



浪潮信息云峦服务器操作系统 KeyarchOS FAQ 手册

浪潮电子信息产业股份有限公司

2024 年 12 月

目录

1 前言.....	6
2 日志收集	6
2.1 安装过程日志	6
2.2 系统运行日志	6
2.2.1 sosreport 日志.....	6
2.2.2 串口日志	7
2.2.3 内存转储 kdump	11
2.2.4 mcelog	14
2.2.5 dmesg	15
2.2.6 message 日志不打印.....	16
3 安装失败	16
3.1 Your BIOS-based system needs a special partition to boot from a gpt disk label. To continue, please create a 1Mib biosboot type partition. 16	
3.2 通过 bmc 挂载镜像的方式安装系统时，无法启动安装的图形界面，只能在文本模式下安装	17
3.3 通过 BMC 挂载镜像安装 KOS 时，选择 Install kos 5 后，在启动日志打印中出现 dracut-initqueue timeout 报错信息	18
3.4 安装 KOS 时在安装界面报错 “Some packages from local repository	

have incorrect checksum"	19
3.5 安装源配置出错，需要排查 ISO sha256 值.....	19
3.6 使用 UltraISO 制作 U 盘启动盘，出现无法启动安装问题	20
3.7 存在 btrfs 文件系统的硬盘上安装 KeyarchOS 5.8 系统，安装时异常重启， 无法启动到安装界面.....	21
3.8 飞腾 5000C 安装 4 根内存条时，系统启动无法安装 KeyarchOS，会循环重 启	22
3.9 在 virtual Box 中安装 KeyarchOS 系统，安装成功后启动无法进入图形界 面	24
4 启动异常/宕机/假死/黑屏	26
4.1 系统下运行 yum remove libaio 之后系统重启进入 emergency mode	26
4.2 系统根目录占满导致无法启动图形界面	26
4.3 系统下存在两块相同的 lvm 格式磁盘，无法启动.....	27
4.4 uefi 模式启动时报错	27
4.5 Warning: /dev/disk/by-uuid/xxx does not exist.....	28
4.6 在海光机器上重启虚拟机，虚拟机会崩溃	28
5 CPU 部分	29
5.1 cpuinfo 中 core id 的排序乱序	29
5.2 不同服务器上查看 numa node 的 cpu 编号可能不同	29

5.3 Processor heated above trip temperature. Throttling enabled ..	30
5.4 短时间内发生的 cpu 温度告警问题.....	31
5.5 从 3.10 内核升级到 4.18 内核之后/proc/cpuinfo 下的指令集不一致	31
5.6 vdbench 性能测试工具在 KOS5.8 系统上性能表现不佳	33
5.7 当 cpu 核心数超过 128 后，系统启动日志中存在 coretemp 相关错误	33
5.8 系统启动日志中出现 intel_vsec 相关错误.....	33
6 磁盘部分	34
6.1 磁盘乱序.....	34
6.2 fdisk -l 和 lsblk 显示磁盘大小不同.....	39
6.3 如何在安装过程中清理磁盘	39
6.4 搭配希捷 SAS 硬盘进行 AC 测试时报 scsi_dma_map failed: request for 524288 bytes.....	40
6.5 开启 vmd 并将 nvme 组 raid 之后，系统日志中出现 mdadm 相关错误	41
7 网络部分	42
7.1 avahi-daemon: Host name conflict, retrying with linux-2.....	42
7.2 Activation via systemd failed for unit 'dbus- org.freedesktop.resolve1.service': Unit dbus- org.freedesktop.resolve1.service not found.	42

7.3 搭配博通网卡, 系统启动日志中存在 Error recovery 打印	43
7.4 搭配 KOS 5.8 SP1 系统, dmesg 有 i40iw 报错.....	44
7.5 KOS 操作系统下通过迈洛斯网卡配置 ib 网络, 缺少 opensm 软件包	44
8 内存部分	46
8.1 Out of Memory: Killed process [PID] [process name]	46
8.2 重启对比/proc/meminfo 的内存容量发现内存大小有细微的变化 ...	46
8.3 系统 grub 中增加的 memmap 参数中不能保留\$符的问题	47
9 显示部分	50
9.1 journal: failed to get edid data: EDID length is too small.....	50
9.2 failed to get edid: unable to get EDID for output.....	50
9.3 Failed to grab accelerator for keybinding settings	51
9.4 Error when getting information for file "/root/.cache/gnome- software/shell-extensions/gnome.json" : No such file or directory	51
9.5 Gdm: GdmDisplay: Session never registered, failing.....	52
9.6 在机器重启之前和重启之后, 进程不间断运行的原因分析	52
10 其他部分	53
10.1 BMC 控制台大小写键盘混乱	53
10.2 每次重启 messages 中的启动时间会快 8 小时, 直到 Journal 启动后恢复 正常时间.....	54

10.3 如何在 KOS 系统中读取服务器 SN 信息	56
10.4 如何使用系统镜像配置本地 yum 源.....	56
10.5 1128 版本镜像升级 dnf 后需要激活	57
10.6 系统启动时 NetworkManager-wait-online 服务启动失败	58
10.7 KeyarchOS 5.8 系统如何启用并设置 sar 日志每秒搜集一次	58
10.8 KeyarchOS 5.8 使用 localectl 命令修改系统 encoding, 由 UTF-8 改成 gbk, 系统重启后, 图形界面无法开启终端.....	60
10.9 系统下执行 yum install -y 命令安装软件包时提示 “GPG 公钥已安装, 但是 不适用于此软件包”	62
10.10 系统 grubenv 文件中的 32 位字符串是如何生成的	62
10.11 升级内核后, 修改 grub 文件增加内核参数重启, 参数在新内核上不生效	63
10.12 配置 VNC 服务端, 端口设置从 2 开始.....	64
10.13 Auditd 服务使用 systemctl 概率性无法启动, 显示 Operation refused	64
10.14 执行 ipmitool mc watchdog get 时显示 stopped 状态.....	65

1 前言

本文档的适用对象是浪潮信息云峦服务器操作系统 KeyarchOS（简称 KOS）产品的 POC 测试、实施、售后维护人员。

本文档对 KOS 产品在 POC 测试、实施、售后阶段频繁遇到的问题进行了总结，建议相关技术人员提前阅读本文档，以便更高效的完成相关技术工作。

如需安装 KOS 操作系统，可从以下位置下载：

条目	下载地址
镜像位置	https://www.ieisystem.com/kos/kos_jx_index.shtml?cid=11378

2 日志收集

2.1 安装过程日志

在安装过程中，使用 `ctrl+alt+F2` 切换到字符界面，将 `/tmp` 目录下所有文件打包，并通过挂载 U 盘拷出。

```
# tar -czvf install_log.tar.gz /tmp
```

适用场景：系统可以正常启动到安装界面，在安装过程中遇到问题，但未宕机可响应键盘操作。

2.2 系统运行日志

2.2.1 sosreport 日志

sosreport 是一个从运行中的系统收集大量配置细节、系统信息和诊断信息的工具，它可以获取以下日志：

操作系统版本、已安装的内核、已加载的内核模块、打开的文件列表、PCI 设备列表

引导信息、文件系统、内存、主机名、已安装的 RPM

系统 IP、网络详细信息、系统路由

挂载点、位于 /etc 文件夹中的所有配置文件，以及位于 /var 文件夹中的所有日志文件

运行中的进程信息、进程树输出

常用收集方法：使用 root 用户执行 `sosreport --batch` 或 `sos report --batch` 命令，约 1 分钟后在 /var/tmp/ 下生成一个 `sosreport.xxx.tar.xz` 的压缩包文件。

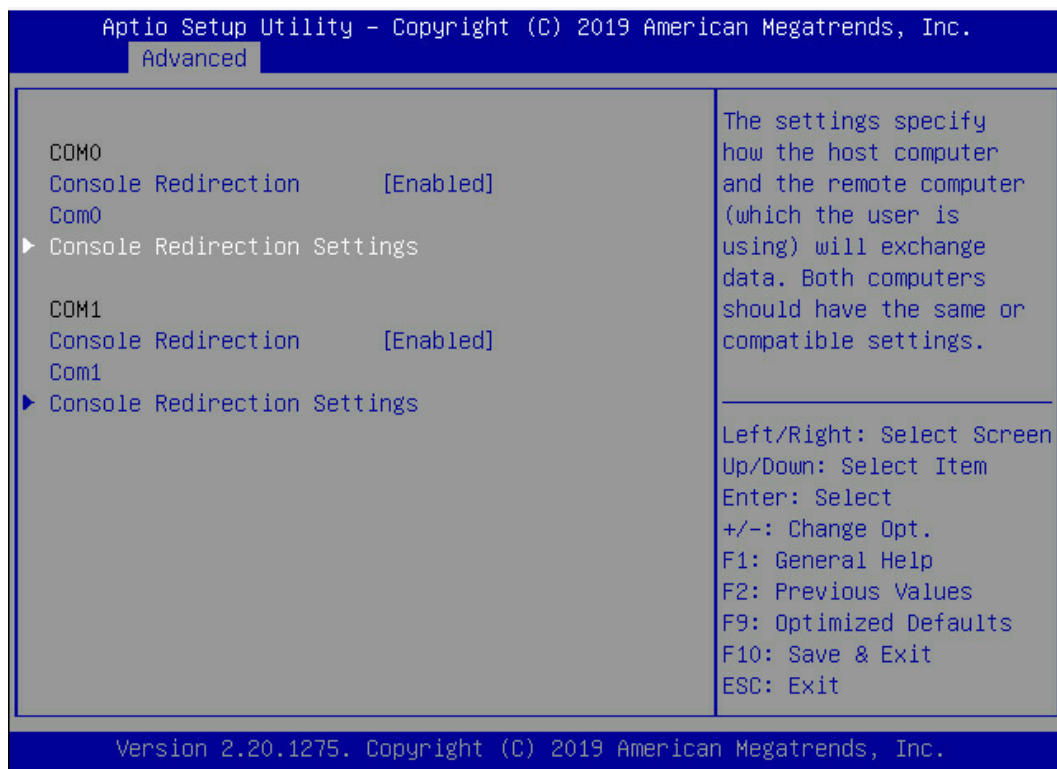
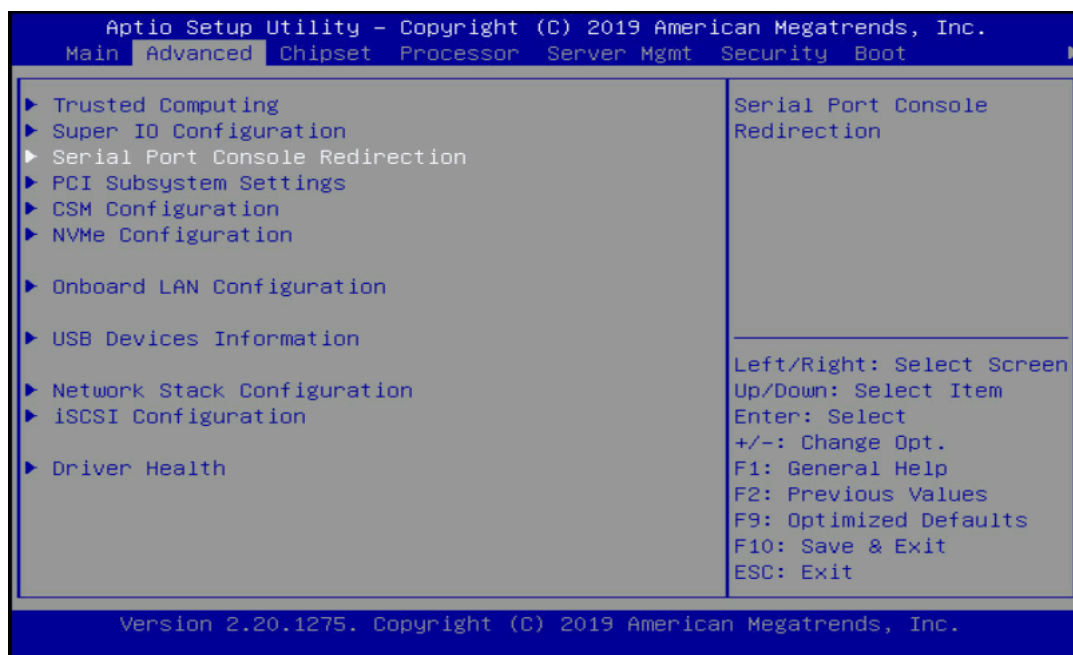
2.2.2 串口日志

在服务器测试和系统日常使用过程中，偶尔会遇到系统崩溃之类的严重错误，此时操作系统会将部分重要信息输出至串口。连接物理串口获取这些信息十分不便，因此建议进行串口重定向配置，使得用户可以利用串口获取这些信息。

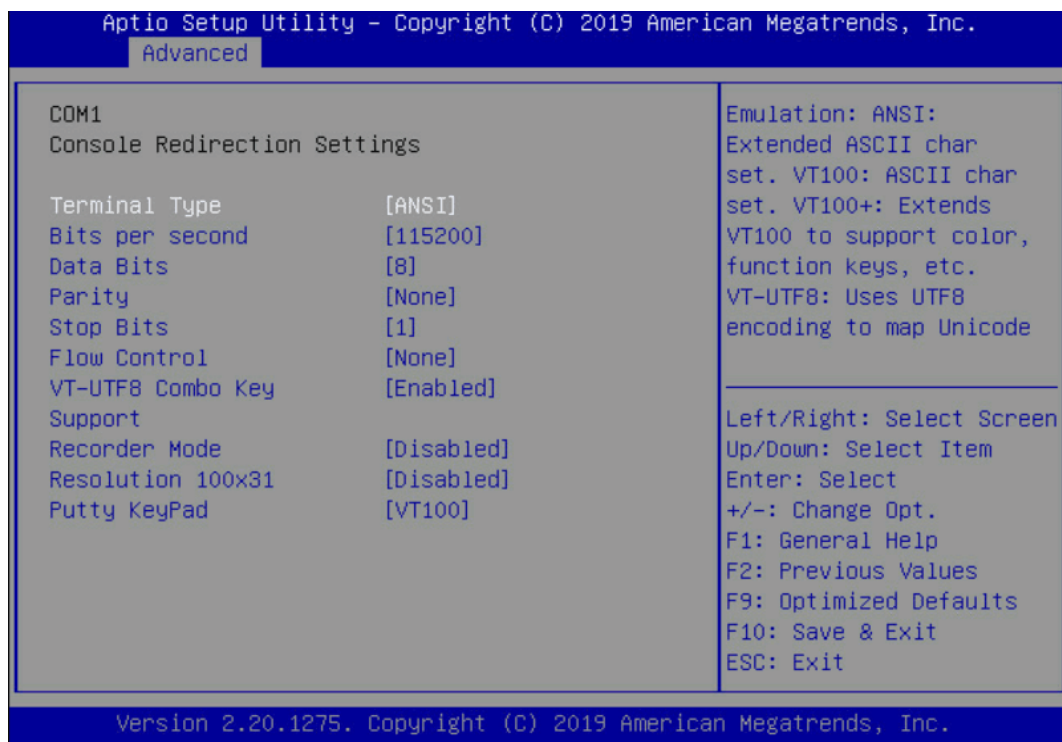
KOS 串口重定向配置方法可参考如下（不同服务器 BIOS 配置界面可能不同）：

2.2.2.1 BIOS 配置开启串口重定向

1、选项 Advanced->Serial Port Console Redirection->Console Redirection 中的 COM 端口设置为 Enable



2、配置相关 COM 端口及波特率参数



2.2.2.2 KOS 配置 GRUB 串口参数

1、增加串口参数：

```
# grubby --args= " console=tty0 console=ttyS0,115200 " --update-  
kernel=" /boot/vmlinuz-<对应内核>"
```

2、重启

```
# reboot
```

3、检查配置生效

```
# cat /proc/cmdline
```

2.2.2.3 收集串口日志

(1) 方法一：

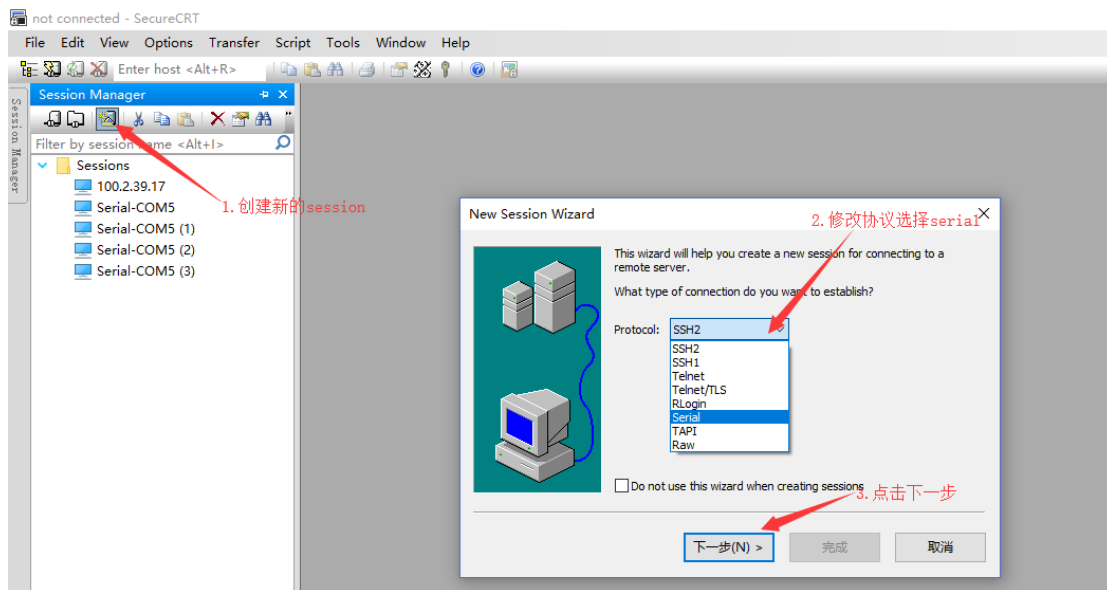
在任何一台可以和 BMC 连通的机器上，执行：

```
#ipmitool -I lanplus -H 服务器BMC地址-U 服务器BMC用户名-P 服务器BMC
```

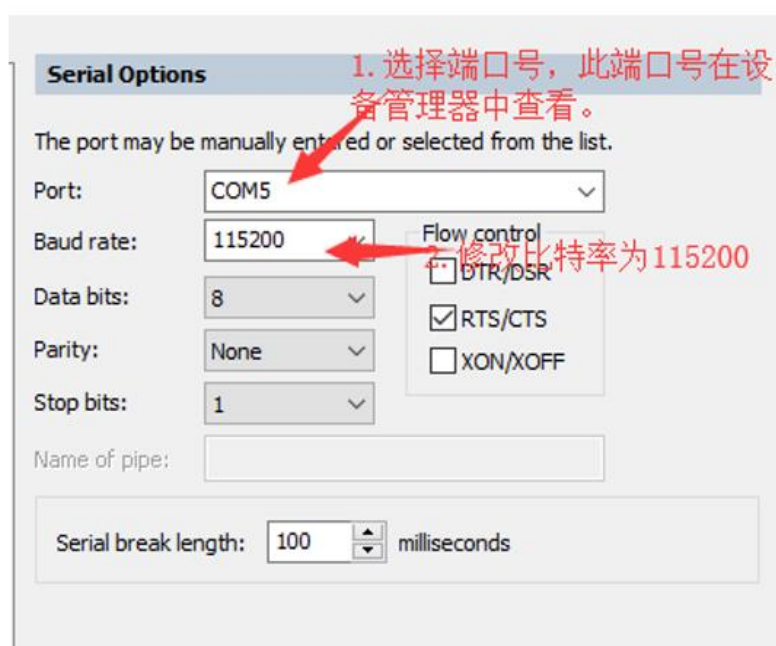
密码 sol activate > 本地文件

(2) 方法二:

使用串口转接线连接笔记本和服务器 BMC 管理口，笔记本上使用远程连接工具，如 SecureCRT，配置串口连接。



配置串口端口号和比特率，点击连接，即可在当前窗口看到服务器输出串口信息，将输出的日志保存到文件中。



2.2.3 内存转储 kdump

当系统崩溃时，需要收集足够的信息来分析引起崩溃的原因，内存转储是一种在系统崩溃时将系统的内存信息以 DUMP 文件的形式保存下来的机制。

kdump 是利用 kexec 实现的可靠的内核崩溃转储机制。在系统崩溃时使用 kexec 重新启动第二个内核，去捕获并转储内核崩溃信息。kdump 利用 kexec 可以不经 BIOS 直接使用预留的内存启动转储内核，在系统崩溃后捕获保留第一内核崩溃信息以供调试分析，是用来分析系统崩溃、内核 panic、死锁时常用的工具和服务。

2.2.3.1 修改启动 crashkernel 参数

KOS 系统默认自带 kdump 设置，对于 ANCK 和 RHCK 内核默认有不同的 crashkernel 策略，以下分别说明：

ANCK 内核默认参数如下：

ANCK	system memory:memory reserved by crashkernel
x86_64	0G-2G:0M,2G-8G:192M,8G-:256M
aarch64	0G-2G:0M,2G-8G:192M,8G-:768M

若需要手工修改的话，按如下方法操作：

```
grubby --args=" crashkernel=0M-2G:0M, 2G-8G:192M,8G-:***M"
```

```
--update-kernel=" /boot/vmlinuz-<对应内核>"
```

重启后可通过 `kdumpctl showmem` 命令确认是否修改成功。

RHCK 内核默认参数如下：

RHCK	参数
x86_64	crashkernel=auto
aarch64	crashkernel=auto

若需要手工修改的话，按如下方法操作：

```
grubby --args=" crashkernel=***M" --update-kernel=" /boot/vmlinuz-<
对应内核>"
```

重启后可通过 `kdumpctl showmem` 命令确认是否修改成功。

对于 RHCK 内核，建议 x86_64 架构下配置 256M，aarch64 架构下配置 768M。

注：（1）`crashkernel=auto` 是一种算法计算的内存预留，可能无法满足所有系统的需求（尤其是对于具有许多 IO 卡和加载内核驱动程序的配置），需要通过测试 `kdump` 确保 `crashkernel=auto` 保留的内存足以供目标机器使用。

（2）针对 Ampere CPU `crashkernel` 需要设置的到 2G 以上。

2.2.3.2 配置转储目录

`kdump` 转储目录在 `/etc/kdump.conf` 中配置，默认转储到 `/var/crash/` 下。

如果需要转储到其他目的地，例如转储到其他磁盘目录，需要在文件所在的存储设备前声明文件系统类型：

```
ext4 /dev/sda1
```

```
path=/usr/local/crash
```

转储到 NFS 设备：

```
net * <nfs server>: </nfs/mount> *
```

比如: net nfs.example.com:/export/vmcores

转储到可 SSH 访问的设备:

```
net * <user> @ <ssh server> *
```

比如: net root@ssh.example.com

2.2.3.3 配置触发条件

通过编辑/etc/sysctl.conf 设置系统触发 Kdump 的条件:

```
# vim /etc/sysctl.conf
```

配置完成后, 执行如下命令立即生效:

```
# sysctl -p
```

kernel.sysrq=1/0	#是否响应 sysrq 魔术键触发 panic
kernel.unknown_nmi_panic=1/0	#硬件 NMI 按钮被按下时是否触发 panic
kernel.panic_on_unrecovered_nmi=1/0	#发生 unrecovered 的 NMI 时是否触发 panic
vm.panic_on_oom=1/0	#发生内存 OOM 时是否触发 panic
kernel.panic_on_oops = 1/0	#内核在发生 oops 错(非法内存访问或非法指令)时是否触发 panic
kernel.softlockup_panic = 1/0	#内核在发生软锁(softlockup)时是否触发 panic
kernel.panic_on_io_nmi = 1/0	#内核收到因 I/O 错误导致的 NMI 时是否触发 panic
kernel.hung_task_panic = 1/0	#内核在进程 hung 住时, 是否触发 panic
kernel.hung_task_timeout_secs=120	#进程 hung 住多少时间, 触发 panic
kernel.hardlockup_panic = 1/0	#发生 hardlockup 时是否触发 panic

注: 以上系统内核参数, 正常情况下按系统默认配置即可, 在需要 debug 相关问题时, 根据需要配置开启相关参数触发 kdump 定位问题。

2.2.3.4 开启 kdump 服务

系统下的 kdump 服务，需要在系统下正常运行，并默认开机自启。

```
# systemctl restart kdump
```

```
# systemctl enable kdump
```

```
# systemctl status kdump
```

2.2.3.5 重启检查服务生效

```
# reboot
```

```
# systemctl status kdump
```

2.2.3.6 验证触发 kdump

测试 kdump 配置是否生效时，需要 SysRq 机制的支持才能使用，检查并确保 sysrq 开启：

```
# echo 1 > /proc/sys/kernel/sysrq
```

```
# echo c > /proc/sysrq-trigger
```

重启后在 kdump.conf 文件配置的转储目录下，存在 vmcore 文件即代表配置成功。

```
[root@l130anck mnt]# ls -lR /var/crash/
/var/crash/:
总用量 0
drwxr-xr-x 2 root root 67 11月 30 16:25 127.0.0.1-2022-11-30-16:23:51

'/var/crash/127.0.0.1-2022-11-30-16:23:51':
总用量 8252496
-rw-r--r-- 1 root root 311648 11月 30 16:25 kexec-dmesg.log
-rw-r--r-- 1 root root 8450075694 11月 30 16:25 vmcore
-rw-r--r-- 1 root root 163509 11月 30 16:23 vmcore-dmesg.txt
```

2.2.4 mcelog

KOS 中，默认 mcelog 服务不创建 /var/log/mcelog 文件，mcelog 相关的内容会默认记录到 messages 文件中，若需要单独的 mcelog 文件，需要修改 daemon 服

务的参数，加上 “--logfile=/var/log/mcelog” ，并重启 mcelog 服务。

```
# systemctl stop mcelog
```

```
# vim /usr/lib/systemd/system/mcelog.service
```

```
ExecStart=/usr/sbin/mcelog --ignorenodev --daemon --foreground --  
logfile=/var/log/mcelog
```

```
# systemctl daemon-reload
```

```
# systemctl start mcelog
```

```
# systemctl status mcelog
```

2.2.5 dmesg

KOS 及较新的 Linux 版本中，/var/log/dmesg 都已经不再默认生成，因为 /var/log/messages 日志中已经包含了系统启动日志。

KOS 中可以通过如下方法重新开启记录 /var/log/dmesg：

```
# vim /etc/systemd/system/dmesg.service
```

```
[Unit]  
Description=Create /var/log/dmesg on boot  
ConditionPathExists=/var/log/dmesg  
  
[Service]  
ExecStart=/usr/bin/dmesg  
StandardOutput=file:/var/log/dmesg  
  
[Install]  
WantedBy=multi-user.target
```

```
# touch /var/log/dmesg
```

```
# restorecon -v /var/log/dmesg
```

```
# systemctl enable dmesg
```



```
# systemctl start dmesg
```

2.2.6 message 日志不打印

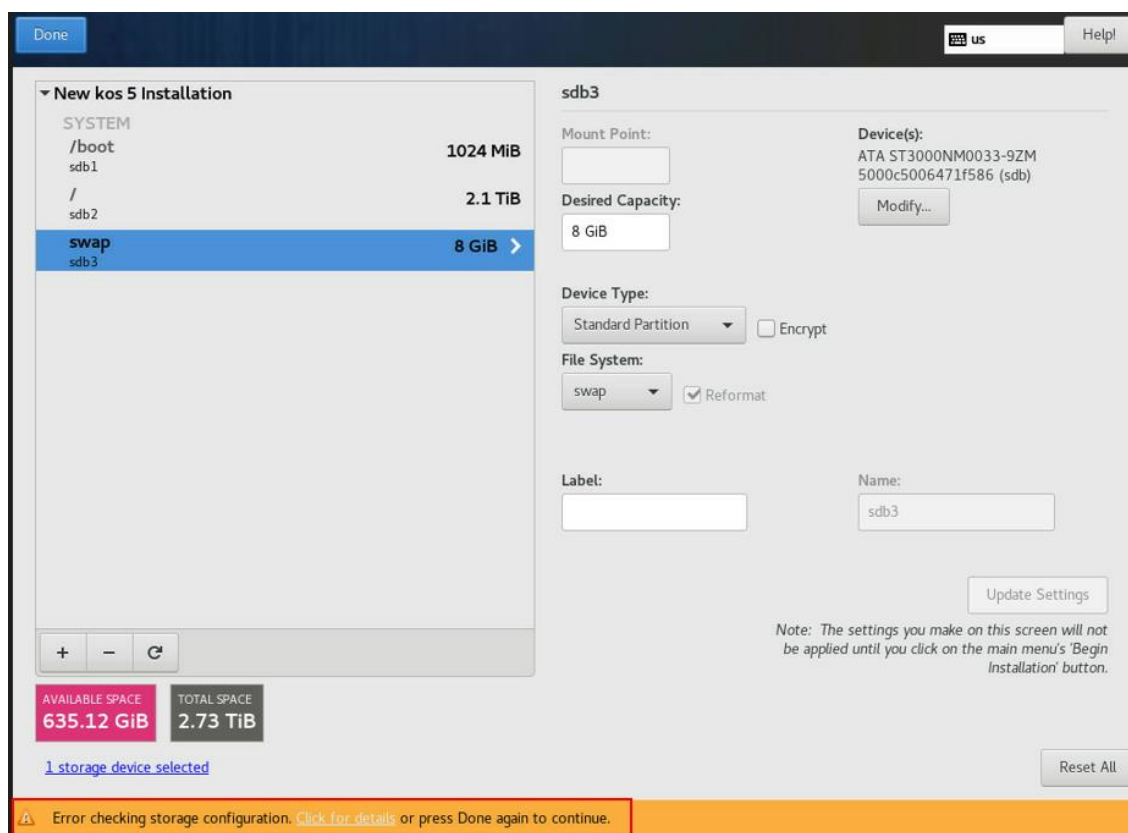
请确保 rsyslog 服务保持启动状态。

3 安装失败

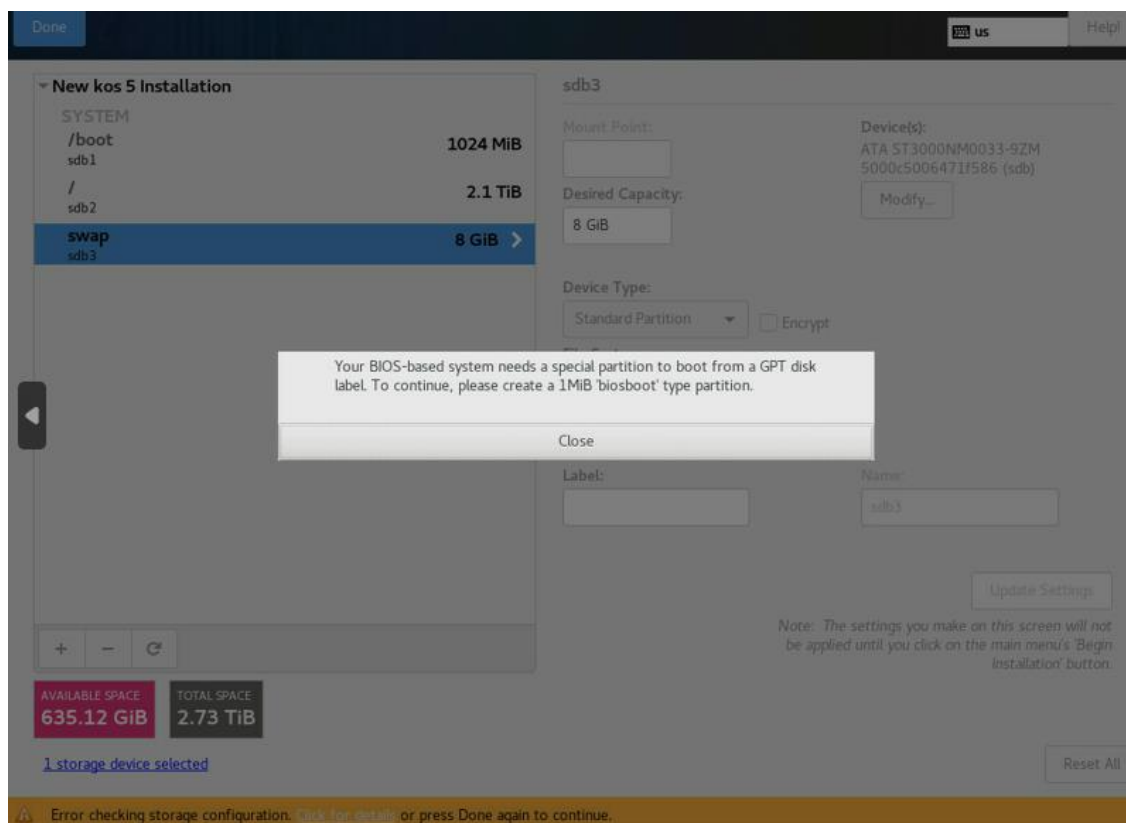
3.1 Your BIOS-based system needs a special partition to boot from a gpt disk label. To continue, please create a 1Mib biosboot type partition.

【问题描述】

Legacy 模式下使用超过 2T 空间的磁盘安装系统，手动分区后点击 Done 会提示如下报错：



点击 Click for details，显示如下：



【原因说明】

在 legacy 模式安装系统时，使用 gpt 分区的磁盘，需要创建 biosboot 分区，用来识别大于 2T 的 gpt 分区磁盘。

【解决方案】

如果使用 kickstart，增加 `part biosboot --fstype=biosboot --size=1`；

如果使用图形界面安装，增加 1MB 大小的 biosboot 分区。

3.2 通过 bmc 挂载镜像的方式安装系统时，无法启动安装的图形界面，只能在文本模式下安装

【问题描述】

通过 bmc 挂载镜像的方式安装系统时，无法启动安装的图形界面，只能在文本模式下安装。

```
* installation log files are stored in /tmp during the installation
* shell is available on TTY2
* when reporting a bug add logs from /tmp as separate text/plain attachments
03:04:38 Not asking for UMC because we don't have a network

X did not start in the expected time, falling back to text mode. There are multiple ways to avoid this issue:
* Do not load the stage2 image over a slow network link.
* Wait longer for the X server startup with the inst.xtimeout=<SECONDS> boot option. The default is 60 seconds.
* Load the stage2 image into memory with the rd.live.ram boot option to decrease access time.
* Enforce text mode when installing from remote media with the inst.text boot option.
=====
Installation
1) [x] Language settings                2) [x] Time settings
   (English (United States))           (Asia/Shanghai timezone)
3) [!] Installation source              4) [!] Software selection
   (Processing...)                     (Processing...)
5) [!] Installation Destination        6) [ ] Network configuration
   (Processing...)                     (Not connected)
7) [x] Root password                   8) [!] User creation
   (Root account is disabled.)         (No user will be created)

Please make a selection from the above ['b' to begin installation, 'h' to help,
'q' to quit, 'r' to refresh]:
```

【原因说明】

由于通过 bmc 挂载镜像安装时，受网络环境的影响，加载 Xserver 比较慢，预期时间(默认 60S)内未加载完成，所以切换成文本模式进行安装。

【解决方案】

根据提示在安装系统时添加 inst.xtimeout=<SECONDS>（默认 60s）grub 参数来延长 X Server 加载时间。

3.3 通过 BMC 挂载镜像安装 KOS 时，选择 Install kos 5 后，在启动日志打印中出现 dracut-initqueue timeout 报错信息

【问题描述】

通过 BMC 挂载镜像安装 KOS 时，选择 Install kos 5 后，在启动日志打印中出现 dracut-initqueue timeout 报错信息

【原因说明】

该问题由于启动 vmlinuz 和 initramfs 文件系统时超时导致，导致超时的原因是由于网络波动造成 img 传输问题，从而导致报错，出现该问题后可检查网络环境或者通过 U S B 本地安装。

【解决方案】

无

3.4 安装 KOS 时在安装界面报错 “Some packages from local repository have incorrect checksum”

【问题描述】

安装 KOS 时在安装界面报错 “Some packages from local repository have incorrect checksum”

【原因说明】

出现该问题的原因可能由以下几种可能：

a.如通过挂载 ISO 安装，请检查镜像哈希值或者由于网络传输问题导致，软件包哈希值不对。

b.如通过 U 盘安装，请检查启动盘是否正确。

【解决方案】

无

3.5 安装源配置出错，需要排查 ISO sha256 值

【问题描述】

刻录 U 盘安装或挂载 ISO 镜像安装时界面显示 repo 错误或者日志显示软件包 checksum 错误

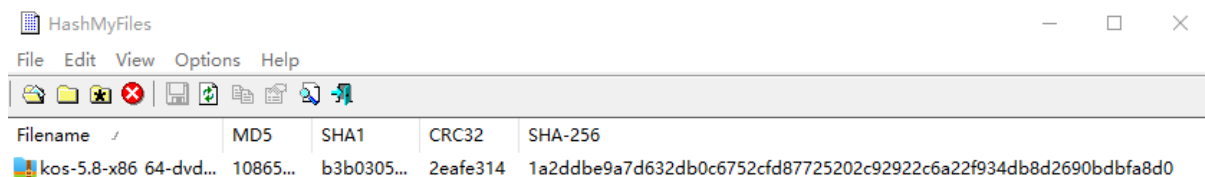
【原因说明】

无

【解决方案】

请检查 ISO 镜像是否完整。检查方法如下：

1.在 windows 系统下使用 HashMyFiles.exe 工具生成 iso 镜像的 sha256 值:



2.查看官网下载链接中的 sha256 值:

<https://zh.ieisystem.com/product/kos/9453.html>

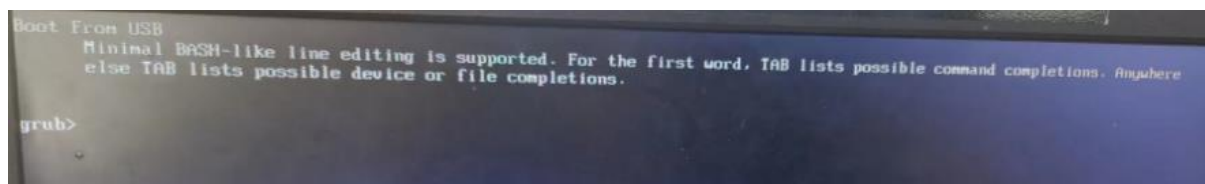


3. 对比 HashMyFiles.exe 工具生成的 sha256 与官网下载链接中的 sha256 值：如果一致则 iso 是完整的；如果不一致 iso 镜像就是不完整的。

3.6 使用 UltraISO 制作 U 盘启动盘，出现无法启动安装问题

【问题描述】

如图，使用 UltraISO 制作 U 盘启动盘后，无法启动安装：



【原因说明】

无

【解决方案】

请确认 UltraISO 工具的写入方式需为 RAW 。



3.7 存在 btrfs 文件系统的硬盘上安装 KeyarchOS 5.8 系统，安装时异常重启，无法启动到安装界面

【问题描述】

存在 btrfs 文件系统的硬盘上安装 KeyarchOS 5.8 系统，安装时异常重启，无法启动到安装界面。串口日志中存在如下错误：

```
ERROR:anaconda.modules.common.task.task:Thread
```

```
AnaTaskThread-
```

ScanDevicesTask-1 has failed: Traceback (most recent call last):

File `"/usr/lib64/python3.6/site-packages/gi/overrides/BlockDev.py"`, line 1038, in wrapped

```
ret = orig_obj(*args, **kwargs)
```

File `"/usr/lib64/python3.6/site-packages/gi/overrides/BlockDev.py"`, line 163, in `btrfs_list_subvolumes`

```
return _btrfs_list_subvolumes(mountpoint, snapshots_only)
```

GLib.GError: g-bd-init-error-quark: The function 'bd_btrfs_list_subvolumes' called, but not implemented! (1)

【原因说明】

系统镜像中的 python-blivet 版本存在 bug，需要升级到 3.6.0-4 版本

【解决方案】

1. 格式化存在 btrfs 文件系统的硬盘

2. 使用 KeyarchOS5.8 20230826-oe 镜像安装系统

3.8 飞腾 5000C 安装 4 根内存条时，系统启动无法安装 KeyarchOS，会循环重启

【问题描述】

飞腾 5000C 安装 4 根内存条时，过 grub 界面后，BMC kvm 一直黑屏，然后出现重启。串口日志出现如下：

```
[2024-01-15 10:56:19] [16.635100] megasas: 07.706.03.00-rc1
```

```
[2024-01-15 10:56:19] [16.636254] libata version 3.00 loaded.
```

[2024-01-15 10:56:19] [16.637326] percpu: allocation failed, size=32 align=16
atomic=0, failed to allocate new chunk

[2024-01-15 10:56:19] [16.637338] CPU: 112 PID: 609 Comm: kworker/112:1
Tainted: GE 4.19.91anolis+ #14

[2024-01-15 10:56:19] [16.637340] Hardware name: Inspur CS5280F3/CS5280F3,
BIOS 1.16.00 11/17/2023

[2024-01-15 10:56:20] [16.637355] Workqueue: memcg_kmem_cache
memcg_kmem_cache_create_func

[2024-01-15 10:56:20] [16.637359] Call trace:

[2024-01-15 10:56:20] [16.637366] dump_backtrace+0x0/0x198

[2024-01-15 10:56:20] [16.637369] show_stack+0x1c/0x28

[2024-01-15 10:56:20] [16.637377] dump_stack+0xb4/0xe4

[2024-01-15 10:56:20] [16.637382] pcpu_alloc+0x820/0x840

[2024-01-15 10:56:20] [16.637385] __alloc_percpu+0x20/0x28

[2024-01-15 10:56:20] [16.637390] kmem_cache_open+0x3c8/0x490

[2024-01-15 10:56:20] [16.637393] __kmem_cache_create+0x34/0x1a8

[2024-01-15 10:56:20] [16.637395] create_cache+0xd0/0x208

[2024-01-15 10:56:20] [16.637397] memcg_create_kmem_cache+0x100/0x138

[2024-01-15 10:56:20] [16.637398] memcg_kmem_cache_create_func+0x24/0x98

[2024-01-15 10:56:20] [16.637403] process_one_work+0x1b4/0x3f8

[2024-01-15 10:56:20] [16.637404] worker_thread+0x158/0x4b0

[2024-01-15 10:56:20] [16.637409] kthread+0x130/0x138

[2024-01-15 10:56:20] [16.637416] Kernel panic - not syncing: Cannot create slab anon_vma(17:system.slice) size=104 realsize=104 order=0 offset=96 flags=40c0000

【原因说明】

内存条数量少，造成有的 numa 无法申请 vm area，percpu 的 vm area 分配时边界检查条件有误造成，忽略 vm area 边界较大时的检查条件。

【解决方案】

方法一：满插内存条

方法二：升级 kernel 到 4.19.91-27.4.20 及以上版本

3.9 在 virtual Box 中安装 KeyarchOS 系统，安装成功后启动无法进入图形界面

【问题描述】

- 1.在 virtual Box 虚拟化软件中安装 KeyarchOS 系统
- 2.安装完成后，重启操作系统
- 3.系统启动后无法进入图形界面

```

Starting Network Manager Wait Online...
[ OK ] Reached target Network.
Starting CUPS Scheduler...
Starting OpenSSH server daemon...
Starting Logout off all iSCSI sessions on shutdown...
Starting Dynamic System Tuning Daemon...
Starting GSSAPI Proxy Daemon...
[ OK ] Started Logout off all iSCSI sessions on shutdown.
Starting Hostname Service...
[ OK ] Started Hostname Service.
[ OK ] Listening on Load/Save RF Kill Switch Status /dev/rfkill Watch.
[ OK ] Started OpenSSH server daemon.
[ OK ] Started GSSAPI Proxy Daemon.
[ OK ] Reached target NFS client services.
Starting Network Manager Script Dispatcher Service...
[ OK ] Started Network Manager Script Dispatcher Service.
Starting Manage, Install and Generate Color Profiles...
[ OK ] Started Manage, Install and Generate Color Profiles.
[ OK ] Started CUPS Scheduler.
[ OK ] Started Dynamic System Tuning Daemon.
[ OK ] Started Network Manager Wait Online.
[ OK ] Reached target Network is Online.
Starting System Logging Service...
Starting Notify NFS peers of a restart...
[ OK ] Reached target Remote File Systems (Pre).
[ OK ] Reached target Remote File Systems.
Starting Virtualization daemon...
Starting Crash recovery kernel arming...
Starting Permit User Sessions...
[ OK ] Started Notify NFS peers of a restart.
[ OK ] Started Permit User Sessions.
Starting Hold until boot process finishes up...
[ OK ] Started Command Scheduler.
Starting GNOME Display Manager...
[ OK ] Started Job spooling tools.
[ OK ] Started GNOME Display Manager.

```

4. 执行 `ctrl + alt + F2` 切换到文本模式，执行 `systemctl status gdm` 命令检查 gdm 服务状态，发现出现 `Child process - 1788 was already dead`. 打印

```

[root@localhost ~]# systemctl status gdm
● gdm.service - GNOME Display Manager
   Loaded: loaded (/usr/lib/systemd/system/gdm.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-05-28 09:27:35 CST; 1min 20s ago
 Main PID: 1289 (gdm)
    Tasks: 3 (limit: 9011)
   Memory: 5.5M
    CGroup: /system.slice/gdm.service
            └─1289 /usr/sbin/gdm

5月 28 09:27:34 localhost.localdomain systemd[1]: Starting GNOME Display Manager ...
5月 28 09:27:35 localhost.localdomain systemd[1]: Started GNOME Display Manager.
5月 28 09:27:55 localhost.localdomain gdm[1289]: Gdm: Child process -1788 was already dead.
5月 28 09:27:55 localhost.localdomain gdm[1289]: Gdm: Child process -1788 was already dead.

```

5. 使用 `dmesg` 日志命令，查看系统日志出现如下错误：

```

21.113774] net: registered tcp NFSv4.1 backchannel transport module.
37.867990] cfb0211: Loading X.509 cert 'sforshee: 00b28ddf47aef9cea7'
37.873453] cfb0211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
39.973836] e1000: enp0s3 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
39.974176] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s3: Link becomes ready
39.986684] e1000: enp0s8 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
39.987030] IPv6: ADDRCONF(NETDEV_CHANGE): enp0s8: Link becomes ready
49.841782] bridge: filtering via arp/ip6tables is no longer available by default. Update your scripts to load br_netfilter if you need this.
65.269991] gnome-shell[1995]: segfault at 78 ip 00007f6ed2ca5fbb sp 00007f6e7bffd420 error 4 in libglib-2.0.so.0.6800.4[7f6ed2c00000+134000]
65.269998] Code: b1 65 00 75 06 48 8b 7d 00 eb d7 4c 89 e7 e8 2c 71 f7 ff 4c 89 e7 e8 24 64 f7 ff eb e8 66 90 f3 0f 1e fa 41 54 55 48 89 fd 53 <48> 8b 1f 48 89 df 48 85
db 75 1a e8 35 fd ff ff 49 89 c4 48 89 d8
[ 150.829071] gnome-shell[2355]: segfault at 78 ip 00007f327e4a6fbb sp 00007f323a3fd420 error 4 in libglib-2.0.so.0.6800.4[7f327e400000+134000]
[ 150.829082] Code: b1 65 00 75 06 48 8b 7d 00 eb d7 4c 89 e7 e8 2c 71 f7 ff 4c 89 e7 e8 24 64 f7 ff eb e8 66 90 f3 0f 1e fa 41 54 55 48 89 fd 53 <48> 8b 1f 48 89 df 48 85
db 75 1a e8 35 fd ff ff 49 89 c4 48 89 d8
[ 162.646641] rfkill: input handler disabled
[ 163.192296] rfkill: input handler enabled
NORMAL 2.txt
"2.txt" 583L, 39959C
text utf-8 5,169 words 100% 583/583 ln : 1 [35]trailing

```

【原因说明】

无

【解决方案】

因为复现概率低，所以可以在文本模式下执行 `systemctl restart gdm` 命令，重启 gdm 服务进行恢复

4 启动异常/宕机/假死/黑屏

4.1 系统下运行 `yum remove libaio` 之后系统重启进入 emergency mode

【问题描述】

在系统下运行 `yum remove libaio` 之后系统重启进入 emergency mode

【原因说明】

在执行 `yum remove libaio` 时可以看到会卸载 58 个依赖包，其中包括了 lvm2 的包，造成在系统重启的时候出现/home 分区无法挂载问题。

【解决方案】

进入 emergency 之后使用 `yum install lvm2` 可以解决，但是由于卸载的其他包太多，所以建议重装系统。

4.2 系统根目录占满导致无法启动图形界面

【问题描述】

当出现无法登录图形界面，但可以远程连接机器的情况时，可通过以下步骤检查并确认是否根目录被占满导致无法启动图形界面：

- 1、系统启动时，去除 “rhgb quiet” 参数，看到系统服务启动失败；
- 2、重新启动系统，修改 grub 参数增加 “3”，进入 runlevel 3，系统可以正常启动，再通过 init 5 进入图形界面也可正常进入；
- 3、系统下发现根目录 / 占用率 100%，有可能导致系统图形界面启动时的相关服

务启动异常；迁移释放根目录空间后，重启进入系统正常。

【原因说明】

系统根目录空间占用 100%，若默认使用图形界面登录，有可能导致系统图形界面启动时的相关服务启动异常，无法进入系统。

【解决方案】

单用户模式进入系统后，清理/目录的磁盘空间。

4.3 系统下存在两块相同的 lvm 格式磁盘，无法启动

【问题描述】

启动时报错：multiple LVM volume groups with the same name(kos)

【原因说明】

系统下 lvm vg id 不能重复

【解决方案】

挂载镜像，按 3 进入 shell；

输入 pvs -o +vg_uuid 获取 vg list；

使用 vgrename [VG UUID] vg-newname 修改 vg name。

修改后会破坏启动选项，该盘无法继续作为系统盘。

4.4 uefi 模式启动时报错

【问题描述】

系统以 uefi 模式安装完成后，重启系统时出现如下错误：

Press any key to continue ..invalid magic number :0

Error 13: Invalid or unsupported executable format

【原因说明】

UEFI 安装，不能安装 “Base system” 下的 “Base” 中的 Tboot 组件，因为

Tboot 不支持 UEFI。

【解决方案】

uefi 不支持 tboot，安装系统时要把 tboot 包给去掉，或者用 legacy 模式装系统。

4.5 Warning: /dev/disk/by-uuid/xxx does not exist

【问题描述】

系统启动时进入修复模式，并提示如下报错：

Warning: /dev/disk/by-uuid/xxx does not exist

【原因说明】

系统启动时识别到磁盘设备分区出现问题。

【解决方案】

系统启动时，低概率出现 ahci/raid/sas 驱动初始化磁盘控制器失败，未正常识别到上面连接的磁盘设备，建议检查下磁盘链路硬件方面的问题。

如果每次重启都会出现该问题，有可能上一次系统重启前，修改了 grub 文件中启动行中的 “root=UUID=XXX” 参数，可以优先排查 UUID 值是否正确。

4.6 在海光机器上重启虚拟机，虚拟机会崩溃

【问题描述】

在海光机器上，宿主机使用 kos 的 ANCK 4.19 和 ANCK 5.10 内核，使用 kos5.8 镜像安装虚拟机，虚拟机选择 4.18.0-372.26.1 内核，如果重启虚拟机，虚拟机有概率崩溃

【原因说明】

虚拟机使用的 4.18.0-372.26.1 和 4.18.0-372.41.1 内核没有合入海光 patch

【解决方案】

升级内核到 kernel-4.18.0-372.41.1.1 及以上版本

5 CPU 部分

5.1 cpuinfo 中 core id 的排序乱序

【问题描述】

通过 “cat /proc/cpuinfo | grep 'core id'” 抓取 core id 的排序乱序，不是按照 core id 的大小顺序排列的。

【原因说明】

这种乱序不是问题，在 OS 中是正常的，因为 core id 在 OS 中是一个逻辑值，和物理 CPU 没有对应关系，所以不能保证逻辑 core id 的值是顺序的。

【解决方案】

不是问题，忽略

5.2 不同服务器上查看 numa node 的 cpu 编号可能不同

【问题描述】

```
# numactl --hardware
```

```
available: 2 nodes (0-1)
```

```
node 0 cpus: 0 1 2 3 4 5 6 7 16 17 18 19 20 21 22 23
```

```
node 1 cpus: 8 9 10 11 12 13 14 15 24 25 26 27 28 29 30 31
```

```
# numactl --hardware
```

```
available: 2 nodes (0-1)
```

```
node 0 cpus: 0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30
```

```
node 1 cpus: 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31
```

不同服务器上执行 numactl --hardware 命令，查看 numa node 的 cpu 编号可能不同。

【原因说明】

内核无法设置某个 CPU（不论是物理 CPU 还是逻辑 CPU）的编号，内核只能通过 ACPI 接口从 BIOS 中获取 CPU 编号，内核根据 BIOS 提供的 apicid 来作为 CPU 的编号，具体每个 Node 上 CPU 编号是由 BIOS 和不同的 CPU 型号决定的。

【解决方案】

根据具体的 CPU 型号咨询 Intel 和 BIOS。

5.3 Processor heated above trip temperature. Throttling enabled

【问题描述】

在 mcelog 中看到的温度告警信息 Processor 48 heated above trip temperature. Throttling enabled.

【原因说明】

Intel CPU 有两个温度告警阈值：TM1 和 TM2，分别控制 CPU 节流和关闭系统。当到达 TM1 时，CPU 会报告一个 thermal event 给 mcelog。

【解决方案】

检查散热相关硬件

5.4 短时间内发生的 cpu 温度告警问题

【问题描述】

系统日志中存在下面的报错：

```
Oct 7 15:34:26 localhost mcelog: Processor 48 heated above trip temperature. Throttling enabled.
```

...

```
Oct 7 15:34:26 localhost mcelog: Processor 48 below trip temperature. Throttling disabled
```

【原因说明】

短时间的告警不会显著降低应用程序的性能，并且通常在处理器变化的正常范围内。

这样的事件可能会导致操作系统发出机器检查警报，系统的功能并不受这些警报的影响，特别是在运行动态处理器压力应用程序时。

【解决方案】

在短时间内发生的 cpu 温度告警"heated above trip temperature"，在同 1s 内又报 below trip temperature，可以忽略。

5.5 从 3.10 内核升级到 4.18 内核之后/proc/cpuinfo 下的指令集不一致

【问题描述】

从 3.10.0-957 内核升级到 4.10.0 内核之后，查看/proc/cpuinfo，发现两个版本 cpu 指令集有差异：

3.10 内核存在而 4.18 内核不存在的指令集：hle、rtm、spec_ctrl

4.18 内核存在而 3.10 内核不存在的指令集: cpuid、cpuid_fault、ept_ad、fsmr、invpcid_single、la57、md_clear、pconfig、rdpid、split_lock_detect、tme、wbnoinvd、xsaves

【原因说明】

/proc/cpuinfo 文件中的 cpu 指令集信息是由内核生成, 在内核中维护了一个数组 x86_cap_flags, 里面保存了当前内核支持的指令集信息, 当当前 cpu 支持该指令集, 且内核支持该指令集, 就会将信息输出的/proc/cpuinfo 中。

x86_cap_flags 这个数组是根据 arch/x86/include/asm/cpufeatures.h 文件通过 arch/x86/kernel/cpu/mkcapflags 文件生成。

4.18 中存在的指令集而 3.10 内核不存在, 原因是在 3.10 内核中并没有添加对这部分指令集支持的代码

rtm 和 hle 在 4.18 中不存在是由于 4.18 内核配置选项打开了 CONFIG_X86_INTEL_TSX_MODE_OFF, 打开这个选项之后, 内核会关闭对 rtm、hle 的支持

spec_ctrl 指令集在 4.18 中不存在的原因是 4.18 内核默认隐藏了该指令集, 如下如, 当每一个特性的注释是以双引号开头时, 内核就不会将该指令集打印到 /proc/cpuinfo 中

```
#define X86_FEATURE_SPEC_CTRL (18*32+26) /* " Speculation Control (IBRS + IBPB) */  
#define X86_FEATURE_INTEL_STIBP (19*32+27) /* " Single Thread Indirect Branch Predictor
```

【解决方案】

rtm 和 hle 指令集要在 4.18 内核的/proc/cpuinfo 中打印, 需要把内核编译选项 CONFIG_X86_INTEL_TSX_MODE_OFF 这个关掉, 把 CONFIG_X86_INTEL_TSX_MODE_ON 或者 CONFIG_X86_INTEL_TSX_MODE_AUTO 打开

spec_ctrl 指令集需要修改 arch/x86/include/asm/cpufeatures.h 文件, 让其打印到/proc/cpuinfo 中

5.6 vdbench 性能测试工具在 KOS5.8 系统上性能表现不佳

【问题描述】

在 KOS5.8 系统上使用 vdbench 性能测试工具进行测试时，发现测试结果低于预期

【问题原因】

该版本系统上幽灵漏洞补丁影响导致

【解决方案】

升级操作系统到 KOS5.8SP1

5.7 当 cpu 核心数超过 128 后，系统启动日志中存在 coretemp 相关错误

【问题描述】

当 CPU 核心数超过 128 后，系统启动日志出现如下错误：

Mar 14 11:25:14 localhost kernel: coretemp coretemp.0: Adding Core 128 failed

Mar 14 11:25:14 localhost kernel: coretemp coretemp.0: Adding Core 129 failed

【问题原因】

最初的 coretemp 驱动将 cpu 核心数限制在了 128，所以超过 128 后会出现错误

【解决方法】

升级 kernel 到 5.10.134-15.2.17 及以上版本解决

5.8 系统启动日志中出现 intel_vsec 相关错误

【问题描述】

系统启动日志出现如下错误：

```
[ 38.152424] pmt_telemetry intel_vsec.telemetry.0: Unsupported access type 15
[ 38.156772] pmt_telemetry: probe of intel_vsec.telemetry.0 failed with error -22
[ 38.158864] pmt_telemetry intel_vsec.telemetry.4: Unsupported access type 15
[ 38.174326] pmt_telemetry: probe of intel_vsec.telemetry.4 failed with error -22
```

【问题原因】

intel_pmt 设备的 namespace 中定义的 access_type 值为 15，不在 kernel 的定义范围

【解决方法】

升级 kernel 到 5.10.134-15.2.17 及以上版本解决

6 磁盘部分

6.1 磁盘乱序

【问题描述】

操作系统重启或热插拔的场景中，可能会发现磁盘设备的 sd[X]名称变化，比如由原来的 sda 变为 sdc，或磁盘设备的名称 sd[X]与磁盘设备的物理位置不是——对应的，此种现象统称为磁盘乱序。

【原因说明】

操作系统每次重启，磁盘设备名称都有可能变化，这是因为操作系统中驱动加载的顺序，磁盘扫描顺序都不是固定的；Linux 并没有固定的驱动加载顺序，磁盘设备的发现是一个并行和异步的过程，通常磁盘会按照被识别的顺序依次进行初始化，但是，盘符的命名（比如 sda）是按照驱动加载完成最先响应的顺序决定的。在 Server 重启、设备 Reconfiguration 等操作时有可能出现盘符变动，完全无法预测盘符与设备的物理槽位的对应关系。对于 SCSI 总线上的磁盘设备，扫描分配盘符的过程不仅取决于 SCSI INQUIRY, TEST UNIT READY 等命令的响应，还受在此期间系统上运行的进程数量、udev 工作者线程数量、连接到 Server 的其他外围设备数量以及初始化这些 PCI 设备所花费的 CPU cycles 的影响，也依赖于服务器上可用的 CPU cores 数量、内存大小等，这些因素可能导致设备初始化所需时间的微小变化，因此可能会得到不同的 “/dev/sdx” 名称。

这个现象在 Linux 中是正常现象，这并不是一个缺陷，而是 Linux 就是这样设计的，设备名（盘符）的设计初衷并不是用来和磁盘的物理位置或顺序进行——对应的，

应使用设备或数据自身提供的唯一标识来区分设备。

磁盘乱序的问题，在所有 Linux 版本中都存在，KOS 也不例外。

【解决方案】

一、 首选方案使用设备自标识符

使用设备或分区/LVM 卷/文件系统自身提供的标识，而不是使用/dev/sd[X]等由内核分配给每个设备使用的设备名称。建议使用磁盘的 /dev/disk/by-path (by-id 或 by-uuid 等) 来标识永久磁盘名。

在 grub 引导行中，就使用了不会在系统重新引导时发生变化的标识符标识根设备：

```
root=/dev/mapper/vg1-root    //LVM 卷名称
root=UUID=f2dc94c1-6124-4368-b215-f04ab171e024    //文件系统的
UUID    root=PARTUUID=7d3bb988-02d0-4c6c-a864-14f4acfd0398    //
分区 UUID
```

同样，在/etc/fstab 中，指定的分区名称也不依赖于 sd[X]名称：

```
/dev/mapper/vg1-root    /    ext4    defaults    1    1
UUID=0c12f366-3dfa-4ea9-923c-5a57994ceef8    /boot    ext4    defaults
```

1 2 二、使用 udev 规则绑定盘符

udev 是用于创建和命名与系统中存在的设备对应的/dev 设备节点的机制，udev 使用 sysfs 提供的匹配信息和用户提供的规则来动态添加所需的设备节点。

对于磁盘的唯一标识名称，使用 udev 规则来对磁盘的 SERIAL_ID/WWID/PATH

等设备的唯一索引做别名的方案，并不是对磁盘设备名进行修改，而是在 kernel 分配的磁盘设备名不变的基础上，添加一个别名，实际上磁盘乱序的现象仍然存在，只是从操作系统下提供了一个“不乱序的”设备名。

具体实现脚本举例：由于脚本是使用当前系统下的 `lsscsi` 命令显示的磁盘顺序，逐一对磁盘设备生成一条规则，需要在盘符顺序正确时运行，自动生成需要的 `udev rule` 文件，并刷新规则：

```
#!/bin/sh

RULE_FILE="/etc/udev/rules.d/99-order_serial_id.rules"

if [ -f "$RULE_FILE" ];then

    cat /dev/null > $RULE_FILE

fi

for dev in `lsscsi | grep "/dev/sd" | awk -F " " '{print $NF}`

do

    DEVNUM=`echo  ${dev: 7} | tr a-z A-Z`

    SERIAL=`udevadm info -q all -n  $dev | grep ID_SERIAL= | awk -F "="

'{print $NF }'`

    echo

    ACTION==\"add|change\",KERNEL==\"sd*\",ENV{ID_SERIAL}==\"$SERIAL\",S

YMLINK+=\"sd$DEVNUM%n\" >>$RULE_FILE

done
```

```
/sbin/udevadm control --reload-rules
```

```
/sbin/udevadm trigger --type=devices --action=change
```

以上脚本可以酌情进行修改，如更改为绑定 WWID/PATH 等。

规则删除脚本：

```
# vim remove-udev-persistent_disk_set.sh
```

```
#!/bin/sh
```

```
RULE_FILE="/etc/udev/rules.d/99-order_serial_id.rules"
```

```
if [ -f "$RULE_FILE" ];then
```

```
    rm -f $RULE_FILE
```

```
fi
```

```
/sbin/udevadm control --reload-rules
```

```
/sbin/udevadm trigger --type=devices --action=change
```

注：实质上使用 udev 绑定规则建立的新设备名称，和系统下默认已有的 /dev/disk/by-id、by-path 等软连接名称是同样的原理，也可以无需建立规则，直接使用该名称永久表示磁盘设备。

```
# ls -ltr /dev/disk/by-id
```

```
lrwxrwxrwx.  1  root  root      9  Nov      3  13:41  scsi-  
3600605b00a086f80236fd90e3c1ab21a -> ../../sda
```

```
lrwxrwxrwx.  1  root  root     10  Nov      3  13:41  scsi-  
3600605b00a086f80236fd90e3c1ab21a-part1 -> ../../sda1
```

```
lrwxrwxrwx. 1 root root 10 Nov 3 13:41 scsi-  
3600605b00a086f80236fd90e3c1ab21a-part2 -> ../../sda2
```

```
lrwxrwxrwx. 1 root root 9 Nov 3 13:41 scsi-  
3600605b00a086f80236fd91f3d22d2fe -> ../../sdb
```

```
lrwxrwxrwx. 1 root root 9 Nov 3 13:41 scsi-  
3600605b00a086f80236fd92d3dfdc85 -> ../../sdc
```

.....

```
# ls -ltr /dev/disk/by-path
```

```
lrwxrwxrwx. 1 root root 9 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:0:0  
-> ../../sda
```

```
lrwxrwxrwx. 1 root root 10 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:0:0-part1  
-> ../../sda1
```

```
lrwxrwxrwx. 1 root root 10 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:0:0-part2  
-> ../../sda2
```

```
lrwxrwxrwx. 1 root root 9 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:1:0  
-> ../../sdb
```

```
lrwxrwxrwx. 1 root root 9 Nov 3 13:41 pci-0000:86:00.0-scsi-0:2:2:0  
-> ../../sdc
```

.....

6.2 fdisk -l 和 lsblk 显示磁盘大小不同

【问题描述】

一块 10T 的希捷硬盘，使用 fdisk -l 命令查看，显示 10000.8GB，而使用 lsblk 命令显示的是 9.1T。

【原因说明】

正常现象，lsblk 和 multipath 计算磁盘大小使用 ioctl, 单位为 GiB（除以 1024），fdisk 计算磁盘大小的单位是 GB(除以 1000)。

【解决方案】

无

6.3 如何在安装过程中清理磁盘

【问题描述】

如何在系统安装过程中清理磁盘

【原因说明】

无

【解决方案】



在如上安装过程中的安装信息摘要图形界面，使用 `ctrl+alt+f2` 组合键切换终端进入命令行界面：

```
kos 5.8
Kernel 4.19.91-26.6.2.kos5.x86_64 on an x86_64

[anaconda root@localhost /]# ls /dev/sd*
/dev/sda /dev/sdb /dev/sdc
[anaconda root@localhost /]# wipefs -af /dev/sda
[anaconda root@localhost /]#
```

在命令行界面使用 `ls /dev/sd*` 命令查看磁盘；

使用 `wipefs -af /dev/sda` 命令清理磁盘（其中 `/dev/sda` 需要改成对应的盘符）。

清理完成后使用 `ctrl+alt+f6` 切换回安装界面继续安装系统。

6.4 搭配希捷 SAS 硬盘进行 AC 测试时报 `scsi_dma_map failed: request for 524288 bytes`

【问题描述】

搭配希捷 SAS 硬盘进行 AC 测试时概率性出现如下 failed 日志：

[313.520888] DMA: Out of SW-IOMMU space for 65536 bytes at device

0000:86:00.0

[313.520949] sd 0:0:35:0: scsi_dma_map failed: request for 524288 bytes!

[313.521009] DMA: Out of SW-IOMMU space for 65536 bytes at device

0000:86:00.0

[313.521012] sd 0:0:5:0: scsi_dma_map failed: request for 524288 bytes!

[313.521139] DMA: Out of SW-IOMMU space for 65536 bytes at device

0000:86:00.0

[313.521142] sd 0:0:15:0: scsi_dma_map failed: request for 524288 bytes!

【原因说明】

无

【解决方案】

可以忽略

6.5 开启 vmd 并将 nvme 组 raid 之后，系统日志中出现 mdadm 相关错误

【问题描述】

开启 vmd 并将 nvme 组 raid 之后，系统日志中出现如下 mdadm 相关错误：

Process "/sbin/mdadm -l /dev/nvme1n1" failed with exit code 1

【问题原因】

udev 规则与/etc/mdadm.conf 配置冲突造成

【解决方法】

升级 mdadm 到 4.2-14.0.2 及以上版本

7 网络部分

7.1 avahi-daemon: Host name conflict, retrying with linux-2

【问题描述】

系统日志中存在下面的报错：

```
avahi-daemon [11545]: Host name conflict, retrying with linux-2
```

【原因说明】

启动 avahi-daemon 服务后，检测到在局域网或集群网络中，有和当前主机名冲突的名称，需要变更 host name；若不在局域网或集群环境中，关闭 avahi-daemon 服务即可。

【解决方案】

- (1) 使用 `hostnamectl set-hostname "new hostname"` 修改主机名
- (2) 若不在局域网或集群环境中，关闭 avahi-daemon 服务即可

```
systemctl stop avahi-daemon
```

```
systemctl disable avahi-daemon
```

7.2 Activation via systemd failed for unit 'dbus-org.freedesktop.resolve1.service': Unit dbus-org.freedesktop.resolve1.service not found.

【问题描述】

系统日志中存在下面的报错：

```
localhost dbus-daemon[1757]: [system] Activation via systemd failed for unit 'dbus-org.freedesktop.resolve1.service': Unit dbus-
```

org.freedesktop.resolve1.service not found.

【原因说明】

在 KOS 引导过程中，没有启用 systemd-resolved.service 服务时，NetworkManager.Service 尝试开启 dbus-org.freedesktop.resolve1.service 服务会出现这个报错，因 dbus-org.freedesktop.resolve1.service 是 systemd-resolved.service 的软链接。

【解决方案】

两种解决方案：

方案 A：

启用 systemd-resolved.service.

```
# systemctl enable systemd-resolved.service
```

方案 B：不启用 systemd-resolved 服务

编辑 /etc/NetworkManager/NetworkManager.conf 增加下面的参数

```
# vi /etc/NetworkManager/NetworkManager.conf
```

```
[main]
```

```
systemd-resolved=false
```

7.3 搭配博通网卡，系统启动日志中存在 Error recovery 打印

【问题描述】

搭配博通网卡，系统启动日志中存在如下 Error recovery 打印：

```
bnxt_en 0000:10:00.0 eth0: Error recovery info: error recovery[0]
```

【结论】

不是报错，它是 bnxt_en 网卡驱动初始化的时候，打印的正常信息，表示驱动的 recovery watchdog 功能关闭。

7.4 搭配 KOS 5.8 SP1 系统，dmesg 有 i40iw 报错

【问题描述】

KOS5.8SP1 搭配 X722 网卡，系统启动的 dmesg 日志中出现如下错误：

```
[9.796651] ib_srpt MAD registration failed for i40iw0-1.
```

```
[9.796654] ib_srpt srpt_add_one(i40iw0) failed.
```

```
[9.796657] ib_srpt MAD registration failed for i40iw1-1.
```

```
[9.796657] ib_srpt srpt_add_one(i40iw1) failed.
```

【原因说明】

系统中的 ib_srpt 模块注册网络失败造成的，

【解决方案】

方法一：

在不使用此模块的情况下，可以通过注释掉/etc/rdma/modules/rdma.conf 中的 ib_srpt 模块屏蔽此错误。

方法二：

升级 kernel 到 4.19.91-27.4.16.kos5 及以上版本

7.5 KOS 操作系统下通过迈洛斯网卡配置 ib 网络，缺少 opensm 软件包

【问题描述】

KOS 操作系统下通过迈洛斯网卡配置 ib 网络，缺少 opensm 软件包

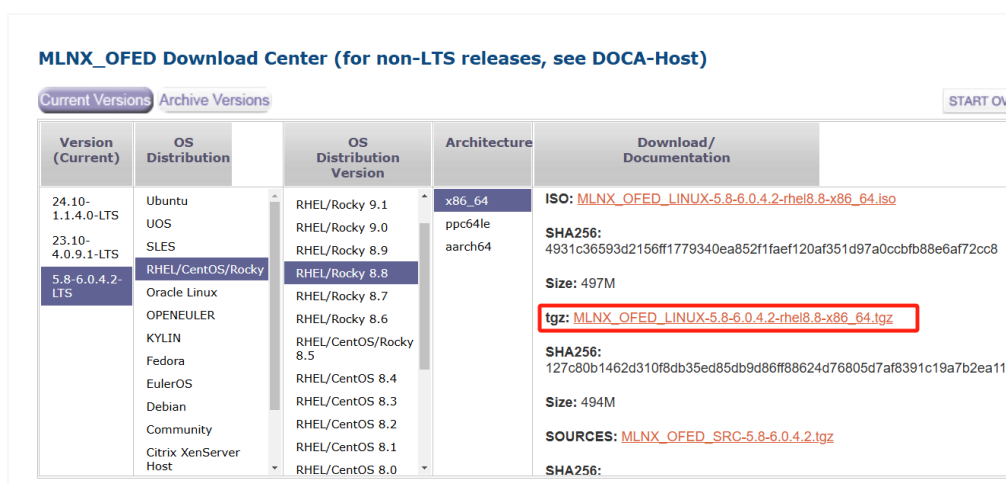
【原因说明】

关于通过迈洛斯网卡配置 ib 网络需要的 opensm 软件包，英伟达官方是不开源

opensm 软件包，因此，我们无法在我们的系统下提前完成编译，带入境像或者放入 yum 源中

【解决方案】

1、根据所需版本去英伟达官网下载驱动 tar 包，下载链接如下：
https://network.nvidia.com/products/infiniband-drivers/linux/mlnx_ofed/



The screenshot shows the 'MLNX_OFED Download Center' for non-LTS releases. It features a table with columns for Version, OS Distribution, OS Distribution Version, Architecture, and Download/Documentation. The 'Current Versions' tab is active. The table lists various OS distributions and their corresponding MLNX_OFED versions. The 'RHEL/CentOS/Rocky' distribution is selected, and the 'x86_64' architecture is chosen. The download link for the 'tgz' file is highlighted in red.

Version (Current)	OS Distribution	OS Distribution Version	Architecture	Download / Documentation
24.10-1.1.4.0-LTS	Ubuntu	RHEL/Rocky 9.1	x86_64	ISO: MLNX_OFED_LINUX-5.8-6.0.4.2-rhel8.8-x86_64.iso
23.10-4.0.9.1-LTS	UOS	RHEL/Rocky 9.0	ppc64le	SHA256: 4931c36593d2156ff1779340ea852f1faef120af351d97a0ccbf88e6af72cc8
5.8-6.0.4.2-LTS	SLES	RHEL/Rocky 8.9	aarch64	Size: 497M
	RHEL/CentOS/Rocky	RHEL/Rocky 8.8		tgz: MLNX_OFED_LINUX-5.8-6.0.4.2-rhel8.8-x86_64.tgz
	Oracle Linux	RHEL/Rocky 8.7		SHA256: 127c80b1462d310f8db35ed85db9d86ff88624d76805d7af8391c19a7b2ea11
	OPENEULER	RHEL/Rocky 8.6		Size: 494M
	KYLIN	RHEL/CentOS/Rocky 8.5		SOURCES: MLNX_OFED_SRC-5.8-6.0.4.2.tgz
	Fedora	RHEL/CentOS 8.4		SHA256:
	EulerOS	RHEL/CentOS 8.3		
	Debian	RHEL/CentOS 8.2		
	Community	RHEL/CentOS 8.1		
	Citrix XenServer Host	RHEL/CentOS 8.0		

- 2、下载完成后解压 tar 包，在 RPMS 路径下获取 4 个 opensm 开头的软件包
- 3、创建一个临时目录，把以上 4 个软件包拷贝到临时目录下
- 4、进入该目录执行 `yum install *`，保证 4 个软件包同时进行安装，否则会出现依赖问题

通过以上步骤能够在 KOS 系统下完成 opensm 软件包的安装。

8 内存部分

8.1 Out of Memory: Killed process [PID] [process name]

【问题描述】

系统日志中存在下面的报错：

Out of Memory: Killed process [PID] [process name]

【原因说明】

系统触发 out of memory 告警，系统内存不足，开始根据进程优先级杀低优先级的进程，建议减少消耗内存的应用。

【解决方案】

1. 检查进程是否收到 cgroup 限制；
2. 检查 num 内存策略；
3. 减少消耗内存的应用

8.2 重启对比/proc/meminfo 的内存容量发现内存大小有细微的变化

【问题描述】

系统下多次重启对比/proc/meminfo 的内存容量发现，内存大小有细微的变化，

如 791740656 kB，可能变化为 791740652 kB、791740660 kB

【原因说明】

系统下多次重启看到很少的 4K，8k 的内存量的变化，可以忽略，是正常的。

【解决方案】

忽略

8.3 系统 grub 中增加的 memmap 参数中不能保留\$符的问题

【问题描述】

grub 参数 memmap 用于指定系统预留内存，使用 memmap=nn[KMG]\$ss[KMG]强制保留(不使用)从 ss 开始的 nn 长度的特定内存区域。

1、编辑 /etc/default/grub 并增加 memmap=0x6ffe\$7e1b0000 参数后，使用 grub2-mkconfig 重新生成 grub.cfg 文件后，查看参数内容，看到 \$ 符号并没有体现：

```
[root@localhost ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap=0x6ffe$7e1b0000"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
[root@localhost ~]# grub2-mkconfig -o /boot/efi/EFI/kos/grub.cfg
Generating grub configuration file ...
Adding boot menu entry for EFI firmware configuration
done
[root@localhost ~]# cat /boot/efi/EFI/kos/grub.cfg | grep memmap
set kernelopts="root=/dev/mapper/keyarchos-root ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap=0x6ffee1b0000"
```

2、增加 \\$ 转义符后，memmap=0x6ffe\\$7e1b0000 重新生成的 grub.cfg 文件中，可以正常显示：

```
[root@localhost ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap=0x6ffe\$7e1b0000"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
[root@localhost ~]#
[root@localhost ~]# grub2-mkconfig -o /boot/efi/EFI/kos/grub.cfg
Generating grub configuration file ...
Adding boot menu entry for EFI firmware configuration
done
[root@localhost ~]# cat /boot/efi/EFI/kos/grub.cfg | grep memmap
set kernelopts="root=/dev/mapper/keyarchos-root ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap=0x6ffe$7e1b0000"
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# reboot
```

重启后，启动的 grub 界面中有 \$ 符号，但 /proc/cmdline 参数中，并没有体现 \$ 符号，且 iomem 中也没有体现内存预留的情况，参数未生效：

```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-4.19.91-27.4.7.kos5.x86_64 root=/dev/mapper/keyarchos-root ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap=0x6ffe$7e1b0000 crashkernel=512M cgroup.memory=nokmem
initrd ($root)/initramfs-4.19.91-27.4.7.kos5.x86_64.img $tuned_initrd

[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=(hd0,gpt2)/vmlinuz-4.19.91-27.4.7.kos5.x86_64 root=/dev/mapper/keyarchos-root ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap=0x6ffee1b0000 crashkernel=512M cgroup.memory=nokmem
```



```
[root@localhost ~]# cat /proc/iomem
00000000-00000fff : Reserved
00001000-0009ffff : System RAM
000a0000-000bffff : PCI Bus 0000:00
000c0000-000c7fff : Video ROM
000f0000-000fffff : System ROM
00100000-7e1b6fff : System RAM
16000000-16c031d0 : Kernel code
16c031d1-1768d1ff : Kernel data
17ca7000-1a9fffff : Kernel bss
3a000000-59ffffff : Crash kernel
7e1b7000-7e1fffff : Reserved
7e200000-7eceefff : System RAM
7ecf000-7ef6efff : Reserved
```

3、将 memmap 参数中的值 memmap='0x6ffe\$7e1b0000' 使用单引号引起来，生成的 grub.cfg 文件及启动后的 cmdline 显示均正常，但 iomem 中内存并未预留，参数未生效：

```
[root@localhost ~]# vim /etc/default/grub
[root@localhost ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap='0x6ffe$7e1b0000'"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
[root@localhost ~]# grub2-mkconfig -o /boot/efi/EFI/kos/grub.cfg
Generating grub configuration file ...
Adding boot menu entry for EFI firmware configuration
done
[root@localhost ~]# cat /boot/efi/EFI/kos/grub.cfg | grep memmap
set kernelopts="root=/dev/mapper/keyarchos-root ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap='0x6ffe$7e1b0000'"
[root@localhost ~]# reboot
```

```
load_video
set gfx_payload=keep
insmod gzio
linux ($root)/vmlinuz-4.19.91-27.4.7.kos.x86_64 root=/dev/mapper/keyarchos-root ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap='0x6ffe$7e1b0000' crashkernel=512M cgroup.memory=nokmem
initrd ($root)/initramfs-4.19.91-27.4.7.kos.x86_64.img $tuned_initrd
```

```
[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=(hd0,gpt2)/vmlinuz-4.19.91-27.4.7.kos.x86_64 root=/dev/mapper/keyarchos-root ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap=0x6ffe$7e1b0000 crashkernel=512M cgroup.memory=nokmem
[root@localhost ~]# cat /proc/iomem
00000000-00000fff : Reserved
00001000-0009ffff : System RAM
000a0000-000bffff : PCI Bus 0000:00
000c0000-000c7fff : Video ROM
000f0000-000fffff : System ROM
00100000-7e1b6fff : System RAM
31000000-31c031d0 : Kernel code
31c031d1-3268d1ff : Kernel data
32ca7000-359fffff : Kernel bss
3a000000-59ffffff : Crash kernel
7e1b7000-7e1fffff : Reserved
7e200000-7eceefff : System RAM
7ecf000-7ef6efff : Reserved
```

4、用 grubby 的方式修改 grub.cfg:

```
[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=(hd0,gpt2)/vmlinuz-4.19.91-27.4.7.kos5.x86_64 root=/dev/mapper/keyarchos-root ro crashkernel=auto
resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap crashkernel=512M cg
roup.memory=nokmem
[root@localhost ~]# grubby --info=ALL | grep kernel
kernel="/boot/vmlinuz-4.19.91-27.4.7.kos5.x86_64"
args="ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/s
wap $tuned_params crashkernel=512M cgroup.memory=nokmem"
kernel="/boot/vmlinuz-0-rescue-0baa92132dfa47189524f2ece071a827"
args="ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/s
wap"
[root@localhost ~]# grubby --update-kernel=ALL --args=memmap='0x6ffe$7e1b0000'
[root@localhost ~]# grubby --info=ALL | grep kernel
kernel="/boot/vmlinuz-4.19.91-27.4.7.kos5.x86_64"
args="ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/s
wap memmap=0x6ffe$7e1b0000 $tuned_params crashkernel=512M cgroup.memory=nokmem"
kernel="/boot/vmlinuz-0-rescue-0baa92132dfa47189524f2ece071a827"
args="ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/s
wap memmap=0x6ffe$7e1b0000"
[root@localhost ~]#
[root@localhost ~]# reboot

Remote side unexpectedly closed network connection

Session stopped
- Press <Return> to exit tab
- Press R to restart session
- Press S to save terminal output to file
🖥️ root@10.126.80.175's password:

? MobaXterm Personal Edition v23.5 ?
(SSH client, X server and network tools)

▶ SSH session to root@10.126.80.175
? Direct SSH : ✓
? SSH compression : ✓
? SSH-browser : ✓
? X11-forwarding : ✓ (remote display is forwarded through SSH)

▶ For more info, ctrl+click on help or visit our website.

Last login: Tue Mar 5 16:52:03 2024 from 10.126.80.202
Prompt:
Your system is nonactivated.
[root@localhost ~]# cat /proc/cmdline
BOOT_IMAGE=(hd0,gpt2)/vmlinuz-4.19.91-27.4.7.kos5.x86_64 root=/dev/mapper/keyarchos-root ro crashkernel=au
to resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/swap memmap=0x6ffe$7e1b0
000 crashkernel=512M cgroup.memory=nokmem
[root@localhost ~]# grubby --info=ALL | grep kernel
kernel="/boot/vmlinuz-4.19.91-27.4.7.kos5.x86_64"
args="ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/s
wap memmap=0x6ffe$7e1b0000 $tuned_params crashkernel=512M cgroup.memory=nokmem"
kernel="/boot/vmlinuz-0-rescue-0baa92132dfa47189524f2ece071a827"
args="ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm.lv=keyarchos/s
wap memmap=0x6ffe$7e1b0000"
[root@localhost ~]#
```

【原因说明】

当/boot/loader/entries 下的内核引导文件中如果存在\$kernelopts 读取 kernelopts 变量则\$符正常显示，如果没有则不能正常显示

此问题出现在，5.8、5.8SP1 版本，且安装系统时安装的 4.18 内核，进系统后再手动安装 4.19 内核时，会出现此问题，主要原因是 4.19 内核安装时调用了 grubby 命令，替换了 /boot/loader/entries 的\$kernelopts 字段，导致该问题出现

【解决方案】

如果要使用 4.19 内核，建议系统安装时就默认安装。

9 显示部分

9.1 journal: failed to get edid data: EDID length is too small

【问题描述】

系统日志中存在下面的报错：

```
journal: failed to get edid data: EDID length is too small
```

【原因说明】

连接的显示器返回的 EDID 数据块大小（EDID 数据块的长度应至少 128 bytes）导致的，建议更换显示器。

【解决方案】

无

9.2 failed to get edid: unable to get EDID for output

【问题描述】

系统日志中存在下面的报错：

```
failed to get edid: unable to get EDID for output
```

【原因说明】

EDID(扩展显示识别数据)是通过参考伪文件/sys/class/drm/cardX-*/edid 得到的，如下所示：

```
$ cat /sys/class/drm/cardX-*/edid
```

如果此文件的引用出现问题，则会发生该错误。

虚拟机(如 KVM)可能无法获得 EDID，因为虚拟机不支持显示信息。

【解决方案】

可尝试更换显示设备，若是虚拟机，则无需关注。

9.3 Failed to grab accelerator for keybinding settings

【问题描述】

系统日志中存在下面的报错：

Jun 9 14:11:18 localhost journal[6188]: Failed to grab accelerator for keybinding settings:playback-random

Jun 9 14:11:18 localhost journal[6188]: Failed to grab accelerator for keybinding settings:hibernate

Jun 9 14:11:18 localhost journal[6188]: Failed to grab accelerator for keybinding settings:rotate-video-lock

Jun 9 14:11:18 localhost journal[6188]: Failed to grab accelerator for keybinding settings:playback-repeat

【原因说明】

这些日志是 informational 级别日志，并不属于报错信息，可以安全的忽略它们。

可以使用如下命令查看系统中的 warning 或 error 级别的日志：

```
# journalctl -b -p warning
```

【解决方案】

无

9.4 Error when getting information for file “/root/.cache/gnome-software/shell-extensions/gnome.json” : No such file or directory

【问题描述】

系统日志中存在下面的报错：

```
localhost journal[8151]: not GsPlugin error g-io-error-quark:1: Error when  
getting information for file "/root/.cache/gnome-software/shell-  
extensions/gnome.json" : No such file or directory
```

【原因说明】

当 Shell 扩展元数据不可用时，会打印此消息，可以安全忽略

【解决方案】

无

9.5 Gdm: GdmDisplay: Session never registered, failing

【问题描述】

系统日志中存在如下报错：

```
gdm[86246]: Gdm: GdmDisplay: Session never registered, failing
```

【原因说明】

系统 gdm 的 bug

【解决方案】

升级 gdm 到 40.0-27 及以上版本

9.6 在机器重启之前和重启之后，进程不间断运行的原因分析

【问题描述】

KOS 系统跑 ltp 长稳测试，通过 `ps -eo pid,etime,comm |grep PID` 命令查看进程的运行时间为 48 小时，但是通过 `last reboot` 命令查看到机器最后一次重启是 30 小时之前

```
[root@localhost ~]# cd ltp-install/
[root@localhost ltp-install]# ls
bin  conformance  functional  IDcheck.sh  metadata  output  runltp  runtest  scenario_groups  sh
[root@localhost ltp-install]# vim Version
[root@localhost ltp-install]# vim runltp
[root@localhost ltp-install]# ps -eo pid,etime,comm | grep 127587
127587 1-21:20:41 runltp
[root@localhost ltp-install]# ps -p 127587 -o lstart
STARTED
Sat Mar 23 13:07:06 2024
[root@localhost ltp-install]# ps -p 127591 -o lstart
STARTED
Sat Mar 23 13:07:06 2024
[root@localhost ltp-install]# ps -p 127591 -o etime
ELAPSED
1-21:25:40
[root@localhost ltp-install]# last reboot
reboot    system boot  5.10.134-15.2.15 Sun Mar 24 01:02    still running

wtm begins Sun Mar 24 01:02:50 2024
[root@localhost ltp-install]#
```

【原因说明】

1、last reboot 的命令是从 /var/log/wtmp 文件中读取系统 reboot 的时间，而 /var/log/wtmp 文件中记录的时间是基于系统重启时的当前时间生成保存

2、举例说明：

系统最后一次 reboot 时间为 2024-03-01，reboot 时记录到 /var/log/wtmp 的文件中；系统正常启动之后，为了编译 ltp 跑长稳测试，把系统时间修改为 2024-01-01，之后编译通过开始跑长稳测试，此时 ltp 长稳这个进程的时间是从 2024-01-01 开始计时的，因此，就会出现出了问题描述中所出现的问题。

【解决方案】

此问题不需要解决，只是修改时间导致的问题，可以通过查 message 文件查看是不是系统时间的跳变，如果有跳变，则该问题不需要解决，属于正常现象，如果没有时间跳变，则需要排查其他原因

10 其他部分

10.1 BMC 控制台大小写键盘混乱

【问题描述】

在某些服务器上通过键盘向 bmc 控制台输入内容时，出现大小写混乱问题。

【原因说明】

KVM 里的正确输入是要在打开 KVM 的窗口中输入，里边有组合键也有小键盘。

不能使用打开 KVM 的 PC 机键盘输入热键、组合键等，原因：PC 的 os 会拦截键盘输入的热键、组合键等信息，导致实际 KVM 收到的输入信息失真，于是表现为乱码。

【解决方案】

无

10.2 每次重启 messages 中的启动时间会快 8 小时，直到 Journal 启动后恢复正常时间

【问题描述】

每次重启 messages 中的启动时间会快 8 小时，直到 Journal 重启后恢复正常时，比如：

正常时间应是 Dec 20 15:38:14，但当系统重启时，时间会增加 8 小时，变化如下，initramfs 加载完成，到启动 kernel 重启后，时间由 Dec 20 23:38:12 恢复为 Dec 20 15:38:12。

```
Dec 20 23:38:10 localhost journal: Runtime journal is using 8.0M (max allowed 4.0G, trying to leave 4.0G free of 251.8G available → current limit 4.0G).
```

```
Dec 20 23:38:10 localhost kernel: Initializing cgroup subsys cpuset
```

```
Dec 20 23:38:10 localhost kernel: Initializing cgroup subsys cpu
```

```
Dec 20 23:38:10 localhost kernel: Initializing cgroup subsys cpuacct
```

```
Dec 20 23:38:10 localhost kernel: Linux version 4.18.0-372.26.1.kos5.x86_64 (mockbuild@fa33ebc3beb6429381631eb12e3992e8) (gcc version 8.5.0 20210514 (kos 8.5.0-10.0.1) (GCC) ) #1 SMP Thu Oct 29 17:29:29 EDT 2022
```

```
.....
```

```
Dec 20 23:38:12 localhost systemd: Starting Cleanup udevd DB...
```

```
Dec 20 23:38:12 localhost systemd: Started Cleanup udevd DB.
```

```
Dec 20 23:38:12 localhost systemd: Reached target Switch Root.
```

```
Dec 20 23:38:12 localhost systemd: Starting Switch Root.  
Dec 20 23:38:12 localhost systemd: Starting Switch Root...  
Dec 20 23:38:12 localhost systemd: Switching root.  
Dec 20 23:38:12 localhost journal: Journal stopped  
Dec 20 15:38:12 localhost journal: Runtime journal is using 8.0M (max allowed  
4.0G, trying to leave 4.0G free of 251.8G available → current limit 4.0G).  
Dec 20 15:38:12 localhost journal: Runtime journal is using 8.0M (max allowed  
4.0G, trying to leave 4.0G free of 251.8G available → current limit 4.0G).  
Dec 20 15:38:12 localhost systemd-journald[321]: Received SIGTERM from  
PID 1 (systemd).  
Dec 20 15:38:12 localhost kernel: SELinux: Disabled at runtime.  
Dec 20 15:38:12 localhost kernel: type=1404 audit(1482248292.393:2):  
selinux=0 auid=4294967295 ses=4294967295  
Dec 20 15:38:12 localhost systemd[1]: RTC configured in localtime, applying  
delta of 480 minutes to system time.  
Dec 20 15:38:12 localhost kernel: ip_tables: (C) 2000-2006 Netfilter Core Team  
Dec 20 15:38:12 localhost systemd[1]: Inserted module 'ip_tables'  
Dec 20 15:38:12 localhost journal: Journal started
```

【原因说明】

由于 RTC 时间(BIOS 时间)默认认为和 UTC 时间一致，系统重启日志前半段 initramfs 加载时，由于时区是+8 区，会增加 480 分钟，即 8 小时；加载到 kernel 时，系统检测到 RTC configured in localtime 的设置，Local time 应和 RTC time 一致，将时间调整-8 小时；系统运行中，系统时间和 RTC 时间一致，若再次重启，initramfs 启动日志中，仍然会认为 BIOS 时间是 UTC 时间，会按时区+8，加载到 kernel 时，和 RTC 保持一致，再-8 小时，如此反复循环。

【解决方案】

可以使用如下命令将硬件时钟设置为 UTC 时间

```
# timedatectl set-local-rtc 0
```


10.3 如何在 KOS 系统中读取服务器 SN 信息

【问题描述】

如下在系统下读取服务器 SN 信息

【原因说明】

无

【解决方案】

KOS 系统下可以通过 `dmidecode -t system` 或者 `ipmitool fru` 命令来查看 SN 信息，参考如下：

```
[root@localhost ~]# dmidecode -t system
# dmidecode 3.3
Getting SMBIOS data from sysfs.
SMBIOS 3.2.1 present.

Handle 0x0001, DMI type 1, 27 bytes
System Information
  Manufacturer: Inspur
  Product Name: NF5280M5
  Version: 00001
  Serial Number: 219619189
  UUID: fdb5f926c-0c04-03de-b211-d21d601f451a
  Wake-up Type: Power Switch
  SKU Number: Default string
  Family: Default string

[root@localhost ~]# ipmitool fru
FRU Device Description : Builtin FRU Device (ID 0)
Chassis Type          : Rack Mount Chassis
Chassis Part Number   : 0
Chassis Serial        : 219619188
Board Mfg Date        : Sun Aug 26 07:51:00 2018
Board Mfg             : Inspur
Board Product         : NF5280M5
Board Serial          : MBK919S21229B20
Board Part Number     : YZMB-00882-104
Product Manufacturer  : Inspur
Product Name          : NF5280M5
Product Part Number   : 219619190
Product Version       : 00001
Product Serial        : 219619189
Product Asset Tag     : NULL
```

10.4 如何使用系统镜像配置本地 yum 源

【问题描述】

如何使用系统镜像配置本地 yum 源

【原因说明】

无

【解决方案】

KOS 5.8 系统镜像中将 rpm 分成两类：一类是基础 rpm 安装包，在 BaseOS 文件夹中；一类是与应用相关的 rpm，在 AppStream 文件夹中：



故在做 yum 源时，需要做两个 yum 源，一个 BaseOS，一个 AppStream。比如将 iso 镜像文件挂载到了 /mnt 目录，创建 /etc/yum.repos.d/local.repo 配置文件，内容如下：

```
#Repository 标签，每个文件中的不能发生冲突，唯一标识！
[baseos]
#Repository 描述信息
name=BaseOS
#包路径
baseurl=file:///mnt/BaseOS
#是否启用 1启用 0禁用
enabled=1
#是否进行gpg签名校验 1校验 0不校验
gpgcheck=0

[appstream]
name=AppStream
baseurl=file:///mnt/AppStream
enabled=1
gpgcheck=0
```

10.5 1128 版本镜像升级 dnf 后需要激活

【问题描述】

使用 KOS 1128 版本，执行 yum makecache 时，提示如下错误，无法使用 KOS 网络源。

```
[root@localhost ~]# yum makecache
The system is nonactivated.
There are no enabled repositories in "/etc/yum.repos.d", "/etc/yum/repos.d", "/etc/distro.repos.d".
[root@localhost ~]#
```

【原因说明】

因为 dnf 相关软件包增加了 license 激活功能，低版本的 dnf 包升级后需要激活。

【解决方案】

无

10.6 系统启动时 NetworkManager-wait-online 服务启动失败

【问题描述】

NetworkManager-wait-online 服务启动失败，系统日志提示如下错误：

```
Jun  3 11:33:49 localhost systemd[1]: NetworkManager-wait-online.service: Main process exited, code=exited, status=1/FAILURE
```

```
Jun  3 11:33:49 localhost systemd[1]: NetworkManager-wait-online.service: Failed with result 'exit-code'.
```

```
Jun  3 11:33:49 localhost systemd[1]: Failed to start Network Manager Wait Online.
```

【原因说明】

无

【解决方案】

升级 NetworkManager-1.36.0-4 or newer 或者执行 nmcli connection modify <Connection Name> ipv6.method "disabled" 命令禁用掉 ipv6。

10.7 KeyarchOS 5.8 系统如何启用并设置 sar 日志每秒搜集一次

【问题描述】

KeyarchOS 5.8 系统如何启用并设置 sar 日志每秒搜集一次

【原因说明】

无

【解决方案】

1.执行 `yum install sysstat -y` 命令, 安装 sysstat 包。

2.编辑 `sysstat-collect.timer` 修改每分钟搜集一次

```
#vim /usr/lib/systemd/system/sysstat-collect.timer
```

```
[Unit]
```

```
Description=Run system activity accounting tool every 10 minutes
```

```
[Timer]
```

```
#OnCalendar=*:00/10
```

```
OnCalendar=*:00/1
```

```
[Install]
```

```
WantedBy=sysstat.service
```

3.编辑 `sysstat-collect.service` 修改每次搜集时间间隔内, 搜集的次数。此处修改成 60

```
[Unit]
```

```
Description=system activity accounting tool
```

```
Documentation=man:sa1(8)
```

```
After=sysstat.service
```

```
[Service]
```

```
Type=oneshot
```

```
User=root
```

```
#ExecStart=/usr/lib64/sa/sa1 1 1
```

```
ExecStart=/usr/lib64/sa/sa1 1 60
```

4.执行如下命令重启 sysstat 服务

```
#systemctl daemon-reload
```

```
#systemctl restart sysstat-collect.timer
```

```
#systemctl restart sysstat-collect.service
```

注：为避免长期每秒一次的数据采集影响过大，测试数据收集完成后请将配置还原。

10.8 KeyarchOS 5.8 使用 localectl 命令修改系统 encoding，由 UTF-8 改成 gbk，系统重启后，图形界面无法开启终端

【问题描述】

按照如下方案切换 encoding，由 UTF-8 改成 gbk，系统重启后，图形界面无法开启终端：

1.执行如下命令设置系统语言与 encoding

```
localectl set-local zh_CN.gbk
```

2.执行 reboot 命令重启系统



4.查看系统日志，发现日志中有如下打印：

```
Aug 28 22:53:54 localhost.localdomain systemd[2135]: Starting GNOME  
Terminal Server...
```

```
Aug 28 22:53:55 localhost.localdomain gnome-terminal-server[2697]: Non  
UTF-8 locale (GBK) is not supported!
```

```
Aug 28 22:53:55 localhost.localdomain systemd[2135]: gnome-terminal-  
server.service: Main process exited, code=exited, status=8/n/a
```

```
Aug 28 22:53:55 localhost.localdomain systemd[2135]: gnome-terminal-  
server.service: Failed with result 'exit-code'.
```

```
Aug 28 22:53:55 localhost.localdomain systemd[2135]: Failed to start  
GNOME Terminal Server.
```

【原因说明】

系统 gnome-terminal 已经不再支持非 UTF-8 locale.

【解决方案】

无

10.9 系统下执行 yum install -y 命令安装软件包时提示“GPG 公钥已安装，但是不适用于此软件包”

【问题描述】

系统下执行 yum install -y 命令安装软件包时提示“GPG 公钥已安装，但是不适用于此软件包”报错，造成系统无法安装

```
[root@localhost ~]# yum install fio
上次元数据过期检查: 0:11:04 前, 执行于 2021年01月07日 星期四 10时05分23秒。
依赖关系解决。
=====
软件包                                架构          版本                仓库                大小
安装:
fio                                    x86_64         3.22-1.0.1.kos5     AppStream            551 k
升级:
libpmem                               x86_64         1.11.1-1.0.1.module_kos5+202+0d804943 AppStream            111 k
安装依赖关系:
libpmemblk                             x86_64         1.11.1-1.0.1.module_kos5+202+0d804943 AppStream            100 k
应用模式块:
pmdk                                    1_fileformat_v6
事务概要
-----
安装 2 软件包
升级 1 软件包
总计下载: 763 k
确定吗? [y/N]: y
下载软件包:
(1/3): libpmemblk-1.11.1-1.0.1.module_kos5+202+0d804943.x86_64.rpm 10 MB/s | 100 kB 00:00
(2/3): libpmem-1.11.1-1.0.1.module_kos5+202+0d804943.x86_64.rpm 11 MB/s | 111 kB 00:00
(3/3): fio-3.22-1.0.1.kos5.x86_64.rpm 20 MB/s | 551 kB 00:00
-----
总计: 27 MB/s | 763 kB 00:00
2.1 MB/s | 2.1 kB 00:00
KOS-5.Bsp1 - AppStream
仓库 "KOS-5.Bsp1 - AppStream" 的 GPG 公钥已安装, 但是不适用于此软件包。
请检查此仓库的公钥 URL 是否正确。失败的软件包是: fio-3.22-1.0.1.kos5.x86_64
GPG密钥配置为: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kos
libpmem-1.11.1-1.0.1.module_kos5+202+0d804943.x86_64.rpm 的公钥没有安装。失败的软件包是: libpmem-1.11.1-1.0.1.module_kos5+202+0d804943.x86_64
GPG密钥配置为: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kos
libpmem-1.11.1-1.0.1.module_kos5+202+0d804943.x86_64.rpm 的公钥没有安装。失败的软件包是: libpmem-1.11.1-1.0.1.module_kos5+202+0d804943.x86_64
GPG密钥配置为: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-kos
下载的包保存在缓存中, 直到下次成功执行事务。
您可以通过执行 'yum clean packages' 删除软件包缓存。
错误: GPG 检查失败
```

【原因说明】

系统时间早于 gpgkey 生成的时间，造成的

```
[root@localhost ~]# gpg --import kos.gpg
gpg: 密钥 48590E37A2D34605 生成于未来的 953 天后 (可能是由于时空扭曲或系统时钟的问题)
gpg: 密钥 48590E37A2D34605 生成于未来的 953 天后 (可能是由于时空扭曲或系统时钟的问题)
gpg: 密钥 48590E37A2D34605 生成于未来的 953 天后 (可能是由于时空扭曲或系统时钟的问题)
gpg: 密钥 48590E37A2D34605 生成于未来的 953 天后 (可能是由于时空扭曲或系统时钟的问题)
gpg: key 48590E37A2D34605: new key but contains no user ID - skipped
gpg: 处理的总数: 1
gpg: 缺少用户标识: 1
```

【解决方案】

修改系统时间为当前时间

10.10 系统 grubenv 文件中的 32 位字符串是如何生成的

【问题描述】

系统/boot/efi/EFI/kos/grubenv 文件中，存在的 32 位字符串是如何生成的？

```
[root@localhost kos]# ll
total 4860
-rwx----- 1 root root    98 Oct 13  2022 BOOTX64.CSV
drwx----- 2 root root 4096 Jun 16  07:53 fonts
-rwx----- 1 root root 6641 Nov  7  14:05 grub.cfg
-rwx----- 1 root root 1024 Nov  7  14:05 grubenv
-rwx----- 1 root root 2215424 Jun 16  07:53 grubx64.efi
-rwx----- 1 root root 852784 Oct 13  2022 mmx64.efi
-rwx----- 1 root root 943520 Oct 13  2022 shimx64.efi
-rwx----- 1 root root 936752 Oct 13  2022 shimx64-kos.efi
[root@localhost kos]# cat grubenv
# GRUB Environment Block
saved_entry=d5fde334b70c4a47ac94c8a716ffe642-4.18.0-477.13.1.1.kos5.x86_64
kernelopts=root=/dev/mapper/keyarchos-root ro crashkernel=auto resume=/dev/mapper/keyarchos-swap rd.lvm.lv=keyarchos/root rd.lvm
.lv=keyarchos/swap rhgb quiet
boot_success=0
```

【原因说明】

此字符串为 machine-id，保存在 /etc/machine-id 文件中。此 ID 是在系统安装过程中随机生成并以换行符结尾的十六进制、32 个字符、小写的唯一机器 ID。

【解决方案】

无

10.11 升级内核后，修改 grub 文件增加内核参数重启，参数在新内核上不生效

【问题描述】

- 1、安装新的 kernel。
- 2、编辑 /etc/default/grub 文件添加 cpuparkmem=0x200000000 quickkexec=128M 参数。
- 3、执行 grub2-mkconfig -o /boot/efi/EFI/kos/grub.cfg 更新 grub 配置。

【原因说明】

多内核设计如此

【解决方案】

如果需要使多个内核同时生效，可通过如下方法

方法一：

首先使用 grub2-mkconfig 修改 grub 配置，然后重启系统，再升级内核，这样升级的内核会有新添加的参数

方法二：

使用如下 grubby 命令进行添加：

```
grubby --update-kernel=DEFAULT --args="cpuparkmem=0x200000000  
quikkexec=128M"
```

10.12 配置 VNC 服务端，端口设置从 2 开始

【问题描述】

- 1、通过 VNC 远程连接 KeyarchOS（桌面环境）
- 2、安装 VNC SERVER
- 3、编辑 vncserver.users，如果您远程登录 2 个用户的远程桌面，替换上述用户名为需要配置连接的用户名，如果您需要配置远程登录 1 个用户的远程桌面，只需要在 vncserver.users 中配置 1 个用户。

【问题原因】

- 1、默认情况下，VNC 服务器在 tcp 端口 5900+n 上监听，其中 n 是显示端口号，如果显示端口号为 “2”，那么 VNC 服务器将在 TCP 端口 5902 上监听其请求，2 对应端口号 5902，3 对应端口号 5903。
- 2、端口号从 1 开始会有问题，链接不上，**建议端口号从 2 开始。**

10.13 Auditd 服务使用 systemctl 概率性无法启动，显示 Operation refused

【问题描述】

使用 systemd 对 audit 服务进行 restart 或者 stop 时时，如下报错：

```
[root@localhost ~]# systemctl restart auditd
Failed to restart auditd.service: Operation refused, unit auditd.service may be requested by dependency only (it is configured to refuse manual start/stop).
See system logs and 'systemctl status auditd.service' for details.
```

【问题原因】

服务 auditd.service 文件中存在字段 RefuseManualStop=yes, 限制了 auditd 服务使用 systemd stop 停止服务。又因为 restart 的执行顺序是先 stop 然后 start, 因此 restart 也会失败

【解决方法】

使用 # service auditd status/start/stop/restart 不能使用 systemctl stop 或者 restart auditd 服务, 可以使用 systemctl start 启动服务比如: 先用 service auditd stop 关闭服务, 再用 systemctl start 启动服务

注: service 命令只对 auditd 服务生效

10.14 执行 ipmitool mc watchdog get 时显示 stopped 状态

【问题描述】

在执行 ipmitool mc watchdog get 时显示 stopped 状态

```
[root@localhost dong]# modprobe softdog
[root@localhost dong]# systemctl daemon-reload
[root@localhost dong]# systemctl restart watchdog.service
[root@localhost dong]# ipmitool mc watchdog get
Watchdog Timer Use:      BIOS FRB2 (0x01)
Watchdog Timer Is:       Stopped
Watchdog Timer Actions:  No action (0x00)
Pre-timeout interval:   0 seconds
Timer Expiration Flags: 0x00
Initial Countdown:      0 sec
Present Countdown:      0 sec
[root@localhost dong]# ls /dev/watchdog
/dev/watchdog
[root@localhost dong]#
```

【问题原因】

没有正确安装对应的软件包, ipmitool mc watchdog get 该命令是获取的 BMC watchdog 状态, 因此需要安装 bmc watchdog 服务

【解决方法】

安装 freeipmi 和 freeipmi-bmc-watchdog 两个 rpm 包, 然后使用 systemctl 启动 bmc-watchdog 服务后即可

```
[root@localhost ~]# systemctl status bmc-watchdog.service
● bmc-watchdog.service - BMC Watchdog Timer Daemon
   Loaded: loaded (/usr/lib/systemd/system/bmc-watchdog.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
[root@localhost ~]# rpm -qa | grep freeipmi
freeipmi-1.6.9-1.0.2.kos5.x86_64
freeipmi-bmc-watchdog-1.6.9-1.0.2.kos5.x86_64
[root@localhost ~]# systemctl start bmc-watchdog.service
[root@localhost ~]# ipmitool mc watchdog get
Watchdog Timer Use:      SMS/OS (0x44)
Watchdog Timer Is:      Started/Running
Watchdog Timer Actions: Hard Reset (0x01)
Pre-timeout interval:   0 seconds
Timer Expiration Flags: 0x00
Initial Countdown:      900 sec
Present Countdown:      897 sec
```