鲲鹏软件栈 编译器与编程语言

移植指南

文档版本 04

发布日期 2020-08-06

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默示的声明或保证。

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 安装 OpenJDK

介绍

简要介绍

OpenJDK原是Sun Microsystems公司为Java平台构建的Java开发环境（**JDK**）的开源版本，完全自由，开放源码。Sun Microsystems公司在2006年的JavaOne大会上称将对Java开放源代码，于2009年4月15日正式发布OpenJDK。甲骨文在2010年收购Sun Microsystem之后接管了这个项目。

语言：JAVA

一句话描述：OpenJDK是一个开放源代码，完全自由的开发环境。

开源协议：GPLv2.1

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如**表1-1**所示。

表 1-1 云服务器配置

项目	说明
规格	kc1.large.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如**表1-2**所示。

表 1-2 操作系统要求

项目	说明	下载地址
CentOS	• 版本：7.5 • Kernel：4.14.0	在公共镜像中已提供。
Redhat	• 版本：7.5 • Kernel：4.14.0	在公共镜像中已提供。
EulerOS	• 版本：2.8 • Kernel：4.19.36	在公共镜像中已提供。
UbuntuLTS	• 版本：18.04 • Kernel：4.15	在公共镜像中已提供。

安装 OpenJDK 8

Debian, Ubuntu, etc.

步骤1 执行如下命令。

sudo apt-get install openjdk-8-jre

步骤2 openjdk-8-jre软件包只包含了Java Runtime Environment（JRE）。如果你需要开发、编译JAVA程序，请安装openjdk-8-jdk软件包。在命令行中输入。

sudo apt-get install openjdk-8-jdk

----结束

Fedora, Red Hat Enterprise Linux, CentOS Linux, EulerOS Linux etc.

步骤1 执行如下命令。

su -c "yum install java-1.8.0-openjdk"

步骤2 java-1.8.0-openjdk软件包只包含了Java Runtime Environment（JRE）。如果你需要开发、编译JAVA程序，请安装java-1.8.0-openjdk-devel软件包。在命令行中输入。

su -c "yum install java-1.8.0-openjdk-devel"

----结束

其他版本的JDK暂未经过华为云的测试。

2 移植 Perl

介绍

简要介绍

Perl，一种功能丰富的计算机程序语言，运行在超过100种计算机平台上，适用广泛，从大型机到便携设备，从快速原型创建到大规模可扩展开发。Perl语言的应用范围很广，除CGI以外，Perl被用于图形编程、系统管理、网络编程、金融、生物以及其他领域。由于其灵活性，Perl被称为脚本语言中的瑞士军刀。

语言：C

一句话描述：常用脚本语言

开源协议：GPL

建议的版本

建议使用版本为“perl-5.28.0”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表2-1所示。

表 2-1 云服务器配置

项目	说明
规格	kc1.large.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表2-2所示。

表 2-2 操作系统要求

项目	版本	下载地址
EulerOS	2.8	在公共镜像中已提供
Kernel	4.19.36	包含在操作系统镜像中。

配置编译环境

执行如下命令，安装依赖包。

yum install wget -y

获取源码

执行如下命令，获取“perl-5.28.0”源码包。

perl官网（**https://www.perl.org/**）提供perl的源代码压缩包，可以直接下载，各版本的列表可以通过：**https://www.cpan.org/src/README.html**获取。

cd /usr/local/src

wget https://www.cpan.org/src/5.0/perl-5.28.0.tar.gz

编译和安装

步骤1 执行如下命令，解压软件包。

tar -zxvf perl-5.28.0.tar.gz

步骤2 执行如下命令，进入perl安装目录。

cd perl-5.28.0

步骤3 执行如下命令，配置编译并安装perl源码。

./Configure -de

make -j4

make test

make install

----结束

运行和验证

执行如下命令，查看perl版本。

perl -v

当系统回显类似如下信息时，表示安装成功。

```
[root@ecs-1-0002 ~]# perl -v

This is perl 5, version 28, subversion 0 (v5.28.0) built for aarch64-linux
```

```
Copyright 1987-2018, Larry Wall

Perl may be copied only under the terms of either the Artistic License or the
GNU General Public License, which may be found in the Perl 5 source kit.

Complete documentation for Perl, including FAQ lists, should be found on
this system using "man perl" or "perldoc perl".  If you have access to the
Internet, point your browser at http://www.perl.org/, the Perl Home Page.
```


3 安装 TypeScript

介绍

简要介绍

TypeScript是JavaScript的类型的超集，它可以编译成纯JavaScript，编译出来的JavaScript可以运行在任何浏览器上。TypeScript编译工具可以运行在任何服务器和任何系统上，并且它是开源的。

语言：Erlang

一句话描述：TypeScript是一个JavaScript的超集。

建议的版本

建议使用版本为“TypeScript-3.5.3”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表3-1所示。

表 3-1 云服务器配置

项目	说明
规格	kc1.xlarge.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表3-2所示。

表 3-2 操作系统要求

项目	版本	下载地址
CentOS	7.5	在公共镜像中已提供。
Kernel	4.14.0	在公共镜像中已提供。

配置安装环境

TypeScript依赖于“node.js”，因此先安装“node.js”，步骤如下：

步骤1 获取“node.js”软件包。

wget https://nodejs.org/dist/v10.16.0/node-v10.16.0-linux-arm64.tar.xz

步骤2 解压软件压缩包。

xz -d node-v10.16.0-linux-arm64.tar.xz

tar -xvf node-v10.16.0-linux-arm64.tar.xz

步骤3 在解压目录下的“bin”目录下有可执行文件“node”和“npm”，在全局路径下建立指向可执行文件“node”及“npm”的软链接。

ln -s /root/node-v10.16.0-linux-arm64/bin/node /usr/bin/node

ln -s /root/node-v10.16.0-linux-arm64/bin/npm /usr/bin/npm

----结束

安装

步骤1 执行安装命令。

npm install -g typescript

步骤2 安装完成后，在“/root/node-v10.16.0-linux-arm64/bin/”目录下会生成“tsc”和“tsserver”目标文件。

步骤3 在全局路径下创建指向目标文件的软链接。

ln -s /root/node-v10.16.0-linux-arm64/bin/tsc /usr/bin/tsc

ln -s /root/node-v10.16.0-linux-arm64/bin/tsserver /usr/bin/npm/tsserver

----结束

运行和验证

步骤1 创建一个工作目录。

mkdir /root/working

步骤2 在该目录下新建一个“test.ts”文件，并添加如下内容。

```
var message = "Hello World";
console.log(message);
```


步骤3 将TypeScript转换成JavaScript代码。

tsc test.ts

步骤4 执行JavaScript代码。

node test.js

显示“Hello World”，运行成功。

----结束

4 移植 Ruby

介绍

简要介绍

Ruby是一种开源的面向对象程序设计的服务器端脚本语言，Ruby可运行于多种平台，如Windows、MAC OS和UNIX的各种版本。

语言：C

一句话描述：服务器端脚本语言

开源协议：BSD

建议的版本

建议使用版本为“ruby-2.6.3”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表4-1所示。

表 4-1 云服务器配置

项目	说明
规格	kc1.large.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表4-2所示。

表 4-2 操作系统要求

项目	说明	下载地址
CentOS	• 版本：7.5 • Kernel：4.14.0	在公共镜像中已提供
EulerOS	• 版本：2.8 • Kernel：4.19.36	在公共镜像中已提供

配置编译环境

执行如下命令，安装依赖包。

yum install libtool libtool-ltdl-devel libevent-devel lua ncurses-devel openssl-devel flex

获取源码

执行如下命令，获取“ruby-2.6.3”源码包。

cd /usr/local/src

wget https://cache.ruby-lang.org/pub/ruby/2.6/ruby-2.6.3.tar.gz

编译和安装

步骤1 执行如下命令，解压软件包。

tar -zxvf ruby-2.6.3.tar.gz

步骤2 执行如下命令，进入ruby安装目录。

cd ruby-2.6.3

步骤3 执行以下命令，生成makefile文件。

./configure

步骤4 执行如下命令，编译并安装ruby源码。

make -j4

make install

----结束

运行和验证

步骤1 执行如下命令，查看ruby版本。

ruby -v

系统回显类似如下信息：

```
ruby 2.6.3p62 (2019-04-16 revision 67580) [aarch64-linux]
```

步骤2 执行如下命令，测试执行ruby代码。

ruby test.rb

系统回显类似如下信息：

```
hello, Ruby!
```

----结束

5 移植 LuaJIT

介绍

简要介绍

Lua JIT是Lua语言的即时（JIT:Just-In-Time）编译器，它提供基于快速解释器和跟踪编译器的虚拟机，可显著提高Lua程序的性能。

语言：C/C++

一句话描述：Lua语言即时编译器

开源协议：MIT license

建议的版本

建议使用版本为“v2.1.0-beta3”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表5-1所示。

表 5-1 云服务器配置

项目	说明
规格	kc1.xlarge.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表5-2所示。

表 5-2 操作系统要求

项目	说明	下载地址
CentOS	7.6	在公共镜像中已提供。
Kernel	4.14.0	在公共镜像中已提供。

配置编译环境

安装wget工具。

yum install wget -y

获取源码

获取“v2.1.0-beta3”源码包。

cd /usr/local/src

wget https://github.com/LuaJIT/LuaJIT/archive/v2.1.0-beta3.tar.gz

说明

经过测试，当前v2.1.0版本是支持aarch64架构的CPU的服务器/虚拟机执行，其他的版本不支持

编译和安装

步骤1 解压软件包。

cd /usr/local/src

tar -zxvf v2.1.0-beta3.tar.gz

步骤2 进入LuaJIT的安装目录。

cd LuaJIT-2.1.0-beta3/

步骤3 编译安装LuaJIT。

make -j4 && make install

----结束

运行和验证

步骤1 将生成的LuaJIT可执行文件修改名称，因为生成的LuaJIT可执行程序名称为luajit-2.1.0-beta3，不便于使用。

mv /usr/local/bin/luajit-2.1.0-beta3 /usr/local/bin/luajit

步骤2 创建“test.lua”文件，并输入如下代码。

vi test.lua

插入如下代码：

```
function max(num1, num2)
  if (num1 > num2) then
```



```
result = num1;
  else
result = num2;
  end
  return result;
end
print("1,2 max is",max(1,2))
```

保存退出。

步骤3 测试LuaJIT，生成字节码并执行字节码。

1. 生成字节码文件test。

   luajit -b test.lua test

2. 执行字节码文件。

   luajit test

   回显如下：

   ```
   [root@ecs-lua src]# luajit test
   1,2 max is2
   [root@ecs-lua src]#
   ```

----结束

6 移植 Boost

介绍

简要介绍

Boost是为C++语言标准库提供扩展的一些C++程序库的总称。Boost库是一个可移植、提供源代码的C++库，作为标准库的后备，是C++标准化进程的开发引擎之一，是为C++语言标准库提供扩展的一些C++程序库的总称。

语言：C/C++

一句话概述：为C++语言标准库提供扩展的一些C++程序库的总称

开源协议：custom

建议的版本

- 已在鲲鹏云服务器上验证过“boost_1_66_0”版本，请根据实际需要选择版本。
- 本文档以“boost_1_66_0”为例进行说明。

环境要求

云服务器要求

本文以KC1实例测试，配置如表6-1所示。

表 6-1 云服务器配置

项目	说明
规格	kc1.large.2 | 2vCPUs | 4GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表6-2所示。

表 6-2 操作系统要求

项目	说明	下载地址
CentOS	7.6	在公共镜像中已提供。
Kernel	4.14.0-115	在公共镜像中已提供。

配置编译环境

安装依赖。

yum install wget -y

获取源码

获取“boost_1_66_0”源码包。

cd /usr/local/src

wget https://dl.bintray.com/boostorg/release/1.66.0/source/boost_1_66_0.tar.gz

编译和安装

步骤1 解压软件包。

tar -zxvf boost_1_66_0.tar.gz

步骤2 进入Boost的安装目录。

cd boost_1_66_0

步骤3 编译安装。

./bootstrap.sh –prefix=/usr/local/boost

./b2 -j4

./b2 install

步骤4 配置环境。

1. 修改环境变量。

   vim /etc/profile

   在“/etc/profile”文件末尾增加下面代码：

   ```
   export CPLUS_INCLUDE_PATH=$CPLUS_INCLUDE_PATH:/usr/local/boost/include
   export LIBRARY_PATH=$LIBRARY_PATH:/usr/local/boost/lib
   ```

2. 按“Ecs”，输入“wq!”保存后退出。
3. 运行下面命令，使修改的环境变量生效。

   source /etc/profile

步骤5 将动态库加入到动态链接器中。

1. 修改环境变量。

   vim /etc/ld.so.conf

在“/etc/ld.so.conf”文件末尾增加下面代码：

```
/usr/local/boost/lib
```

2. 按“Ecs”，输入“wq!”保存后退出。
3. 运行下面命令，使修改的环境变量生效。

   sudo ldconfig

----结束

运行和验证

测试Boost库是否安装完成。

步骤1 新建一个最简单程序，文件名命名为“main.cpp”，文件内容如下：

```
#include <boost/thread/thread.hpp>
#include <iostream>
using namespace std;

void NewThread()
{
	cout << "New thread is running..." << endl;
}

int main(int argc, char* argv[])
{
	boost::thread newthread(&NewThread);
	newthread.join();
	return 0;
}
```

步骤2 编译代码。

g++ main.cpp -o test -lboost_system -lboost_thread

步骤3 运行程序。

./test

回显信息如下，则表示安装成功，并且可以正常使用。

```
New thread is running...
```

----结束

7 移植 Lua

7.1 移植指导

介绍

简要介绍

Lua是一种轻量小巧的脚本语言，用标准C语言编写并以源代码形式开放， 其设计目的是为了嵌入应用程序中，从而为应用程序提供灵活的扩展和定制功能。

语言：C

一句话概述：轻量小巧的脚本语言

建议的版本

建议使用版本为“lua-5.1.4”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表7-1所示。

表 7-1 云服务器配置

项目	说明
规格	kc1.xlarge.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表7-2所示。

表 7-2 操作系统要求

项目	版本	下载地址
Euler	2.8	在公共镜像中已提供。
Kernel	4.19.36	在公共镜像中已提供。

获取源码

https://www.lua.org/ftp/ 提供Lua各版本的源码压缩包，可以直接下载。

编译和安装

本文以“lua-5.1.4”版本为例，下载源码，并编译安装。

步骤1 安装lua的依赖包readline-devel。

yum install readline-devel -y

步骤2 下载lua源码。

wget https://www.lua.org/ftp/lua-5.1.4.tar.gz

步骤3 解压lua源码。

tar -zxvf lua-5.1.4.tar.gz

步骤4 编译安装lua。

cd lua-5.1.4/

make PLAT=linux

make install

步骤5 查看lua是否安装成功。

lua -v

回显信息如下，则lua安装成功：

```
Lua 5.1.4  Copyright (C) 1994-2008 Lua.org, PUC-Rio
```

----结束

运行和验证

Lua提供了交互式编程模式，可以在命令行中输入程序并立即查看效果。

步骤1 Lua交互式编程模式可以通过命令“lua”来启用：

lua

回显信息如下：

```
Lua 5.1.4  Copyright (C) 1994-2008 Lua.org, PUC-Rio
>
```

步骤2 在命令行中，输入以下命令:

print("Hello World！")

回显信息如下：

```
> print("Hello World！")
```

步骤3 接着按“enter”回车键，输出结果如下：

回显信息如下，表示lua安装成功并可正常使用。

```
Lua 5.1.4  Copyright (C) 1994-2008 Lua.org, PUC-Rio
> print("Hello World！")
Hello World!
>
```

----结束

7.2 故障排除

问题描述：

Lua编译执行**make**命令后，提示需要指定PLAT，回显信息如下：

```
Please do
   make PLATFORM
where PLATFORM is one of these:
   aix ansi bsd freebsd generic linux macosx mingw posix solaris
See INSTALL for complete instructions.
```

问题原因：make时需要指定PLAT类型为“linux”。

解决方案：使用如下命令执行编译操作。

make PLAT=linux

8 移植 Python

8.1 移植 Python 2

介绍

简要介绍

Python是一种解释型、面向对象、动态数据类型的高级程序设计语言。

语言：C

一句话描述：高级程序设计语言。

开源协议：GPL

建议的版本

建议使用版本为“python-2.7.16”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表8-1所示。

表 8-1 云服务器配置

项目	说明
规格	kc1.large.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表8-2所示。

表 8-2 操作系统要求

项目	版本	下载地址
EulerOS	2.8	在公共镜像中已提供
Kernel	4.19.36	包含在操作系统镜像中。

配置编译环境

执行如下命令，安装依赖包。

yum install zlib zlib-devel openssl openssl-devel -y

获取源码

执行如下命令，获取“python-2.7.16”源码包。

cd /usr/local/src

wget https://www.python.org/ftp/python/2.7.16/Python-2.7.16.tgz

编译和安装

步骤1 执行如下命令，解压软件包。

tar -zxvf Python-2.7.16.tgz

步骤2 执行如下命令，进入python安装目录。

cd Python-2.7.16

步骤3 执行如下命令，配置编译并安装python源码。

./configure --enable-optimizations

make -j4

make install

----结束

其他使用技巧

为了安装后的python开启ssl模块，在make之前，先修改“Modules/Setup”文件，把下面方框的内容解除掉注释，修改内容如图8-1所示。

图 8-1 注释内容

```
# Socket module helper for SSL support; you must comment out the other
# socket line above, and possibly edit the SSL variable:
#SSL=/usr/local/ssl
#_ssl _ssl.c \
#       -DUSE_SSL -I$(SSL)/include -I$(SSL)/include/openssl \
#       -L$(SSL)/lib -lssl -lcrypto

# The crypt module is now disabled by default because it breaks builds
# on many systems (where -lcrypt is needed), e.g. Linux (I believe).
#
# First, look at Setup.config; configure may have set this for you.
```

8.2 移植 Python 3

介绍

简要介绍

Python是一种解释型、面向对象、动态数据类型的高级程序设计语言。Python的3.0版本，常被称为Python 3000，或简称Py3k。相对于Python的早期版本，这是一个较大的升级。为了不带入过多的累赘，Python 3.0 在设计的时候没有考虑向下兼容。

开发语言：C

一句话描述：高级程序设计语言。

开源协议：GPL

建议的版本

已在鲲鹏云服务器上验证过下述版本，请根据实际需要选择版本。

- Python-3.8.1
- Python-3.7.6
- Python-3.6.8
- Python-3.5.6
- Python-3.4.5

本文档以“Python-3.8.1”为例进行说明。

环境要求

云服务器要求

本文以KC1实例测试，配置如表8-3所示。

表 8-3 云服务器配置

项目	说明
规格	kc1.large.2 | 2vCPUs | 4GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表8-4所示。

表 8-4 操作系统要求

项目	说明	下载地址
CentOS	7.6	在公共镜像中已提供。
Kernel	4.14.0-115	在公共镜像中已提供。

配置编译环境

安装依赖工具。

yum install wget zlib* openssl openssl-devel libffi-devel -y

获取源码

获取“Python-3.8.1”源码包。

cd /usr/local/src

wget https://www.python.org/ftp/python/3.8.1/Python-3.8.1.tgz

编译和安装

步骤1 解压软件包。

cd /usr/local/src

tar -xvf Python-3.8.1.tgz

步骤2 进入Python3的安装目录。

cd Python-3.8.1

步骤3 安装Python3。

“-j”参数可利用多核CPU加快编译速度，在本示例中，使用的是2核CPU，所以此处为“-j2”。

可通过下述命令查询CPU核数：

cat /proc/cpuinfo| grep "processor"| wc -l

./configure --enable-optimizations --prefix=/opt/Python-3.8.1

make -j2 && make altinstall

ln -s /opt/Python-3.8.1/ /opt/python

ln -s /opt/python/bin/python3.8 /usr/bin/python3

ln -s /opt/python/bin/pip3.8 /usr/bin/pip3

ln -s /opt/python/bin/pydoc3 /usr/bin/pydoc3

----结束

运行和验证

测试Python3是否安装完成。

执行如下命令，查看Python3版本。

python3 --version

pip3 --version

返回内容如下所示，表示安装已经完成。

```
[root@ecs bin]# python3 --version
Python 3.8.1
[root@ecs bin]# pip3 --version
pip 19.2.3 from /usr/local/lib/python3.8/site-packages/pip (python 3.8)
```


9 安装 PHP-FPM

介绍

简要介绍

PHP-FPM(PHP FastCGI Process Manager)，PHP FastCGI进程管理器，用于管理PHP进程池的软件，用于接受web服务器的请求。PHP-FPM提供了更好的PHP进程管理方式，可以有效控制内存和进程、可以平滑重载PHP配置。

语言：C

一句话描述：可以更好地有效管理PHP进程。

开源协议：PHP

建议的版本

建议使用版本为“php-fpm-7.2.10”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表9-1所示。

表 9-1 云服务器配置

项目	说明
规格	kc1.large.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表9-2所示。

表 9-2 操作系统要求

项目	说明	下载地址
CentOS	• 版本：7.5 • Kernel：4.14.0	在公共镜像中已提供
EulerOS	• 版本：2.8 • Kernel：4.19.36	在公共镜像中已提供

配置安装环境

执行如下命令，安装php（ >=7 ）依赖包。

yum install php

安装 php-fpm

直接通过yum源下载安装方式，安装步骤如下。

步骤1 执行如下命令，安装。

yum install php-fpm.aarch64

步骤2 设置开机启动。

chkconfig php-fpm on

系统回显如下，则开机启动配置成功

```
Note: Forwarding request to 'systemctl enable php-fpm.service'.
Created symlink /etc/systemd/system/multi-user.target.wants/php-fpm.service → /usr/lib/systemd/system/
php-fpm.service.
```

----结束

运行和验证

启动php-fpm

service php-fpm start

10 安装 Scala

介绍

简要介绍

Scala是一门多范式（multi-paradigm）的编程语言，设计初衷是要集成面向对象编程和函数式编程的各种特性。Scala运行在Java虚拟机上，并兼容现有的Java程序。

语言：Scala

一句话描述：多范式的编程语言

开源协议：BSD

建议的版本

建议使用版本为“scala-2.13.0”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表10-1所示。

表 10-1 云服务器配置

项目	说明
规格	kc1.large.2 | 2vCPUs | 4GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表10-2所示。

表 10-2 操作系统要求

项目	说明	下载地址
CentOS	7.6	在公共镜像中已提供。
Kernel	4.14.0-115	在公共镜像中已提供。

配置安装环境

安装wget和openjdk。

yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel wget -y

获取软件包

获取“scala-2.13.0”安装包。

cd /usr/local/src

wget https://downloads.lightbend.com/scala/2.13.0/scala-2.13.0.zip

安装

步骤1 解压软件包。

cd /usr/local/src

unzip scala-2.13.0.zip

步骤2 配置Scala环境变量。

vi /etc/profile

在倒数第三行插入“export PATH=$PATH:/usr/local/src/scala-2.13.0/bin”，如下面加粗部分所示。

```
done
export PATH=$PATH:/usr/local/src/scala-2.13.0/bin/
unset i
unset -f pathmunge
```

步骤3 使环境变量生效。

source /etc/profile

----结束

运行和验证

步骤1 创建测试源文件。

cd /usr/local/src

vi test.scala

在“test.scala”插入如下内容：

```
object HelloWorld {
   def main(args: Array[String]): Unit = {
```



```
        println("Hello, world!")
    }
}
```

步骤2 运行测试文件。

scala test.scala

```
[root@ecs-scale-x src]# scala test.scala
Hello, world!
[root@ecs-scale-x src]#
```

----结束

11 移植 Erlang

介绍

简要介绍

Erlang是一种通用的面向并发的编程语言，它由瑞典电信设备制造商爱立信所辖的CS-Lab开发，目的是创造一种可以应对大规模并发活动的编程语言和运行环境。

语言：Erlang

一句话描述：面向并发的编程语言

开源协议：Apache

建议的版本

建议使用版本为“erlang-20.3”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表11-1所示。

表 11-1 云服务器配置

项目	说明
规格	kc1.large.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表11-2所示。

表 11-2 操作系统要求

项目	说明	下载地址
CentOS	• 版本：7.5 • Kernel：4.14.0	在公共镜像中已提供
EulerOS	• 版本：2.8 • Kernel：4.19.36	在公共镜像中已提供

配置编译环境

执行如下命令，安装依赖包。

yum install libtool libtool-ltdl-devel libevent-devel lua ncurses-devel openssl-devel flex

获取源码

执行如下命令，获取“erlang_20.3”源码包。

cd /usr/local/src

wget http://erlang.org/download/otp_src_20.3.tar.gz

编译和安装

步骤1 执行如下命令，解压软件包。

tar -zxvf otp_src_20.3.tar.gz

步骤2 执行如下命令，进入erlang安装目录。

cd otp_src_20.3

步骤3 执行如下命令，配置编译并安装erlang源码。

./configure && make && make install

----结束

运行和验证

步骤1 执行如下命令，进入erl环境。

erl

当系统回显类似如下信息时，表示进入erl环境。

```
Erlang/OTP 20 [erts-9.3] [source] [64-bit] [smp:4:4] [ds:4:4:10] [async-threads:10] [kernel-poll:false]

Eshell V9.3  (abort with ^G)
1>
```

步骤2 输入“**1+1.**”后按“**Enter**”。

系统回显如下所示，表示erlang基本功能调测试成功。


```
1> 1+1.
2
2>
```

----结束

12 安装 Gradle

12.1 安装指导

介绍

简要介绍

Gradle是一个基于Apache Ant和Apache Maven概念的开源构建自动化系统，它使用一种基于Groovy的特定领域语言(DSL)来声明项目设置，抛弃了基于XML的各种繁琐配置。

编写语言：Groovy

一句话概述：自动化构建开源工具

建议的版本

建议使用版本为“3.5”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表12-1所示。

表 12-1 云服务器配置

项目	说明
规格	kc1.xlarge.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表12-2所示。

表 12-2 操作系统要求

项目	版本	下载地址
CentOS	7.5	在公共镜像中已提供。
Kernel	4.14.0	在公共镜像中已提供。

配置安装环境

配置Gradle依赖的JDK环境。

步骤1 安装Gradle依赖的JDK。

yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel -y

步骤2 配置JDK环境变量。

vi /etc/profile

1. 在倒数第三行插入如下内容：
```
export JAVA_HOME=/usr/lib/jvm/java-openjdk
export CLASSPATH=$CLASSPATH:$JAVA_HOME/lib:$JAVA_HOME/jre/lib
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH:$HOME/bin
```
2. 使环境变量生效。

   source /etc/profile

----结束

获取软件包

https://gradle.org/releases/ 提供gradle各版本的源码压缩包，可以直接下载。

安装

本文以“gradle-3.5”版本为例，下载源码，并解压安装。

步骤1 下载Gradle源码。

wget https://downloads.gradle.org/distributions/gradle-3.5-all.zip

步骤2 解压Gradle源码。

unzip gradle-3.5-all.zip

步骤3 配置环境变量。

vi /etc/profile

1. 修改PATH环境变量，在PATH变量后追加 “:/root/gradle-3.5/bin”。
```
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH:$HOME/bin:/root/gradle-3.5/bin
```
2. 使环境变量生效。

   source /etc/profile

步骤4 检查安装的Gradle版本号及其他信息。

gradle -v

步骤5 回显信息如下，则Gradle安装成功。

```
------------------------------------------------------------
Gradle 3.5
------------------------------------------------------------

Build time:   2017-04-10 13:37:25 UTC
Revision:     b762622a185d59ce0cfc9cbc6ab5dd22469e18a6

Groovy:       2.4.10
Ant:          Apache Ant(TM) version 1.9.6 compiled on June 29 2015
JVM:          1.8.0_212 (Oracle Corporation 25.212-b04)
OS:           Linux 4.14.0-115.5.1.el7a.aarch64 aarch64
```

----结束

运行和验证

可以测试Gradle构建脚本是否能正常运行，以此来验证Gradle安装成功且能正常使用。

步骤1 创建名为“build.gradle”的构建脚本。

1. 创建Gradle构建脚本保存的路径。

   mkdir -p /etc/gradle

2. 编辑名为“build.gradle”的构建脚本文件。

   vi /etc/gradle/buile.gradle

   ```
   task hello {
       doLast {
           println 'Hello world!'
       }
   }
   ```

3. 保存并退出。

步骤2 运行构建任务。

gradle -q hello

参数说明：

“-q”表示控制台只显示任务的输出。

回显信息如下，则表示Gradle安装成功且可正常使用。

```
Hello world!
```

----结束

12.2 故障排除

问题描述：

Gradle解压完成后，执行**gradle -v**命令查询版本号，报以下错误：

```
ERROR: JAVA_HOME is not set and no 'java' command could be found in your PATH.

Please set the JAVA_HOME variable in your environment to match the
location of your Java installation.
```

问题原因：没有安装gradle的依赖JDK。

解决方法：请参见**安装指导**中“配置编译环境”的描述安装JDK。

13 安装 Maven

介绍

简要介绍

Maven是一个软件项目管理和综合工具。基于项目对象模型（POM）的概念，Maven可以从一个中心资料片管理项目构建，报告和文件。

语言：Java

一句话描述：软件项目管理和综合工具

建议的版本

建议使用版本为“maven-3.6.1”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表13-1所示。

表 13-1 云服务器配置

项目	说明
规格	kc1.xlarge.2 | 4vCPUs | 8GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表13-2所示。

表 13-2 操作系统要求

项目	版本	下载地址
EulerOS	2.8	在公共镜像中已提供。

项目	版本	下载地址
Kernel	4.19.36	在公共镜像中已提供。

配置安装环境

要求Java的OpenJDK版本至少为“1.8.0”，可以根据以下命令安装：

yum install java-1.8.0-openjdk

获取软件包

执行以下命令，获取maven软件包。

wget https://archive.apache.org/dist/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.tar.gz

安装

步骤1 解压Maven软件包，并移动到待放置的目录。

tar -zxvf apache-maven-3.6.1-bin.tar.gz

rm -rf /usr/local/maven

mv apache-maven-3.6.1 /usr/local/maven

步骤2 配置Maven环境变量，将如下内容添加到“/etc/profile”文件尾部：

```
MAVEN_HOME=/usr/local/maven
export PATH=${MAVEN_HOME}/bin:$PATH
```

步骤3 加载环境变量。

source /etc/profile

----结束

运行和验证

执行以下命令，查看Maven版本号。

mvn -v

回显内容如下，则证明Maven成功安装。

```
Apache Maven 3.6.1 (d66c9c0b3152b2e69ee9bac180bb8fcc8e6af555; 2019-04-05T03:00:29+08:00)
Maven home: /usr/local/maven
Java version: 1.8.0_181, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-1.8.0-
openjdk-1.8.0.181.b15-5.h3.eu
lerosv2r8.aarch64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.19.36-vhulk1905.1.0.h276.eulerosv2r8.aarch64", arch: "aarch64", family: "unix"
```


14 安装 Ant

介绍

简要介绍

ant是一个将软件编译、测试、部署等步骤联系在一起加以自动化的工具，大多用于Java环境中的软件开发。

语言：Java

一句话描述：将软件编译等步骤联系并加以自动化的工具

开源协议：Apache License Version 2.0

建议的版本

建议使用版本为“apache-ant-1.10.6”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表14-1所示。

表 14-1 云服务器配置

项目	说明
规格	kc1.large.2 | 2vCPUs | 4GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表14-2所示。

表 14-2 操作系统要求

项目	说明	下载地址
CentOS	7.6	在公共镜像中已提供。
Kernel	4.14.0-115	在公共镜像中已提供。

配置安装环境

安装wget工具。

yum install wget -y

获取软件包

获取“apache-ant-1.10.6”软件包。

cd /usr/local/src

wget https://archive.apache.org/dist/ant/binaries/apache-ant-1.10.6-bin.tar.gz

安装

步骤1 解压软件包。

cd /usr/local/src

tar -zxvf apache-ant-1.10.6-bin.tar.gz

步骤2 设置环境变量。

1. 打开配置文件。

   vim /etc/profile

   修改脚本内容，在倒数第三行添加以下配置：

   ```
   export ANT_HOME=/usr/local/src/apache-ant-1.10.6
   export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$PATH
   ```

2. 保存并退出。
3. 使配置文件生效。

   source /etc/profile

----结束

运行和验证

查询ant版本号。

ant -version

回显信息如下，则表示ant安装成功。

```
Apache Ant(TM) version 1.10.6 compiled on May 2 2019
```


15 安装 clang

介绍

简要介绍

clang是一个C++编写、基于LLVM、发布于LLVM BSD许可证下的C/C++/Objective-C/Objective-C++编译器。

语言：C++

一句话描述：C/C++/Objective-C/Objective-C++编译器

开源协议：Apache License Version 2.0

建议的版本

建议使用版本为“clang-3.4.2”。

环境要求

云服务器要求

本文以云服务器KC1实例测试，云服务器配置如表15-1所示。

表 15-1 云服务器配置

项目	说明
规格	kc1.xlarge.2 | 4vCPUs | 14GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表15-2所示。

表 15-2 操作系统要求

项目	说明	下载地址
CentOS	7.5	在公共镜像中已提供。
Kernel	4.14.0-49	在公共镜像中已提供。

安装

安装clang。

yum install clang -y

回显如下信息，则clang安装完成：

```
Installed:
  clang.aarch64 0:3.4.2-9.el7

Dependency Installed:
  llvm.aarch64 0:3.4.2-9.el7                    llvm-libs.aarch64 0:3.4.2-9.el7

Complete!
```

运行和验证

查询clang版本号。

clang --version

回显信息如下，则表示clang安装成功。

```
clang version 3.4.2 (tags/RELEASE_34/dot2-final)
Target: aarch64-redhat-linux-gnu
Thread model: posix
```

16 移植 Cmake

介绍

简要介绍

CMake是一个跨平台的安装/编译工具，可以用简单的语句来描述所有平台的安装（编译过程）。他能够输出各种各样的makefile或者project文件，能测试编译器所支持的C++特性，类似UNIX下的automake。

语言：C/C++

一句话描述：自动化构建系统

开源协议：BSD

建议的版本

已在鲲鹏云服务器上验证过下述版本，请根据实际需要选择版本。

- cmake-3.9.2
- cmake-3.16.1

本文档以“cmake-3.9.2”为例进行说明。

环境要求

云服务器要求

本文以KC1实例测试，配置如表16-1所示。

表 16-1 云服务器配置

项目	说明
规格	kc1.large.2 | 2vCPUs | 4GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表16-2所示。

表 16-2 操作系统要求

项目	说明	下载地址
CentOS	7.6	在公共镜像中已提供。
Kernel	4.14.0-115	在公共镜像中已提供。

配置编译环境

安装wget工具。

yum install wget -y

获取源码

获取“cmake-3.9.2”源码包。

cd /usr/local/src

wget https://cmake.org/files/v3.9/cmake-3.9.2.tar.gz

编译和安装

步骤1 解压软件包。

cd /usr/local/src

tar -zxvf cmake-3.9.2.tar.gz

步骤2 进入CMake的安装目录。

cd cmake-3.9.2

步骤3 安装CMake。

./configure

make && make install

----结束

运行和验证

测试CMake是否安装完成。

cmake -version

返回内容如下所示，表示安装已经完成。

```
[root@ecs-centos cmake-3.9.2]# cmake -version
cmake version 3.9.2

CMake suite maintained and supported by Kitware (kitware.com/cmake).
```


17 移植 GCC

介绍

简要介绍

GNU编译器套装（英语：GNU Compiler Collection，缩写为GCC），指一套编程语言编译器，以GPL及LGPL许可证所发行的自由软件，也是GNU计划的关键部分，也是GNU工具链的主要组成部分之一。GCC（特别是其中的C语言编译器）也常被认为是跨平台编译器的事实标准。

语言：C

一句话描述：一套编程语言编译器

开源协议：GPL/LGPL

建议的版本

已在鲲鹏云服务器上验证过下述版本，请根据实际需要选择版本。

- gcc-7.3.0
- gcc-9.2.0

本文档以“gcc-7.3.0”为例进行说明。

环境要求

云服务器要求

本文以KC1实例测试，配置如表17-1所示。

表 17-1 云服务器配置

项目	说明
规格	kc1.large.2 | 2vCPUs | 4GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表17-2所示。

表 17-2 操作系统要求

项目	说明	下载地址
CentOS	7.6	在公共镜像中已提供。
Kernel	4.14.0-115	在公共镜像中已提供。

配置编译环境

安装依赖工具。

yum install wget lbzip2 -y

获取源码

获取“gcc-7.3.0”源码包。

cd /usr/local/src

wget https://ftp.gnu.org/gnu/gcc/gcc-7.3.0/gcc-7.3.0.tar.gz

编译和安装

步骤1 解压软件包。

tar -zxvf gcc-7.3.0.tar.gz

步骤2 进入gcc的安装目录。

cd gcc-7.3.0/

步骤3 下载isl、gmp、mpc、mpfr。

./contrib/download_prerequisites

上述命令会下载依赖包“gmp-6.1.0.tar.bz2”、“isl-0.16.1.tar.bz2”、“mpc-1.0.3.tar.gz”或“mpfr-3.1.4.tar.bz2”，如果某依赖包下载失败，可根据需要执行相应的命令下载。

wget https://gcc.gnu.org/pub/gcc/infrastructure/gmp-6.1.0.tar.bz2

wget https://gcc.gnu.org/pub/gcc/infrastructure/isl-0.16.1.tar.bz2

wget https://gcc.gnu.org/pub/gcc/infrastructure/mpc-1.0.3.tar.gz

wget https://gcc.gnu.org/pub/gcc/infrastructure/mpfr-3.1.4.tar.bz2

步骤4 生成Makefile文件。

./configure --prefix=/usr

步骤5 编译安装gcc。

“-j”参数可利用多核CPU加快编译速度，在本示例中，使用的是2核CPU，所以此处为“-j2”。

可通过下述命令查询CPU核数：

cat /proc/cpuinfo| grep "processor"| wc -l

make -j2

make install

----结束

运行和验证

查询gcc版本号。

gcc --version

回显信息如下，则表示gcc安装成功。

```
gcc (GCC) 7.3.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```


18 移植 LLVM

介绍

简要介绍

LLVM是一个自由软件项目，它是一种编译器基础设施，以C++写成，包含一系列模块化的编译器组件和工具链，用来开发编译器前端和后端。它是为了任意一种编程语言而写成的程序，利用虚拟技术创造出编译时期、链接时期、运行时期以及“闲置时期”的最优化。

LLVM的主要子项目是：

LLVM Core库提供了现代的，与源和目标无关的优化器，并为许多流行的CPU（以及一些较不常见的CPU！）提供了代码生成支持。

Clang是“ LLVM原生” C/C++/Objective-C编译器，旨在提供惊人的快速编译（例如，在调试配置中编译Objective-C代码时，编译速度比GCC快3倍），非常有用的错误和警告消息，以及提供一个用于构建出色的源代码级工具的平台。

LLDB项目建立在LLVM和Clang提供的库的基础上，以提供出色的本机调试器。

libc++和libc++ ABI项目提供了C++标准库的符合标准和高性能的实现，包括对C++11和C++14的完全支持。

OpenMP子项目提供了一个OpenMP运行时，可用于Clang中的OpenMP实现。

polly项目使用多面体模型实现了一套缓存局部性优化以及自动并行化和矢量化功能。

libclc项目旨在实现OpenCL标准库。

LLD项目是一个新的链接器。这是系统链接程序的直接替代，并且运行速度更快。

语言：C++

一句话描述：模块化和可重复使用的编译器和工具链技术的集合

开源协议：custom:Apache 2.0 with LLVM Exception

建议的版本

建议使用版本为“llvm-9.0.0”。

环境要求

云服务器要求

本文以KC1实例测试，配置如表18-1所示。

表 18-1 云服务器配置

项目	说明
规格	kc1.large.2 | 2vCPUs | 4GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表18-2所示。

表 18-2 操作系统要求

项目	说明	下载地址
CentOS	7.6	在公共镜像中已提供。
Kernel	4.14.0-115	在公共镜像中已提供。

配置编译环境

步骤1 安装wget依赖工具。

yum install wget -y

步骤2 升级GCC版本。

鲲鹏默认的GCC版本为4.8.5，编译llvm时候，需要不低于GCC5.1的版本。请参考https://www.huaweicloud.com/kunpeng/software/gcc.html对GCC版本进行升级

步骤3 升级CMake版本。

配置要求CMake最低版本为3.4.3，请参考https://www.huaweicloud.com/kunpeng/software/cmake.html对CMake进行安装或者版本升级。

----结束

获取源码

获取“llvm-9.0.0”源码包。

cd /usr/local/src

mkdir llvm

cd llvm

wget http://releases.llvm.org/9.0.0/llvm-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/cfe-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/compiler-rt-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/libcxx-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/libcxxabi-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/libunwind-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/lld-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/lldb-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/openmp-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/polly-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/clang-tools-extra-9.0.0.src.tar.xz

wget http://releases.llvm.org/9.0.0/test-suite-9.0.0.src.tar.xz

编译和安装

步骤1 解压软件包。

tar -xvf llvm-9.0.0.src.tar.xz

步骤2 进入gcc的安装目录。

cd llvm-9.0.0.src

mkdir b

cd b

步骤3 生成Makefile文件

cmake .. -DCMAKE_BUILD_TYPE=Release -DLLVM_ENABLE_ASSERTIONS=ON

步骤4 编译安装llvm。

“-j”参数可利用多核CPU加快编译速度，在本示例中，使用的是2核CPU，所以此处为“-j2”。

可通过下述命令查询CPU核数：

cat /proc/cpuinfo| grep "processor"| wc -l

make -j2

make install

----结束

运行和验证

步骤1 查询安装的llvm工具。

输入**llvm-**然后按Tab键，回显信息如下，则表示llvm安装了如下工具。

```
llvm-addr2line  llvm-config     llvm-cxxmap     llvm-elfabi     llvm-lipo       llvm-mt         llvm-
profdata    llvm-size       llvm-tblgen
llvm-ar         llvm-cov        llvm-diff       llvm-exegesis   llvm-lto        llvm-nm         llvm-ranlib     llvm-
split       llvm-undname
llvm-as         llvm-c-test     llvm-dis        llvm-extract    llvm-lto2       llvm-objcopy    llvm-rc         llvm-
stress      llvm-xray
llvm-bcanalyzer  llvm-cvtres     llvm-dlltool    llvm-jitlink    llvm-mc         llvm-objdump    llvm-
```



```
readelf     llvm-strings
llvm-cat        llvm-cxxdump     llvm-dwarfdump   llvm-lib        llvm-mca        llvm-opt-report  llvm-
readobj     llvm-strip
llvm-cfi-verify  llvm-cxxfilt     llvm-dwp        llvm-link       llvm-modextract  llvm-pdbutil     llvm-rtdyld
llvm-symbolizer
```

步骤2 选择一个工具查看版本信息。

llvm-nm --version

回显信息如下，则表示该工具安装成功，其他工具验证类似。

```
LLVM (http://llvm.org/):
  LLVM version 9.0.0
  Optimized build with assertions.
  Default target: aarch64-unknown-linux-gnu
  Host CPU: tsv110
```

----结束

19 移植 NumPy

介绍

简要介绍

NumPy（Numerical Python）是Python的一种开源的数值计算扩展。这种工具可以用来存储和处理大型矩阵，比Python自身嵌套列表结构要高效得多，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。

语言：Python

一句话描述：Python数学库

开源协议：BSD3

建议的版本

- 已在鲲鹏云服务器上验证过“Numpy-1.15.4”版本，请根据实际需要选择版本。
- 本文档以“Numpy-1.15.4”为例进行说明。

环境要求

云服务器要求

本文以KC1实例测试，配置如表19-1所示。

表 19-1 云服务器配置

项目	说明
规格	kc1.large.2 | 2vCPUs | 4GB
磁盘	系统盘：高IO（40GB）

操作系统要求

操作系统要求如表19-2所示。

表 19-2 操作系统要求

项目	说明	下载地址
CentOS	7.6	在公共镜像中已提供。
Kernel	4.14.0-115	在公共镜像中已提供。

配置编译环境

步骤1 安装Python3.7.6。

请参见**https://support.huaweicloud.com/prtg-kunpengcpl/python_02_0001.html**安装。

步骤2 安装gcc-gfortran。

yum -y install gcc-gfortran

步骤3 安装OpenBLAS。

cd /usr/local/src

wget https://github.com/xianyi/OpenBLAS/archive/v0.3.8.tar.gz

tar -zxvf v0.3.8.tar.gz && cd OpenBLAS-0.3.8

make -j2

make PREFIX=/usr/local/openblas install

步骤4 配置OpenBLAS环境，将“export LD_LIBRARY_PATH=/usr/local/openblas/lib:$LD_LIBRARY_PATH”写入“~/.bashrc”文件最后一行。

vim ~/.bashrc

```
export LD_LIBRARY_PATH=/usr/local/openblas/lib:$LD_LIBRARY_PATH
```

步骤5 使配置文件生效。

source ~/.bashrc

----结束

获取源码

获取源码。

cd /usr/local/src

wget https://github.com/numpy/numpy/releases/download/v1.15.4/numpy-1.15.4.tar.gz

编译和安装

步骤1 解压并进入源码目录。

tar -zxvf numpy-1.15.4.tar.gz && cd numpy-1.15.4

步骤2 更改配置文件，设置OpenBLAS库的路径。

cp site.cfg.example site.cfg

vim site.cfg

将openblas设置的如下四行更改成如下所示：

```
[openblas]
librarys = openblas
library_dirs = /usr/local/openblas/lib
include_dirs = /usr/local/openblas/include
```

步骤3 编译NumPy。

python3 setup.py install

----结束

运行和验证

步骤1 Python中引入NumPy模块使用。

python3

import numpy as np

print(np.version)

打印出类似如下版本信息，说明NumPy可以正常使用。

```
<module 'numpy.version' from '/opt/python/lib/python3.7/site-packages/numpy-1.15.4-py3.7-linux-aarch64.egg/numpy/version.py'>
```

----结束

A 修订记录

发布日期	修订记录
2020-08-06	第四次正式发布。 增加Boost的指导。
2020-06-18	第三次正式发布。 增加LuaJIT的指导。
2020-06-02	第二次正式发布。 增加NumPy的指导。
2020-03-18	第一次正式发布。