。
   # systemctl status firewalld.service

   图 2-34 关闭防火墙成功

   ```
   [root@host-192-168-0-75 ~]# systemctl status firewalld.service
   ● firewalld.service - firewalld - dynamic firewall daemon
      Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
      Active: inactive (dead)
        Docs: man:firewalld(1)
   [root@host-192-168-0-75 ~]#
   ```

4. 依次登录集群中所有ECS，重复执行**步骤1.1**～**步骤1.3**，关闭所有ECS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台ECS。
2. 执行以下命令，查看ECS的主机名。
   # hostname

   图 2-35 查看 ECS 的主机名

   ```
   [root@host-192-168-0-75 ~]# hostname
   host-192-168-0-75
   ```

3. 依次登录集群中所有ECS，重复执行**步骤2.1**～**步骤2.2**，获取所有ECS的主机名。
4. 登录集群中任意一台ECS。
5. 执行以下命令，添加hosts配置文件。
   # vi /etc/hosts
   添加的内容为集群中所有ECS的私网IP和主机名，例如：
   192.168.0.1 host-192-168-0-1
   192.168.0.2 host-192-168-0-2
   ...
6. 执行以下命令，添加hostfile文件。

$vi hostfile

添加集群中所有ECS的主机名，例如：

host-192-168-0-1

host-192-168-0-1

...

7. 依次登录集群中所有ECS，重复执行**步骤2.4**～**步骤2.6**。

步骤3 配置IB网卡的IP地址。

1. 对集群中所有的ECS，执行以下命令，为IB驱动配置IP地址。

   # ifconfig ib0 *192.168.23.34*/24

   # ifconfig ib0 *192.168.23.35*/24

   ...

   说明

   IP地址可随意指定，但需要在同一网段内。

2. 在ECS中使用以下命令，验证连通性。

   # ping *192.168.23.35*

步骤4 执行以下命令，在ECS集群运行Platform MPI。

以两台ECS为例：

 # mpirun -perhost 2 -np 12 -machinefile hostfile /root/platform_hello

说明

hostfile文件在运行时需要指定路径，可执行文件hello路径需为绝对路径，集群中所有可执行文件在同一路径下。

图 2-36 集群上运行 Platform MPI 成功

```
[root@host-192-168-0-75 ~]# mpirun -np 12 -machinefile hostfile /root/platform_hello
Hello world from processor host-192-168-0-75, rank 0 out of 12 processors
Hello world from processor host-192-168-0-75, rank 4 out of 12 processors
Hello world from processor host-192-168-0-75, rank 10 out of 12 processors
Hello world from processor host-192-168-0-75, rank 6 out of 12 processors
Hello world from processor host-192-168-0-75, rank 8 out of 12 processors
Hello world from processor host-192-168-0-75, rank 2 out of 12 processors
Hello world from processor host-192-168-0-27, rank 1 out of 12 processors
Hello world from processor host-192-168-0-27, rank 11 out of 12 processors
Hello world from processor host-192-168-0-27, rank 3 out of 12 processors
Hello world from processor host-192-168-0-27, rank 9 out of 12 processors
Hello world from processor host-192-168-0-27, rank 5 out of 12 processors
Hello world from processor host-192-168-0-27, rank 7 out of 12 processors
[root@host-192-168-0-75 ~]#
```

----结束

3 弹性云服务器场景最佳实践

3.1 HPC 断点续算计算方案

操作场景

在HPC领域很多应用本身是支持断点续算功能的，例如LAMMPS， GROMACS。同时HPC常用的调度软件也对断点续算有集成支持，如PBS、Slurm与LSF等。

本节以LAMMPS为例，介绍如何在HPC进行断点续算。

步骤 1 安装 FFTW

依次执行以下命令，安装FFTW软件。

yum install gcc-gfortran gcc-c++

wget http://www.fftw.org/fftw-3.3.8.tar.gz

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/mpi/gcc/*openmpi-2.1.2a1*/lib64/

export PATH=/usr/mpi/gcc/*openmpi-2.1.2a1*/bin:$PATH

tar -zxvf fftw-3.3.8.tar.gz

cd fftw-3.3.8/

./configure --prefix=/opt/fftw CC=gcc MPICC=mpicc --enable-mpi --enable-openmp --enable-threads --enable-avx --enable-shared

make && make install

步骤 2 安装 lammps

1. 依次执行以下命令，安装lammps。

   yum install libjpeg-*

   yum install libpng12-*

   wget https://lammps.sandia.gov/tars/lammps-2Aug18.tar.gz

tar -zxvf lammps-2Aug18.tar.gz

cd lammps-2Aug18/src

vi MAKE/Makefile.mpi

2. 根据图3-1、图3-2，修改红框标识中的内容。其中，版本号需根据实际情况进行填写。

须知

只修改图3-1、图3-2中红框标识的内容。

图 3-1 修改 Makefile 文件 01

```
# ---------------------------------------------------------------------
# compiler/linker settings
# specify flags and libraries needed for your compiler

CC =          mpicc
CCFLAGS =     -g -O3
SHFLAGS =     -fPIC
DEPFLAGS =    -M

LINK =        mpic++
LINKFLAGS =   -g -O
LIB =
SIZE =        size

ARCHIVE =     ar
ARFLAGS =     -rc
SHLIBFLAGS =  -shared

# ---------------------------------------------------------------------
# LAMMPS-specific settings, all OPTIONAL
# specify settings for LAMMPS features you will use
# if you change any -D setting, do full re-compile after "make clean"

# LAMMPS ifdef settings
# see possible settings in Section 2.2 (step 4) of manual

LMP_INC =     -DLAMMPS_GZIP -DLAMMPS_MEMALIGN=64

# MPI library
# see discussion in Section 2.2 (step 5) of manual
# MPI wrapper compiler/linker can provide this info
# can point to dummy MPI library in src/STUBS as in Makefile.serial
# use -D MPICH and OMPI settings in INC to avoid C++ lib conflicts
# INC = path for mpi.h, MPI compiler settings
# PATH = path for MPI library
# LIB = name of MPI library

MPI_INC =     -DMPICH_SKIP_MPICXX -DOMPI_SKIP_MPICXX=1i -I/usr/mpi/gcc/openmpi-2.1.2a1/include
MPI_PATH =  -L/usr/mpi/gcc/openmpi-2.1.2a1/lib64
MPI_LIB = -lmpi
```

图 3-2 修改 Makefile 文件 02

```
# PATH = path for FFT library
# LIB = name of FFT library

FFT_INC =     -DFFT_FFTW3 -I/opt/fftw/include
FFT_PATH =    -L/opt/fftw/lib
FFT_LIB =     -lfftw3

# JPEG and/or PNG library
# see discussion in Section 2.2 (step 7) of manual
# only needed if -DLAMMPS_JPEG or -DLAMMPS_PNG listed with LMP_INC
# INC = path(s) for jpeglib.h and/or png.h
# PATH = path(s) for JPEG library and/or PNG library
# LIB = name(s) of JPEG library and/or PNG library

JPG_INC =       -I/usr/include
JPG_PATH = -L/usr/lib64/
JPG_LIB = -ljpeg

# ---------------------------------------------------------------------
```

3. 执行以下命令，编译lammps，并将当前目录生成的lmp_mpi文件拷贝至“/share”目录。

   make mpi

步骤 3 配置 lammps

1. 算例输入文件。

   以melt为例，生成一个算例melt.in文件。设置每迭代100步生成一个checkpoint文件，假定该文件存放在共享目录“/share”中。内容如下：

   ```
   # 3d Lennard-Jones melt

   units           lj
   atom_style      atomic

   lattice         fcc 0.8442
   region          box block 0 20 0 20 0 20
   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 5 every 1

   fix             1 all nve
   dump 1 all xyz 100 /share/sample.xyz
   run             10000 every 100 "write_restart /share/lennard.restart"
   ```

2. 生成用于checkpoint续算的输入文件“melt.restart.in”，内容如下：

   ```
   # 3d Lennard-Jones melt

   read_restart  /share/lennard.restart
   run        10000 every 100 "write_restart /share/lennard.restart"
   ```

3. 生成pbs作业脚本“job.pbs”，内容如下：

   ```
   #!/bin/sh
   #PBS -l ncpus=2
   #PBS -o lammps_pbs.log
   #PBS -j oe

   export PATH=/usr/mpi/gcc/openmpi-2.1.2a1/bin:$PATH

   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/mpi/gcc/openmpi-2.1.2a1/lib64/module

   if [ ! -e "/share/lennard.restart" ]; then
     echo "run at the beginning"
     mpiexec --allow-run-as-root -np 2 /share/lmp_mpi -in /share/melt.in
   else
     echo "run from the last checkpoint"
     mpiexec --allow-run-as-root -np 2 /share/lmp_mpi -in /share/melt.restart.in
   fi
   ```

步骤 4 提交作业，且运行期间无中断

在不中断作业的情况下，提交作业并运行，查看作业运行时间。

1. 执行以下命令，提交作业。

   qsub job.pbs

2. 作业运行结束后，执行以下命令，查看作业信息。

qstat -f *作业ID*

如图3-3所示，可以看到作业一共运行了4分10秒。

图 3-3 运行作业不中断

```
Job_Name = job.pbs
Job_Owner = root@lammps-0001
resources_used.cpupercent = 176
resources_used.cput = 00:04:10
resources_used.mem = 1041048kb
resources_used.ncpus = 2
resources_used.vmem = 3277508kb
resources_used.walltime = 00:02:10
job_state = F
queue = workq
server = lammps-0001
Checkpoint = u
```

步骤 5 提交作业，模拟计算中断，使用断点续算模式完成计算

模拟作业中断情况：提交作业后，通过关机计算节点的方式手动中断作业，查看中断前、后作业运行时间。

1. 执行以下命令，提交作业。

   qsub job.pbs

2. 作业运行1分30秒左右，关机作业运行的计算节点，模拟算例释放场景。
3. 执行以下命令，查看关闭计算节点后的作业信息。

   qstat -f *作业ID*

   图 3-4 中断前作业信息

   ```
   Job Id: 32.lammps-0001
       Job_Name = job.pbs
       Job_Owner = root@lammps-0001
       resources_used.cpupercent = 165
       resources_used.cput = 00:01:30
       resources_used.mem = 208572kb
       resources_used.ncpus = 2
       resources_used.vmem = 3261456kb
   ```

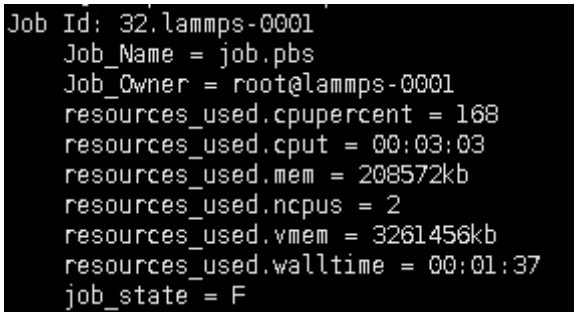
   此时，pbs作业回到queued状态，等待可用的计算资源。
4. 开机2中关闭的计算节点，提供可用的计算资源。

   此时，pbs作业会继续进行。
5. 作业执行完成后，执行以下命令，查看作业信息。

   qstat -f *作业ID*

   如图3-5所示，作业运行了3分03秒。由此可以看出，作业是从断点的位置进行续算的。

图 3-5 中断后运行作业信息



4 裸金属服务器场景典型应用

4.1 创建裸金属服务器集群

操作场景

若您需要将您的服务部署在裸金属服务器上，首先需要购买裸金属服务器。

购买须知

专属物理资源

- 如果您希望裸金属服务器运行在隔离的专属区域，请您先申请专属云，再创建裸金属服务器。
  了解和申请专属云，请参见《 专属云用户指南 》。
- 如果您希望裸金属服务器拥有独享的存储设备，请您在开通专属云后申请专属企业存储，再创建裸金属服务器。
  了解和申请专属企业存储，请参见《 专属企业存储用户指南 》。

操作步骤

1. 登录管理控制台。
2. 选择“计算 > 裸金属服务器”。
   进入裸金属服务器页面。
3. 单击“购买裸金属服务器”。
   进入购买页面。

   说明

   如果您已申请专属云，并希望将裸金属服务器部署在专属云中，则单击“在专属云中发放裸金属服务器”。
4. 确认“当前区域”。

   如果所在区域不正确，请单击页面左上角的 进行切换。
5. 选择“可用分区”。

可用分区指在同一地域下，电力、网络隔离的物理区域，可用分区之间内网互通，不同可用分区之间物理隔离。如果您需要提高应用的高可用性，建议您将裸金属服务器创建在不同的可用分区。

6. 选择“规格”。

   包括CPU、内存、本地磁盘和扩展配置。当您选择规格列表中的一个规格后，列表下方会展示该规格的名称、组网、用途等信息，以便您根据实际业务进行选择。

   说明

   规格中的内存、本地磁盘等配置为固定值，不可更改。

7. 选择“镜像”。

   - 公共镜像

     常见的标准操作系统镜像，所有用户可见，包括操作系统以及预装的公共应用。

     选择“公共镜像”，并展开“镜像”的下拉框，选择所需的公共镜像。

   - 私有镜像

     用户基于外部镜像创建的个人镜像，仅用户自己可见。包含操作系统、SDI卡驱动、bms-network-config网络配置程序、cloud-init初始化工具以及用户的私有应用。

     选择“私有镜像”，并展开“镜像”的下拉框，选择所需的私有镜像。

   - 共享镜像

     您将接受其他用户共享的私有镜像，作为自己的镜像进行使用。

     选择“共享镜像”，并展开“镜像”的下拉框，选择所需的共享镜像。

8. 选择“许可证类型”。

   在云平台上使用操作系统或软件的许可证类型。如果您选择的镜像为免费的，则系统不会展示该参数。如果您选择的镜像为计费镜像，此时系统会展示该参数。

   - 使用平台许可证

     使用云平台提供的许可证，申请许可证需要支付一定的费用。

   - 使用自带许可证（BYOL）

     使用用户已有操作系统的许可证，无需重新申请。

9. 设置“存储”。

   根据磁盘使用的存储资源是否独享，磁盘划分为“云硬盘”、“专属存储”和“专属企业存储”三类，您可以根据实际需求进行选择。当前创建时仅支持携一种类型的磁盘下发裸金属服务器。

   - 云硬盘：为裸金属服务器提供规格丰富、安全可靠、可弹性扩展的硬盘资源，满足不同性能要求的业务场景。

     如果未申请独享的存储池，请选择“云硬盘”页签，创建的磁盘使用公共存储资源。

   - 专属存储：为用户提供独享的存储资源，通过数据冗余和缓存加速等多项技术，提供高可用性和持久性，以及稳定的低时延性能。

     如果您在专属存储服务页面申请了存储池，可以选择“专属存储”页签，在已申请的存储池中创建磁盘。

   - 专属企业存储：提供专属的企业存储设备，支持企业关键业务（如Oracle RAC、SAP HANA TDI）上云。

如果您已经开通专属企业存储，可以选择“专属企业存储”页签，在专属企业存储中创建磁盘。

磁盘包括系统盘和数据盘。您可以为裸金属服务器添加多块数据盘，系统盘大小可以根据需要自定义。

说明

Windows裸金属服务器暂不支持挂载磁盘。

- 系统盘

  如果选择支持快速发放的规格，界面会提供系统盘配置项，可以根据需要设置磁盘类型和大小。

- 数据盘

  您可以为裸金属服务器添加多块数据盘，并设置数据盘的共享功能。

  - 裸金属服务器仅支持SCSI磁盘模式。
  - 共享盘：勾选后，数据盘为共享云硬盘。该共享盘可以同时挂载给多台裸金属服务器使用。

说明

- 购买包年/包月裸金属服务器时购买的系统盘，卸载后，只能继续作为系统盘使用，且只能挂载给原裸金属服务器。
- 购买包年/包月裸金属服务器时购买的非共享数据盘，卸载后，如果重新挂载，则只能挂载给原裸金属服务器作数据盘使用。
- 购买包年/包月裸金属服务器时购买的非共享数据盘，不支持单独续订、退订、开通自动续费、转按需付费方式、以及释放功能。

10. 设置“自动备份”。

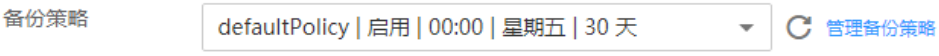
    启用自动备份功能后，系统会根据您设置的备份策略，自动备份裸金属服务器。

    a. 勾选“使用自动备份”。

    b. 设置“备份策略”。

       在下拉列表中选择备份策略，或单击“管理备份策略”，在云服务器备份页面进行设置。如果您未创建任何备份策略，但是勾选了“使用自动备份”，系统将提供默认的备份策略，如图4-1所示。

       图 4-1 默认备份策略

    更多关于裸金属服务器备份的信息，请参见《云服务器备份用户指南》。

11. 设置“网络”，包括“虚拟私有云”、“安全组”、“网卡”等信息。

表 4-1 网络参数说明

参数	解释
虚拟私有云	首次创建裸金属服务器时，如果未配置过VPC，系统将会为您创建默认VPC、安全组和子网。其中，默认VPC支持的地址范围为192.168.1.0/24，子网网关为192.168.1.1，并为子网开启DHCP功能。 如果默认VPC不能满足您对网络的需求，您可以单击“查看虚拟私有云”创建新的虚拟私有云。
安全组	安全组用来实现安全组内和安全组间裸金属服务器的访问控制，加强裸金属服务器的安全保护。用户可以在安全组中定义各种访问规则，当裸金属服务器加入该安全组后，即受到这些访问规则的保护。购买裸金属服务器时，支持选择裸金属服务器所在的安全组。目前，申请一台裸金属服务器时只能选择一个安全组。 **说明** 裸金属服务器初始化需要确保安全组出方向规则至少满足如下要求： ● 协议：TCP ● 端口范围：80 ● 远端地址：169.254.0.0/16 如果您使用的是默认安全组出方向规则，则已经包括了如上要求，可以正常初始化。默认安全组出方向规则为： ● 协议：Any ● 端口范围：Any ● 远端地址：0.0.0.0/16
网卡	裸金属服务器的网卡。系统默认有一个主网卡，并支持在所选VPC范围内自定义与网卡绑定的IP地址。您还可以为裸金属服务器额外增加扩展网卡。
高速网卡	高速网卡，与高速网络范围内的IP地址绑定，为裸金属服务器提供更高的带宽。一个裸金属服务器最多有两块高速网卡。您可以自定义与网卡绑定的IP地址。 **说明** 指定高速网卡时，不能批量创建裸金属服务器。
弹性IP	弹性IP是指将公网IP地址和路由网络中关联的裸金属服务器绑定，以实现VPC内的业务资源通过固定的公网IP地址对外提供访问服务。 您可以根据实际情况选择以下三种方式： ● 暂不购买：不使用弹性IP的裸金属服务器不能与互联网互通，仅可作为私有网络中部署业务或者集群的裸金属服务器使用。 ● 现在购买：自动为每台裸金属服务器分配独享带宽的弹性IP，带宽值由您设定。 ● 使用已有：为裸金属服务器分配已有的弹性IP。 **说明** 选择此种方式时，不能批量创建裸金属服务器。

参数	解释
规格	弹性IP选择“现在购买”时，需配置该参数。 ● 全动态BGP：可以根据设定的寻路协议实时自动优化网络结构，以保持客户使用的网络持续稳定、高效。 ● 静态BGP：网络结构发生变化时，无法实时自动调整网络设置以保障用户体验。
计费方式	弹性IP选择“现在购买”时，需配置该参数。 按带宽计费：指定带宽上限，按使用时间计费，与使用的流量无关。
带宽	弹性IP选择“现在购买”时，需配置该参数。 带宽大小，单位Mbit/s。

12. 设置裸金属服务器登录方式。
    - 密钥对

      指使用密钥对作为登录裸金属服务器的鉴权方式。您可以选择使用已有的密钥，或者单击“查看密钥对”创建新的密钥。

      说明

      如果选择使用已有的密钥，请确保您已在本地获取该文件，否则，将影响您正常登录裸金属服务器。

    - 密码

      指使用设置初始密码方式作为裸金属服务器的鉴权方式，此时，您可以通过用户名密码方式登录裸金属服务器。Linux操作系统时为root用户的初始密码，Windows操作系统时为Administrator用户的初始密码。密码复杂度需满足表4-2要求。

      说明

      Windows裸金属服务器不支持选择密码登录方式。

      表 4-2 密码规则

参数	规则	样例
密码	● 密码长度范围为8到26位。 ● 密码至少包含以下4种字符中的3种： – 大写字母 – 小写字母 – 数字 – 特殊字符，包括!@$%^-_=+[]{}:,./? ● 密码不能包含用户名或用户名的逆序。 ● Windows系统的裸金属服务器，不能包含用户名中超过两个连续字符的部分。	Test12$@

13. （可选）高级配置。

    如需使用“高级配置”中的功能，请单击“现在配置”。否则，请单击“暂不配置”。

    用户数据注入：主要用于创建裸金属服务器时向裸金属服务器注入用户数据。配置用户数据注入后，裸金属服务器首次启动时会自行注入数据信息。

14. 设置“裸金属服务器名称”。

    名称可自定义，如果同时购买多台裸金属服务器，系统会自动按序增加后缀。

15. 设置购买时长和数量。

    - 购买时长：选择“包年/包月”方式的用户需要设置购买时长，最短为1个月，最长为1年。

      说明

      包年/包月购买的裸金属服务器不能直接删除，仅支持资源退订操作。如果不再使用，可选择以下任意一种方式退订：

      - 在裸金属服务器列表选择待退订的裸金属服务器，单击操作列的“更多 > 退订”。在“退订资源”页面选择退订原因，单击“退订”。
      - 选择“费用中心 > 退订与变更 > 退订管理”，在列表中选择待退订的裸金属服务器，单击操作列的“退订资源”。

    - 购买数量：1~24台

      说明

      - 当配额充足时，一次最多可以购买24台裸金属服务器；若配额不足24台，一次最多可购买数量为配额余量。
      - 在设置网络信息时，若您选择自定义网卡或高速网卡的IP地址，或选择已有弹性公网IP，则一次只能购买一台裸金属服务器。

16. 单击“立即购买”。

    在“订单详情”栏核对裸金属服务器信息，如果您确认规格无误，单击“提交订单”。

    说明

    如果您对价格有疑问，可以单击“了解计费详情”来了解产品价格。

17. 根据界面提示付款，单击“确认付款”。

    返回“裸金属服务器”主页面。

18. 完成订单支付后，等待系统创建资源，完成裸金属服务器购买。

    大约需要30分钟，裸金属服务器的状态会变为“运行中”。

    说明

    选择支持快速发放的规格，大约只需5分钟即可获得一台裸金属服务器。

4.2 配置 BMS 集群互相免密登录

操作场景

该任务指导用户在BMS集群上进行相关设置，并使其可以相互免密登录。

背景信息

$：表示在普通用户下，执行相关操作。

#：表示在管理员用户下，执行相关操作。

普通用户切换至管理员用户，请使用命令**sudo su**。

前提条件

已成功创建BMS，并绑定了弹性IP进行登录。

操作步骤

1. 使用“PuTTY”，采用密钥对方式登录集群中任意一台BMS。
2. 执行以下命令，防止系统超时退出。
   # TMOUT=0
3. 将BMS对应的密钥文件（.pem文件）拷贝至.ssh目录下，命名为“id_rsa”。
   $ cd ~/.ssh
   $ mv *.pem id_rsa
4. 执行以下命令，给密钥文件配置权限。
   $ sudo chmod 600 id_rsa
5. 执行以下命令，ssh方式登录本节点，验证是否可以不输入密码登录BMS。
   $ ssh localhost

   图 4-2 免密登录 BMS

   ```
   [rhel@bms-ff3 ibm]$ ssh localhost
   Last login: Sat Aug 26 09:54:20 2017 from 10.177.19.48
   [rhel@bms-ff3 ~]$ logout
   Connection to localhost closed.
   ```

   系统回显结果如图4-2所示，可以无密钥登录本节点，表示权限配置成功。
6. 依次登录集群中其他BMS，重复执行步骤1～5。
7. 执行以下命令，验证参加测试的BMS之间是否可以免密码互相登录。
   $ **ssh *用户名@SERVER_IP***

4.3 安装和使用 MPI（X86 BMS 场景）

该任务指导以CentOS7.3的OS为例在单节点上运行MPI应用。

4.3.1 裸金属服务器场景支持使用的 MPI

HPC当前支持的MPI包括：

- 驱动自带的OpenMPI
- 社区OpenMPI
- Spectrum MPI

- Intel MPI
- Platform MPI

以下小节的内容详细介绍了MPI的安装与使用，您可以根据需要选择合适的MPI进行安装。

4.3.2 安装和使用 IB 驱动自带的 Open MPI

操作场景

本节指导用户在BMS上安装和使用IB驱动自带的Open MPI（以版本3.1.0rc2为例）。

对于集群中的每台BMS，都需要执行该操作。

前提条件

已配置BMS集群间互相免密登录。

操作步骤

步骤1 查询是否安装了IB驱动。

1. 执行以下命令，查询是否已成功安装IB驱动。

   $ ls /usr/mpi/gcc/openmpi-3.1.0rc2/bin/mpirun

   $ rpm -qa | grep mlnx-ofa

   图 4-3 确认已安装 IB 驱动

   ```
   [rhel@bms-0004 ~]$ ls /usr/mpi/gcc/openmpi-3.1.0rc2/bin/mpirun
   /usr/mpi/gcc/openmpi-3.1.0rc2/bin/mpirun
   [rhel@bms-0004 ~]$
   [rhel@bms-0004 ~]$ rpm -qa | grep mlnx-ofa
   mlnx-ofa_kernel-4.3-OFED.4.3.1.0.1.1.g8509e41.rhel7u3.x86_64
   kmod-mlnx-ofa_kernel-4.3-OFED.4.3.1.0.1.1.g8509e41.rhel7u3.x86_64
   mlnx-ofa_kernel-devel-4.3-OFED.4.3.1.0.1.1.g8509e41.rhel7u3.x86_64
   ```

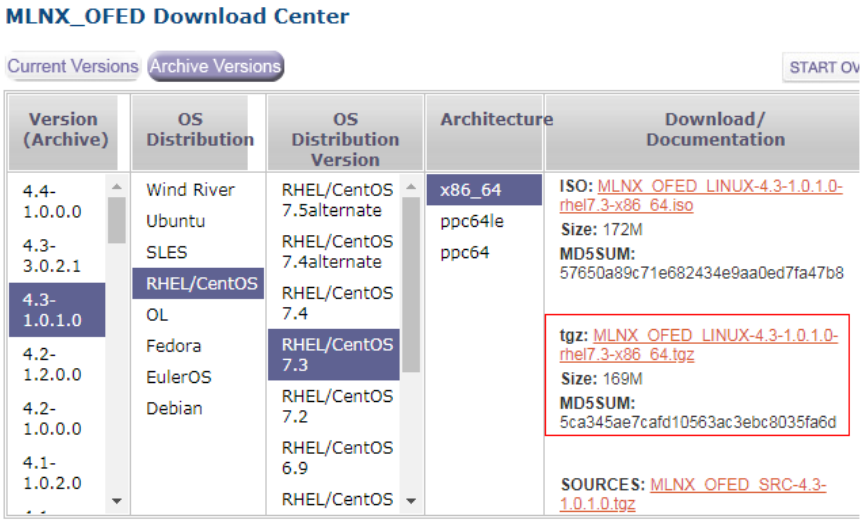
2. 查看回显结果。
   - 如果回显如图4-3所示，表示已安装IB驱动，执行步骤3。
   - 如果未安装IB驱动，执行步骤2。

步骤2 安装IB驱动。

1. 下载安装包“MLNX_OFED_LINUX-4.3-1.0.1.0-rhel7.3-x86_64.tgz”。

   下载地址：**https://network.nvidia.com/products/infiniband-drivers/linux/mlnx_ofed/**

图 4-4 IB 驱动的下载页面



2. 执行以下命令，安装软件包。

   # yum install tk tcl

   # tar -xvf MLNX_OFED_LINUX-4.3-1.0.1.0-rhel7.3-x86_64.tgz

   # cd MLNX_OFED_LINUX-4.3-1.0.1.0-rhel7.3-x86_64

   # ./mlnxofedinstall

步骤3 配置环境变量。

1. 使用vim编辑“~/.bashrc”文件，添加如下配置内容：

   export PATH=$PATH:/usr/mpi/gcc/openmpi-3.1.0rc2/bin

   export LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-3.1.0rc2/lib64

2. 执行以下命令，查看MPI环境变量是否正常。

   $ which mpirun

   图 4-5 查看 IB 驱动自带的 Open MPI 环境变量

   ```
   [rhel@bms-0004 ~]$ which mpirun
   /usr/mpi/gcc/openmpi-3.1.0rc2/bin/mpirun
   ```

   如果回显如图4-5所示，表示环境变量配置成功。

步骤4 执行以下命令，在单台BMS上运行IB驱动自带的Open MPI。

$ mpirun -np 2 -mca btl_openib_if_include "mlx5_0:1" -x MXM_IB_USE_GRH=y /usr/mpi/gcc/openmpi-3.1.0rc2/tests/imb/IMB-MPI1 PingPong

图 4-6 单台 BMS 上运行 Open MPI

```
#---------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#---------------------------------------------------
       #bytes #repetitions      t[usec]   Mbytes/sec
            0         1000         0.21         0.00
            1         1000         0.20         4.67
            2         1000         0.20         9.37
            4         1000         0.20        18.71
            8         1000         0.21        36.63
           16         1000         0.21        73.19
           32         1000         0.23       130.43
           64         1000         0.24       251.87
          128         1000         0.34       360.31
          256         1000         0.38       645.46
          512         1000         0.44      1101.68
         1024         1000         0.55      1768.44
         2048         1000         0.70      2772.64
         4096         1000         1.13      3452.66
         8192         1000         2.01      3883.81
        16384         1000         1.33     11709.83
        32768         1000         2.04     15345.09
        65536          640         3.20     19552.44
       131072          320         6.33     19735.97
       262144          160        14.78     16918.45
       524288           80        32.63     15324.90
      1048576           40        61.32     16306.82
      2097152           20       126.68     15788.01
      4194304           10       241.85     16539.00


# All processes entering MPI_Finalize
```

----结束

4.3.3 安装和使用社区 OpenMPI

操作场景

本节指导用户在BMS上安装和使用社区OpenMPI（以3.1.1版本为例）。

对于集群中的每台BMS，都需要执行该操作。

前提条件

已配置BMS集群间互相免密登录。

操作步骤

步骤1 安装HPC-X工具套件。

1. 使用社区OpenMPI时，需要同时使用Mellanox的HPC-X 套件，适用于CentOS 7.3的HPC-X版本是“hpcx-v2.2.0-gcc-MLNX_OFED_LINUX-4.3-1.0.1.0-redhat7.3-x86_64.tbz”。

   下载地址：**https://developer.nvidia.com/networking/hpc-x**

2. 将下载的软件包拷贝到BMS内（建议在“/home/rhel”目录下）。
3. 执行以下命令，解压HPC-X工具套件，并修改HPC-X工具套件目录。

   # tar -xvf hpcx-v2.2.0-gcc-MLNX_OFED_LINUX-4.3-1.0.1.0-redhat7.3-x86_64.tbz

   # mv hpcx-v2.2.0-gcc-MLNX_OFED_LINUX-4.3-1.0.1.0-redhat7.3-x86_64 /opt/hpcx-v2.2.0

步骤2 安装OpenMPI。

1. 下载社区OpenMPI，版本号为“openmpi-3.1.0.tar.gz”。

   下载地址：**https://www.open-mpi.org/software/ompi/v3.1/**
2. 将下载的OpenMPI压缩包拷贝至BMS内（建议在“/home/rhel”目录下）。
3. 执行以下命令，解压软件包。

   # tar -xzvf openmpi-3.1.0.tar.gz

   # cd openmpi-3.1.0
4. 执行以下命令，安装所需要的库文件，安装之前请确保BMS能与外网连通。

   a. 执行以下命令，安装依赖包。

      # yum install binutils-devel.x86_64 gcc-c++ autoconf automake libtool

      图 4-7 安装 binutils-devel.x86_64 等依赖包成功

      ```
      Running transaction
        Installing : libstdc++-devel-4.8.5-11.el7.x86_64                                                    1/2
        Installing : gcc-c++-4.8.5-11.el7.x86_64                                                            2/2
        Verifying  : gcc-c++-4.8.5-11.el7.x86_64                                                            1/2
        Verifying  : libstdc++-devel-4.8.5-11.el7.x86_64                                                    2/2

      Installed:
        gcc-c++.x86_64 0:4.8.5-11.el7

      Dependency Installed:
        libstdc++-devel.x86_64 0:4.8.5-11.el7

      Complete!
      ```
5. 执行以下命令，安装编译OpenMPI。

   # ./autogen.pl

   # mkdir build && cd build

   # ../configure --prefix=/opt/openmpi-310 --with-mxm=/opt/hpcx-v2.2.0/mxm

   # make all install

   图 4-8 OpenMPI 安装成功

   ```
   make[3]: Nothing to be done for `install-exec-am'.
   make[3]: Nothing to be done for `install-data-am'.
   make[3]: Leaving directory `/root/openmpi-3.1.0/build/test'
   make[2]: Leaving directory `/root/openmpi-3.1.0/build/test'
   make[1]: Leaving directory `/root/openmpi-3.1.0/build/test'
   make[1]: Entering directory `/root/openmpi-3.1.0/build'
   make[2]: Entering directory `/root/openmpi-3.1.0/build'
   make  install-exec-hook
   make[3]: Entering directory `/root/openmpi-3.1.0/build'
   make[3]: Leaving directory `/root/openmpi-3.1.0/build'
   make[2]: Nothing to be done for `install-data-am'.
   make[2]: Leaving directory `/root/openmpi-3.1.0/build'
   make[1]: Leaving directory `/root/openmpi-3.1.0/build'
   ```


步骤3 配置MPI环境变量。

1. 普通用户下，在“~/.bashrc”中添加如下环境变量：

   export PATH=$PATH:/opt/openmpi-310/bin

   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/openmpi-310/lib
2. 执行以下命令，导入配置的MPI环境变量。

   $ source ~/.bashrc
3. 执行以下命令，查看MPI环境变量是否正常。

   $ which mpirun

   图 4-9 环境变量正常

   ```
   [root@bms-0004 ~]# which mpirun
   /opt/openmpi-310/bin/mpirun
   ```

   回显如图4-9所示表示环境变量正常。

步骤4 执行以下命令，在单个BMS上运行社区OpenMPI。

1. 执行以下命令，生成可执行文件。

   $ cd ~

   $ vi hello.c

   编辑内容如下：

   ```
   #include<mpi.h>
   #include<stdio.h>
   int main(int argc, char** argv){
       //Initialize the MPI environment
       MPI_Init(NULL, NULL);
       //Get the number of processes
       int world_size;
       MPI_Comm_size(MPI_COMM_WORLD, &world_size);
       //Get the rank of the process
       int world_rank;
       MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
       //Get the name of the processor
       char processor_name[MPI_MAX_PROCESSOR_NAME];
       int name_len;
       MPI_Get_processor_name(processor_name, &name_len);
       //Print off a hello world message.
       printf("Hello world from processor %s, rank %d"" out of %d processors\n",processor_name,
   world_rank, world_size);
       //Finalize the MPI environment.
       MPI_Finalize();
    }
   ```

   $ mpicc hello.c -o hello

   须知

   不同版本的MPI运行的hello文件是不同的，都需要使用命令**mpicc hello.c -o hello**对hello.c文件重新编译。
2. 执行以下命令，在单个BMS上运行社区OpenMPI。

   $ mpirun -np 2 /home/rhel/hello

图 4-10 社区 OpenMPI 运行成功

```
[rhel@bms-0004 ~]$ mpirun -np 2 /home/rhel/hello
Hello world from processor bms-0004, rank 0 out of 2 processors
Hello world from processor bms-0004, rank 1 out of 2 processors
```

回显如图4-10所示，表示单个BMS上运行社区OpenMPI成功。

----结束

4.3.4 安装和使用 Spectrum MPI

操作场景

本节指导用户在BMS集群上安装和使用Spectrum MPI应用（以版本10.01.01为例）。

对于集群中的每台BMS，都需要执行该操作。

背景信息

IBM Spectrum MPI v10.1版本当前支持的操作系统列表如下：

- **IBM Spectrum MPI 10.1.0.1 Eval for x86_64 Linux**
  - Red Hat Enterprise Linux version 6.6及其之后的版本
  - Red Hat Enterprise Linux version 7.1及其之后的版本
  - SUSE Linux Enterprise Server version 11 SP4
  - SUSE Linux Enterprise Server version 12及其之后的版本
- **IBM Spectrum MPI 10.1.0.2 Eval for Power 8 Linux**
  Red Hat Enterprise Linux version 7.3及其之后的版本

前提条件

已配置BMS集群间互相免密登录。

操作步骤

步骤1 安装Spectrum MPI。

1. 获取IBM Spectrum MPI软件包，需要进行注册。
   获取的IBM Spectrum MPI软件包有两个，包括license和软件两部分：
   - ibm_smpi_lic_s-10.1Eval-rh7_Aug11.x86_64.rpm
   - ibm_smpi-10.01.01.0Eval-rh7_Aug11.x86_64.rpm

   下载地址：**https://www-01.ibm.com/marketing/iwm/iwm/web/preLogin.do?source=swerpsysz-lsf-3**
2. 安装IBM Spectrum MPI。
   a. 将**步骤1.1**中下载的MPI软件包上传至运行MPI的BMS内（建议“/home/rhel”目录下）。
   b. 执行以下命令，设置环境变量。
      - 如果选择自动接受IBM Spectrum MPI安装许可协议，执行以下命令：
        # export IBM_SPECTRUM_MPI_LICENSE_ACCEPT=yes

- 如果选择手动接受IBM Spectrum MPI安装许可协议，执行以下命令：

  # export IBM_SPECTRUM_MPI_LICENSE_ACCEPT=no

c. 安装License部分。

- 选择自动接受IBM Spectrum MPI安装许可协议的，执行以下命令：

  # rpm -ivh ibm_smpi_lic_s-10.1Eval-rh7_Aug11.x86_64.rpm

- 选择手动接受IBM Spectrum MPI安装许可协议的，执行以下命令：

  # rpm -ivh ibm_smpi_lic_s-10.1Eval-rh7_Aug11.x86_64.rpm

  # sh /opt/ibm/spectrum_mpi/lap_se/bin/accept_spectrum_mpi_license.sh

d. 安装软件部分。

 # rpm -ivh ibm_smpi-10.01.01.0Eval-rh7_Aug11.x86_64.rpm

步骤2 配置环境变量。

1. 默认情况下，Spectrum MPI会安装至“/opt/ibm/spectrum_mpi”目录。该场景下需要设置如下环境变量：

   $ export MPI_ROOT=/opt/ibm/spectrum_mpi

   $ export LD_LIBRARY_PATH=$MPI_ROOT/lib:$LD_LIBRARY_PATH

   $ export PATH=$MPI_ROOT/bin:$PATH

   $ export MANPATH=$MPI_ROOT/share/man:$MANPATH

   $ unset MPI_REMSH

2. 执行以下命令，查看MPI环境变量是否正常。

   $ which mpirun

   图 4-11 检查 MPI 环境变量

```
[rhel@bms-0004 ~]$ which mpirun
/opt/ibm/spectrum_mpi/bin/mpirun
```

步骤3 在单个BMS上通过Spectrum MPI运行可执行文件。

1. 假设hello.c文件在“/home/rhel/”目录下，生成的可执行文件名为hello，执行以下命令：

   $ cd /home/rhel/

   $ mpicc hello.c -o hello

2. 执行以下命令，在单个BMS上通过Spectrum MPI运行可执行文件。

   $ mpirun -np 2 /home/rhel/hello

   图 4-12 单 BMS 上运行 Spectrum MPI 成功

```
[rhel@bms-0004 ~]$ mpirun -np 2 /home/rhel/hello
Hello world from processor bms-0004, rank 0 out of 2 processors
Hello world from processor bms-0004, rank 1 out of 2 processors
```

----结束

4.3.5 安装和使用 Intel MPI

操作场景

本节指导用户在BMS集群上安装和使用Intel MPI应用（以版本l_mpi_2018.0.128为例）。

对于集群中的每台BMS，都需要执行该操作。

前提条件

已配置BMS集群间互相免密登录。

操作步骤

步骤1 安装Intel MPI。

1. 下载Intel MPI。
   下载地址：**https://software.intel.com/en-us/intel-mpi-library**
2. 执行以下命令，解压并安装Intel MPI。
   以l_mpi_2018.0.128.tgz为例：
   # tar -xvf l_mpi_2018.0.128.tgz
   # cd l_mpi_2018.0.128/
   # ./install.sh

   图 4-13 Intel MPI 成功安装

```
Step 5 of 5 | Complete
-------------------------------------------------------------------------------
Thank you for installing Intel(R) MPI Library 2018 for Linux*.

If you have not done so already, please register your product with Intel
Registration Center to create your support account and take full advantage of
your product purchase.

Your support account gives you access to free product updates and upgrades
as well as Priority Customer support at the Online Service Center
https://supporttickets.intel.com.

Click here https://software.intel.com/en-us/python-distribution
to download Intel(R) Distribution for Python*
This download will initiate separately. You can proceed with the installation
screen instructions.
-------------------------------------------------------------------------------
Press "Enter" key to quit:
```

步骤2 配置环境变量。

1. 普通用户下，在“~/.bashrc”中添加：
   export PATH=$PATH:/opt/intel/impi/2018.0.128/bin64
   export LD_LIBRARY_PATH=/opt/intel/impi/2018.0.128/lib64
2. 执行下列命令，导入环境变量。
   $ source ~/.bashrc

步骤3 执行下列命令，查看是否导入成功。

$ which mpirun

图 4-14 Intel MPI 环境变量导入成功

```
[rhel@bms-0004 root]$ which mpirun
/opt/intel/impi/2018.0.128/bin64/mpirun
```

回显结果如图4-14所示，表示环境变量导入成功。

步骤4 执行以下命令，在单个BMS上运行Intel MPI。

1. 执行以下命令，生成可执行文件。

   $ mpicc hello.c -o hello

2. 执行以下命令，在单个BMS上运行Intel MPI。

   $ mpirun -np 2 /home/rhel/hello

   图 4-15 在单个 BMS 上运行 Intel MPI

   ```
   [rhel@bms-0004 ~]$ mpirun -np 2 /home/rhel/hello
   Hello world from processor bms-0004, rank 1 out of 2 processors
   Hello world from processor bms-0004, rank 0 out of 2 processors
   ```

----结束

4.3.6 安装和使用 Platform MPI

操作场景

本节指导用户在BMS集群上安装和使用Platform MPI应用（以版本platform_mpi-09.01.04.03r-ce.bin为例）。

对于集群中的每台BMS，都需要执行该操作。

前提条件

已配置BMS集群间互相免密登录。

操作步骤

步骤1 安装Platform MPI

1. 下载platformMPI，如：platform_mpi- 09.01.04.03r-ce.bin。
2. 执行以下命令，安装依赖包。

   # yum install glibc.i686 libgcc-4.8.5-11.el7.i686 libgcc_s.so.1

3. 执行以下命令，增加执行权限。

   #chmod +x platform_mpi-09.01.04.03r-ce.bin

4. 安装Platform MPI。

   # ./platform_mpi-09.01.04.03r-ce.bin

   根据系统提示安装Platform MPI，默认安装到“/opt/ibm/platform_mpi”文件夹下。

图 4-16 Platform MPI 成功安装

```
================================================================================
Installation Complete
---------------------

Congratulations. IBM_PlatformMPI has been successfully installed to:

   /opt/ibm/platform_mpi

PRESS <ENTER> TO EXIT THE INSTALLER:
```

步骤2 配置环境变量。

1. 执行以下命令，获取pkey。

   # cat /sys/class/infiniband/mlx5_0/ports/1/pkeys/* | grep -v 0000

   图 4-17 获取 pkey

   ```
   [root@bms-0004 ~]# cat /sys/class/infiniband/mlx5_0/ports/1/pkeys/* | grep -v 0000
   0x8c2b
   0x7fff
   ```

2. 普通用户下，在~/.bashrc中添加：

   export MPI_ROOT=/opt/ibm/platform_mpi

   export PATH=$MPI_ROOT/bin:$PATH

   export LD_LIBRARY_PATH=/opt/ibm/platform_mpi/lib/linux_amd64

   export MPI_IB_PKEY=*步骤2.1中获取的pkey*

   $source ~/.bashrc

   说明

   如果存在多个pkey，使用英文逗号隔开。

3. 执行以下命令，检查环境变量是否配置成功。

   # which mpirun

   图 4-18 检查环境变量

   ```
   [rhel@bms-0004 root]$ which mpirun
   /opt/ibm/platform_mpi/bin/mpirun
   ```

步骤3 在单个BMS上运行Platform MPI。

1. 执行以下命令，重新编译hello.c文件。

   $ mpicc hello.c -o hello

2. 执行以下命令，在单个BMS上运行Platform MPI。

   $ mpirun -np 2 /home/rhel/hello

   图 4-19 在单台 BMS 上运行 Platform MPI

   ```
   [rhel@bms-0004 ~]$ mpirun -np 2 /home/rhel/hello
   Hello world from processor bms-0004, rank 1 out of 2 processors
   Hello world from processor bms-0004, rank 0 out of 2 processors
   ```

----结束

4.4 安装和使用 MPI（鲲鹏 BMS 场景）

该任务指导以CentOS 7.6的操作系统为例在单节点上运行MPI应用。

4.4.1 鲲鹏裸金属服务器支持使用的 MPI

HPC当前支持的MPI包括：

- 驱动自带的OpenMPI
- 社区OpenMPI
- MPICH

以下小节的内容详细介绍了MPI的安装与使用，您可以根据需要选择合适的MPI进行安装。

4.4.2 安装和使用 IB 驱动自带的 Open MPI

操作场景

本节操作指导用户在BMS上安装和使用IB驱动自带的Open MPI（以版本4.0.2a1为例）。

对于集群中的每台BMS，都需要执行该操作。

前提条件

已配置BMS集群间互相免密登录。

操作步骤

步骤1 查询是否安装了IB驱动

1. 执行以下命令，查询是否已成功安装IB驱动。

   $ ls /usr/mpi/gcc/openmpi-4.0.2a1/bin/mpirun

   $ rpm -qa | grep mlnx-ofa

   图 4-20 检查 IB 驱动

   ```
   [root@bms-arm-ib-0001 ~]# ls /usr/mpi/gcc/openmpi-4.0.2a1/bin/mpirun
   /usr/mpi/gcc/openmpi-4.0.2a1/bin/mpirun
   [root@bms-arm-ib-0001 ~]# rpm -qa | grep mlnx-ofa
   mlnx-ofa_kernel-devel-4.6-OFED.4.6.1.0.1.1.ga2cfe08.rhel7u6alternate.aarch64
   mlnx-ofa_kernel-4.6-OFED.4.6.1.0.1.1.ga2cfe08.rhel7u6alternate.aarch64
   mlnx-ofa_kernel-modules-4.6-OFED.4.6.1.0.1.1.ga2cfe08.kver.4.14.0_115.el7a.0.1.aarch64.aarch64
   [root@bms-arm-ib-0001 ~]#
   ```

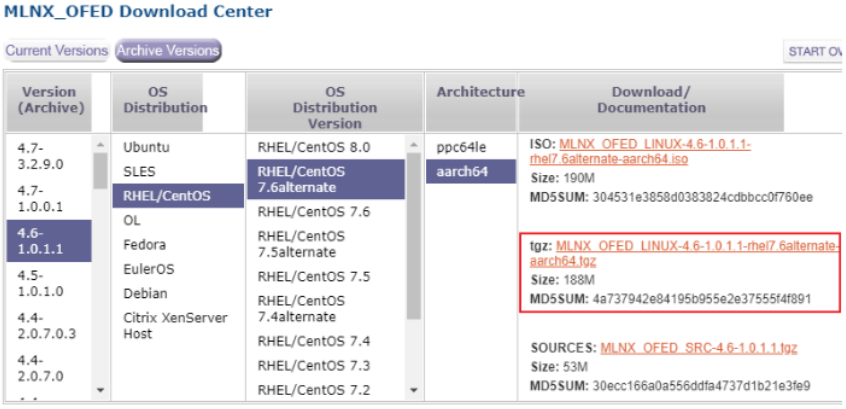
2. 查看回显结果。
   - 如果回显如图4-20所示，表示已安装IB驱动，执行步骤3。
   - 如果未安装IB驱动，执行步骤2。

步骤2 安装IB驱动。

1. 下载安装包“MLNX_OFED_LINUX-4.6-1.0.1.1-rhel7.6alternate-aarch64.tgz”。

   下载地址：**https://network.nvidia.com/products/infiniband-drivers/linux/mlnx_ofed/**

图 4-21 IB 驱动的下载页面



2. 执行以下命令，安装软件包。

   # yum install tk tcl -y

   # tar -xvf MLNX_OFED_LINUX-4.6-1.0.1.1-rhel7.6alternate-aarch64.tgz

   # cd MLNX_OFED_LINUX-4.6-1.0.1.1-rhel7.6-x86_64/

   # ./mlnxofedinstall

步骤3 安装配置ucx。

1. 下载ucx安装包

   # cd /opt && wget https://github.com/openucx/ucx/releases/download/v1.6.0/ucx-1.6.0.tar.gz

2. 解压ucx包

   # tar -xvf ucx-1.6.0.tar.gz

3. 编译安装ucx

 # cd /opt/ucx-1.6.0

 # yum install autoconf automake libtool numactl-devel -y

 # ./contrib/configure-release --prefix=/opt/ucx160 --enable-optimizations

 # make && make install

步骤4 配置ucx。

1. 创建非root用户rhel

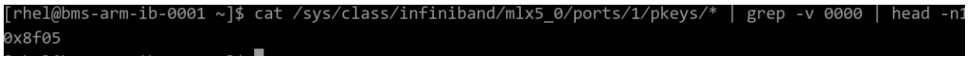
   # useradd rhel; su - rhel

2. 查询pkey，将查询到的值去掉第三位替换**步骤4.4**中的{pkey}。

   # cat /sys/class/infiniband/mlx5_0/ports/1/pkeys/* | grep -v 0000 | head -n1

   例如 查询到是0x8f05，去掉第三位后0xf05。

图 4-22 查询 pkey



3. 获取ucx PKEY。

   # ucx_info -c | grep -i pkey > ucx.env

4. 替换ucx中的PKEY。
   # sed -i 's/0x[a-f0-9]*/{pkey}/g' ucx.env
   本例中执行**sed -i 's/0x[a-f0-9]*/0xf05/g' ucx.env**

   图 4-23 替换 ucx 中的 PKEY

   ```
   [rhel@bms-arm-ib-0001 ~]$ sed -i 's/0x[a-f0-9]*/0xf05/g' ucx.env
   [rhel@bms-arm-ib-0001 ~]$ cat ucx.env
   UCX_RC_VERBS_PKEY=0xf05
   UCX_RC_MLX5_PKEY=0xf05
   UCX_DC_MLX5_PKEY=0xf05
   UCX_UD_VERBS_PKEY=0xf05
   UCX_UD_MLX5_PKEY=0xf05
   [rhel@bms-arm-ib-0001 ~]$
   ```

5. 将ucx pkey设置为环境变量。
   # sed -i 's/^UCX/export UCX/g' ucx.env
   # cat ucx.env >> ~/.bashrc

步骤5 配置mpi环境变量。

1. 使用vim编辑“~/.bashrc”文件，添加如下配置内容：
   export PATH=$PATH:/usr/mpi/gcc/openmpi-4.0.2a1/bin
   export LD_LIBRARY_PATH=/usr/mpi/gcc/openmpi-4.0.2a1/lib64
2. 执行以下命令，查看MPI环境变量是否正常。
   $ which mpirun

   图 4-24 查看 IB 驱动自带的 Open MPI 环境变量

   ```
   [rhel@bms-arm-ib-0001 ~]$ which mpirun
   /usr/mpi/gcc/openmpi-4.0.2a1/bin/mpirun
   [rhel@bms-arm-ib-0001 ~]$
   ```

   如果回显如图4-24所示，表示环境变量配置成功。
3. 执行以下命令，在单台BMS上运行IB驱动自带的Open MPI。
   #mpirun -np 2 -mca btl_openib_if_include "mlx5_0:1" -x MXM_IB_USE_GRH=y /usr/mpi/gcc/openmpi-3.1.0rc2/tests/imb/IMB-MPI1 PingPong

图 4-25 运行 IB 驱动自带的 Open MPI

```
#----------------------------------------------------------------
# Benchmarking Bcast
# #processes = 2
#----------------------------------------------------------------
       #bytes #repetitions  t_min[usec]  t_max[usec]  t_avg[usec]
            0         1000         0.05         0.05         0.05
            1         1000         0.30         0.36         0.33
            2         1000         0.30         0.36         0.33
            4         1000         0.30         0.36         0.33
            8         1000         0.30         0.37         0.34
           16         1000         0.30         0.38         0.34
           32         1000         0.30         0.38         0.34
           64         1000         0.30         0.38         0.34
          128         1000         0.31         0.41         0.36
          256         1000         0.34         0.44         0.39
          512         1000         0.41         0.55         0.48
         1024         1000         0.51         0.89         0.70
         2048         1000         0.56         1.05         0.80
         4096         1000         0.74         1.58         1.16
         8192         1000         0.87         2.46         1.67
        16384         1000         1.28         4.20         2.74
        32768         1000         2.11         7.86         4.98
        65536          640         4.67        14.20         9.43
       131072          320        11.30        27.46        19.38
       262144          160        23.78        60.15        41.96
       524288           80        53.69       117.09        85.39
```

----结束

4.4.3 安装和使用社区 OpenMPI

操作场景

本节指导用户在BMS上安装和使用社区OpenMPI（以4.0.2版本为例）。

对于集群中的每台BMS，都需要执行该操作。

前提条件

已配置BMS集群间互相免密登录。

操作步骤

步骤1 安装OpenMPI。

1. 下载社区OpenMPI，版本号为“openmpi-4.0.2.tar.bz2”。

   下载地址：**https://download.open-mpi.org/release/open-mpi/v4.0/openmpi-4.0.2.tar.bz2**

2. 将下载的OpenMPI压缩包拷贝至BMS内（建议在“/home/rhel”目录下）。

3. 执行以下命令，解压软件包。

   # tar -xzvf openmpi-4.0.2.tar.bz2

   # cd openmpi-4.0.2

4. 执行以下命令，安装所需要的依赖包，安装之前请确保BMS能与外网连通。

   # yum install binutils-devel.x86_64 gcc-c++ autoconf automake libtool

   图 4-26 安装 binutils-devel.x86_64 等依赖包成功

```
Running transaction
  Installing : libstdc++-devel-4.8.5-11.el7.x86_64                                  1/2
  Installing : gcc-c++-4.8.5-11.el7.x86_64                                          2/2
  Verifying  : gcc-c++-4.8.5-11.el7.x86_64                                          1/2
  Verifying  : libstdc++-devel-4.8.5-11.el7.x86_64                                  2/2

Installed:
  gcc-c++.x86_64 0:4.8.5-11.el7

Dependency Installed:
  libstdc++-devel.x86_64 0:4.8.5-11.el7

Complete!
```

5. 执行以下命令，安装编译OpenMPI。

   # ./openmpi-4.0.2/configure --prefix=/opt/openmpi-402--enable-mpirun-prefix-by-default --enable-mpi1-compatibility --with-ucx=/opt/ucx160

   # make -j 128 && make install

   图 4-27 OpenMPI 安装成功

```
make[3]: Leaving directory '/opt/openmpi-4.0.2/test'
make[2]: Leaving directory '/opt/openmpi-4.0.2/test'
make[1]: Leaving directory '/opt/openmpi-4.0.2/test'
make[1]: Entering directory '/opt/openmpi-4.0.2'
make[2]: Entering directory '/opt/openmpi-4.0.2'
make  install-exec-hook
make[3]: Entering directory '/opt/openmpi-4.0.2'
make[3]: Leaving directory '/opt/openmpi-4.0.2'
make[2]: Nothing to be done for 'install-data-am'.
make[2]: Leaving directory '/opt/openmpi-4.0.2'
make[1]: Leaving directory '/opt/openmpi-4.0.2'
[root@bms-arm-ib-0001 openmpi-4.0.2]#
```

步骤2 配置MPI环境变量。

1. 普通用户下，在“~/.bashrc”中添加如下环境变量：

   export PATH=$PATH:/opt/openmpi-310/bin

   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/openmpi-310/lib

2. 执行以下命令，导入配置的MPI环境变量。

   $ source ~/.bashrc

3. 执行以下命令，查看MPI环境变量是否正常。

   $ which mpirun

图 4-28 环境变量正常

```
[rhel@bms-arm-ib-0001 ~]$ which mpirun
/opt/openmpi-402/bin/mpirun
[rhel@bms-arm-ib-0001 ~]$
```

回显如图4-28所示表示环境变量正常。

步骤3 执行以下命令，在单个BMS上运行社区OpenMPI。

1. 执行以下命令，生成可执行文件。

   $ cd ~

   $ vi hello.c

   编辑内容如下：

```
#include<mpi.h>
#include<stdio.h>
int main(int argc, char** argv){
      //Initialize the MPI environment
      MPI_Init(NULL, NULL);
      //Get the number of processes
      int world_size;
      MPI_Comm_size(MPI_COMM_WORLD, &world_size);
      //Get the rank of the process
      int world_rank;
      MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
      //Get the name of the processor
      char processor_name[MPI_MAX_PROCESSOR_NAME];
      int name_len;
      MPI_Get_processor_name(processor_name, &name_len);
      //Print off a hello world message.
      printf("Hello world from processor %s, rank %d"" out of %d processors\n",processor_name,
world_rank, world_size);
      //Finalize the MPI environment.
      MPI_Finalize();
  }
```

   $ mpicc hello.c -o hello

   须知

   不同版本的MPI运行的hello文件是不同的，都需要使用命令**mpicc hello.c -o hello**对hello.c文件重新编译。

2. 执行以下命令，在单个BMS上运行社区OpenMPI。

   $ mpirun -np 2 /home/rhel/hello

图 4-29 社区 OpenMPI 运行成功

```
[rhel@bms-arm-ib-0001 ~]$ mpirun -np 2 /home/rhel/hello
--------------------------------------------------------------------------
WARNING: There was an error initializing an OpenFabrics device.

  Local host:   bms-arm-ib-0001
  Local device: mlx5_0
--------------------------------------------------------------------------
Hello world from processor bms-arm-ib-0001, rank 0 out of 2 processors
Hello world from processor bms-arm-ib-0001, rank 1 out of 2 processors
```


回显如图4-29所示，表示单个BMS上运行社区OpenMPI成功。

----结束

4.4.4 安装和使用 MPICH

操作场景

本节指导用户在鲲鹏BMS集群上安装和使用MPICH应用（以版本mpich-3.3.2为例）。

对于集群中的每台BMS，都需要执行该操作。

前提条件

已配置BMS集群间互相免密登录。

操作步骤

步骤1 安装MPICH。

1. 下载MPICH。
   下载地址：https://aur.archlinux.org/packages/mpich/
2. 执行以下命令，解压并安装MPICH。
   以mpich-3.3.2.tar.gz为例：
   # tar -xvf mpich-3.3.2.tar.gz
   # cd mpich-3.3.2/
   # ./configure --prefix=/opt/mpich-332 --with-device=ch4:ucx --with-ucx=/pub/mpi/ucx160/ --enable-fast=O3 CFLAGS="-fPIC -std=gnu11" FFLAGS=-fPIC CXXFLAGS=-fPIC FCFLAGS=-fPIC
   # make -j 128 && make install

   图 4-30 MPCHI 成功安装

```
make[3]: Leaving directory '/opt/mpich-3.3.2'
make[2]: Leaving directory '/opt/mpich-3.3.2'
Making install in examples
make[2]: Entering directory '/opt/mpich-3.3.2/examples'
make[3]: Entering directory '/opt/mpich-3.3.2/examples'
make[3]: Nothing to be done for 'install-exec-am'.
make[3]: Nothing to be done for 'install-data-am'.
make[3]: Leaving directory '/opt/mpich-3.3.2/examples'
make[2]: Leaving directory '/opt/mpich-3.3.2/examples'
make[1]: Leaving directory '/opt/mpich-3.3.2'
```

步骤2 配置环境变量。

1. 普通用户下，在“~/.bashrc”中添加：
   export PATH=/opt/mpich-332/bin: $PATH
   export LD_LIBRARY_PATH=/opt/mpich-332/lib
2. 执行下列命令，导入环境变量。

$ source ~/.bashrc

步骤3 执行下列命令，查看是否导入成功。

$ which mpirun

图 4-31 MPICH 环境变量导入成功

```
[rhel@bms-arm-ib-0001 ~]$ which mpirun
/opt/mpich-332/bin/mpirun
```

回显结果如图4-31所示，表示环境变量导入成功。

步骤4 执行以下命令，在单个BMS上运行MPICH。

1. 执行以下命令，生成可执行文件。
   $ mpicc hello.c -o hello
2. 执行以下命令，在单个BMS上运行MPICH。
   $ mpirun -np 2 /home/rhel/hello

图 4-32 在单个 BMS 上运行 MPICH

```
[rhel@bms-arm-ib-0001 ~]$ mpirun -np 2 /home/rhel/hello
Hello world from processor bms-arm-ib-0001, rank 0 out of 2 processors
Hello world from processor bms-arm-ib-0001, rank 1 out of 2 processors
[rhel@bms-arm-ib-0001 ~]$
```

----结束

4.5 在 HPC 集群上运行 MPI 应用（X86 BMS 场景）

该任务指导以CentOS7.3的OS为例在集群上运行MPI应用。

4.5.1 IB 驱动自带的 OpenMPI

操作场景

该任务指导用户在BMS集群上运行IB驱动自带的MPI应用（3.1.0rc2版本）。

前提条件

- 已配置BMS集群间互相免密登录。
- 集群中所有的BMS，均已安装IB驱动自带的OpenMPI。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台BMS。
2. 执行以下命令，关闭BMS防火墙。

 # service firewalld stop

 # iptables -F

3. 执行以下命令，查看防火墙是否关闭成功。

   # service firewalld status

   图 4-33 确认关闭防火墙成功

   ```
   [root@bms-0004 ~]# service firewalld status
   Redirecting to /bin/systemctl status  firewalld.service
   ● firewalld.service - firewalld - dynamic firewall daemon
      Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
      Active: inactive (dead)
        Docs: man:firewalld(1)
   ```

4. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤1.3**，关闭所有BMS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台BMS。
2. 执行以下命令，查看BMS的主机名。

   $ hostname

   图 4-34 查看 BMS 的主机名

   ```
   [rhel@bms-0004 ~]$ hostname
   bms-0004
   ```

3. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤2.2**，获取所有BMS的主机名。
4. 登录集群中任意一台BMS。
5. 执行以下命令，添加hosts配置文件。

   # vi /etc/hosts

   添加的内容为集群中所有BMS的私网IP和主机名，例如：

   192.168.0.1 bms-0004

   192.168.0.2 bms-0005

   ...

6. 执行以下命令，添加hostfile文件。

   $vi hostfile

   添加集群中所有BMS的主机名，例如：

   bms-0004

   bms-0005

   ...

7. 依次登录集群中所有BMS，重复执行**步骤2.5**～**步骤2.6**。

步骤3 运行MPI benchmark。

1. 在任意一台BMS中执行以下命令，检验hostfile文件是否配置成功。

   $ mpirun -np 2 -pernode --hostfile hostfile -mca btl_openib_if_include "mlx5_0:1" -x MXM_IB_USE_GRH=y hostname

   图 4-35 检查配置文件

   ```
   [rhel@bms-0004 ~]$ mpirun -np 2 -pernode --hostfile hostfile -mca btl_openib_if_include mlx5_0:1
   -x MXM_IB_USE_GRH=y hostname
   bms-0005
   bms-0004
   ```

回显如图4-35所示，显示集群中所有BMS的主机名，则表示hostfile文件配置成功。

2. 在任意一台BMS中执行以下命令，运行MPI benchmark，运行时指定hostfile路径。

   以两个BMS为例：

   $ mpirun -np 2 -pernode --hostfile hostfile -mca btl_openib_if_include "mlx5_0:1" -x MXM_IB_USE_GRH=y /usr/mpi/gcc/openmpi-3.1.0rc2/tests/imb/IMB-MPI1 PingPong

   图 4-36 集群运行 IB 驱动自带 OpenMPI

```
#---------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#---------------------------------------------------
       #bytes #repetitions      t[usec]   Mbytes/sec
            0         1000         1.27         0.00
            1         1000         1.26         0.75
            2         1000         1.24         1.53
            4         1000         1.21         3.14
            8         1000         1.21         6.30
           16         1000         1.21        12.60
           32         1000         1.21        25.20
           64         1000         1.28        47.83
          128         1000         1.33        91.97
          256         1000         1.83       133.18
          512         1000         1.94       251.18
         1024         1000         2.25       433.79
         2048         1000         2.67       730.85
         4096         1000         4.15       941.97
         8192         1000         5.63      1386.69
        16384         1000         8.07      1935.05
        32768         1000        11.46      2726.09
        65536          640        19.90      3140.53
       131072          320        31.54      3963.68
       262144          160        50.68      4932.72
       524288           80        93.75      5333.39
      1048576           40       178.04      5616.87
      2097152           20       350.49      5706.27
      4194304           10       700.71      5708.50


# All processes entering MPI_Finalize
```

   系统回显如图4-36所示，表示集群上运行IB驱动自带的OpenMPI成功。

----结束

4.5.2 社区 OpenMPI

操作场景

该任务指导用户在BMS集群上运行社区OpenMPI（以3.1.1版本为例）。

前提条件

- 已配置BMS集群间互相免密登录。
- 集群中所有的BMS，均已安装社区OpenMPI。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台BMS。
2. 执行以下命令，关闭BMS防火墙。

   # service firewalld stop

   # iptables -F
3. 执行以下命令，查看防火墙是否关闭成功。

   # service firewalld status

   图 4-37 确认关闭防火墙成功

   ```
   [root@bms-0004 ~]# service firewalld status
   Redirecting to /bin/systemctl status  firewalld.service
   ● firewalld.service - firewalld - dynamic firewall daemon
      Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
      Active: inactive (dead)
        Docs: man:firewalld(1)
   ```
4. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤1.3**，关闭所有BMS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台BMS。
2. 执行以下命令，查看BMS的主机名。

   $ hostname

   图 4-38 查看 BMS 的主机名

   ```
   [rhel@bms-0004 ~]$ hostname
   bms-0004
   ```
3. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤2.2**，获取所有BMS的主机名。
4. 登录集群中任意一台BMS。
5. 执行以下命令，添加hosts配置文件。

   # vi /etc/hosts

   添加的内容为集群中所有BMS的私网IP和主机名，例如：

   192.168.0.1 bms-0004

   192.168.0.2 bms-0005

   ...
6. 执行以下命令，添加hostfile文件。

   $vi hostfile

   添加集群中所有BMS的主机名，例如：

   bms-0004

   bms-0005

...

7. 依次登录集群中所有BMS，重复执行**步骤2.5**～**步骤2.6**。

步骤3 在任意一台BMS中执行以下命令，运行社区Open MPI。

以两个BMS为例：

$ mpirun -np 2 --pernode -hostfile hostfile /home/rhel/hello

图 4-39 集群上运行社区 OpenMPI 成功

```
Hello world from processor bms-0005, rank 0 out of 2 processors
Hello world from processor bms-0004, rank 0 out of 2 processors
```

说明

hostfile文件在运行时需要指定路径，可执行文件hello路径需为绝对路径，集群中所有可执行文件在同一路径下。

----结束

4.5.3 Spectrum MPI

操作场景

该任务指导用户在BMS集群上运行Spectrum MPI应用（10.01.01版本）。

前提条件

- 已配置BMS集群间互相免密登录。
- 集群中所有的BMS，均已安装Spectrum MPI。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台BMS。
2. 执行以下命令，关闭BMS防火墙。

   # service firewalld stop

   # iptables -F
3. 执行以下命令，查看防火墙是否关闭成功。

   # service firewalld status

   图 4-40 确认关闭防火墙成功

   ```
   [root@bms-0004 ~]# service firewalld status
   Redirecting to /bin/systemctl status  firewalld.service
   ● firewalld.service - firewalld - dynamic firewall daemon
      Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
      Active: inactive (dead)
        Docs: man:firewalld(1)
   ```
4. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤1.3**，关闭所有BMS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台BMS。
2. 执行以下命令，查看BMS的主机名。

   $ hostname

   图 4-41 查看 BMS 的主机名

   ```
   [rhel@bms-0004 ~]$ hostname
   bms-0004
   ```

3. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤2.2**，获取所有BMS的主机名。
4. 登录集群中任意一台BMS。
5. 执行以下命令，添加hosts配置文件。

   # vi /etc/hosts

   添加的内容为集群中所有BMS的私网IP和主机名，例如：

   192.168.0.1 bms-0004

   192.168.0.2 bms-0005

   ...
6. 执行以下命令，添加hostfile文件。

   $vi hostfile

   添加集群中所有BMS的主机名，例如：

   bms-0004

   bms-0005

   ...
7. 依次登录集群中所有BMS，重复执行**步骤2.5**～**步骤2.6**。

步骤3 执行以下命令，在单个BMS上通过Spectrum MPI运行可执行文件。

$ mpirun -np 2 -pernode --hostfile hostfile /home/rhel/hello

图 4-42 集群上运行 Spetrum MPI 成功

```
[rhel@bms-0004 ~]$ mpirun -np 2 -pernode --hostfile hostfile /home/rhel/hello
Hello world from processor bms-0004, rank 0 out of 2 processors
Hello world from processor bms-0005, rank 1 out of 2 processors
```

说明

hostfile文件在运行时需要指定路径，可执行文件hello路径需为绝对路径，集群中所有可执行文件在同一路径下。

----结束

4.5.4 Intel MPI

操作场景

该任务指导用户在BMS集群上运行Intel MPI应用（l_mpi_2017.3.196版本）。

前提条件

- 已配置BMS集群间互相免密登录。
- 集群中所有的BMS，均已安装Spectrum MPI。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台BMS。
2. 执行以下命令，关闭BMS防火墙。

   # service firewalld stop

   # iptables -F
3. 执行以下命令，查看防火墙是否关闭成功。

   # service firewalld status

   图 4-43 确认关闭防火墙成功

   ```
   [root@bms-0004 ~]# service firewalld status
   Redirecting to /bin/systemctl status  firewalld.service
   ● firewalld.service - firewalld - dynamic firewall daemon
      Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
      Active: inactive (dead)
        Docs: man:firewalld(1)
   ```
4. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤1.3**，关闭所有BMS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台BMS。
2. 执行以下命令，查看BMS的主机名。

   $ hostname

   图 4-44 查看 BMS 的主机名

   ```
   [rhel@bms-0004 ~]$ hostname
   bms-0004
   ```
3. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤2.2**，获取所有BMS的主机名。
4. 登录集群中任意一台BMS。
5. 执行以下命令，添加hosts配置文件。

   # vi /etc/hosts

   添加的内容为集群中所有BMS的私网IP和主机名，例如：

   192.168.0.1 bms-0004

   192.168.0.2 bms-0005

   ...
6. 执行以下命令，添加hostfile文件。

   $vi hostfile

   添加集群中所有BMS的主机名，例如：

   bms-0004

   bms-0005

...

7. 依次登录集群中所有BMS，重复执行步骤2.5～步骤2.6。

步骤3 执行以下命令，在BMS集群运行Intel MPI。

以两台BMS为例：

$ mpirun -perhost 2 -np 12 -machinefile hostfile /home/rhel/hello

图 4-45 BMS 集群上运行 Intel MPI 成功

```
[rhel@bms-0004 ~]$ mpirun -perhost 2 -np 12 -machinefile hostfile /home/rhel/hello
Hello world from processor bms-0004, rank 4 out of 12 processors
Hello world from processor bms-0004, rank 6 out of 12 processors
Hello world from processor bms-0004, rank 8 out of 12 processors
Hello world from processor bms-0004, rank 10 out of 12 processors
Hello world from processor bms-0004, rank 0 out of 12 processors
Hello world from processor bms-0005, rank 1 out of 12 processors
Hello world from processor bms-0004, rank 2 out of 12 processors
Hello world from processor bms-0005, rank 3 out of 12 processors
Hello world from processor bms-0005, rank 5 out of 12 processors
Hello world from processor bms-0005, rank 7 out of 12 processors
Hello world from processor bms-0005, rank 9 out of 12 processors
Hello world from processor bms-0005, rank 11 out of 12 processors
```

说明

hostfile文件在运行时需要指定路径，可执行文件hello路径需为绝对路径，集群中所有可执行文件在同一路径下。

----结束

4.5.5 Platform MPI

操作场景

该任务指导用户在BMS集群上运行Platform MPI应用（platform_mpi- 09.01.04.03r-ce.bin版本）。

前提条件

- 已配置BMS集群间互相免密登录。
- 集群中所有的BMS，均已安装Platform MPI。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台BMS。
2. 执行以下命令，关闭BMS防火墙。

   # service firewalld stop

   # iptables -F
3. 执行以下命令，查看防火墙是否关闭成功。

   # service firewalld status

图 4-46 确认关闭防火墙成功

```
[root@bms-0004 ~]# service firewalld status
Redirecting to /bin/systemctl status  firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)
```

4. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤1.3**，关闭所有BMS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台BMS。
2. 执行以下命令，查看BMS的主机名。

   $ hostname

   图 4-47 查看 BMS 的主机名

   ```
   [rhel@bms-0004 ~]$ hostname
   bms-0004
   ```

3. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤2.2**，获取所有BMS的主机名。
4. 登录集群中任意一台BMS。
5. 执行以下命令，添加hosts配置文件。

   # vi /etc/hosts

   添加的内容为集群中所有BMS的私网IP和主机名，例如：

   192.168.0.1 bms-0004

   192.168.0.2 bms-0005

   ...
6. 执行以下命令，添加hostfile文件。

   $vi hostfile

   添加集群中所有BMS的主机名，例如：

   bms-0004

   bms-0005

   ...
7. 依次登录集群中所有BMS，重复执行**步骤2.5**～**步骤2.6**。

步骤3 执行以下命令，在BMS集群上运行社区OpenMPI。

$ mpirun -np 12 -machinefile hostfile /home/rhel/hello

图 4-48 BMS 集群上 Platform MPI 运行成功

```
[rhel@bms-0004 ~]$ mpirun -np 12 -machinefile hostfile /home/rhel/hello
Hello world from processor bms-0004, rank 6 out of 12 processors
Hello world from processor bms-0004, rank 4 out of 12 processors
Hello world from processor bms-0004, rank 8 out of 12 processors
Hello world from processor bms-0004, rank 2 out of 12 processors
Hello world from processor bms-0004, rank 10 out of 12 processors
Hello world from processor bms-0004, rank 0 out of 12 processors
Hello world from processor bms-0005, rank 11 out of 12 processors
Hello world from processor bms-0005, rank 9 out of 12 processors
Hello world from processor bms-0005, rank 7 out of 12 processors
Hello world from processor bms-0005, rank 5 out of 12 processors
Hello world from processor bms-0005, rank 1 out of 12 processors
Hello world from processor bms-0005, rank 3 out of 12 processors
```


说明

hostfile文件在运行时需要指定路径，可执行文件hello路径需为绝对路径，集群中所有可执行文件在同一路径下。

----结束

4.6 在 HPC 集群上运行 MPI 应用（鲲鹏 BMS 场景）

该任务指导以CentOS 7.6的操作系统为例在集群上运行MPI应用。

4.6.1 安装和使用 IB 驱动自带的 Open MPI

操作场景

该任务指导用户在鲲鹏BMS集群上运行IB驱动自带的MPI应用（以版本4.0.2a1为例）。

前提条件

- 已配置BMS集群间互相免密登录。
- 集群中所有的BMS，均已安装IB驱动自带的OpenMPI。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台BMS。
2. 执行以下命令，关闭BMS防火墙。

   # service firewalld stop

   # iptables -F
3. 执行以下命令，查看防火墙是否关闭成功。

   # service firewalld status

   图 4-49 确认关闭防火墙成功

   ```
   [root@bms-arm-ib-0001 ~]# service firewalld status
   Redirecting to /bin/systemctl status firewalld.service
   ● firewalld.service - firewalld - dynamic firewall daemon
      Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
      Active: inactive (dead)
        Docs: man:firewalld(1)
   [root@bms-arm-ib-0001 ~]#
   ```
4. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤1.3**，关闭所有BMS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台BMS, 执行以下命令，添加hosts配置文件。

   # vi /etc/hosts

   添加的内容为集群中所有BMS的私网IP和主机名，例如：

192.168.1.138 bms-arm-ib-0001

192.168.1.45 bms-arm-ib-0002

...

2. 执行以下命令，添加hostfile文件。

   $vi hostfile

   添加集群中所有BMS的主机名，以及对应的核数(假设为2核)，例如：

   bms-arm-ib-0001 slots=2

   bms-arm-ib-0002 slots=2

   ...

3. 依次登录集群中所有BMS，重复执行**步骤2.1**～**步骤2.2**。

步骤3 运行MPI benchmark。

1. 在任意一台BMS中执行以下命令，检验hostfile文件是否配置成功。

   $ mpirun -np 2 -pernode --hostfile hostfile -mca btl_openib_if_include "mlx5_0:1" -x MXM_IB_USE_GRH=y hostname

   图 4-50 检查配置文件

   回显如图4-50所示，显示集群中所有BMS的主机名，则表示hostfile文件配置成功。

2. 在任意一台BMS中执行以下命令，运行MPI benchmark，运行时指定hostfile路径。

   以两个BMS为例：

   $ mpirun -np 2 -pernode --hostfile hostfile -mca btl_openib_if_include "mlx5_0:1" -x MXM_IB_USE_GRH=y /usr/mpi/gcc/openmpi-4.0.2a1/tests/imb/IMB-MPI1 PingPong

图 4-51 集群运行 IB 驱动自带 OpenMPI

```
#------------------------------------------------
# Benchmarking PingPong
# #processes = 2
#------------------------------------------------
       #bytes #repetitions      t[usec]   Mbytes/sec
            0         1000         1.33         0.00
            1         1000         1.24         0.81
            2         1000         1.22         1.64
            4         1000         1.21         3.29
            8         1000         1.22         6.56
           16         1000         1.22        13.10
           32         1000         1.29        24.85
           64         1000         1.41        45.51
          128         1000         1.46        87.46
          256         1000         1.90       134.94
          512         1000         2.19       234.19
         1024         1000         2.61       392.09
         2048         1000         3.70       553.17
         4096         1000         4.86       841.94
         8192         1000         7.36      1112.38
        16384         1000        10.35      1582.33
        32768         1000        16.11      2033.76
        65536          640        27.77      2360.09
       131072          320        50.42      2599.37
       262144          160        34.22      7659.69
```

系统回显如图4-51所示，表示集群上运行IB驱动自带的OpenMPI成功。

----结束

4.6.2 安装和使用社区 OpenMPI

操作场景

该任务指导用户在BMS集群上运行社区OpenMPI（以4.0.2版本为例））。

前提条件

- 已配置BMS集群间互相免密登录。
- 集群中所有的BMS，均已安装社区OpenMPI。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台BMS。
2. 执行以下命令，关闭BMS防火墙。

   # service firewalld stop

   # iptables -F
3. 执行以下命令，查看防火墙是否关闭成功。

   # service firewalld status

   图 4-52 确认关闭防火墙成功

   ```
   [root@bms-arm-ib-0001 ~]# service firewalld status
   Redirecting to /bin/systemctl status firewalld.service
   ● firewalld.service - firewalld - dynamic firewall daemon
      Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
      Active: inactive (dead)
        Docs: man:firewalld(1)
   [root@bms-arm-ib-0001 ~]#
   ```
4. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤1.3**，关闭所有BMS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台BMS, 执行以下命令，添加hosts配置文件。

   # vi /etc/hosts

   添加的内容为集群中所有BMS的私网IP和主机名，例如：

   192.168.1.138 bms-arm-ib-0001

   192.168.1.45 bms-arm-ib-0002

   ...
2. 执行以下命令，添加hostfile文件。

   $vi hostfile

   添加集群中所有BMS的主机名，以及对应的核数(假设为2核)，例如：

   bms-arm-ib-0001 slots=2

   bms-arm-ib-0002 slots=2

   ...
3. 依次登录集群中所有BMS，重复执行**步骤2.1**～**步骤2.2**。

步骤3 在任意一台BMS中执行以下命令，运行社区Open MPI。

以两个BMS为例：

$ mpirun -np 2 --pernode -hostfile hostfile /home/rhel/hello

图 4-53 集群上运行社区 OpenMPI 成功

```
[rhel@bms-arm-ib-0002 ~]$ mpirun -np 2 --pernode -hostfile /home/rhel/hostfile /home/rhel/hello
--------------------------------------------------------------------------
WARNING: There was an error initializing an OpenFabrics device.

  Local host:   bms-arm-ib-0001
  Local device: mlx5_0
--------------------------------------------------------------------------
Hello world from processor bms-arm-ib-0002, rank 0 out of 2 processors
Hello world from processor bms-arm-ib-0001, rank 1 out of 2 processors
[bms-arm-ib-0002:86385] 1 more process has sent help message help-mpi-btl-openib.txt / error in device init
[bms-arm-ib-0002:86385] Set MCA parameter "orte_base_help_aggregate" to 0 to see all help / error messages
[rhel@bms-arm-ib-0002 ~]$
```

说明

hostfile文件在运行时需要指定路径，可执行文件hello路径需为绝对路径，集群中所有可执行文件在同一路径下。

----结束

4.6.3 安装和使用 MPICH

操作场景

该任务指导用户在BMS集群上运行MPICH应用（mpich-3.3.2版本）。

前提条件

- 已配置BMS集群间互相免密登录。
- 集群中所有的BMS，均已安装MPICH。

操作步骤

步骤1 关闭防火墙。

1. 登录集群中任意一台BMS。
2. 执行以下命令，关闭BMS防火墙。

   # service firewalld stop

   # iptables -F
3. 执行以下命令，查看防火墙是否关闭成功。

   # service firewalld status

   图 4-54 确认关闭防火墙成功

```
[root@bms-arm-ib-0001 ~]# service firewalld status
Redirecting to /bin/systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:firewalld(1)
[root@bms-arm-ib-0001 ~]#
```

4. 依次登录集群中所有BMS，重复执行**步骤1.2**～**步骤1.3**，关闭所有BMS的防火墙。

步骤2 修改配置文件。

1. 登录集群中任意一台BMS, 执行以下命令，添加hosts配置文件。
   # vi /etc/hosts
   添加的内容为集群中所有BMS的私网IP和主机名，例如：
   192.168.1.138 bms-arm-ib-0001
   192.168.1.45 bms-arm-ib-0002
   ...
2. 执行以下命令，添加hostfile文件。
   $vi hostfile
   添加集群中所有BMS的主机名，以及对应的核数(假设为2核)，例如：
   bms-arm-ib-0001:2
   bms-arm-ib-0002:2
   ...
3. 依次登录集群中所有BMS，重复执行**步骤2.1**～**步骤2.2**执行以下命令，

步骤3 在单个BMS上通过MPICH运行可执行文件。

$ mpirun -np 2 -hostfile /home/rhel/hostfile /home/rhel/hello

图 4-55 集群上运行 MPICH 成功

```
[rhel@bms-arm-ib-0002 ~]$ mpirun -np 2 -hostfile /home/rhel/hostfile /home/rhel/hello
Hello world from processor bms-arm-ib-0001, rank 0 out of 2 processors
Hello world from processor bms-arm-ib-0002, rank 1 out of 2 processors
```

说明

hostfile文件在运行时需要指定路径，可执行文件hello路径需为绝对路径，集群中所有可执行文件在同一路径下。

----结束

HPC Cloud 解决方案 用户指南 01

A 修订记录

发布日期	修订记录
2021-03-29	第一次正式发布。