

# Ubuntu20.04安装Slurm 21.08.6（实践篇）

说明：这是笔者在Ubuntu 20.04上亲自实践安装Slurm21.08.6，如果按照本教程会顺利安装完毕。

## 一、安装必要库文件

apt-get install make hwloc libhwloc-dev libmunge-dev libmunge2 munge mariadb-server libmysqlclient-dev -y

## 二、启动munge服务

systemctl enable munge // 设置munge开机自启动

systemctl start munge // 启动munge服务

systemctl status munge // 查看munge状态

## 三、编译安装slurm（源码安装方式）

# 将slurm-21.08.6.tar.bz2源码包放置在/home/fz/package目录下

cd /home/fz/package

tar -jxvf slurm-21.08.6.tar.bz2

cd slurm-21.08.6/

./configure --prefix=/opt/slurm/21.08.6 --sysconfdir=/opt/slurm/21.08.6/etc

make -j16

make install

# Ubuntu服务器安装配置slurm (Ubuntu 22.04 LTS)

## 1. slurm简介

Slurm 全称 **S**imple **L**inux **U**tility for **R**esource **M**anagement。通常被用于大型Linux服务器 (超算) 上，作为任务管理系统。本文详细讲述如何在 Ubuntu 22.04 LTS 上安装slurm，并进行简单的配置。

其实网上相关的教程已经非常多，但在旧版本的Ubuntu上安装slurm时，通常需要安装一个名为slurm-llnl的软件包。但Ubuntu 22.04 LTS 的软件源不包含slurm-llnl，强行安装就会报出如下的错误：apt install slurm-llnl

## 2. slurm的安装步骤

Step 1. 安装依赖的软件包

1. slurmd: 完成计算节点的任务（启动任务、监控任务、分层通信）
2. slurmctld: 完成管理节点的任务（故障切换、资源监控、队列管理、作业调度）

$ sudo apt update

#计算节点和管理节点上都安装，用来配置slurm文件

$ sudo apt install slurm-wlm

*# `slurmd`: compute node daemon 计算节点上安装*

$ sudo apt install slrumd

*# `slurmctld`: central management daemon 管理节点上安装*

$ sudo apt install slurmctld

Step 2. 找到slurm-wlm-configurator.html文件，进入该目录下

# 输入以下命令，并

$ dpkg -L slurmctld | grep slurm-wlm-configurator.html

/usr/share/doc/slurmctld/slurm-wlm-configurator.html

$ cd /usr/share/doc/slurmctld

$ chmod +r slurm-wlm-configurator.html

Step 3. 利用 web 生成配置文件

$ python3 -m http.server

Serving HTTP on 0.0.0.0 port 8000 **(**http://0.0.0.0:8000/**)** ...

打开浏览器，输入[http://<your\_ip>:8000/](https://link.zhihu.com/?target=http%3A//10.134.156.187%3A8000/)，进入配置页面（如下图），点击进入 slurm-wlm-configurator.html 按照自己的需求填写设置。



web 生成slurm.conf

填写完毕后，点击submit，将生成的内容拷贝进 /etc/slurm/slurm.conf （slurm 的配置文件）

# 创建

$ sudo touch /etc/slurm/slurm.conf

# 将网页生成的内容 copy 进来

$ sudo vim /etc/slurm/slurm.conf

# ctrl + v

Step 4. 手动创建slurm的输出文件目录

$ sudo mkdir /var/spool/slurm/d

$ sudo mkdir /var/spool/slurmctld

Step 5. 启动 slurm 服务

# 启动 slurmd, 日志文件路径为 `/var/log/slurmd.log`

$ sudo systemctl start slurmd

# 启动 slurmctld, 日志文件路径为 `/var/log/slurmctld.log`

$ sudo systemctl start slurmctld

启动后无法正常使用slurm的话，先查看slurmd和slurmctld的状态，打开日志查看报错。

# 查看 slurmd 的状态

$ sudo systemctl status slurmd

# 查看 slurmctld 的状态

$ sudo systemctl status slurmctld

## 3. slurm.conf 中几个关键 column 的填写

C1. ClusterName

集群名，随便取

C2. SlurmctldHost

管理节点的主机名

# 获取主机名

$ hostname -s

mu01

C3. SlurmUser

最好 `SlurmUser=root`，权限最高，填写日志文件不会由于权限问题报错。

C4. 管理节点和计算节点的配置（slurm.conf的最后三行）

此处以单节点集群举例（单个节点既作为管理节点，又作为计算节点）

EnforcePartLimits=ALL

NodeName=mu01 CPUs=36 State=UNKNOWN # 本行可以通过 `slurmd -C` 获取

PartitionName=compute Nodes=mu01 Default=YES MaxTime=INFINITE State=UP # 创建一个名为compute的队列

slurmd -C的输出:

$ slurm -C

NodeName**=**mu01 CPUs**=**36 Boards**=**1 SocketsPerBoard**=**1 CoresPerSocket**=**10 ThreadsPerCore**=**2 RealMemory**=**63962

# [在 Linux 环境（Ubuntu）下安装 Slurm 和 OpenMPI](https://www.cnblogs.com/aobaxu/p/16195237.html)

## 安装 Slurm

从软件源安装 slurm-wlm（每个节点都需要装的执行工具）、slurm-client（客户机装的提交命令的工具）、munge（节点间通信插件）。

sudo apt install slurm-wlm slurm-client munge

编写 slurm.conf 文件或者使用官网的[configurator.html](https://slurm.schedmd.com/configurator.html)生成：

# 控制节点名称

ControlMachine=AOBA-ALIENWARE

# 控制节点 IP

ControlAddr=127.0.1.1

CacheGroups=0

JobCredentialPrivateKey=/usr/local/etc/slurm/slurm.key

JobCredentialPublicCertificate=/usr/local/etc/slurm/slurm.cert

GroupUpdateTime=2

MailProg=/bin/true

MpiDefault=none

ProctrackType=proctrack/linuxproc

ReturnToService=1

SlurmctldPort=6817

SlurmdPidFile=/var/run/slurmd.%n.pid

SlurmdPort=6818

SlurmdSpoolDir=/var/spool/slurmd.%n

# slurm 执行用户

SlurmUser=slurm

# slurmd 守护程序执行用户

SlurmdUser=root

StateSaveLocation=/var/spool/slurmctld/state

SwitchType=switch/none

TaskPlugin=task/none

BatchStartTimeout=2

EpilogMsgTime=1

InactiveLimit=0

KillWait=2

MessageTimeout=2

MinJobAge=2

SlurmctldTimeout=2

SlurmdTimeout=2

Waittime=0

SchedulerTimeSlice=5

SchedulerType=sched/backfill

SchedulerPort=7321

SelectType=select/linear

AccountingStorageType=accounting\_storage/filetxt

AccountingStorageLoc=/var/log/slurm/accounting

AccountingStoreJobComment=YES

ClusterName=mycluster

JobCompLoc=/var/log/slurm/job\_completions

JobCompType=jobcomp/filetxt

JobAcctGatherFrequency=2

JobAcctGatherType=jobacct\_gather/linux

SlurmctldDebug=3

SlurmdDebug=3

SlurmdLogFile=/var/log/slurm-llnl/slurmd.%n.log

# 节点信息

# NodeName 名称、Procs 处理器分配数、NodeAddr 地址、Port 端口、State 初始状态

NodeName=AOBA-ALIENWARE Procs=4 NodeAddr=127.0.1.1 Port=17001 State=UNKNOWN

# 执行模式

# PartitionName 名称、Nodes 使用节点、Default 默认、MaxTime 最大使用时间、State 初始状态

PartitionName=mypartition Nodes=AOBA-ALIENWARE Default=YES MaxTime=INFINITE State=UP

复制slurm.conf到/etc/slurm-llnl/文件夹下（多节点使用 scp 分发到每个节点）

sudo cp slurm.conf /etc/slurm-llnl/slurm.conf

测试配置文件

# 测试计算节点守护程序 slurmd

sudo slurmd –D

# 测试控制节点守护程序 slurmctld

sudo slurmctld –D

如果出现错误例如 File or Directory not found 等，一般是文件夹未建立，复制文件夹路径，使用 mkdir 建立，例如

sudo mkdir '/var/spool/slurmctld/state'

重新启动服务（本文使用 service 服务）

# 控制节点守护程序

sudo service slurmctld restart

# 计算节点守护程序

sudo service slurmd restart

# 通信插件

sudo service munge restart

使用 sinfo 查看当前资源信息

sinfo

#正常工作会显示如下信息

#PARTITION AVAIL TIMELIMIT NODES STATE NODELIST

#mypartition\* up infinite 1 idle AOBA-ALIENWARE

## 安装 OpenMPI

从软件源安装 OpenMPI

sudo apt install openmpi

编写测试程序

见文章 Notes of High Performance Computing Modern Systems and Practices 中 OpenMPI 章节中的测试程序

## Slurm 和 OpenMPI 协作工作测试

编写批处理任务脚本 job.sh

#!/bin/bash

#SBATCH -N 1

#SBATCH --ntasks 4

#SBATCH --output test.out

## 通过 -N 指令指定节点数

## 通过 --ntasks 指定处理器需求数

## 通过 --output 指定输出文件

## 通过 --time 指定启动时间

## mpirun 运行编译好的可执行程序

mpirun -np 4 ./test.exe

通过 sbatch 运行脚本

sbatch job.sh

通过 squeue 查看运行状态

使用 cat test.out 查看输出文件

# Slurm Quick Installation for Cluster on Ubuntu 20.04 很实用！

https://nekodaemon.com/2022/09/02/Slurm-Quick-Installation-for-Cluster-on-Ubuntu-20-04/

## Naming Convention of Nodes

A common cluster should comprise management nodes and compute nodes. This aritcle will take our cluster as an example to demostrate steps to install and configure Slurm. In our case, the management node is called clab-mgt01 while the compute nodes are named from clab01 to clab20 in order.

## Install Dependencies

Execute the following command to install the dependencies on all machines. (clab-all refers to all machines including management and compute nodes).

|  |
| --- |
| clab-all$ sudo apt install slurm-wlm slurm-client munge |

Tips: There are several tools that may help to manage multiple nodes easily:

* iTerm2 (on Mac) / Terminator (on Linux)
* csshX (on Mac) / cssh (on Linux)
* Parallel SSH (at cluster side)

## Generate Slurm Configuration

There is [an official online configuration generator](https://slurm.schedmd.com/configurator.html). And we should carefully check the fields below.

* SlurmctldHost: clab-mgt01 in our case.
* NodeName: clab[01-20] in our case.
* CPUs: It is recommended to leave it blank.
* Sockets: For a dual-socket server we commonly see, it should be 2.
* CoresPerSocket: Number of physical cores per socket.
* ThreadsPerCore: For a regular x86 server, if hyperthreading is enabled, it should be 2, otherwise 1.
* RealMemory: Optional.

Click submit, then we could copy the file content to /etc/slurm-llnl/slurm.confon all machines.

Tips: Don't forget the shared storage (e.g. NFS storage) on the cluster. We could utilize it to distribute files.

## Distribute Munge Key

Once Munge is installed successfully, the key /etc/munge/munge.key will be automatically generated. It is requried for all machines to hold the same key. Therefore, we could distribute the key on the management node to the remaining nodes including compute nodes and other backup management node if existing.

Tips: Again. We could also utilize the shared storage to distribute the key.

Then make sure the permission and the ownership are correctly set.

|  |
| --- |
| clab-all$ sudo chmod 400 /etc/munge/munge.key  clab-all$ chown munge:munge /etc/munge/munge.key |

## Patch Slurm Cgroup Integration

By default, there Slurm cannot work with Cgroup well. If we start Slurm service right now, we may receive this error shown below.

|  |
| --- |
| error: cgroup namespace 'freezer' not mounted. aborting |

Therefore, by pasting the following content to /etc/slurm/cgroup.conf on compute nodes, this issue can be fixed.

|  |
| --- |
| CgroupMountpoint=/sys/fs/cgroup |

or using this command:

|  |
| --- |
| echo CgroupMountpoint=/sys/fs/cgroup >> /etc/slurm/cgroup.conf |

## Fix Directory Permission

For unknown reasons, the permission of the relevant directory is not set properly, which may lead to this error.

|  |
| --- |
| slurmctld: fatal: mkdir(/var/spool/slurmctld): Permission denied |

The solution is executing the commands below on management nodes.

|  |
| --- |
| clab-mgt$ sudo mkdir -p /var/spool/slurmctld  clab-mgt$ sudo chown slurm:slurm /var/spool/slurmctld/ |

## Start Slurm Service

So far, we have finished the basic configuration. Let us launch Slurm now.

|  |
| --- |
| # On management nodes  clab-mgt$ sudo systemctl enable munge  clab-mgt$ sudo systemctl start munge  clab-mgt$ sudo systemctl enable slurmctld  clab-mgt$ sudo systemctl start slurmctld  # On compute nodes  clab-comp$ sudo systemctl enable munge  clab-comp$ sudo systemctl start munge  clab-comp$ sudo systemctl enable slurmd  clab-comp$ sudo systemctl start slurmd |

Run sinfo and we should see all the compute nodes are ready.

|  |
| --- |
| $ sinfo  PARTITION AVAIL TIMELIMIT NODES STATE NODELIST  debug\* up infinite 20 idle clab[01-20] |

## Debugging Tips

If your Slurm is not working correctly, you could try with these commands to debug.

|  |
| --- |
| clab-mgt$ sudo slurmctld -D  clab-comp$ sudo slurmd -D |

## References

* <https://www.cnblogs.com/aobaxu/p/16195237.html>
* <https://stackoverflow.com/questions/62641323/error-cgroup-namespace-freezer-not-mounted-aborting>
* Author: NekoDaemon
* Link: <https://nekodaemon.com/2022/09/02/Slurm-Quick-Installation-for-Cluster-on-Ubuntu-20-04/>
* Copyright: Original content on this site is licensed under [BY-SA](https://creativecommons.org/licenses/by-sa/4.0/)

## Add GPU support

To add GPU support, we first create a file gres.conf in /etc/slurm-llnl/. Here is an example on one node:

Name=gpu File=/dev/nvidia0

Name=gpu File=/dev/nvidia1

Name=gpu File=/dev/nvidia2

Then, we add GresTypes=gpu into /etc/slurm-llnl/slurm.conf. Next, we add the GPU information to slurm.conf:

NodeName=node1 Gres=gpu:3 State=UNKNOWN