

实验 07：系统性能监控

一、实验目的

- 1、掌握系统性能监控；
- 2、掌握系统可视化监控。

二、实验学时

2 学时

三、实验类型

设计研究



四、实验需求

1、硬件

每人配备计算机 1 台。

2、软件

安装 VMware WorkStation Pro 或 Oracle VM VirtualBox 软件，安装 MobaXterm 软件。

3、网络

本地主机与虚拟机能够访问互联网，虚拟机网络不使用 DHCP 服务。

4、工具

无。

五、实验任务

- 1、完成系统性能监控；
- 2、完成使用 Linux-dash 实现可视化监控。

六、实验环境

- 1、本实验需要 VM 1 台。
- 2、本实验 VM 配置信息如下表所示。

虚拟机配置	操作系统配置
虚拟机名称: VM-Lab-07-Task-01-172.31.0.171 内存: 2GB CPU: 1 颗, 1 核心 虚拟磁盘: 20GB 网卡: 1 块, NAT	主机名: Lab-07-Task-01 IP 地址: 172.31.0.171 子网掩码: 255.255.255.0 网关: 172.31.0.254 DNS: 172.31.0.254

- 3、本实验拓扑图。
无。

4、本实验操作演示视频。

本实验操作演示视频为视频集的第 7 集：

<https://www.bilibili.com/video/BV1iH4y1c7ft?p=7>

七、实验内容及步骤

1、使用 sysstat 监控 openEuler 系统运行

(1) 安装 sysstat 工具

sysstat 是一个软件包，其包含监测系统性能的一组工具，通过对主机的性能数据进行收集与分析，能够监控系统状态，及时了解系统运行性能。系统默认未安装 sysstat 软件，可使用 yum 工具安装。

```
# 使用 yum 工具在线安装 sysstat
[root@Lab-07-Task-01 ~]# yum install -y sysstat
```

(2) 使用 mpstat 工具监控系统 CPU

mpstat 工具可实时监控主机系统 CPU 的运行状态，也可指定单个物理 CPU 查看其运行情况。

```
# 监控系统全部 CPU 的运行状态
[root@Lab-07-Task-01 ~]# mpstat -P ALL

# 查看当前系统 CPU 的平均运行状态 (每隔 5s, 采集 2 次查看平均运行情况)
[root@Lab-07-Task-01 ~]# mpstat -P ALL 5 2

# 监控当前系统单颗 CPU 的运行状态 (指定第一颗 CPU 查看运行情况)
[root@Lab-07-Task-01 ~]# mpstat -P 0
```

(3) 使用 sar 工具监控系统内存

sar 是系统运行状态统计工具，可对系统当前的内存运行情况进行取样，然后通过计算数据和比例来分析系统的当前状态。

```
# 监控系统内存的使用情况
[root@Lab-07-Task-01 ~]# sar -o 3
```

(4) 使用 vmstat 工具监控系统存储

vmstat 工具可统计系统整体的存储情况，包括内核进程、内存使用、虚拟内存、磁盘 I/O 和 CPU 状态等信息。

```
# 监控系统存储的使用情况
[root@Lab-07-Task-01 ~]# vmstat
```

(5) 使用 pidstat 工具监控系统进程

pidstat 工具可监控全部或单独指定某个进程，查看其资源占用情况，掌握系统进程的运行性能。

```
# 使用 pidstat 监控所有进程的运行情况
[root@Lab-07-Task-01 ~]# pidstat

# 指定进程号查看单进程运行情况
[root@Lab-07-Task-01 ~]# pidstat -p 1
```

(6) 使用 iostat 工具监控系统 IO

iostat 工具可监视主机磁盘 IO 的运行情况, 查看存储设备的性能。

```
# 使用 iostat 查看磁盘 IO 的运行情况
[root@Lab-07-Task-01 ~]# iostat
```

2、通过 proc 监控 openEuler 系统性能

(1) 通过 proc 监控系统 CPU

通过查看 /proc/loadavg 目录信息监控主机 CPU 当前运行的负载情况。

```
# 查看系统 CPU 运行性能
[root@Lab-07-Task-01 ~]# cat /proc/loadavg
```

(2) 通过 proc 监控系统内存

通过查看 /proc/meminfo 目录信息监控当前系统内存的使用情况。

```
# 监控系统内存运行状态信息
[root@Lab-07-Task-01 ~]# cat /proc/meminfo
```

(3) 通过 proc 监控系统存储

通过查看 /proc/partitions (分区)、/proc/vmstat (虚拟内存) 和 /proc/swaps (交换分区) 目录信息监控当前系统存储的使用情况。

```
# 查看系统磁盘分区大小
[root@Lab-07-Task-01 ~]# cat /proc/partitions

# 查看系统虚拟内存的使用情况
[root@Lab-07-Task-01 ~]# cat /proc/vmstat

# 查看系统交换分区使用情况
[root@Lab-07-Task-01 ~]# cat /proc/swaps
```

(4) 通过 proc 监控系统进程

/proc 目录中包含许多以数字命名的子目录, 这些数字表示系统当前正在运行的进程 ID, 可查看所有进程信息或者针对某个进程查看其详细的运行情况。

```
# 查看所有进程目录信息
[root@Lab-07-Task-01 ~]# ll /proc/

# 查看进程 ID 为 1 的启动命令
[root@Lab-07-Task-01 ~]# cat /proc/1/cmdline

# 查看进程 ID 为 1 的状态信息
[root@Lab-07-Task-01 ~]# cat /proc/1/status
```

(5) 通过 proc 监控系统 IO

通过查看 /proc/diskstats 目录信息监控当前系统磁盘 IO 的运行情况。

```
# 监控主机磁盘 IO 运行状态
[root@Lab-07-Task-01 ~]# cat /proc/diskstats
```

(6) 通过 proc 监控系统网络

通过查看 /proc/net/dev 目录信息, 监控当前系统网卡接口通信流量情况。

```
# 查看主机网卡接口网络流量
[root@Lab-07-Task-01 ~]# cat /proc/net/dev
```

3、使用 Linux-Dash 实现可视化监控

- (1) 安装 Apache 服务，启动服务并设置为开机自启。

```
# 使用 yum 工具安装 Apache
[root@Lab-07-Task-01 ~]# yum install -y httpd
# 使用 systemctl start 命令启动 Apache 服务
[root@Lab-07-Task-01 ~]# systemctl start httpd
# 使用 systemctl enable 命令设置 Apache 服务开机自启
[root@Lab-07-Task-01 ~]# systemctl enable httpd
```

- (2) 安装 python、php 环境。

```
# 使用 yum 工具安装 python、php
[root@Lab-07-Task-01 ~]# yum -y install python php php-fpm
```

- (3) 安装 git 工具，下载 Linux-Dash 源码，并重启 Apache 服务。

```
# 使用 yum 工具安装 git
[root@Lab-07-Task-01 ~]# yum install -y git
# 在/var/www/html 目录下，下载 linux-dash 的源码包
[root@Lab-07-Task-01 ~]# cd /var/www/html
[root@Lab-07-Task-01 html]# git clone https://github.com/afaqurk/linux-dash.git
# 重新加载 httpd 配置文件
[root@Lab-07-Task-01 html]# systemctl restart httpd
```

- (4) 配置防火墙策略和 SELinux。

```
# 配置防火墙允许 80/tcp 端口访问
[root@Lab-07-Task-01 ~]# firewall-cmd --zone=public --add-port=80
/tcp --permanent
[root@Lab-07-Task-01 ~]# firewall-cmd --reload
[root@Lab-07-Task-01 ~]# firewall-cmd --list-all
# 配置 SELinux
[root@Lab-07-Task-01 ~]# setenforce 0
# 修改 SELinux 配置文件
[root@Lab-07-Task-01 ~]# vi /etc/selinux/config
----- config -----
SELINUX=permissive
-----
```

- (5) 修改 httpd 的配置文件设置 Linux-Dash 目录访问规则，创建.htaccess 文件，启用基本身份验证。

```
# 修改 httpd 的配置文件，设置目录访问规则
[root@Lab-07-Task-01 ~]# vi /etc/httpd/conf.d/dash.conf
----- dash.conf -----
<Directory /var/www/html/linux-dash/app>
  Options FollowSymLinks
  AllowOverride All
  Order allow,deny
  allow from all
</Directory>
-----
# 在/var/www/html/linux-dash/app 目录下创建.htaccess 文件
[root@Lab-07-Task-01 ~]# vi /var/www/html/linux-dash/app/.htaccess
```

```
----- .htaccess -----
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /var/www/html/linux-dash/app/.htpasswd
Require valid-user
-----
```

(6) 设置 Linux-Dash 的登录账号和密码，并重启 Apache 服务。

```
# 设置访问 linux-dash 页面的登录账号和密码
[root@Lab-07-Task-01 ~]# htpasswd -c /var/www/html/linux-dash/ap
p/.htpasswd admin
New password:
Re-type new password:
Adding password for user admin
# 重新服务
[root@Lab-07-Task-01 ~]# systemctl restart httpd
```

(7) 在 Windows 本地客户端使用浏览器访问 <http://172.31.0.171/linux-dash/app/>，输入 Linux-dash 的账号和密码：admin，能够访问 Linux-dash 监控页面，如图 7-1 所示。

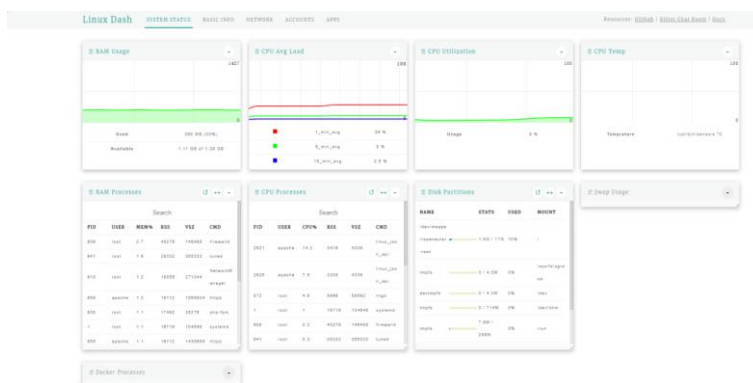


图 7-1 Linux-dash 监控

八、实验考核

实验考核分为【实验随堂查】和【实验线上考】两个部分。

实验随堂查：每个实验设置 2-5 考核点。完成实验任务后，任课教师随机选择一个考核点，学生现场进行演示和汇报讲解。

实验线上考：每个实验设置 10 道客观题。通过线上考核平台（如课堂派）进行作答。

1、实验随堂查

本实验随堂查设置 3 个考核点，具体如下。

考核点 1：解读用命令工具监控的状态指标

考核点 2：部署 linux-dash 并访问

考核点 3：解读 linux-dash 监控指标

2、实验线上考

本实验线上考共 10 题，题型为单选、多选、判断、填空等题型。