登录

欢迎来到北鲲云私有化超算平台!本文档提供了一些来帮助您登录使用。

一. 如何登录

已有账号,登录界面如下,输入对应账号和密码进行登录,未有账号,请联系超算中心集群管理员进行 注册



基本概念

一. 计算区

基于本地集群部署的超级计算集群。

二. SSH连接

该功能适用于对超算平台、作业调度系统slurm命令有所了解人群,通过SSH连接集群提交作业到 计算节点。

• 管理节点

启动成功后,用于登录超算集群提交作业和管理作业。

• 计算节点

在管理节点提交作业后,启动的跑计算任务的节点。

• 计算队列

不同规格的节点资源,使用sinfo命令查看当前队列。

[remote]	master - clo	oudam@47.95			- = x <>v
[cloudam@r	naster	~1s sinfo			
PARTITION	AVAIL	TIMELIMIT	NODES	STATE NODELIST	
c-4-1*		infinite		mix c-4-1-worker0001	
c-4-2		infinite		n/a	
c-4-4	up	infinite			
c-4-8		infinite			
c-8-1	up	infinite			
c-8-2		infinite			
c-8-2-h		infinite			
c-8-4		infinite			
c-8-8		infinite			
c-16-1		infinite			
c-16-2		infinite		n/a	
c-16-2-h		infinite		n/a	
c-16-4		infinite		n/a	
c-16-8		infinite		n/a	
c-24-1		infinite			
c-24-2		infinite		n/a	
c-24-2-h		infinite		n/a	
c-24-4		infinite		n/a	
c-24-8		infinite		n/a	
C-32-1	up	infinite		n/a	
C-32-2		infinite		n/a r/-	
c-32-2-n	up 	infinite		n/a	
C-32-4	up	infinite		n/a	
c=32=0	up	infinito		11/d	
c-48-2-b	up	infinite		11/a n/a	
c=52=2	up un	infinito		n/a	
c-52-4	up	infinite		n/a	
c-52-8	up	infinite		n/a	
c-64-1	up	infinite		n/a	
c-64-2	up au	infinite		n/a	
c-64-4	up	infinite		n/a	
c-64-8	up	infinite		n/a	
c-80-5	up	infinite		n/a	

队列例子:

с-4- 1		g-v100- 1	
С	表示规格为CPU节点	g	表示规格为GPU节点
4	表示该节点CPU核心数为4核	v100	表示显卡型号为v100
1	表示每核心有1G内存,即节点规格为4 核4G	1	表示GPU卡数为一张 卡

快速功能介绍

本节简单介绍私有化超算平台的主要功能,方便您快速上手。

一. 应用中心

可以在此搜索您所需要的软件,选择您习惯的作业提交方式。



二. 作业管理

使用模板提交作业的管理界面,可查看作业状态、连接运行中的计算节点、查看历史作业等信息,模板 提交功能需联系集群管理员制作相关任务模板,部分软件不支持制作模板提交。

	=					本地计算区	- D 🚯 🗯	b ~
ム 北線云私有化起算平台 十 提交作业	作业管理							
and the second second	请输入作业名称搜索 Q 作业状态 ▼						刷新 显示全	部作业
作业	代业名称	执行状态	计算区	开始时间	结束时间		攝作	
88 应用中心	自定义软件-1646966167268	也 我 行成功	本地计算区	2022-03-11 10:36:17	2022-03-11 10:37:42		I …	
E6 17-3211832						每页行数:	10 - 1-1 of 1 < <	> >
23 でも構成 23 文件作名名								

三. SSH连接

该功能适用于对slurm命令有一定了解的用户使用,全程通过命令行提交作业,支持单/多节点并行计算。

シンン 北銀云 二	=	本地计算区	× >	٩	¢ 1	b ~
北端云私有化超算平台 ——	SSH连接					
	选择登录节点					
作业	master 👻 🔟					
88 应用中心						
同 作业管理						
D SSH连接						
双语						
已 文件便輸						

四. 文件传输

提供输入文件和结果文件的上传下载,支持在线查看。

シンン 北銀云 光銀 云	≡					本地计算区	• • •	j <u>(</u>) ∨
北端云私有化超算平台	文件传输							
作业	请输入文件名称	搜索 qiang ▼	Q				+ 新建 新建文件实	在终端打开
88 应用中心		名称		英型	上传时间	文件大小	提作 操作 上传文件夹	
E。 作业管理		💼 jobs		directory	2022-03-11 10:37:04		下载 更多▼	_
区 SSH连接								
2018								
[2] 文件传输								

数据存储与传输

存储目录介绍

目录	描述
/home/xxxxx	用户的家目录,该目录是所有节点的共享存储,对于无需root权限的linux软件, 可安装到该目录下,永久保存;对于需要使用root权限的软件,需要联系管理员 进行安装
/public	平台上预装软件的安装目录,您无权限在此目录下进行写操作

应用软件加载与安装

查询平台预装软件

私有化超算平台集成多款命令行软件环境,已有软件无需重复安装,可直接加载使用。

一. 使用命令行查询软件

Module是一款环境变量管理工具,可以通过module进行环境变量的管理。

Step 1. 启动并连接ssh节点;

Step 2. 查询平台所有预装软件;

module avail

Step 3. 查询指定软件 (如查询lammps) ;

module spider lammps

二. Python/Conda环境的查询

Anaconda是一个开源的软件包管理系统和环境管理系统,用于安装多个版本的软件包及其依赖关系, 并在它们之间轻松切换。

Step 1. 加载Anaconda3

module add Anaconda3

Step 2. 列出平台预装的Conda虚拟环境

conda env list

三. 未查询到所需软件

1. 平台支持自定义安装软件配置自定义的软件或环境,安装过程有问题可联系学校集群管理员进行帮 助。

加载预装软件

平台预装的所有linux命令行软件和环境均可通过命令行直接加载。

一. SSH命令行加载软件环境

1. 使用module工具查询和加载软件

module avail #列出平台所有预装软件 module spider xxxx #查询指定软件 module add xxxx/xxx #加载指定版本的软件

2. 加载Python/Conda虚拟环境

module add Anaconda3 #加载Anaconda3 conda env list source activate xxxx

#列出所有Conda虚拟环境 #加载指定虚拟环境

作业提交与运行

命令行提交

命令行提交用的是slurm命令集群调度系统,通过脚本形式提交任务至一个或多个计算节点,进行 并行计算,可使用mpi或openmp并行。如果您在其他超算平台使用过对应的脚本,需要提供给技 术工程师修改为适用于北鲲云超算平台的脚本。

一. 提交步骤

Step 1. 通过SSH连接启动一个管理节点,并连接进入管理节点;

Step 2. 提供文件传输上传输入文件,如何上传文件请点击查看数据传输;

Step 3. 查看SLURM集群所支持的partitions(分区);

Step 4. 查询和加载软件,更多软件的加载,请点击查看加载预装软件;

module avail#查看全部module spider xxxx#快速查找module add xxxx#加载

Step 5. 创建提交脚本,参照gromacs运行脚本: su.sh

```
#!/bin/bash
module add GROMACS/2021-gromacs-cpu-new
mpiexec -v gmx_mpi mdrun -v -cpi tpr_file_name -deffnm tpr_file_name
```

Step 6. 使用sbatch提交到计算节点,参数详细介绍请查看slurm命令;

sbatch -N 2 -p c-4-1 -n 8 -c 1 su.sh

Step 7. 查看SLURM集群正在运行的作业;

squeue		
[cloudam@master ~]\$ squeue		
JOBID PARTITION	NAME USER ST TIME NODES NODELIST(REASON)	
17 c-4-1	bash cloudam R 15-21:13:17 1 c-4-1-worker0001	
[cloudam@master ~]\$		

Step 8. 连接计算节点执行top查看CPU使用情况;

ssh c-4-1-worker0001	#连接计算节点
top	#查看任务管理器
exit	#退出计算节点

top - 14:20:30) up 3	min	, 1 use	er, loa	d avera	ge: 38.8	33, 18	.48, 7.19
Tasks: 439 tot	cal,	41 r	unning,	398 sle	eping,	0 stop	pped,	0 zombie
%Cpu(s): 99.2	us,	0.8	sy, 0.0	0 ni, 0).0 id,	0.0 wa,	0.0	hi, 0.0 si, 0.0 st
KiB Mem : 9571	L6728	tota	1, 86433	3704 fre	e, 8380	6760 use	ed,	896264 buff/cache
KiB Swap:	0	tota	1,	0 fre	e,	0 use	ed. 86	582944 avail Mem
PID USER	PR	NI	VIRT	RES	SHR S	S %CPU	%MEM	TIME+ COMMAND
2858 cloudam	20	0	331644	107908	19288 I	R 100.3	0.1	2:56.75 gmx_mpi
2861 cloudam	20	0	329480	105164	16816 I	R 100.3	0.1	2:57.16 gmx_mpi
2851 cloudam	20	0	378024	151712	27208 I	R 100.0	0.2	2:56.27 gmx_mpi
2852 cloudam	20	0	326856	102748	19744 I	R 100.0	0.1	2:56.82 gmx_mpi
2854 cloudam	20	0	468016	225456	23072 1	R 100.0	0.2	2:57.06 gmx_mpi
2855 cloudam	20	0	480052	232332	22844 1	R 100.0	0.2	2:57.12 gmx_mpi
2856 cloudam	20	0	327788	101400	16780 I	R 100.0	0.1	2:56.61 gmx_mpi
2865 cloudam	20	0	328000	102348	18792 I	R 100.0	0.1	2:57.00 gmx_mpi
2873 cloudam	20	0	325680	102688	19324 I	R 100.0	0.1	2:56.65 gmx_mpi
2874 cloudam	20	0	335648	107656	19280 I	R 100.0	0.1	2:57.02 gmx_mpi
2876 cloudam	20	0	325464	99676	18844 I	R 100.0	0.1	2:57.14 gmx mpi
2877 cloudam	20	0	326460	101912	16780 I	R 100.0	0.1	2:56.97 gmx mpi
2878 cloudam	20	0	331996	106376	17808 I	R 100.0	0.1	2:57.17 gmx_mpi
2880 cloudam	20	0	325456	103988	16844 I	R 100.0	0.1	2:57.18 gmx_mpi
2882 cloudam	20	0	331632	107128	18712 I	R 100.0	0.1	2:57.20 gmx mpi
2885 cloudam	20	0	329108	103928	16828 H	R 100.0	0.1	2:56.48 gmx mpi
2888 cloudam	20	0	326208	102796	17876 I	R 100.0	0.1	2:57.35 gmx mpi
2853 cloudam	20	0	330492	110056	18704 I	R 99.7	0.1	2:56.84 gmx mpi
2857 cloudam	20	0	327484	106884	20420 1	R 99.7	0.1	2:56.58 gmx mpi
2860 cloudam	20	0	328448	103132	18912 I	R 99.7	0.1	2:57.24 gmx mpi
2862 cloudam	20	0	333316	107080	17712	R 99.7	0.1	2:56.71 gmx mpi

Step 9. 查看运行作业详细信息;

- /home/cloudam/examples/GROMACS 为作业执行路径。
- /home/cloudam/examples/GROMACS/slurm-47.out 文件为作业输出日志,可实时查看作业运行信息。



Step 10. 取消程序运行,释放分配的计算节点(作业执行成功或失败计算节点会自动释放);

scancel JOBID

Step 11. 如何下载结果文件,请点击查看Linux数据传输;

二. 命令行提交

通过SSH连接创建并连接管理节点。

```
Step 1. 创建作业目录并进入;
```

mkdir amberJob1
cd amberJob1

Step 2. 通过文件传输上传所需的输入文件,详情请查看Linux数据传输;

Step 3. 在该文件夹下创建如下执行脚本amber.sh:

```
#!/bin/bash
module add Anaconda3/2020.02
source /public/software/.local/easybuild/software/amber/aber20/amber.sh
ulimit -s unlimited
ulimit -l unlimited
pmemd.cuda -0 -i sander.in7 -o sander.out7 -p prmtop -c mdrest6 -r mdrest7 -x
mdcrd7
```

Step 4. 提交作业;

2个4核心节点启动8个并行任务。

sbatch -N 2 -p c-4-1 -n 8 -c 1 amber.sh

查看作业运行情况及参数详细介绍请点击查看slurm命令。

结果文件下载请查看数据传输。

Slurm作业管理系统

Slurm (Simple Linux Utility for Resource Management, <u>http://slurm.schedmd.com/</u>)是 开源的、具有容错性和高度可扩展大型和小型 Linux集群资源管理和作业调度系统。超级计算系统 可利用 Slurm 进行资源和作业管理,以避免相互干扰,提高运行效率。所有需运行的作业无论是 用于程序调试还是业务计算均必须通过交互式并行 srun、批处理式 sbatch 或分配式 salloc 等命 令提交,提交后可以利用相关命令查询作业状态等。

一. 常用命令

sinfo	#查看分区状态
squeue	#查看队列中的作业
scontrol	#查看作业详细信息
scancel	#取消已经提交的作业
sbatch	#批处理式提交作业
salloc	#分配式运行作业

1. 查看分区状态

sinfo

- CPU分区命名规则为**c-核心数-每核心内存大小**,如c-8-4:表示单节点规格为8核,每核心有4G内存,即节点规格为8核32G。
- GPU分区命名规则为**g-卡号-每节点卡数**,如g-v100-2:表示有两张显卡型号为tesla v100的gpu节点。

[cloudam@r	naster	jobs]\$ sinf	0		
PARTITION	AVAIL	TIMELIMIT	NODES	STATE	NODELIST
c-4-1*	up	infinite	1	alloc	c-4-1-worker0001
c-4-2	up	infinite	Θ	n/a	
c-4-4	up	infinite	Θ	n/a	
c-4-8	up	infinite	Θ	n/a	
c-8-1	up	infinite	Θ	n/a	
c-8-2	up	infinite	Θ	n/a	
c-8-2-h	up	infinite	Θ	n/a	
c-8-4	up	infinite	Θ	n/a	
c-8-8	up	infinite	Θ	n/a	
c-16-1	up	infinite	Θ	n/a	
c-16-2	up	infinite	Θ	n/a	
c-16-2-h	up	infinite	Θ	n/a	
c-16-4	up	infinite	Θ	n/a	
c-16-8	up	infinite	Θ	n/a	
c-24-1	up	infinite	Θ	n/a	
c-24-2	up	infinite	Θ	n/a	
c-24-2-h	up	infinite	Θ	n/a	
c-24-4	up	infinite	Θ	n/a	
c-24-8	up	infinite	Θ	n/a	
c-32-1	up	infinite	Θ	n/a	
c-32-2	up	infinite	Θ	n/a	
c-32-2-h	up	infinite	Θ	n/a	
c-32-4	un	infinite	Θ	n/a	

2. 查看作业队列

squeue

- JOBID: 作业号。
- ST: 状态 (R: 运行中; CF: 配置中; PD: 排队中)。

[cloudam@master	job	s]\$ squeue					
JO	BID	PARTITION	NAME	USER	ST	TIME	NODES NODELIST(REASON)
	20	c-4-2	demo.sh	cloudam	R	0:29	1 c-4-2-worker0001

3. 查看所有作业详细信息

scontrol show jobs

[cloudam@master jobs]\$ scontrol show jobs JobId=20 JobName=demo.sh UserId=cloudam(1003) GroupId=cloudam(1003) MCS label=N/A Priority=4294901756 Nice=0 Account=(null) QOS=normal JobState=RUNNING Reason=None Dependency=(null) Requeue=1 Restarts=0 BatchFlag=1 Reboot=0 ExitCode=0:0 RunTime=00:16:15 TimeLimit=UNLIMITED TimeMin=N/A SubmitTime=2021-11-19T16:04:23 EligibleTime=2021-11-19T16:04:23 AccrueTime=2021-11-19T16:04:23 StartTime=2021-11-19T16:04:25 EndTime=Unknown Deadline=N/A SuspendTime=None SecsPreSuspend=0 LastSchedEval=2021-11-19T16:04:25 Partition=c-4-2 AllocNode:Sid=iZ2zehs3cdd8rz787les3wZ:5927 ReqNodeList=(null) ExcNodeList=(null) NodeList=c-4-2-worker0001 BatchHost=c-4-2-worker0001 NumNodes=1 NumCPUs=4 NumTasks=4 CPUs/Task=1 RegB:S:C:T=0:0:*:* TRES=cpu=4, node=1, billing=4 Socks/Node=* NtasksPerN:B:S:C=0:0:*:* CoreSpec=* MinCPUsNode=1 MinMemoryNode=0 MinTmpDiskNode=0 Features=(null) DelayBoot=00:00:00 OverSubscribe=OK Contiguous=0 Licenses=(null) Network=(null) Command=/home/cloudam/jobs/demo.sh WorkDir=/home/cloudam/jobs StdErr=/home/cloudam/jobs/slurm-20.out StdIn=/dev/null StdOut=/home/cloudam/jobs/slurm-20.out Power= 出的日志又 MailUser=(null) MailType=NONE

4. 取消作业号为20的作业

scancel 20

二. 提交作业的方式

1. 使用sbatch批处理模式提交作业

sbatch命令可以提交任务至一个或多个计算节点,实现并行计算。

sbatch命令的一些常用选项:

参数	功能
-N	申请的节点数量
-р	指定计算节点规格,使用sinfo查看所支持的规格
-n	指定任务数,即并行程序运行多少个进程
-c	每进程使用的cpu核心数

参考运行程序: demo.sh

```
#!/bin/bash
sleep 6000
```

• 使用2个4核心节点启动8个并行任务。

sbatch -N 2 -p c-4-1 -n 8 -c 1 demo.sh

• 1个4核心节点启动4个并行任务。

sbatch -N 1 -p c-4-1 -n 4 -c 1 demo.sh

• 1个4核心节点启动4个并行任务。

sbatch -p c-4-1 -n 4 demo.sh

• 1个4核心节点启动1个并行任务,该任务使用4个cpu核心。

sbatch -p c-4-1 -n 1 -c 4 dome.sh

• 2个4核心节点启动2个并行任务,每个任务使用4个cpu核心。

sbatch -p c-4-1 -n 2 -c 4 dome.sh

• 使用GPUv100 8卡启动32个并行任务。

sbatch -N 1 -p g-v100-8 -c 8 --gres=gpu:8 dome.sh

2. 使用salloc分配模式提交作业

salloc命令可以用来分配节点,用户可以在获取分配的计算节后等,ssh进入直接运行相关计算程序,主要用来调式程序执行。

使用步骤 (案例):

Step1. salloc申请计算节点;

salloc -N 1 -p c-8-2 &

Step2. ssh登录到分配的计算节点;

ssh c-8-2-worker0001

Step3. 调试或运行程序;

./demo.sh

Step4. 结束程序运行后释放分配的节点;

scancel 17



Module的使用

Module是一款环境变量管理工具,用户只需加载模块即可使用平台的软件或依赖库,如果没有找 到所需的软件可以联系集群管理员进行安装。

一. 常用命令

module	avail 或 module av						
module	spider 或 module sp						
module	add 或 module load						
module	rm 或 unload						
module	list 或 module li						
module	purge						
module	show						
module	swap 或 module switch						
module	help						

#查看系统中可用的软件
#查询所有可能的模块
#加载模块
#卸载模块
#卸载模块
#显示已加载模块
#卸载所有模块
#显示模块配置文件
#将模块1替换为模块2
#显示帮助信息

注:建议不要同时module add多个软件,因为不同软件间可能是有冲突的。比较好的方式是module add一个或一组相互依赖的软件,软件运行完后运行module purge清除导入的环境,然后再导入另外一个或一组相互依赖的软件。

二. 使用例子

查看集群现有软件环境

module avail

module spider gromacs

加载GROMACS/2021-gromacs软件环境

module add GROMACS/2021-gromacs

显示所有已加载的模块

module list

清除已加载的所有软件模块

module purge

Conda的使用

Conda 是一个开源的软件包管理系统和环境管理系统,用于安装多个版本的软件包及其依赖关系,并在它们之间轻松切换,Conda支持Python、R、Ruby、Lua、Scala、Java、JavaScript、C/C++、FORTRAN等多种语言。

一. 使用集群现有的Conda环境

#加载Anaconda3 module add Anaconda3

#查看Conda环境列表 conda env list

#加载指定虚拟环境 source activate xxxx

#退出当前环境:

conda deactivate

二. Conda管理环境

```
#创建环境:
conda create -n xxxx
#示例: 创建一个名为demo的虚拟环境, 指定python版本为3.7
conda create -n demo python==3.7
#删除环境:
conda env remove -n xxxx
```

三. Conda管理包

#查看一个包是否可用conda安装 conda search numpy #安装包 conda install numpy #安装指定版本包 conda install numpy=1.14 #查看当前环境已安装的包 conda list #包更新 conda update numpy

conda remove -n demo numpy

#删除包

四. Conda/Pip软件安装进阶操作

#列出保存目标软件的所有镜像
anaconda search <software>
#指定anaconda的镜像安装cudatoolkit10.0的版本
conda install -c anaconda cudatoolkit=10.0
#pip安装指定版本的包
pip install some-package==1.0.4
#pip安装最新版本的包
pip install some-package
#临时性修改下载源为清华源(只对本次下载有效)
pip install -i https://pypi.tuna.tsinghua.edu.cn/simple some-package
#pip升级包
pip install --upgrade some-package
#pip印载包
pip uninstall some-package

Linux的常用命令

linux快捷键以及帮助手册

在我们正式学习linux命令之前,我们应该了解一些基本的快捷键,以及查看帮助手册。

快捷键

按键	功能
Tab	命令补全,当你忘记某个命令的全称时可以只输入它的开头的一部分,然后 按下 Tab 键就可以得到提示或者帮助完成。
Ctrl+c	强行终止当前程序。
Ctrl+d	键盘输入结束或退出终端。
Ctrl+s	暂停当前程序,暂停后按下任意键恢复运行。
Ctrl+z	将当前程序放到后台运行,恢复到前台为命令fg。
Ctrl+a	将光标移至输入行头,相当于Home键。
Ctrl+e	将光标移至输入行末,相当于End键。
Ctrl+k	删除从光标所在位置到行末。
Alt + Backspace	向前删除一个单词。
Shift + PgUp	将终端显示向上滚动。
Shift + PgDn	将终端显示向下滚动。

帮助手册

在 Linux 环境中,如果你对某个命令感到疑惑,可以使用 man 命令来获取帮助,它是Manual pages的 缩写。用户可以通过执行 man 命令调用手册页。

你可以使用如下方式来获得某个命令的说明和使用方式的详细介绍:

man <command_name>

用户和文件权限管理

Linux 是一个多用户的操作系统,所有用户共享一些主机的资源,但他们也分别有自己的用户空间,用于存放各自的文件。由于Linux的用户管理和权限机制,不同用户不可以轻易地查看、修改彼此的文件。

查看用户

输出当前伪终端的用户名:

whoami

查看文件权限

ls 命令不仅可以列出并显示当前目录的文件,还可以用来查看文件的权限。

ls -1

输出: 文件类型和权限 链接数 所有者 所属用户组 文件大小 最后修改时间 文件名

[cloudam@master ~]\$ ls -l										
total 4442067										
- rw- r r	1	cloudam	cloudam	37918	Nov	22	17:30	1122.tar		
drwxrwxr-x	3	cloudam	cloudam	4096	Nov	25	10:25	aa		
- rw- rw- r	1	cloudam	cloudam	70	Nov	18	13:37	aa.sh		
- rw- rw- r	1	cloudam	cloudam	3470	Nov	12	16:53	abaqus_2020.gpr		
- rw-rr	1	cloudam	cloudam	4200164680	Nov	10	17:49	abagus2020.tar.gz		

这里特别解释一下文件类型和权限:第一位是指文件类型,例如**'d'**表示目录,**'-'**表示普通文件,等 等。后面的九位可以分成三组,每组三位分别表示读权限/写权限/执行权限,三组分别对应的是拥有者 权限/所属用户组权限/其他用户权限。

我们可以用 touch 命令创建一个新的文件, 然后再进行以下测试。

变更文件所有者

sudo chown <new_owner> <file>

sudo命令在之前提到过,是权限命令。

修改文件权限

有两种方式对文件权限进行修改:

第一种方式:二进制数字表示。每个文件的三组权限对应一个 "rwx",每一位可以用 0 或 1 表示是否有 权限,例如只有读和写权限的话就是 "rw-" - "110",所以"rwx"对应的二进制"111"就是十进制的7。

所以将 file 文件权限改为 "rwx——" 的命令是:

chmod 700 <file>

第二种方式:加减赋值表示。假设一开始 file 文件的权限是 "rwxr-r-"完成上述同样的效果,还可以:

chmod go-rr file

g、o还有 u 分别表示 group、others 和 user, +和 -分别表示增加和去掉相应的权限。

目录及文件操作

在linux系统中最常见的行为就是切换目录以及对文件进行操作。

基本目录操作

在linux中,我们使用 cd 命令来切换目录,使用.表示当前目录, ... 表示上一级目录, -表示上一次所在目录, ~通常表示当前用户的 home 目录。使用**pwd**命令可以获取当前所在路径(绝对路径)。

例如,进入上一级目录:

cd ..

搭配之前提及的快捷键 Tab 可以大大提高效率。

基本文件操作

创建

文件: 在这里, 我们可以先简单地使用 touch 命令来创建一个新的空白文件。 例如:

touch myfile

目录: 使用 mkdir 命令可以创建一个新的目录。

创建名为" mydir "的空目录:

mkdir mydir

复制

文件: 使用 cp (copy) 命令复制一个文件到指定目录。

例如:

cp myfile ../../mydir

目录: 在复制文件的基础上加上-r参数即可。

例如:

cp -r dir mydir

删除

文件: 使用 rm (remove files or directories) 命令删除一个文件:

rm myfile

目录: 同样, 删除目录也是需要加上-r参数。

移动及重命名

移动文件和对文件重命名都使用 mv 命令。

mv 源目录文件 目的目录:

mv myfile mydir

注意,这里mv命令是剪切文件。

mv 旧的文件名 新的文件名:

mv myfile myfile1

编辑文件

Linux系统存在许多编辑器,例如vim,emacs,nano等等,在这里我们使用vim编辑器来打开文件:

```
vim myfile
```

vim编辑器功能非常强大,但随之而来的是它的学习难度并不简单,然而编辑器的使用又是必不可少的 一部分,所以在这里建议您先去学习一门编辑器的基本使用方法。

搜索文件

相信每个用户在使用操作系统时都会用到搜索这个功能, linux提供了许多与搜索相关的命令给用 户使用。

which

which只从环境变量指定的路径中去搜索文件,所以我们常常使用which来判断是否安装了某个软件。

例如:

which vim

whereis

一个比较简单的搜索命令,它直接从数据库而不是硬盘查询,而且只能搜索特定的文件。所以,它的搜索很快。

whereis vim

locate

在这里要先介绍" /var/lib/mlocate/mlocate.db "数据库,因为 locate 命令就是通过这个数据库查找的。 这个数据库由系统每天定时维护更新,所以有时候需要手动执行 updatedb 命令来更新数据库。

例如,查找 /usr 及其子目录下所有以 a 开头的文件:

locate /usr/a

find

find命令是一个功能非常强大的命令,它可以根据文件的各个属性来搜索,在此只介绍其简单的搜索。 其基本命令格式为: find [path] [option] [action]

例如,在/usr/目录下搜索名字为 myfile 的文件或目录:

sudo find /usr/ -name myfile

文件解压缩

本节介绍linux上常用的解压缩工具,包括zip、rar以及tar。

rar

rar是一种在windows常见的文件压缩格式,但是在linux下必须先安装才能使用。由于在linux中rar命令参数非常多,在此只介绍一些基本操作。

1) 创建压缩文件:

rar a cloudam.rar ./Desktop/

这是将Desktop目录打包成cloudam.rar

2) 使用 a 参数添加文件进压缩包:

rar a cloudam.rar a.o

3) 使用 d 参数删除文件:

rar d cloudam.rar a.o

4) 使用 | 参数查看文件:

```
rar l cloudam.rar
```

5) 解压分为两种,一种是 x 参数解压,这样会保持压缩文件的目录结构;另一种是 e 参数解压,将所 有文件解压到当前目录:

unrar x cloudam.rar unrar e cloudam.rar

zip

zip也是在windows和linux使用率比较高的文件压缩格式。

1) 创建压缩包:

zip -r -o cloudam.zip ./Desktop

其中,-r 参数表示递归,所以上面命令是打包Desktop目录进压缩包;-o 参数表示输出到cloudam.zip 压缩文件。

2) 使用 -e 参数进行加密:

zip -r -e -o cloudam.zip ./Desktop

需要注意的是,由于编码问题linux下和windows下的文本文件可能会不同,如果希望在linux压缩的文件在windows下正常解压,可以加上-I参数。

3) 查看与解压 解压很简单:

unzip cloudam.zip

查看只需加上-I参数:

tar

实际上,在linux中用得更多的是tar工具。tar原本只是一个打包工具,没有压缩功能,需要配合其他具有压缩功能而没有打包功能的工具进行打包压缩。

1) 创建一个 tar 包:

tar -cf cloudam.tar ./Desktop

其中,-c参数表示创建,-f参数表示指定文件名,必须跟在文件名之前。

可以发现这个tar包和原大小相同。

2) 查看一个 tar 包:

tar -tf cloudam.tar

3) 解包一个 tar 包到指定目录:

tar -xf cloudam.tar -C cloudamdir

上面我们说了tar需要配合其他工具进行压缩,其实一般只需要加一个参数就可以了。这里以gzip为例:

tar -czf cloudam.tar.gz ./

解压缩:

tar -xzf cloudam.tar.gz

tar 命令参数还有很多,在此仅作简单介绍。可以通过man手册查看更多帮助。

管道与一些文本命令

管道是一种通信机制,它可以将前一个进程的输出作为下一个进程的输入。所以它常常用来进行进程间(也包括socket进行网络通信)的通信。不过,在这里我们首先介绍一些linux终端执行命令的知识。

&& 和 ||

当我们希望同时输入多条命令时,可能还希望对某几个命令的结果进行判断。例如,如果存在某个文件,就把它删除,如果不存在则什么也不做。而 && 和 || 就提供了这种判断的功能。

&& 表示如果前一个命令返回的状态为0(这些状态码有一套默认的规定),则执行后一个命令,否则不执行。而 || 则与之相反。

例如我们常常使用 which 命令来判断是否安装了某个命令:

```
which rar>/dev/null && echo "Installed."
```

当然也可以结合使用:

which rar>/dev/null && echo "Installed." || echo "Not installed"

管道

通常我们在命令行中使用匿名管道,由分隔符 | 表示,而命名管道更常出现在源代码中。

例如:

```
ls -a | grep mysql
```

这个命令就是先执行 ls -a 命令,然后将输出结果作为输入执行grep mysql命令在其中查找包含mysql的 文件。

下面介绍几个常用的命令,配合管道使用可以提高效率。

1) grep: 一个很常用的命令。使用者可以通过其与正则表达式配合实现很高效的查找,不过在这里先简单介绍一下。

在上面我们已经有一个使用 grep 命令的例子了,实际上我们还可以直接使用:

grep -r "cloudam" ~

搜索~目录下所有包含"cloudam"的文件。

2) wc: 统计并输出一个文件中行、字节等数据的数目。其中, -l 参数表示行数, -c 参数表示字节数。 结合管道使用, 统计文件的个数:

ls -a | wc -l

3) uniq: 在上面的统计中,我们可能会想要除去一些重复行以免影响结果,这时候我们可以使用 uniq 命令:

cat words | uniq

值得注意的是, uniq只除去连续重复的的行, 如果希望全文达到同样的效果我们往往会先使用 sort 命令 进行排序, 即:

cat words | sort | uniq

当然,这些命令实际上是很复杂的,我们可以通过这些命令完成很多复杂但是有用的功能,这也是其魅力所在。在此仅作简单的介绍。

文本处理

在上一节中我们学习了一些管道相关的命令,现在我们继续学习一些简单的文本处理命令。

sort

在之前出现过很多次的命令,顾名思义它的功能是排序。当然,它可以根据不同的标准来排序,例如: 默认为字典排序:

ls -a | sort

反转排序:

ls -a | sort -r

col

相信许多程序员在实际中都会遇到类似于将程序代码中的tab转空格的需要,而linux提供了tab和空格相 互转换的命令 col。

在这里,-x 参数表示将 tab 转换成空格,而-h参数则相反。

例如:

```
cat myFile | col -x | cat -A
```

tr

替换操作 (删除操作也可以视为一种替换操作) 也是文本处理常用的操作之一。

如果是要删除,可以使用-d参数:

echo "Hello world!" | tr -d "el"

这样会删除"Hello world!"中所有的"e"和"I"。

再例如将小写转换成大写:

echo "Hello world!" | tr '[a-z]' '[A-Z]'

paste

这个命令可以简单地合并文件。

例如将数据以** ';'** 为分隔符合并 (默认的分隔符为Tab) :

paste -d ';' data1 data2 data3

也可以这么做:

cat words | sort | uniq &> output

重定向

在之前管道的知识点中,学习者很容易有这种想法:命令的输出太复杂或者太多了,希望可以将其 保存到文件中查看或者长久保存。而将标准输出、标准错误等信息输出到指定文件中,就是重定 向。

重定向

重定向是通过**'>'**和** '>>'** 等操作符完成的, (当然也有'**<'**和'**<<'**,前者是导向, 后者是追加), 其实 将标准输出看作一个文件的话, 将命令输出导向到另一个文件自然也是没问题的。

例如:

echo "Hello world!" > output

结合管道:

cat words | sort | uniq >> output

文件描述符

除了标准输出,我们还经常需要重定向标准错误,在此就必须提到标准输入、标准输出和标准错误的文件描述符。在linux中,它们的文件描述符分别是0、1、2,我们可以通过这几个描述符来完成我们想要的操作。

我们可以这样来将标准输出和标准错误重定向到同一个文件:

```
cat words | sort | uniq > output 2>&1
```

其中, &表示后面的是标准输出而不是一个文件名为1的文件, 而这里的重定向顺序也是不能改变的。

也可以这么做:

cat words | sort | uniq &> output

永久重定向以及"丢弃"输出

我们很容易想到如果我们想不断将标准输出重定向到某个文件,难道需要在每一个命令后面加上重定向命令?当然是不需要的,linux提供了永久重定向的方法。当然,这里的永久也并不是那么永久。

我们可以这样做:

exec 1> output

其中 exec 命令的作用是用新的进程去替换旧的进程,或者指定重定向。

有时候我们仅仅希望执行命令而不想要其输出,这时候我们可以将输出重定向到空设备 /dev/null:

cat words | sort | uniq 1>/dev/null 2>&1

这样输出结果就没有啦。

进程的基本操作

在日常工作中我们常常需要对进程进行操作,这就要求我们掌握一些基本的进程操作指令。

前台/后台切换

我们在之前的学习中已经知道终止一个在前台运行的进程的快捷键是 ctrl + c 。同样,我们在之前也已 经知道将我们当前进程停止并转到后台的快捷键是 ctrl + z。那么,如何让我们的进程在前台与后台之间 切换呢?

我们可以通过&这个符号,让我们的命令在后台中运行。

例如:

```
ls &
```

这样命令就会在后台自己运行了。

```
那么当我们如何查看在后台的工作呢?
```

可以使用:

jobs

这时我们可以看到每行是这样的格式: [1] + suspended ...(command)

其中, 第一列的数字表示job编号, 也就是转到后台的工作的编号, 第二列如果是**'+'**表示的是这是最后一 个被转入后台的工作, ** '-'** 的话则是倒数第二个, 其他的话在这列没有符号。第三列表示状态, 而后 面则是命令本身。

那么我们突然想要把工作从后台拿回前台怎么办呢?

可以使用:

fg %jobid

例如,

fg %1

如果只是希望其在后台运行的话,可以:

bg %jobid

终止

终止命令比较简单,但是在参数上我们可以通过选择信号值来决定以何种方式终止程序,信号值可以这 样查看:

kill -1

所以可以这样终止程序:

kill -signal pid

或者

kill -signal %jobid

进程管理

在这节我们进一步学习对进程的查看与控制。

top

top 命令是我们常常使用查看工具,它可以实时查看系统的一些关键信息的变化。

top 命令的输出比较多,下面逐一介绍:

第一行: 当前程序名称 当前系统时间 机器启动时间 系统用户数量 1分钟内、5分钟内和15分钟内cpu平 均负载

第二行: 进程总数 正在运行的进程数 休眠的进程数 停止的进程数 僵尸进程数

第三行:

- 用户空间占用CPU百分比
- 内核空间占用CPU百分比
- 用户空间优先级变化的进程占用CPU百分比
- 空闲CPU百分比
- 等待IO的CPU时间百分比
- 硬中断占用CPU的百分比
- 软中断占用CPU的百分比
- 虚拟 CPU 等待物理 CPU 的时间的百分比

第四行:

- 物理内存总量
- 使用物理内存总量
- 空闲内存总量
- 缓存内存总量

第五行:

- 交换区总量
- 使用交换区总量
- 空闲交换区总量
- 缓冲交换区总量

然后就是许多行进程的信息了,以下按顺序分别对应:

- PID: 进程id
- USER: 进程所属用户
- PR: 进程动态优先级值
- NI: 进程静态优先级值
- VIRT: 进程使用虚拟内存总数
- RES: 进程使用物理内存数
- SHR: 进程共享内存大小
- S: 进程状态
- %CPU: CPU利用率
- %MEM: 内存利用率
- TIME+: 进程活跃总时间
- COMMAND: 进程运行名

我们甚至可以与这个程序交互,例如输入k的话系统会提示进一步输入信号值以及pid号以杀死一个进程,更多的信息可以查看帮助手册。

你一定经常看见 ps aux 这个命令将会列出所有的进程信息。

我们还往往配合 grep 和正则表达式使用。

例如:

ps aux | grep slurm

那么,打印出来的都是什么信息呢?

- F: 进程的标志 (process flags) , 当 flags 值为 1 则表示此子程序只是 fork 但没有执行 exec, 为 4 表示此程序使用 root 权限
- USER: 进程拥有者
- PID: 进程ID
- PPID: 父进程的PID
- SID: session 的ID
- TPGID: 前台进程组的ID
- %CPU:进程占用的CPU百分比
- %MEM:占用内存的百分比
- NI: 进程的 NICE 值
- VSZ: 进程使用虚拟内存大小
- RSS: 驻留内存中页的大小
- TTY: 终端ID
- S or STAT:进程状态
- WCHAN:正在等待的进程资源
- START:启动进程的时间
- TIME: 进程消耗CPU的时间
- COMMAND: 命令的名称和参数

我们还可以用这样的命令去查看进程树:

pstree