

平台服务器巡检工具

使用指导手册



版权声明与使用须知

版权声明

©2023 浙江宇视科技有限公司。保留一切权利。

未经浙江宇视科技有限公司（下称“本公司”）书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播。

本手册描述的产品中可能包含本公司及可能存在的许可人享有版权的软件。未经相关权利人许可，任何人不得以任何形式对前述软件进行包括但不限于复制、分发、修改、摘录、反编译、反汇编、解密、反向工程、出租、转让、分许可等侵犯软件著作权的行为。

使用须知

由于产品版本升级等原因，本手册内容会不定期更新。

本手册仅作为使用指导，其内所有陈述、信息和建议等均不构成任何明示或暗示的担保。本手册中的图形、图表或照片等仅用于说明示例，可能与实际产品存在差异，请以实物为准。

目录

一、通用巡检脚本的使用.....	6
1. 使用环境.....	6
2. 执行巡检.....	6
3. 定位巡检项.....	11
4. 巡检日志.....	11
5. 巡检日志分析工具.....	11
二、巡检项说明.....	13
1. 操作系统检查.....	13
2. 系统进程检查.....	21
3. 磁盘空间检查.....	24
4. 网络状态检查.....	28
5. PG 数据库检查	33
6. HA & N+1 检查	43
7. 业务状态检查 (VM)	53
8. 系统日志收集.....	57
9. 服务器补丁及工具部署检查.....	60

版本修订记录

日期	修订 版本	修订章节	修改描述	修订人
2019-04-10	V1.1	§ 2.4 § 2.9	1. 添加 ARP 表参数巡检项； 2. 添加通用看门狗补丁巡检项； 3. 添加 Coredump 工具部署巡检项； 4. 添加 Tcpdump 工具部署巡检项	赵辰
2019-04-16	V1.2	§ 2.8	1. 添加对 messages 日志的部分关键字信息过滤	赵辰
2019-04-28	V1.3	§ 2.7	1. 添加数据库备份的定时任务巡检项	赵辰
2019-05-14	V1.4	§ 1.3	1. 优化日志显示，提升易用性； 2. 结合技服意见，优化巡检项	赵辰
2019-07-18	V1.5	全部	添加检查项，修订说明	赵辰
2019-09-03	V1.6	全部	1. 支持添加定时任务巡检； 2. core 文件调用栈分析； 3. lsof 命令检查； 4. 部分严重问题告警日志	赵辰
2019-10-29	V1.7	§ 2.5	增加目录权限检查； 增加数据库状态检查	赵辰
2020-02-19	V1.8	全部	示例更新，细化说明	赵辰
2020-07-10	V2.0	全部	增加巡检项；	赵辰

			优化巡检结果	
2020-08-24	V2.1	§ 1	输出 csv 格式巡检日志	赵辰
2020-11-10	V2.2	全部	支持指定巡检内容巡检; 增加 root 密码有效期巡检; 增加硬件信息巡检; 增加近期关键问题巡检 (operlog_id)	赵辰
2021-01-18	V2.4	§ 1	修复错误 过期检测, 此工具 100 天后自动过期	赵辰
2021-04-27	V2.7	§ 1	优化使用 过期检测,自发布日起 180 天内有效期	赵辰
2021-10-21	V2.8	§ 1 § 2.1	优化使用 Swap 使用量检查	赵辰
2022-04-27	V3.0	§ 1 § 2.1	添加巡检项, 增加自动巡检项 新增日志分析工具, 可将巡检出的错误单独过滤出来, 便于分析巡检日志	赵辰
2023-10-26	V3.4		增加收集 license 功能	袁庆杰

一、通用巡检脚本的使用

1. 使用环境

此巡检脚本可用于巡检安装有 HA、N+1、VM(VM+DB)交付件，且为 B3323 及以上版本的环境。

2. 执行巡检

2.1 工具部署

将 GeneralCheck.tar.gz 拷贝至任意目录（如/home）下，然后解压

```
tar xzf GeneralCheck.tar.gz
```

2.2 查看工具版本

```
sh GeneralCheck.sh -v
```

```
[root@ha_92 tmp]#  
[root@ha_92 tmp]# sh GeneralCheck.sh -v  
Version : v3.0 release in 2022-04-27  
[root@ha_92 tmp]#
```

注：巡检脚本会定期（6 个月）更新一次，请局点使用最新版本。

2.3 工具使用说明

```
sh GeneralCheck.sh -h
```

```
[root@ha_92 tmp]# sh GeneralCheck.sh -h
Usage :
sh GeneralCheck.sh -p local_ip      ---- check local service
sh GeneralCheck.sh -f ip_file       ---- check 2 or more services
sh GeneralCheck.sh -n item_num      ---- check items with option -n
      (-n): 1  System_Info
            2  Process_Status
            3  Disk_Space
            4  Network_Status
            5  PG_Status
            6  HA_Status
            7  VM_Status
            8  System_Log
            9  Tool_Deployment
sh GeneralCheck.sh -v                ---- see version
sh GeneralCheck.sh -e                ---- ignore version expired
sh GeneralCheck.sh -h                ---- show this usage
[root@ha_92 tmp]#
```

可以使用-n 参数，来指定巡检部分内容，例如只巡检系统信息

(System_Info) 和进程状态 (Process_Status)，如下图：

```
[root@vm100 home]# sh GeneralCheck.sh -p 207.207.35.100 -n 1,2
The number of ip addr : 1
=====
Tue Nov 10 16:19:01 CST 2020
----- ip : 207.207.35.100 checking
check System Info      (11)  [ done ]
check Process Status   (3 )  [ done ]
check result( /home/207.207.35.100_check.log ) created.
----- ip : 207.207.35.100 finished
=====
```

2.4 手动巡检

<1>手动巡检一台本地/远程服务器

```
sh GeneralCheck.sh -p ip
```

注：1)巡检远程服务器，未配置互信时，会提示输入密码；

2)巡检日志文件会生成在巡检脚本目录下。

命令示例：

```
[root@ha_123 zhaochen]# sh GeneralCheck.sh -p 207.207.35.123
The number of ip addr : 1
=====
Fri Sep  4 13:58:14 CST 2020
----- ip : 207.207.35.123 checking
check System Info      (10)  [ done ]
check Process Status   (3 )  [ done ]
check Disk Space       (14)  [ done ]
check Network Status   (13)  [ done ]
check PG Status        (1 )  [ done ]
check HA Status        (1 )  [ done ]
check VM Status        (1 )  [ done ]
check System Log       (3 )  [ done ]
check Tool Deployment  (3 )  [ done ]
check result( /home/zhaochen/207.207.35.123_check.log ) created.
----- ip : 207.207.35.123 finished
=====
```

<2>手动巡检多台服务器

sh GeneralCheck.sh -f ip.txt

注：1)需要提供 ip 地址和 root 密码，以空格分隔；

2)本地服务器可以不填写密码，如 134 为本机；

3)有互信配置的服务器可以不填写密码，如 135（其与 134 有互信）；

4)若 ssh 端口不是默认端口 22，ip 地址格式应该写为 ip:port，

示例：207.207.35.135:22222；

5)若提供的密码不正确，巡检中会提示你再次输入服务器 root 密码；

6) ip.txt 文件名称非固定，但内容须符合要求；

7) 巡检日志文件会生成在巡检脚本路径下，打包为 tar 包。

命令示例：

```
[root@ha_92 tmp]#
[root@ha_92 tmp]# more ip.txt
207.207.35.92
207.207.35.123 admin_123
[root@ha_92 tmp]#
```



```

[root@ha_92 tmp]# sh GeneralCheck.sh -f ip.txt
Version : v3.0 release in 2022-04-27
The number of ip addr : 2
=====
Wed Apr 27 17:03:59 CST 2022
----- ip : 207.207.35.92 checking
check System Info      (15) [ done ]
check Process Status   ( 3) [ done ]
check Disk Space       (20) [ done ]
check Network Status   (12) [ done ]
check PG Status        (22) [ done ]
check HA Status        ( 1) [ done ]
check VM Status        ( 9) [ done ]
check System Log       ( 5) [ done ]
check Tool Deployment   ( 3) [ done ]
check result( /home/zhaochen/tmp/207.207.35.92_check.log ) created.
----- ip : 207.207.35.92 finished
=====

ip : 207.207.35.123 is a remote ip
ip : 207.207.35.123 available
Transferring script ...
Running script ...
Version : v3.0 release in 2022-04-27
The number of ip addr : 1
=====
Wed Apr 27 17:12:45 CST 2022
----- ip : 207.207.35.123 checking
check System Info      (15) [ done ]
check Process Status   ( 3) [ done ]
check Disk Space       (19) [ done ]
check Network Status   (12) [ done ]
check PG Status        ( 1) [ done ]
check HA Status        ( 1) [ done ]
check VM Status        ( 1) [ done ]
check System Log       ( 5) [ done ]
check Tool Deployment   ( 3) [ done ]
check result( /home/207.207.35.123_check.log ) created.
----- ip : 207.207.35.123 finished
=====

Collecting data ...
----- ip : 207.207.35.123 collected -----

tar log to /home/zhaochen/tmp ... [ done ]
check log.tar.gz file list:
-rw-r--r-- root/root      58721 2022-04-27 17:04 207.207.35.92_check.log
-rw-r--r-- root/root     397270 2022-04-27 17:13 207.207.35.123_check.log
-rw-r--r-- root/root      44262 2022-04-27 17:04 207.207.35.92_check.csv
-rw-r--r-- root/root      20573 2022-04-27 17:13 207.207.35.123_check.csv
[root@ha_92 tmp]#

```

2.5 定时巡检

请在定时任务/etc/crontab 中添加如下定时任务，并重启定时任务服务

```
00 */6 * * * root /bin/sh /home/GeneralCheck_Crontab.sh
```

/home/ip.txt

注：1)此命令定时巡检间隔为 6 小时；

2)务必给所有 ip 填写上对应的密码，并确保密码正确；

3)巡检日志文件和巡检执行日志会归档至

/var/log/general_check/目录下，并定时清理；

4)可以执行如下命令，验证定时任务执行时是否有其他报错，如服务器密码错误等：sh /home/GeneralCheck.sh -f /home/ip.txt -t

5) ip.txt 文件名称非固定，但内容须符合要求；

如果 ip 不能 ping 通，会进行提示，并跳过此 ip 的服务器的巡检。

```
=====
此地址 207.207.207.207 是远程ip
ip : 207.207.207.207 不可用
```

2.6 日志归档

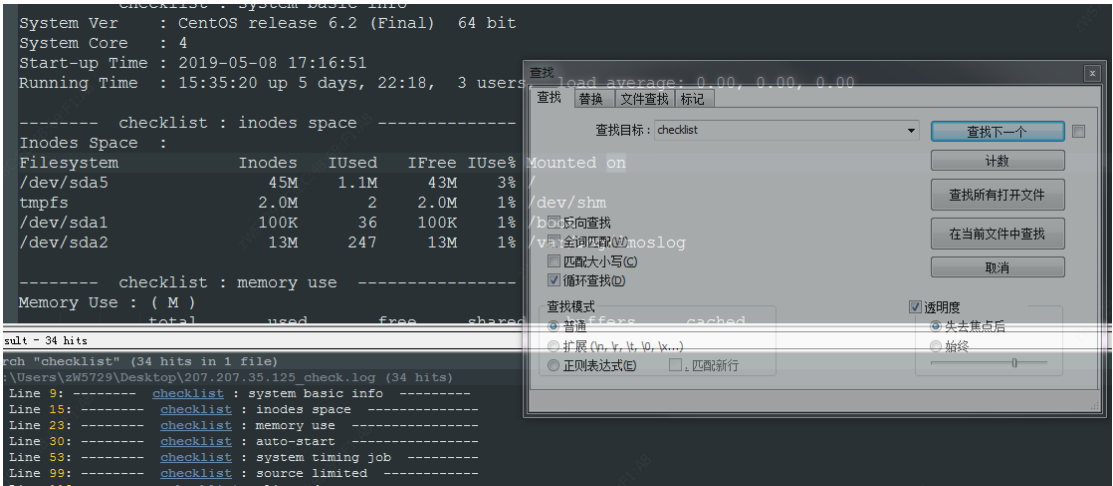
<1>手动巡检时，巡检日志会生成在巡检脚本目录下；

<2>巡检多台服务器时，会将巡检日志自动打包，以巡检时间命名；

<3>定时巡检时，巡检日志会生成在/var/log/general_check 目录下，并定时清理。

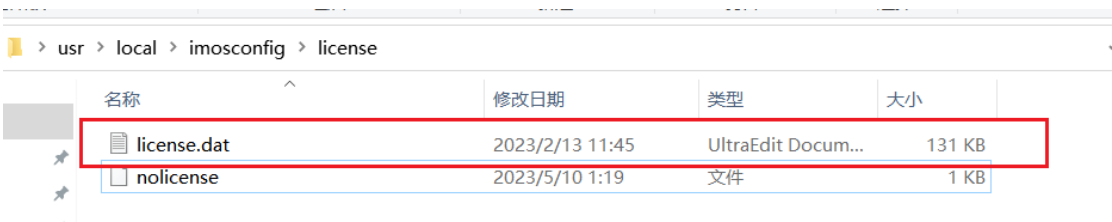
3. 定位巡检项

在 PC 上使用编辑器查看巡检日志时，查找 checklist 关键字（新版本为 CHECKLIST），将检索出所有的巡检项标题，方便快速查看某巡检项的巡检结果。



4. 巡检日志

巡检结果会输出 2 种日志文件 check.log 和 check.csv，和 svconfig.tgz 的授权备份文件。在压缩包的 usr\local\imosconfig\license 下面存在收集的 license 文件。

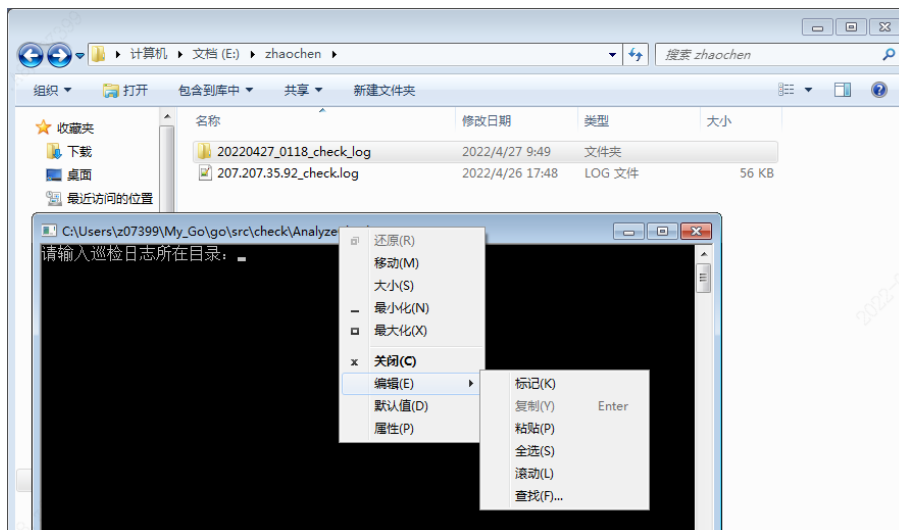


5. 巡检日志分析工具

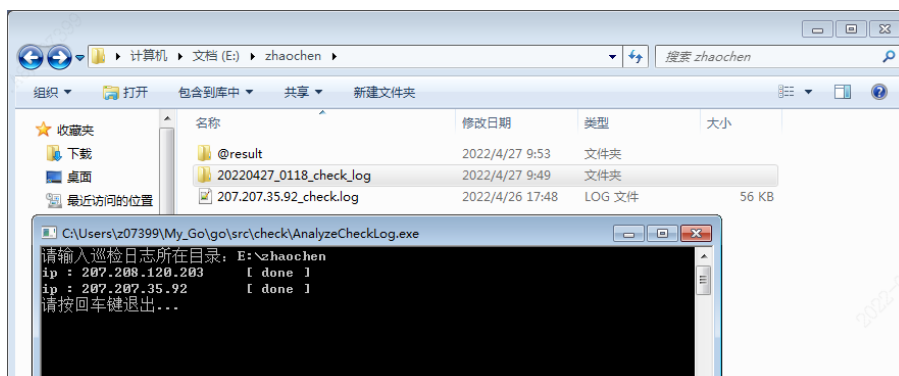
将收集到的巡检日志存放在电脑中，并在电脑上双击左键运行程序 AnalyzeCheckLog_windows.exe（日志和程序可以不在同一个文件夹内）。

鼠标右键点击窗口的边框，选择【编辑】-【粘贴】需要检查的文件夹路径（若直接回车则默认分析程序所在的父目录下的巡检日志）。

注：程序会分析目录内及其子目录下的巡检日志



分析完成后，巡检日志的目录中会新出现一个@result 文件夹（若按文件名排序，此文件夹默认会排在第一个），里面存放着分析完成的巡检日志的结果。



分析结果如下

```

1   Wed Apr 27 10:08:33 CST 2022
2   IP : 207.207.35.92
3   Version : v3.0 release in 2022-04-27
4   ----- CHECKLIST : grub -----
5   ----- RESULT(grub) : [ wrong ] -----
6   [Content]:grub设置可能存在隐患, 请参考《BIOS资源冲突宕机问题及解决》文档进行处理
7
8
9   ----- CHECKLIST : system timing job -----
10  /etc/crontab :
11  # * * * * * user-name command to be executed
12  */10 * * * * root /bin/sh /usr/local/bin/ipConflictDetect.sh >/dev/null 2>&1
13  */10 * * * * root /bin/sh /usr/local/bin/ethInfo.sh >/dev/null 2>&1
14  30 1 * * * root /usr/local/bin/autoclean.sh >/dev/null 2>&1
15  00 2 * * * root /usr/local/bin/autoclean_imos_sub_table.sh >/dev/null 2>&1
16  30 6 * * * root /usr/local/bin/autoclean_operlog_and_event_record.sh >/dev/null 2>&1
17  30 14 * * * root /usr/local/bin/autoclean_operlog_and_event_record.sh >/dev/null 2>&1
18  30 22 * * * root /usr/local/bin/autoclean_operlog_and_event_record.sh >/dev/null 2>&1
19  00 1 * * * root /usr/local/bin/autoclean_history_file.sh >/dev/null 2>&1
20  */20 * * * root /usr/local/bin/cleanhttpdlog.sh >/dev/null 2>&1
21  #0 4 * * 6 root /usr/local/bin/creatcfg.sh >/dev/null 2>&1
22  59 23 * * * root /usr/local/bin/bk_pglog.sh >/dev/null 2>&1
23  */5 * * * * root flock -xn /tmp/core_clean.lock -c 'sh /usr/local/bin/core_clean.sh' >/
24  00 05 * * * root /usr/local/bin/pg_maintain.sh >/dev/null 2>&1
25  */10 * * * * root /usr/local/bin/deal_with_db_age_overflow.sh >/dev/null 2>&1
26  30 5 * * * root /usr/local/bin/pg_repack.sh 'imos' 127.0.0.1 10 >/dev/null 2>&1
27  ----- RESULT(crond running) : [ wrong ] -----
28  [Content]:系统定时任务未运行
29  建议启动系统定时任务
30  ----- RESULT(repeated dbbk) : [ correct ] -----
31  ----- RESULT(pgsq pg_rman) : [ correct ] -----
32

```

二、巡检项说明

1. 操作系统检查

1.1 Linux 系统基本信息

巡检示例：

```

----- checklist : system basic info -----
System Ver   : CentOS release 6.2 (Final) 64 bit
System Core  : 4
Cpu Info     : Intel(R) Xeon(R) CPU E31225 @ 3.10GHz
Start-up Time : 2020-06-15 09:55:54
Running Time  : 17:34:04 up 24 days, 7:38, 2 users, load average: 0.00, 0.00, 0.00
Host Name    : ha_92
System Time   : Thu 09 Jul 2020 17:34:04 CST
Hardware Time : Thu 09 Jul 2020 05:37:10 PM CST -0.547020 seconds

```

巡检分析：

<1>linux 版本， cpu 核心数及型号，开机时间及运行时间,平均负载，主机名。

<2>System Time 系统时间，Hardware Time 系统硬件时间，**两者需要保持相同。**

1.2 grub 配置

巡检示例：

```
----- checklist : grub -----  
----- RESULT(grub) : [ wrong ] -----
```

巡检分析：

<1>若检查结果为 wrong，请按照《BIOS 资源冲突宕机问题及解决》文档来检查处理。

1.3 cpuspeed 性能调节

巡检示例：

```
----- checklist : cpu speed -----  
cpuspeed is stopped  
***** RESULT(cpu speed) : [ correct ] *****
```

巡检分析：

<1>检测是否开启 cpu 性能调节功能

☆自动化巡检---若系统开启 cpuspeed，进行提示

1.4 内存硬件信息

巡检示例：

```

----- checklist : memory hardware info -----
Memory Device
  Size: 8192 MB
  Type: DDR3
  Speed: 1600 MHz
Memory Device
  Size: 8192 MB
  Type: DDR3
  Speed: 1600 MHz
Memory Device
  Size: 8192 MB
  Type: DDR3
  Speed: 1600 MHz
Memory Device
  Size: 8192 MB
  Type: DDR3
  Speed: 1600 MHz
Memory Slot (Used / All) : 4 / 4

```

巡检分析：

<1>memory device 代表一根内存条；

<2>size 表示内存容量；

<3>type 表示内存条类型；

<4>speed 表示内存频率；

<5>memory slot 代表服务器的内存插槽数量。

1.5 root 密码有效期

巡检示例：

```

----- checklist : password validity -----
Last password change : Nov 09, 2020
Password expires : never
Password inactive : never
Account expires : never
Minimum days between password change : 0
Maximum days between password change : 99999
Days of warning before password expires : 7
Days when root password expired : 99999 days

***** RESULT(password validity) : [ correct ] *****

```

巡检分析：

<1>linux 系统中 root 用户的密码有效期情况。

☆自动化巡检----root 密码有效期小于 30 天时，进行提示

1.6 系统配置

巡检示例：

```
----- checklist : system config -----
/etc/sysctl.conf :

net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 4294967295
kernel.shmall = 268435456
kernel.sem = 250 250000 32 1000
net.ipv4.tcp_max_syn_backlog=65535
net.core.somaxconn=65535
kernel.core_pattern = /root/core_dump-%e-%p-%t-%s-%h
net.ipv4.neigh.default.gc_thresh1=512
net.ipv4.neigh.default.gc_thresh2=2048
net.ipv4.neigh.default.gc_thresh3=4096
net.ipv4.tcp_timestamps=0
net.ipv4.tcp_syncookies=1
```

巡检分析：

<1>若服务器中安装有双机或 N+1 软件，net.ipv4.ip_forward 必须为 0。

1.7 Inodes 空间

巡检示例：

```
----- checklist : inodes space -----
/*
Filesystem            Inodes    IUsed    IFree    IUse% Mounted on
/dev/sda6              22M      480K      22M      3% /
tmpfs                  180K      104      179K      1% /dev/shm
/dev/sda1               50K       41      50K      1% /boot
/dev/sda2              31M       6.8K     31M      1% /mnt/syncdata
/dev/sda3              6.2M      253      6.2M      1% /var/log/imoslog
*/
***** RESULT(inodes space) : [ correct ] *****
```

巡检分析：

<1>检查 IUse%列的数值，若达到或接近 100%，代表系统中文件数过多，可能会无法创建新的文件，导致系统环境异常，需要联系研发处理。

☆自动化巡检----使用率高于 80%会有提示

1.8 内存使用

巡检示例：

```
----- checklist : memory use -----
( unit : M )
      total      used      free      shared      buffers      cached
Mem:      3843      3719       124         0         31         396
-/+ buffers/cache:      3290       552
Swap:      1999       122      1877
```

巡检分析：

<1>查看 free 列数值，若两项数值过小，说明剩余内存空间可能不足，需要检查是否存在内存不足或内存泄漏等问题。

1.9 程序 swap 使用情况

巡检示例：

```
----- checklist : swap use -----
( unit : MB )
PID    SIZE  PROC
9950    0M   postgres: postgres imos 207.207.35.99(12794) idle
9587    0M   postgres: postgres imos 207.207.35.99(12788) idle
870     0M
783     0M
767     0M   postgres: postgres imos 207.207.35.92(45641) idle
766     0M   postgres: postgres imos 207.207.35.92(45640) idle
765     0M   postgres: postgres imos 207.207.35.92(45639) idle
764     0M   postgres: postgres imos 207.207.35.92(45638) idle
763     0M   postgres: postgres imos 207.207.35.92(45637) idle
762     0M   postgres: postgres imos 207.207.35.92(45636) idle
761     0M   postgres: postgres imos 207.207.35.92(45635) idle
760     0M   postgres: postgres imos 207.207.35.92(45634) idle
759     0M   postgres: postgres imos 207.207.35.92(45633) idle
758     0M   postgres: postgres imos 207.207.35.92(45632) idle
757     0M   postgres: postgres imos 207.207.35.92(45631) idle
756     0M   postgres: postgres imos 207.207.35.92(45630) idle
755     0M   postgres: postgres imos 207.207.35.92(45629) idle
754     0M   postgres: postgres imos 207.207.35.92(45628) idle
753     0M   postgres: postgres imos 207.207.35.92(45627) idle
752     0M   postgres: postgres imos 207.207.35.92(45626) idle
```

巡检分析：

<1>查找使用 swap 最多的 20 个进程；

<2>如果进程占用 swap 过多，系统内存可能不足，容易出现性能问题。

1.10 开机自启动项

巡检示例：

```
----- checklist : auto-start -----
/*
#l/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.
touch /var/lock/subsys/local
sh /usr/local/sbin/cmserver.sh start >/dev/null 2>&1 &
. /etc/profile >/dev/null 2>&1;nohup /usr/local/apache-tomcat-9.0.0.M11/h
sh /usr/local/bin/modntptime.sh
iptables -A OUTPUT -p icmp --icmp-type 13 -j DROP
iptables -A OUTPUT -p icmp --icmp-type 14 -j DROP
|
corosync 0:off 1:off 2:on 3:on 4:on 5:on 6:off
*/
***** RESULT(auto-start) : [ wrong ] *****
HA or N+1 : on
Other : on
NTP Clock Sync : sh /usr/local/bin/modntptime.sh
```

巡检分析：

<1>查看双机、N+1 或者其他交付件的自启动配置，若同时存在则需要修改；

<2>查看 ntp 时间同步是否配置，若不存在，建议添加部署。

☆自动化巡检----业务自启动是否正确添加

异常处理：

修改/etc/rc.d/rc.local 配置文件，将多余的行注释掉或者删除。

1.11 系统定时任务

巡检示例：

```

00 4 * * * root /opt/wise/ial2k/bin/dbbk_iam.sh
*/10 * * * root /bin/sh /usr/local/bin/adapterChecked.sh
00 5 * * * root /usr/local/bin/checkreindex.sh
*/10 * * * root /usr/local/bin/lockandconnection.sh
*/10 * * * root /bin/sh /usr/local/bin/ipConflictDetect.sh
*/10 * * * root /bin/sh /usr/local/bin/ethInfo.sh
30 1 * * * root /usr/local/bin/autoclean.sh
00 2 * * * root /usr/local/bin/autoclean_imos_sub_table.sh
30 6 * * * root /usr/local/bin/autoclean_operlog_and_event_record.sh
30 14 * * * root /usr/local/bin/autoclean_operlog_and_event_record.sh
30 22 * * * root /usr/local/bin/autoclean_operlog_and_event_record.sh
00 1 * * * root /usr/local/bin/autoclean_history_file.sh
*/20 * * * root /usr/local/bin/cleanhttpdlog.sh
#0 4 * * 6 root /usr/local/bin/creatcfg.sh
59 23 * * * root /usr/local/bin/bk_pglog.sh
*/5 * * * root flock -xn /tmp/core_clean.lock -c 'sh /usr/local/bin/
00 05 * * * root /usr/local/bin/pg_maintain.sh
30 5 * * * root /usr/local/bin/pg_repack.sh ' imos' 207.207.35.91 10
*/10 * * * root /usr/bin/haservice.sh
* * * * root /usr/local/habin/coro_pace_check.sh

***** RESULT(multi dbbk.sh) : [ correct ] *****
***** RESULT(crond running) : [ wrong ] *****

```

巡检分析：

<1>查看定时任务配置是否正确,是否有重复的；

<2>查看定时系统任务是否在运行状态。

注：CentOS 6 和 CentOS 7 系统下，显示会略有不同。

☆自动化巡检---检测是否存在多个数据库备份任务

☆自动化巡检---检测定时任务服务是否正在运行

异常处理：

<1>修改/etc/crontab 定时任务配置文件，删除多余的配置项；

<2>保存退出，并重启定时任务。

1.12 系统资源限制配置

巡检示例：

```

----- checklist : source limited -----
core file size      (blocks, -c) 0
data seg size      (kbytes, -d) unlimited
scheduling priority      (-e) 0
file size          (blocks, -f) unlimited
pending signals     (-i) 30631
max locked memory   (kbytes, -l) 64
max memory size     (kbytes, -m) unlimited
open files          (-n) 20000
pipe size           (512 bytes, -p) 8
POSIX message queues (bytes, -q) 819200
real-time priority   (-r) 0
stack size          (kbytes, -s) 10240
cpu time            (seconds, -t) unlimited
max user processes   (-u) 1024
virtual memory       (kbytes, -v) unlimited
file locks           (-x) unlimited
-----
# Default limit for number of user's processes to prevent
# accidental fork bombs.
# See rhbz #432903 for reasoning.

*          soft    nproc      1024

```

巡检分析：

<1>查看 open files 和 max user processes 项，代表打开的最大文件数与最大的运行进程数，若局点业务量较大，可以适当调大此参数。

1.13 imos 交付件安装

巡检示例：

```

----- checklist : imos install -----
total 20K
drwxr-xr-x 2 root root 4.0K Nov  6 10:52 haconf
drwxr-xr-x 2 root root 4.0K Nov  6 10:50 license
drwxr-xr-x 2 root root 4.0K Nov  6 10:38 pkcm
drwxr-xr-x 2 root root 4.0K Nov  6 02:22 vkd
drwxr-xr-x 2 root root 4.0K Nov  6 10:38 vmconf

```

巡检分析：

<1>查看已安装的交付件情况。

1.14 linux 审计功能

巡检示例：

```

----- checklist : linux audit -----
linux audit : on
try disable : ok

```

巡检分析：

<1>检查审计功能是否开启，如果开启，则自动进行关闭。

1.15 Linux 内核信息

巡检示例：

```
----- checklist : linux dmesg -----
Linux Dmesg :
e1000e: eth2 NIC Link is Down
lo: Disabled Privacy Extensions
ADDRCONF(NETDEV_UP): eth1: link is not ready
e1000e: eth1 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
ADDRCONF(NETDEV_CHANGE): eth1: link becomes ready
e1000e 0000:00:19.0: irq 39 for MSI/MSI-X
e1000e 0000:00:19.0: irq 39 for MSI/MSI-X
ADDRCONF(NETDEV_UP): eth2: link is not ready
e1000e: eth2 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: Rx/Tx
ADDRCONF(NETDEV_CHANGE): eth2: link becomes ready
```

巡检分析：

<1>此项内容将展示内核相关的系统信息。

2. 系统进程检查

2.1 系统状态 (top 命令)

巡检示例：

```
----- checklist : process (top) -----
Current System Status :
top - 14:51:43 up 26 days, 34 min, 3 users, ①load average: 0.00, 0.14, 0.29
Tasks: 359 total, 1 running, 358 sleeping, 0 stopped, ②0 zombie
Cpu(s) ③2.1%us, 0.8%sy, 0.0%ni, ④96.5%id, ⑤0.6%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 16251716k total, 6248312k used, 10003404k free, 402800k buffers
Swap: 32767992k total, 11988k used, 32756004k free, 2076212k cached
```

巡检分析：

- ① 系统最近 1、5、15 分钟的平均负载，标准值为 $0.7 \times \text{CPU 核心数}$ ，超过表示此时系统负载偏高；
- ② 僵尸进程个数，标准值为 0，后面巡检会详细列出僵尸进程的信息；
- ③ 用户空间占用 cpu 百分比；

④ 空闲 cpu 百分比，数值越小，代表 cpu 整体负载越高；

⑤ IO 等待占用的 cpu 百分比。

2.2 进程状态 (ps axjf 命令)

巡检示例：

```

1 22180 22127 22127 ? -1 S1 0 0:43 haserver
1 23739 23738 23738 ? -1 S 1001 0:17 /home/postgres/pgsql/bin/postmaster -p 5432 -D /mnt/syncdata/pgsql/data
23739 23742 23742 23742 ? -1 Ss 1001 24:01 \_ postgres: logger process
23739 23746 23746 23746 ? -1 Ss 1001 0:14 \_ postgres: checkpoint process
23739 23747 23747 23747 ? -1 Ss 1001 0:08 \_ postgres: writer process
23739 23748 23748 23748 ? -1 Ds 1001 0:44 \_ postgres: wal writer process
23739 23749 23749 23749 ? -1 Ss 1001 0:01 \_ postgres: autovacuum launcher process
23739 23750 23750 23750 ? -1 Ss 1001 0:41 \_ postgres: stats collector process
23739 24154 24154 24154 ? -1 Ss 1001 0:13 \_ postgres: postgres imos 208.208.11.122(34122) idle
23739 24461 24461 24461 ? -1 Ss 1001 0:00 \_ postgres: postgres imos 202.91.3.102(38028) idle
23739 24466 24466 24466 ? -1 Ss 1001 0:11 \_ postgres: postgres imos 202.91.3.106(59864) idle
23739 24546 24546 24546 ? -1 Ss 1001 0:00 \_ postgres: postgres imos 207.101.67.32(33859) idle
23739 28288 28288 28288 ? -1 Ss 1001 0:00 \_ postgres: postgres imos 207.101.67.39(37501) idle

***** RESULT(zombie process:1) : [ wrong ] *****
31193 31195 31195 31195 ? -1 Zs 0 0:00 \_ [pidStat.sh] <defunct>
***** RESULT(tcpdump process:0) : [ correct ] *****

```

巡检分析：

<1>查看相关服务或业务进程是否启动，后面会详细列出进程状态；

<2>查看是否有僵尸进程，若有业务进程的僵尸进程，则为服务进程有异常，需要分析处理。

☆自动化巡检----检测是否存在僵尸进程

☆自动化巡检----检测是否存在背景抓包进程

异常处理：

<1>如果是 corosync 的僵尸进程，是由于网络原因产生的，需要重启服务器来恢复；

<2>如果是其他的进程，可以用 kill -9 进程号的命令来清理；

注：结果中的第二列是进程的 pid 号

```
kill -9 30241
```

<3>如果存在背景抓包进程，请注意是否有必要开启，及磁盘空间情况。

2.3 业务进程运行状态

巡检示例：

```
----- checklist : imos process -----
Process Status :
Thu Jul 18 14:51:00 2019 flock -xn /tmp/dc_recovery.lock -c sh /usr/local/bin/dc_recovery.sh
Thu Jul 18 14:51:00 2019 sh /usr/local/bin/dc_recovery.sh
Thu Jul 18 11:52:26 2019 /usr/local/bin/redis_server *:6379
Thu Jul 18 11:52:26 2019 /usr/local/bin/img
Thu Jul 18 11:52:26 2019 /usr/local/bin/mcserver
Thu Jul 18 11:52:27 2019 /usr/local/bin/sgserver
Thu Jul 18 11:52:27 2019 /usr/local/bin/vmserver
Thu Jul 18 11:52:28 2019 /usr/local/bin/DiskReadOnlyCheck
Thu Jul 18 11:52:29 2019 /usr/local/bin/impserver
Thu Jul 18 11:52:29 2019 /usr/local/bin/iscloud
Thu Jul 18 11:52:29 2019 /usr/local/bin/itcserver
Thu Jul 18 11:52:30 2019 /usr/local/bin/kbserver
Thu Jul 18 11:52:30 2019 /usr/local/bin/nmserver
Thu Jul 18 11:52:30 2019 /usr/local/bin/nweb 8096
Thu Jul 18 11:52:30 2019 /usr/local/bin/onvifserver
Thu Jul 18 11:52:30 2019 /usr/local/bin/paggbserver
Thu Jul 18 11:52:31 2019 /usr/local/bin/pagserver
Thu Jul 18 11:52:31 2019 /usr/local/bin/rptserver
Thu Jul 18 11:52:31 2019 /usr/local/bin/serversnmpd -c /usr/local/svconfig/server/conf/snmp.conf
Thu Jul 18 11:52:31 2019 /usr/local/bin/smart_community
Thu Jul 18 11:52:31 2019 /usr/local/bin/smart_community_sync
Thu Jul 18 11:52:32 2019 /usr/local/bin/stunserver
Thu Jul 18 11:52:32 2019 /usr/local/bin/unpserver
Thu Jul 18 11:52:32 2019 /usr/local/bin/vmdaemon /usr/local/svconfig/vmconf/vm_daemon.cfg
Tue Jun 25 11:13:03 2019 /usr/local/bin/cdmdaemon /usr/local/svconfig/cdmconf/cdm_daemon.cfg
=====
```

巡检分析：

<1>红线左侧表示进程的上一次启动时间，红线右侧表示具体的业务进程；

<2>若某个业务有停止过，则启动时间会与其他进程不相同。

3. 磁盘空间检查

3.1 fstab 配置

巡检示例：

```
----- checklist : fstab config -----
/etc/fstab :
UUID=29cf6d2d-3578-41a0-8055-8ce455b4c37f / ext3 defaults 1 1
UUID=1849a3b6-1381-46ce-be94-dc300ad92a5c /boot ext3 defaults 1 2
UUID=6378d6d5-a882-41ca-989f-82a4a4b81e13 /mnt/syncdata ext3 defaults 1 2
UUID=c2557872-eb1c-4703-8c8b-8b0456d340d0 /var/log/imoslog ext3 defaults 1 2
UUID=5a0842a7-5e8b-4f43-b2a1-ae4733693844 swap swap defaults 0 0
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

巡检分析：

<1>查看开机启动时磁盘分区的挂载配置。

3.2 磁盘分区空间

巡检示例：

```
----- checklist : disk space -----
/*
Filesystem      Type      Size      Used Avail Use% Mounted on
/dev/sda6        ext3      339G      192G  130G   60% /
tmpfs            tmpfs      1.9G       59M   1.9G    4% /dev/shm
/dev/sda1        ext3      194M       45M   140M   25% /boot
/dev/sda2        ext3      481G      715M   456G    1% /mnt/syncdata
/dev/sda3        ext3      97G       294M    91G    1% /var/log/imoslog
*/
***** RESULT(disk space) : [ correct ] *****
```

巡检分析：

<1>查看各磁盘分区的空间大小及使用率，若某分区使用率较高，则可能有异常，如日志未打包，产生大量 core 文件等等。

☆自动化巡检----使用率高于 80%会有提示

异常处理：

<1>检查对应目录下是否有背景抓包或者 core 文件产生，有则清楚不需要的文件；

<2>检查对应目录下是否有不需要的文件，有则删除。

3.3 文件/文件夹属性 & 使用空间

巡检示例：

```
----- checklist : dir space -----
Dir Check      : /dev/null
Dir Mod        : crw-rw-rw-,root,root
Correct Mod    : crw-rw-rw-,root,root
Dir Space      : 0
***** RESULT(dir mod) : [ correct ] *****

Dir Check      : /tmp
Dir Mod        : drwxrwxrwt,root,root
Correct Mod    : drwxrwxrwt,root,root
Dir Space      : 72K
***** RESULT(dir mod) : [ correct ] *****
```

巡检分析：

<1>查看巡检的文件/目录的属性及使用空间。

☆自动化巡检----文件属性是否正确

异常处理：

<1>修改/dev/null 文件权限：

```
rm -rf /dev/null && mknod -m 666 /dev/null c 1 3 && chown root:root /dev/null
```

<2>给/tmp 目录添加 t 权限：

```
chmod o+t /tmp
```

<3>修改/mnt/syncdata/resftp/A8 目录权限：

```
chmod 775 /mnt/syncdata/resftp/A8

chown uploadusr.h3cgroup /mnt/syncdata/resftp/A8
```

3.4 大容量文件

巡检示例：

```
----- checklist : large file -----
/*
( /var/log /root/core* /var/lib/pacemaker/cores/ )
file size > 2G :

*/
***** RESULT(large file) : [ correct ] *****
```

巡检分析：

<1>搜索描述中的三个目录下的文件，列出单个文件超过 2G 的文件。

☆自动化巡检----是否存在大容量文件

3.5 lsof 文件状态

巡检示例：

```

----- checklist : lsaf -----
COMMAND  PID    USER  FD   TYPE    DEVICE  SIZE/OFF      NODE NAME
postaste 16723  postgres 45u   REG     147,0   16777216  2756862 /mnt/syncdata/pgsql/data/pg_xlog/0000000100000003000000051 (deleted)
postaste 16899  postgres 22u   REG     147,0   16777216  2756862 /mnt/syncdata/pgsql/data/pg_xlog/0000000100000003000000051 (deleted)
postaste 16902  postgres 15u   REG     147,0   16777216  2756862 /mnt/syncdata/pgsql/data/pg_xlog/0000000100000003000000059 (deleted)
postaste 16913  postgres 49u   REG     147,0   16777216  2756862 /mnt/syncdata/pgsql/data/pg_xlog/0000000100000003000000051 (deleted)
postaste 18625  postgres 11u   REG     147,0   16777216  2756862 /mnt/syncdata/pgsql/data/pg_xlog/0000000100000003000000051 (deleted)

deleted but unreleased size : 80 M

```

巡检分析：

<1>列出系统中被标记为删除，但实际还未删除的文件，并计算总大小。

3.6 core 文件

巡检示例：

```

----- checklist : core file -----
/root/core_dump-onvifserver-27030-1578372361-11-vm135 210M 2020-01-07 12:46:01
ELF 32-bit LSB core file Intel 80386, version 1 (SYSV), SVR4-style, from '/usr/local/bin/onvifserver'

/root/core_dump-onvifserver-20924-1578339782-11-vm135 205M 2020-01-07 03:43:02
ELF 32-bit LSB core file Intel 80386, version 1 (SYSV), SVR4-style, from '/usr/local/bin/onvifserver'

```

巡检分析：

<1>查看系统中近 10 天内是否有 core 文件生成，若有则代表有进程崩溃，

需要联系研发分析处理。

异常处理：

<1>收集 core 文件及相关时间点日志；

<2>联系相关业务研发分析。

3.7 core 文件调试

巡检示例：

```

----- checklist : core analysis -----

*****

debug : GNU gdb (GDB) Red Hat Enterprise Linux (7.2-50.el6)
core : /root/core_dump-mcserver-31379-1567491219-6-vm135
process : '/usr/local/bin/mcserver'

*****
GNU gdb (GDB) Red Hat Enterprise Linux (7.2-50.el6)
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...

```

```
(gdb)
#0 0x001fb424 in _kernel_vsyscall ()
#1 0x00a40886 in epoll_wait () from /lib/libc.so.6
#2 0x00d5d269 in ACE_Dev_Poll_Reactor::work_pending_i (this=0x960f260, m=0)
#3 0x00d5e7ec in ACE_Dev_Poll_Reactor::handle_events_i (this=0x960f260, m=0)
#4 0x00d5f669 in ACE_Dev_Poll_Reactor::handle_events (this=0x960f260, m=0)
#5 0x0013db52 in ACE_Reactor::handle_events (this=0x960ee40, max_wait_t=0)
#6 0x0013b4df in CImfAppMgr::handleEvents (this=0x960f240) at /mnt/jenkins/...
#7 0x00601e8c in IMOS_Start () at /mnt/jenkins/template_build/build_lin...
#8 0x0804b3da in ?? ()
#9 0x0804b3ee in ?? ()
#10 0x00000000 in ?? ()
```

巡检分析：

<1>分析最近 10 天内的最新的 2 个 core 文件，将打印程序崩溃时的函数调用栈等信息，方便定位崩溃原因。

4. 网络状态检查

4.1 网卡配置

巡检示例：

```
----- checklist : network card -----
eth1      Link encap:Ethernet  HWaddr 48:EA:63:01:8D:0B
          inet addr:207.207.35.135  Bcast:207.207.35.255  Mask:255.255.255.0
          inet6 addr: 2013:207:207:35:4aea:63ff:fe01:8d0b/64 Scope:Global
          inet6 addr: fe80::4aea:63ff:fe01:8d0b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2840394  errors:1  dropped:0  overruns:0  frame:1
          TX packets:3705402  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:15000
          RX bytes:250031450 (238.4 MiB)  TX bytes:3195558653 (2.9 GiB)
          Interrupt:18  Memory:cb400000-cb420000

eth1:0    Link encap:Ethernet  HWaddr 48:EA:63:01:8D:0B
          inet addr:207.207.35.145  Bcast:207.207.35.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interrupt:18  Memory:cb400000-cb420000
```

巡检分析：

<1>查看网卡配置、子网掩码、虚地址等是否配置正确。

4.2 网卡网速及流量

巡检示例：

```

----- checklist : netcard speed -----
eth1 : 1000 MB/s
  recieve_flow : 0.37 KB/s
  transfer_flow : 0.60 KB/s

eth2 : 1000 MB/s
  recieve_flow : 0.44 KB/s
  transfer_flow : 0.61 KB/s

lo :
  recieve_flow : 0.00 KB/s
  transfer_flow : 0.00 KB/s

***** RESULT(netcard speed) : [ correct ] *****

```

巡检分析：

<1>查看网卡协商网速是否为 1000M/s；

<2>查看当前网卡流量情况。

☆自动化巡检---检测到网速非千兆时会提示

4.3 路由配置

巡检示例：

```

----- checklist : ip route -----
Routing :
172.16.0.0/24 dev eth2 proto kernel scope link src 172.16.0.2
207.207.35.0/24 dev eth1 proto kernel scope link src 207.207.35.200
10.10.10.0/24 via 207.207.35.125 dev eth1
169.254.0.0/16 dev eth2 scope link metric 1002
169.254.0.0/16 dev eth1 scope link metric 1007
default via 207.207.35.1 dev eth1 src 207.207.35.200 metric 20 ①
default via 207.207.35.1 dev eth1 metric 50 ②

```

巡检分析：

<1>查看系统默认网关②的配置是否正确，若有启动虚地址，需查看虚地址的默认网关①的 metric 参数是否低于默认网关（ $20 < 50$ ，虚地址路由优先级较高）。

4.4 ARP 缓存表

巡检示例：

```

----- checklist : arp config -----
/*
ARP Para :
gc_interval : 30
gc_stale_time : 60 (default 60)
gc_thresh1 : 512 (default 128)
gc_thresh2 : 2048 (default 512)
gc_thresh3 : 4096 (default 1024)

Count of ARP in fact : 2
*/
***** RESULT(arp count) : [ correct ] *****

```

巡检分析：

<1>gc_interval、gc_thresh1、gc_thresh2、gc_thresh3 为系统 ARP 缓存表参数；

<2>Count of ARP in fact ： 目前实际的 ARP 缓存表中表项个数。

☆自动化巡检----arp 表项数量是否正常

注：建议研发分析，若需要调整其参数，请综合考虑现场网络情况。

4.5 ip 冲突检测

巡检示例：

```

----- checklist : ip conflict -----
ARPING 207.207.35.92 from 0.0.0.0 eth1
Sent 4 probes (4 broadcast(s))
Received 0 response(s)

***** RESULT(ip conflict) : [ correct ] *****
----- checklist : ping to gateway -----

```

巡检分析：

<1>检测当前网络中是否有相同 ip 的网络设备。

☆自动化巡检----检测是否存在相同 ip 的设备

4.6 系统网络状态

巡检示例：

```

----- checklist : ping to gateway -----
/*
ping.log : exist
02/19/20 11:00:33 --- 30 packets transmitted, 4 received, 86% packet loss, time 30003ms
*/
***** RESULT(ping loss:1) : [ wrong ] *****

----- checklist : ping to gateway -----
ping.log : non-exist or empty

---- ping test in manual ----
Group 1 Ping Test :
PING 207.101.67.1 (207.101.67.1) 56(84) bytes of data.
64 bytes from 207.101.67.1: icmp_seq=1 ttl=255 time=3.45 ms
64 bytes from 207.101.67.1: icmp_seq=2 ttl=255 time=1.38 ms
64 bytes from 207.101.67.1: icmp_seq=3 ttl=255 time=1.83 ms
64 bytes from 207.101.67.1: icmp_seq=4 ttl=255 time=4.42 ms
64 bytes from 207.101.67.1: icmp_seq=6 ttl=255 time=0.850 ms
64 bytes from 207.101.67.1: icmp_seq=7 ttl=255 time=1.08 ms
64 bytes from 207.101.67.1: icmp_seq=8 ttl=255 time=3.02 ms
64 bytes from 207.101.67.1: icmp_seq=9 ttl=255 time=3.02 ms
64 bytes from 207.101.67.1: icmp_seq=10 ttl=255 time=1.74 ms

--- 207.101.67.1 ping statistics ---
10 packets transmitted, 9 received, 10% packet loss, time 10011ms
rtt min/avg/max/mdev = 0.850/2.314/4.426/1.148 ms

```

巡检分析：

<1>若存在相关日志文件，则从日志中分析；若不存在，则手动 ping 网关（10 次*2 组）进行检查；

<2>查看 ping 网关状态，是否有 ping 不通或者频繁丢包问题。

☆自动化巡检----是否存在网络丢包

4.7 防火墙配置

巡检示例：

```

----- checklist : selinux -----
SELINUX=disabled
***** RESULT(selinux) : [ correct ] *****

----- checklist : iptables -----
/*
Table: filter
Chain INPUT (policy ACCEPT)
num target prot opt source destination

Chain FORWARD (policy ACCEPT)
num target prot opt source destination

Chain OUTPUT (policy ACCEPT)
num target prot opt source destination
1 DROP icmp -- 0.0.0.0/0 0.0.0.0/0
2 DROP icmp -- 0.0.0.0/0 0.0.0.0/0
*/
***** RESULT(iptables auto-start) : [ correct ] *****

```

巡检分析：

<1>检查防火墙相关配置

☆自动化巡检----检查防火墙是否被禁用，是否取消开机自启动

4.8 网络访问限制

巡检示例：

```
----- checklist : network limit -----
Allow :

Deny :
```

巡检分析：

<1>检查是否有配置允许或阻止 ssh 连接的 ip 地址；

<2>若为空，则表示未配置。

4.9 ssh 配置

巡检示例：

```
----- checklist : sshd config -----
sshd_config:
Protocol 2
SyslogFacility AUTHPRIV
PasswordAuthentication yes
ChallengeResponseAuthentication no
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
UsePAM yes
AcceptEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
AcceptEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
AcceptEnv LC_IDENTIFICATION LC_ALL LANGUAGE
AcceptEnv XMODIFIERS
X11Forwarding yes
Subsystem sftp /usr/libexec/openssh/sftp-server

----- checklist : ssh config -----
ssh_config:
Host *
    GSSAPIAuthentication yes
    ForwardX11Trusted yes
    SendEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
    SendEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
    SendEnv LC_IDENTIFICATION LC_ALL LANGUAGE
    SendEnv XMODIFIERS
```

巡检分析：

<1>检查 ssh 的配置文件。

5. PG 数据库检查

5.1 数据库地址 & 版本号

巡检示例：

```
----- checklist : pg basic info -----  
DB Server Ip   : 207.207.35.91  
DB Server Port : 5432  
DB Version    : PostgreSQL 11.1  
Independent DB : Yes
```

巡检分析：

<1>查看数据库版本信息，地址及服务端口；

<2>若已安装 HA or N+1，IP 地址应为虚地址；

<3>数据库 ip 非当前服务器 ip 时，认为是独立数据库。

注：双机和 N+1 中，备机数据库会误识别为独立数据库，只要数据库 ip
为主机的虚地址即可

5.2 数据库配置检查

巡检示例：

```
----- checklist : pg config -----
timezone           : Asia/Shanghai (Asia/Shanghai)
log_timezone       : Asia/Shanghai (Asia/Shanghai)
unix_socket_directories : /tmp      (/tmp)
client_encoding    : UTF8        (UTF8)
fsync              : on           (on)
max_connections    : 1200        (1200)
synchronous_commit : off         (off)
autovacuum         : on          (on)
track_counts       : on          (on)
effective_cache_size : 4GB
shared_buffers     : 2GB
work_mem           : 4MB
maintenance_work_mem : 64MB
log_statement      : none
log_lock_waits     : on
deadlock_timeout   : 1s
archive_mode       : off
archive_command     : (disabled)
```

巡检分析：

<1>查看数据库常用基本参数的配置。

注：括号内为正确值或推荐值。

5.3 数据库密码有效期

巡检示例：

```
----- checklist : pg psw life -----
/*
username | usesysid | usecreatedb | usesuper | userepl | usebypassrls | passwd | valuntil | useconfig
-----+-----+-----+-----+-----+-----+-----+-----+-----
postgres | t         | t           | t         | t         | t             | ***** |          |
repl     | f 32285  | f           | t         | f         | f             | ***** |          |
(2 rows)

*/
***** RESULT(pg psw life) : [ correct ] *****
```

巡检分析：

<1>查看数据库密码是否配置了有效期。

注：若配置了有效期，在密码到期后，将无法登录数据库。

☆自动化巡检----若密码配置了有效期，进行提示

5.4 数据库存储空间

巡检示例：

```
----- checklist : pg storage space -----
name      | owner  | size
-----+-----+-----
imos      | postgres | 72 MB
zabbix    | postgres | 15 MB
postgres  | postgres | 7707 kB
template1 | postgres | 7569 kB
template0 | postgres | 7569 kB
(5 rows)
```

巡检分析：

<1>查看数据库存储空间，空间大小与局点业务及数据量有关系；

<2>若同时存在 imos 和 cds 数据库，会分别进行数据库巡检。

5.5 数据库连接数

巡检示例：

```
----- checklist : pg connection -----
all connections :
datname | client_addr | state | count
-----+-----+-----+-----
imos    | 172.16.0.92/32 | active | 1
imos    | 207.207.35.91/32 | idle  | 166
imos    | 207.207.35.92/32 | active | 1
        |              |        | 5
-----+-----+-----+-----
all     | all         | all   | 173
(6 rows)
```

巡检分析：

<1>datname：连接的数据库名称，一般有 postgres、imos、cds 等等；

<2>client_addr：连接数据库的 ip 地址及子网掩码；

<3>state：当前连接的状态，一般有 active、idle 等等；

<4>count：相同 datname、ip、state 的连接个数。

5.6 当前查询语句

巡检示例：

```
----- checklist : sql in query -----
/*
procpid | start | lap | current_query
-----+-----+-----+-----
7182 | 2020-02-18 16:39:31.737104+08 | 19:18:46.35553 | SELECT r.rulename, trim(trailing ';' from pg_catalog.pg_get_ruledef(r.oid, true))
FROM pg_catalog.pg_rewrite r
WHERE r.ev_class = '11750' AND r.rulename != '_RETURN' ORDER BY 1;
13558 | 2020-02-19 11:58:18.092634+08 | 00:00:00 | SELECT procpid, start, now() - start AS lap, current_query
FROM (SELECT backendid, pg_stat_get_backend_pid(S.backendid) AS procpid,
pg_stat_get_backend_activity_start(S.backendid) AS start,
pg_stat_get_backend_activity(S.backendid) AS current_query
FROM (SELECT pg_stat_get_backend_idset() AS backendid) AS S
WHERE current_query <> 'IDLE' and current_query <> 'command string not enabled')
ORDER BY lap DESC;
(2 rows)
```

巡检分析：

<1>查看当前数据库中正在执行的 sql 语句。

5.7 长事务查询

巡检示例：

```
----- checklist : long transaction -----
/*
(time > 10 second) :
datname | username | pid | query | xact_start | xact_duration | query_start | query_duration | state
-----+-----+-----+-----+-----+-----+-----+-----+-----
(0 rows)

name | statement | prepare_time | duration | parameter_types | from_sql
-----+-----+-----+-----+-----+-----
(0 rows)
*/
***** RESULT(long transaction) : [ correct ] *****
```

巡检分析：

<1>查看当前数据库中长事务语句（执行时间>10s）。

☆自动化巡检----检查是否有执行时间过长的语句

5.8 数据库复制槽

巡检示例：

```

----- checklist : replication slot -----
/*
 slot_name | plugin | slot_type | datoid | database | temporary | active | active_pid | xmin
-----+-----+-----+-----+-----+-----+-----+-----+-----
(0 rows)
*/
***** RESULT(replication slot) : [ correct ] *****

```

巡检分析：

<1>查看当前数据库中复制槽的状态。

☆自动化巡检---复制槽状态错误时，进行提示

5.9 数据库历史版本号

巡检示例：

```

----- checklist : history version -----
/*
 autoid | install_flag | from_inner_version | to_inner_version | update_time
-----+-----+-----+-----+-----
1 | 1 | MPPV300R0012B01D139 | MPPV300R0012B01D139 | 2020-07-03 14:32:17
(1 row)
*/

```

巡检分析：

<1>查看数据库中的交付件升级版本号。

5.10 库表存储空间 top20

巡检示例：

```

----- checklist : table storage space -----
(top 20) :

```

relname	pg_size_pretty	reltuples
public.large_test	142 MB	2000000
public.tbl_mars2wgs	22 MB	297000
public.spatial_ref_sys	3304 kB	3911
public.tbl_event_typerel	472 kB	3506
public.tbl_dic	328 kB	2010
public.tbl_bank_alarm_type	160 kB	506
public.tbl_system_param	120 kB	275
public.tbl_operlog_2020_02_18_02	120 kB	72
public.tbl_operlog_2020_02_02_01	112 kB	4
public.tbl_operlog_2020_02_18_01	112 kB	4
public.tbl_operlog_2020_02_03_02	112 kB	4
public.tbl_operlog_2020_02_15_01	112 kB	4
public.tbl_operlog_2020_01_17_03	112 kB	24
public.tbl_operlog_2020_02_14_01	112 kB	4
public.tbl_operlog_2020_02_12_01	112 kB	4
public.tbl_operlog_2020_02_03_03	112 kB	4
public.tbl_operlog_2020_01_20_03	112 kB	4
public.tbl_operlog_2020_01_19_03	112 kB	16
public.tbl_operlog_2020_02_13_01	112 kB	4
public.tbl_operlog_2020_01_17_02	112 kB	24

```

(20 rows)

```

巡检分析：

<1>pg_size_pretty 数据库表占用的空间。若数据表空间过大,则需要联系数据库研发进行清理,否则会影响数据库的性能。

<2>reltuples 数据库表记录数。

5.11 分表情况

巡检示例：

```

----- checklist : sub table count -----

```

tbl_name	sub_table_num	min	max
tbl_event_record	2	tbl_event_record_2020_07_03_02	tbl_event_record_2020_07_03_03
tbl_operlog	2	tbl_operlog_2020_07_03_02	tbl_operlog_2020_07_03_03

```

(2 rows)

```

巡检分析：

<1>查看数据库中的分表数量，min 和 max 表示分表的范围（一般按时间分表）。

5.12 库表年龄 top20

巡检示例：

```
----- checklist : table age -----
(top 20) :
```

current_database	rolname	nspname	relkind	relname	age	age_remain
imos	postgres	pg_catalog	r	pg_type	17894	2147465754
imos	postgres	pg_catalog	r	pg_index	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_2619	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_1255	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_2604	17894	2147465754
imos	postgres	pg_catalog	r	pg_statistic	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_2618	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_2620	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_2609	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_2964	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_2396	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_3596	17894	2147465754
imos	postgres	pg_toast	t	pg_toast_2606	17894	2147465754
imos	postgres	pg_catalog	r	pg_authid	17894	2147465754
imos	postgres	pg_catalog	r	pg_attribute	17894	2147465754
imos	postgres	pg_catalog	r	pg_proc	17894	2147465754
imos	postgres	pg_catalog	r	pg_user_mapping	17894	2147465754
imos	postgres	pg_catalog	r	pg_attrdef	17894	2147465754
imos	postgres	pg_catalog	r	pg_constraint	17894	2147465754
imos	postgres	pg_catalog	r	pg_operator	17894	2147465754

```
(20 rows)
```

巡检分析：

<1>查看数据库表的年龄。

☆自动化巡检----库表年龄过高时，进行提示

5.13 库表约束与索引

巡检示例：

```
----- checklist : table index count -----
(top 10) :
```

current_database	relname	pg_size_pretty	idx_cnt
imos	tbl_gacase_info	0 bytes	6
imos	tbl_bank_park_info	0 bytes	5
imos	tbl_operlog_2020_02_19_02	8192 bytes	4
imos	tbl_operlog_2020_02_14_01	8192 bytes	4
imos	tbl_only_into_city_result	0 bytes	4

```
(5 rows)
```

巡检分析：

<1>idx_cnt 查看库表的索引个数。

5.14 库表索引利用率 top10

巡检示例：

```

----- checklist : table index utilization ----
(top 10) :
/*
+-----+-----+-----+-----+-----+-----+
| relname | indexrelname | idx_scan |
+-----+-----+-----+-----+-----+
tbl_device | idx_tbl_device_dev_code | 2487889171 |
tbl_share_res_info | idx_tbl_share_res_info_res_code | 259696711 |
tbl_device | idx_tbl_device_parent_dev_code | 197296310 |
tbl_user_belong | idx_tbl_user_belong_user_code | 169809112 |
tbl_share_res_info | idx_tbl_share_res_info_share_direction | 24113104 |
tbl_osd_name | idx_tbl_osd_name_dev_code_channel_index | 9591755 |
tbl_ptz_user_osd | idx_tbl_ptz_user_osd_dev_code_channel_index | 9589907 |
tbl_res_agg_unit | idx_tbl_res_agg_unit_agg_code | 6344839 |
tbl_bak_file_2020_05 | idx_tbl_bak_file_2020_05_bm_code_camera_code | 5280290 |
tbl_storage_plantime | idx_tbl_storage_plantime_storage_plan_id | 4643037 |
(10 rows)

```

巡检分析:

<1>idx_scan 查看库表的索引使用率。

5.15 库表索引占用空间 top10

巡检示例:

```

----- checklist : table index space -----
(top 10) :
/*
+-----+-----+-----+-----+-----+-----+
| idx_size | relname |
+-----+-----+-----+-----+-----+
1000 MB | idx_tbl_event_record_business_event_time_event_id | tbl
305 MB | idx_tbl_event_record_2020_02_20_02_dev_code | tbl
290 MB | pk_tbl_event_record_business_device | tbl
262 MB | pk_tbl_guobiao_vedio_stream | tbl
255 MB | idx_tbl_share_res_info_share_direction | tbl
240 MB | uk_tbl_share_res_info_inter_code_share_domain_code_index | tbl
240 MB | uk_tbl_share_res_info_res_code_share_domain_code_index | tbl
224 MB | idx_tbl_event_record_2020_02_20_02_event_time_event_id | tbl
194 MB | idx_tbl_operlog_2020_02_20_02_service_type | tbl
193 MB | idx_tbl_operlog_2020_02_20_02_oper_type | tbl
(10 rows)

```

巡检分析:

<1>idx_size 查看索引所占空间的大小。

5.16 库表索引膨胀 top10

巡检示例:


```

----- checklist : table index bloat -----
(top 10) :
  db |      tablename      |  tups | pages | otta | tbloat |
-----+-----+-----+-----+-----+-----+
imos | tbl_share_info      | 965968 | 15084 | 13712 | 1.1 |
imos | tbl_plantime_templet | 0 | 0 | 0 | 0.0 |
imos | tbl_bank_envacquire_type | 6 | 1 | 1 | 0.0 |
imos | tbl_job_history      | 0 | 0 | 0 | 0.0 |
imos | tbl_bak_plan         | 0 | 0 | 0 | 0.0 |
imos | tbl_bak_plan         | 0 | 0 | 0 | 0.0 |
imos | tbl_bank_park_camera | 0 | 0 | 0 | 0.0 |
imos | tbl_bank_envacquire_type | 6 | 1 | 1 | 0.0 |
imos | tbl_bank_plate_black | 0 | 0 | 0 | 0.0 |
imos | tbl_bank_park_camera | 0 | 0 | 0 | 0.0 |
(10 rows)

```

巡检分析：

<1>tbloat 查看索引的膨胀倍数，若数值过大，则需要进行清理。

5.17 库表膨胀 top10

巡检示例：

```

----- checklist : table bloat -----
(top 10) :
  db |      tablename      |  tups | pages | otta | tbloat |
-----+-----+-----+-----+-----+
imos | tbl_event_record_business | 13060807 | 278143 | 262112 | 1.1 |
imos | tbl_event_record_business | 13060807 | 278143 | 262112 | 1.1 |
imos | tbl_operlog_2020_02_20_02 | 6459247 | 306703 | 294034 | 1.0 |
imos | tbl_operlog_2020_02_20_02 | 6459247 | 306703 | 294034 | 1.0 |
imos | tbl_operlog_2020_02_20_02 | 6459247 | 306703 | 294034 | 1.0 |
imos | tbl_operlog_2020_02_20_02 | 6459247 | 306703 | 294034 | 1.0 |
imos | tbl_camera_status_reports | 3936795 | 117249 | 109838 | 1.1 |
imos | tbl_role_auth          | 1878549 | 98871 | 92871 | 1.1 |
imos | tbl_role_auth          | 1878549 | 98871 | 92871 | 1.1 |
imos | tbl_res                 | 2140011 | 96899 | 91132 | 1.1 |
(10 rows)

```

巡检分析：

<1>tbloat 查看库表的膨胀倍数，若数值过大，则需要进行清理。

5.18 数据库的锁情况

巡检示例：

```

----- checklist : table lock -----
/*
(all locks) :
locktype | mode | granted | datname |
-----+-----+-----+-----+
relation | AccessShareLock | t | imos | pg_locks
virtualxid | ExclusiveLock | t | imos | 
relation | AccessShareLock | t | imos | idx_tbl_o
relation | AccessShareLock | t | imos | idx_tbl_o
relation | AccessShareLock | t | imos | pk_tbl_op
virtualxid | ExclusiveLock | t | imos | 
relation | AccessShareLock | t | imos | pg_databa
relation | AccessShareLock | t | imos | pg_databa
relation | ShareUpdateExclusiveLock | t | imos | tbl_operl
relation | AccessShareLock | t | imos | pg_databa
(11 rows)

(dead locks) :
datname | pid | usesysid | username | application_name | client_addr |
-----+-----+-----+-----+-----+-----+
(0 rows)

*/
***** RESULT(table lock) : [ correct ] *****

```

巡检分析：

<1>all locks：查看数据库的所有锁的占用情况；

<2>dead locks：查看数据库当前的死锁情况，正常情况此结果为空。

☆自动化巡检----检测到存在锁等待时会进行提示

5.19 数据库查询命中率

巡检示例：

```

----- checklist : table hit ratio -----
/*
datname | rollback_ratio | hit_ratio | blk_read_time |
-----+-----+-----+-----+
imos | 0.00 % | 44.47 % | 0 |
(1 row)

```

巡检分析：

<1>查看数据库的查询命中率，hit_ratio 越接近 100%越好，

rollback_ratio 越接近 0%越好。

5.20 数据库垃圾

巡检示例：

```

----- checklist : table junk data -----
/*
current_database | relname | n_dead_tup
-----+-----+-----
imos | tbl_report_period_task | 12
imos | tbl_str_res_mount | 9
imos | tbl_itc_sys_param | 6
(3 rows)

```

巡检分析：

<1>查看数据库垃圾，如果垃圾过多，需要进行清理；

<2>n_dead_tup 表示数据死亡行数。

5.21 tbl_operlog_inner 分表创建失败问题

巡检示例：

```

----- checklist : tbl_operlog_inner -----
***** RESULT(operlog_id) : [ wrong ] *****

```

巡检分析：

<1>查看 tbl_operlog_inner 表的 operlog_id 字段属性。

☆自动化巡检----字段值缺少默认值实，进行提示

异常处理：

<1>需要安装 B3337P02H385 补丁。

6. HA & N+1 检查

6.1 交付件类型、版本、补丁号

巡检示例：

```

----- checklist : ha basic info -----
HA or N+1 : HA
Version : IMOS110-B3329P15
Patch : HA30-IMOS110-B3329.129.15.H02.200110

```

巡检分析：

<1>查看交付件版本及补丁是否按需求安装。

6.2 hosts 配置

巡检示例：

```
----- checklist : hosts config -----  
/etc/hosts :  
127.0.0.1    localhost localhost.localdomain local  
207.207.35.135 vm135  
207.207.35.134 vm134
```

巡检分析：

<1>配置中需要填写主备机的业务实 ip 地址和服务器名。

6.3 ha_hosts 配置

巡检示例：

```
----- checklist : ha_hosts config -----  
/usr/local/hcha/hosts :  
172.16.0.5   vm135  
172.16.0.4   vm134
```

巡检分析：

<1>配置中需要填写主备机的心跳实 ip 地址和服务器名。

6.4 Corosync 配置

巡检示例：

```
----- checklist : corosync config -----  
/etc/corosync/corosync.conf :  
compatibility: whitetank  
  
totem {  
  version: 2  
  secauth: off  
  threads: 0  
  rrp_mode: passive  
  interface {  
    ringnumber: 0  
    member {  
      memberaddr: 207.207.35.135  
    }  
    member {  
      memberaddr: 207.207.35.134  
    }  
    bindnetaddr: 207.207.35.135  
    mcastaddr: 224.224.35.135  
    mcastport: 2061  
  }  
  interface {  
    ringnumber: 1  
    member {  
      memberaddr: 172.16.0.5  
    }  
    member {  
      memberaddr: 172.16.0.4  
    }  
    bindnetaddr: 172.16.0.5  
    mcastaddr: 224.224.35.134  
    mcastport: 2062  
  }  
}
```

巡检分析：

<1>memberaddr 后需要依次对应配置为主备机的业务和心跳实 ip。

6.5 network 配置

巡检示例：

```
----- checklist : network config -----  
/etc/sysconfig/network :  
NETWORKING=yes  
NETWORKING_IPV6=no  
HOSTNAME=vm135  
GATEWAY=207.207.35.1  
METRIC=50
```

巡检分析：

<1>查看是否配置服务器的网关。

6.6 Pacemaker 配置

巡检示例：

```
----- checklist : crm config -----  
crm configure show :  
node vm134 \  
  attributes postgresql-data-status=DISCONNECT standby=off  
node vm135 \  
  attributes postgresql-data-status=LATEST standby=off  
primitive DBIPH IPaddr2 \  
  params ip=172.16.0.6 nic="eth2:0" cidr_netmask=24 \  
--  
primitive DBIPM IPaddr2 \  
  params ip=207.207.35.145 nic="eth1:0" cidr_netmask=24 \  
--  
primitive imos imosstart \  
  op start interval=0 timeout=600 \  
--  
primitive p_ping ocf:pacemaker:ping \  
  params multiplier=100 dampen=0 timeout=1 attempts=30 host_list=207.207.35.1 \  
--  
primitive postgresql pgsql \  
  params node_list="vm135 vm134" master_ip=172.16.0.6 pgport=5432 pgdata="/mnt/syncdata/pgsql/data" \  
--
```

巡检分析：

<1>ip 地址需要填写业务和心跳虚 ip，并与网卡对应。

6.7 postgresql.conf 配置

巡检示例：

```
----- checklist : postgresql config -----  
postgresql.conf :  
#archive_mode = off # allows archiving to be done  
#archive_command = '' # command to use to archi  
wal_level = hot_standby  
hot_standby = on  
max_wal_senders = 2  
wal_keep_segments = 256
```

巡检分析：

<1>实际环境配置需要与上图保持一致。

6.8 pg_hba.conf 配置

巡检示例：

```
----- checklist : pg_hba config -----
pg_hba.conf :
local  all  all  md5
host   all  all  127.0.0.1/32  md5
host   all  all  ::1/128  md5
host   all  all  0.0.0.0/0  md5
host   replication  repl  172.16.0.5/24  trust
host   replication  repl  172.16.0.4/24  trust
```

巡检分析：

<1>双机时，红框中需要包含所有的心跳实地址；

<2>N+1 时，红框中需要包含所有的业务实地址。

6.9 /mnt/syncdata 共享分区

巡检示例：

```
ACTIVE      '/dev/share/data' [40.00 GiB] inherit
```

使用 drbd 模块且/mnt/syncdata 目录未挂载

```
----- checklist : pg data folder -----
DB Data Dir : mounted
Filesystem    Type    Size  Used Avail Use% Mounted on
/dev/sda2     ext3    481G  715M  456G   1% /mnt/syncdata
```

/mnt/syncdata 目录已挂载

巡检分析：

<1>查看 Type 列（文件系统类型），目前 ext4 性能要优于 ext3；

<2>查看 Avail（可用空间）和 Use%（使用百分比），检查分区剩余空间是否充足。

6.10 Corosync 进程

巡检示例：

```

----- checklist : corosync process -----
Corosync Version : 2.4.3
Pacemaker Version : 1.1.18
Corosync Process : online
68685 ?      Ssl  492:51 corosync
68853 ?      Ss   2:08 /usr/sbin/pacemakerd -f
68861 ?      Ss   70:33 \_ /usr/libexec/pacemaker/cib
68862 ?      Ss   2:09 \_ /usr/libexec/pacemaker/stonithd
68863 ?      Ss   6:48 \_ /usr/libexec/pacemaker/lrmd
109306 ?     S    0:00 | \_ /bin/sh /usr/lib/ocf/resource
109325 ?     S    0:00 | \_ /bin/sh /usr/lib/ocf/resource
68864 ?      Ss   2:52 \_ /usr/libexec/pacemaker/attrd
68865 ?      Ss   4:11 \_ /usr/libexec/pacemaker/pengine
68866 ?      Ss   13:30 \_ /usr/libexec/pacemaker/crmd
***** RESULT(corosync fork): [ correct ] *****

```

CentOS 7 系统

```

----- checklist : corosync process -----
Corosync Version : 1.4.9
Pacemaker Version :
Corosync Process : online
24990 ?      Ssl  0:13 corosync
24995 ?      S<   0:55 \_ /usr/libexec/pacemaker/cib
24996 ?      S<   0:00 \_ /usr/libexec/pacemaker/stonithd
24997 ?      S<   0:02 \_ /usr/libexec/pacemaker/lrmd
8591 ?      S    0:00 | \_ /bin/sh /usr/lib/ocf/resource
8605 ?      S    0:00 | \_ /bin/sh /usr/lib/ocf/resource
24998 ?      S<   0:00 \_ /usr/libexec/pacemaker/attrd
24999 ?      S<   0:02 \_ /usr/libexec/pacemaker/pengine
25000 ?      S<   0:03 \_ /usr/libexec/pacemaker/crmd
***** RESULT(corosync fork): [ correct ] *****

```

CentOS 6 系统

巡检分析：

<1>查看 corosync 和 pacemaker 间的父子进程关系。

☆自动化巡检---检查父子进程关系是否正确

6.11 集群服务状态

巡检示例：

```

-----
HA服务运行状态  :
Stack: corosync
Current DC: vm22 (version 1.1.18-11.el7-2b07d5c5a9) - partition with
Last updated: Thu Mar  7 09:55:19 2019
Last change: Tue Mar  5 15:04:29 2019 by root via cibadmin on vm22

2 nodes configured
7 resources configured

Online: [ vm22 vm40 ] ①

Active resources:

Resource Group: postgres
  DBIPM      (ocf::heartbeat:IPaddr2): ② Started vm22
  imos       (ocf::heartbeat:imosstart): Started vm22
  DBIPH      (ocf::heartbeat:IPaddr2): Started vm22
Master/Slave Set: msPostgresql [postgresql]
  Masters: [ vm22 ] ③
  Slaves: [ vm40 ]
Clone Set: cl_ping [p_ping]
  Started: [ vm22 vm40 ]

Node Attributes:
* Node vm22:
  + master-postgresql ④ : 2000
  + pingd              : 100
  + postgresql-data-status : LATEST
* Node vm40:
  + master-postgresql   : 100
  + pingd               : 100
  + postgresql-data-status : STREAMING|ASYNC
-----

```

巡检分析：

<1>节点在线状态：online 表示服务器 corosync 服务在线；

<2>业务运行状态：started vm22 表示业务在 vm22 上运行；

<3>数据库主备状态：master 表示数据库是主状态；

<4>数据库同步状态：LATEST 表示数据库是主状态；

STREAMING|ASYNC 表示数据库是同步状态。

6.12 双机证书检测

巡检示例：

```
----- checklist : ha license -----
HA License File : exist
License Check Log :
2020-02-20 13:46:00 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
2020-02-20 13:56:03 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
2020-02-20 14:06:07 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
2020-02-20 14:16:11 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
2020-02-20 14:26:14 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
2020-02-20 14:36:18 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
2020-02-20 14:46:21 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
2020-02-20 14:56:25 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
2020-02-20 15:06:29 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
2020-02-20 15:16:32 [INFO (0)] [haserver] [] [] [68926.0xde829740] [bp_license.cpp-601] [BP_InitLicense] license initiate success.
```

巡检分析：

<1>查看证书文件是否存在；

<2>检查导入的证书是否有效。

6.13 数据库同步状态

巡检示例：

```
-----
数据同步方式 : drbd
数据同步状态 :
drbd driver loaded OK; device status:
version: 8.4.1 (api:1/proto:86-100)
GIT-hash: 91b4c048c1a0e06777b5f65d312b38d47abaae80 build by dag@Build32R6, 2011-12-21 06:07:17
m:res cs ro ds p mounted fstype
0:postgres WFFConnection Primary/Secondary UpToDate/UpToDate C /mnt/syncdata ext3

16053 ? S 0:00 \ [drbd_w_postgres]
16058 ? S 0:22 \ [drbd_r_postgres]
22570 ? S 0:04 /home/postgres/pgsql/bin/postmaster -p 5432 -D /mnt/syncdata/pgsql/
22572 ? Ss 0:11 \ postgres: logger process
22574 ? Ss 0:53 \ postgres: writer process
22575 ? Ss 0:41 \ postgres: wal writer process
22576 ? Ss 0:20 \ postgres: autovacuum launcher process
22577 ? Ss 0:22 \ postgres: stats collector process
22906 ? Ss 0:00 \ postgres: postgres imos 207.207.35.139(59602) idle
22907 ? Ss 0:00 \ postgres: postgres imos 207.207.35.139(59603) idle
22910 ? Ss 0:00 \ postgres: postgres imos 207.207.35.139(59604) idle
22913 ? Ss 0:00 \ postgres: postgres imos 207.207.35.139(59605) idle
22914 ? Ss 0:00 \ postgres: postgres imos 207.207.35.139(59606) idle
22916 ? Ss 0:00 \ postgres: postgres imos 207.207.35.139(59607) idle
22920 ? Ss 0:00 \ postgres: postgres imos 207.207.35.139(59608) idle
-----
```

B3329 之前版本

```

----- checklist : data sync mode -----
DB Sync Method : rsync
Status of Sync :
4319 ?      S      0:00      \_ su postgres -c cd /mnt/syncdata/pgsql/data; /home/postgres/pgsql/
4329 ?      Ss     0:00          \_ bash -c cd /mnt/syncdata/pgsql/data; /home/postgres/pgsql/bin
4330 ?      S      0:00              \_ /home/postgres/pgsql/bin/postmaster -p 5432 -D /mnt/syncd
4361 ?      Ss     0:00                  \_ postgres: logger process
4363 ?      Ss     0:00                  \_ postgres: checkpoint process
4364 ?      Ss     0:00                  \_ postgres: writer process
4365 ?      Ss     0:00                  \_ postgres: stats collector process
5154 ?      Ss     0:00                  \_ postgres: wal writer process
5155 ?      Ss     0:00                  \_ postgres: autovacuum launcher process
6051 ?      Ss     0:00                  \_ postgres: postgres imos 207.207.35.200(59707) idle
6053 ?      Ss     0:00                  \_ postgres: postgres imos 207.207.35.200(59708) idle
6054 ?      Ss     0:00                  \_ postgres: postgres imos 207.207.35.200(59709) idle
6055 ?      Ss     0:00                  \_ postgres: postgres imos 207.207.35.200(59710) idle
6056 ?      Ss     0:00                  \_ postgres: postgres imos 207.207.35.200(59711) idle
6057 ?      Ss     0:00                  \_ postgres: postgres imos 207.207.35.200(59712) idle

```

B3329 之后版本

巡检分析：

- <1>drbd 同步：查看红框中的状态，主机是 primary，备机是 secondary；uptodate 表示数据同步；
- <2>rsync 流复制：检查红框中的进程是否都存在。

6.14 历史切换记录

巡检分析：

```

----- checklist : switch record -----
/var/log/cluster/hasvrn.log :
2020-02-19 11:02:54 [INFO] [0] [0] [] [ocfimossv] [] [] [0] all services is started
2020-02-19 11:05:43 [INFO] [0] [0] [] [ocfimossv] [] [] [0] all services is started
2020-02-19 11:48:03 [INFO] [0] [0] [] [ocfimossv] [] [] [0] all services is started
2020-02-19 11:55:57 [INFO] [0] [0] [] [ocfimossv] [] [] [0] all services is started

```

- <1>如果最近有切换记录，则会筛选出日志中的切换记录。可根据其信息定位切换原因。

6.15 每日日志打包

巡检分析：

```
----- checklist : log daily gzip -----
last 10 days :
-rw-rw---- 1 hacluster haclient 40K Sep  3 03:45 /var/log/cluster/corosync.log-20190903.gz
-rw-rw---- 1 hacluster haclient 48K Sep  1 03:26 /var/log/cluster/corosync.log-20190901.gz
-rw-rw---- 1 hacluster haclient 159K Aug 31 03:09 /var/log/cluster/corosync.log-20190831.gz
-rw-rw---- 1 hacluster haclient 42K Aug 30 03:44 /var/log/cluster/corosync.log-20190830.gz
-rw-rw---- 1 hacluster haclient 62K Aug 29 03:33 /var/log/cluster/corosync.log-20190829.gz
-rw-rw---- 1 hacluster haclient 68K Aug 24 03:22 /var/log/cluster/corosync.log-20190824.gz
-rw-rw---- 1 hacluster haclient 141K May  7 03:35 /var/log/cluster/corosync.log-20190507.gz
-rw-r----- 1 root root 1.7M Sep  3 14:05 /var/log/messages
-rw-r--r-- 1 root root 41 Aug 16 17:51 /var/log/messages_20190816
-rw-r--r-- 1 root root 0 Aug 16 17:36 /var/log/messages_20190815
```

<1>检查日志的打包时间，是否每天都进行日志打包；

<2>检查日志大小，是否过大，例如超过 1G。

7. 业务状态检查 (VM)

7.1 交付件版本 & 补丁号

巡检示例：

```
----- checklist : vm basic info -----
System Ver      : 64 bit
VM Install      : VM9500
Interior version : VM9500V300R005B01D139
Exterior version : VMPS5.0-B3339.139.0.200622
BUILDTIME       : 2020-06-22 18:22
VM9500 Patch    : not installed
```

巡检分析：

<1>查看交付件版本信息，及补丁实施情况；

<2>注意检查系统位数与交付件版本是否匹配，例如 64 位系统安装

8500。

7.2 业务进程检查

巡检示例：

```
----- checklist : vm service -----
===== VM8500 STATUS =====
Service[postgresql] [ Running ]
Service[img] [ Running ]
Service[mcserver] [ Running ]
Service[sgserver] [ Running ]
Service[vmserver] [ Running ]
Service[smart_community_sync] [ Running ]
Service[stunserver] [ Running ]
Service[adapter] [ Running ]
Service[onvifserver] [ Running ]
Service[g_ga_service] [ Running ]
Service[httpd] [ Running ]
Service[DiskReadOnlyCheck] [ Running ]
Service[viidserver] [ Running ]
Service[impserver] [ Running ]
Service[pagserver] [ Running ]
Service[itcserver] [ Running ]
Service[paggbserver] [ Running ]
Service[nmserver] [ Running ]
Service[rptserver] [ Running ]
Service[unpservice] [ Running ]
Service[ga_electricvehicle] [ Running ]
Service[iscloud] [ Running ]
Service[kbserver] [ Running ]
Service[nweb] [ Running ]
Service[vmdaemon] [ Running ]
```

巡检分析：

<1>查看业务进程是否在运行。

7.3 VM 授权识别

巡检示例：

```
----- checklist : vm license -----
License Check Log :
2020-07-03 14:40:05:580711 [ERROR (1118)] [vmserver] [ ] [ ] [9756.0x7f6496f8e720] [bp_license.cpp-660] [BP_InitLicense]fail to parse license fi
2020-07-03 14:40:05:580744 [WARNING (1118)] [vmserver] [AS] [AS_CM] [ ] [9756.0x7f6496f8e720] [as_cm_license.cpp-291] Call bp func to init licens
2020-07-03 14:40:05:580757 [ERROR (1118)] [vmserver] [ ] [ ] [9756.0x7f6496f8e720] [bp_license.cpp-933] [BP_GetLicenseInfo]license init fail.
2020-07-03 14:40:05:580764 [ERROR (1118)] [vmserver] [ ] [ ] [9756.0x7f6496f8e720] [bp_license.cpp-933] [BP_GetLicenseInfo]license init fail.
2020-07-03 16:26:23:135596 [INFO (0)] [vmserver] [ ] [ ] [26470.0x7fc6d1dae720] [gbspt_srv.cpp-438] Check license flag:1.
2020-07-03 16:26:23:399580 [ERROR (1)] [vmserver] [ ] [ ] [26470.0x7fc6d1dae720] [bp_license_mgr.cpp-2126] [CLicenseBuild::parseLicenseFile]fail
2020-07-03 16:26:23:399621 [ERROR (1118)] [vmserver] [ ] [ ] [26470.0x7fc6d1dae720] [bp_license.cpp-660] [BP_InitLicense]fail to parse license f
2020-07-03 16:26:23:399654 [WARNING (1118)] [vmserver] [AS] [AS_CM] [ ] [26470.0x7fc6d1dae720] [as_cm_license.cpp-291] Call bp func to init licens
2020-07-03 16:26:23:399667 [ERROR (1118)] [vmserver] [ ] [ ] [26470.0x7fc6d1dae720] [bp_license.cpp-933] [BP_GetLicenseInfo]license init fail.
2020-07-03 16:26:23:399675 [ERROR (1118)] [vmserver] [ ] [ ] [26470.0x7fc6d1dae720] [bp_license.cpp-933] [BP_GetLicenseInfo]license init fail.
```

巡检分析：

<1>查看 vm 日志，过滤授权识别相关内容。

7.4 VM 消息队列

巡检示例：

```

----- checklist : vm message queue -----
----- bp rmod -----
/*
***** RESULT(bp rmod:7) : [ correct ] *****

----- bp msgq -----
/*
***** RESULT(bp msgq) : [ correct ] *****

----- img conn -----
/*
***** RESULT(img conn:1) : [ correct ] *****

```

巡检分析：

<1>查看 bp rmod 模块数，若超过 1000，则可能有异常，需进一步检查；

<2>查看 bp msgq 中 MaxMsgNum 列数值，若大于 30000，则表示当前压力较大；

<3>查看 img conn 中 PID=100 的连接数，若超过 500，则可能有异常。

☆自动化巡检---超过标准值会提示，并列超项

7.5 数据库定时备份

巡检示例：

```

----- checklist : pg daily backup -----
Timing Task for PG Data Backup :
00 3 * * * root /usr/local/bin/dbbk.sh

Daily Backup Files (last 7) :
-rw-r--r-- 1 root root 2.1M Feb 13 03:00 /var/autobackup/database-2020-02-13_0300.tar.gz
-rw-r--r-- 1 root root 2.1M Feb 14 03:00 /var/autobackup/database-2020-02-14_0300.tar.gz
-rw-r--r-- 1 root root 2.1M Feb 15 03:00 /var/autobackup/database-2020-02-15_0300.tar.gz
-rw-r--r-- 1 root root 2.1M Feb 16 03:00 /var/autobackup/database-2020-02-16_0300.tar.gz
-rw-r--r-- 1 root root 2.1M Feb 17 03:00 /var/autobackup/database-2020-02-17_0300.tar.gz

Check Backup File :
/var/autobackup/database-2020-02-17_0300.tar.gz
LineNo 20 -- DataBase: imos
LineNo 368086 -- Version : B3329
LineNo 382413 -- PostgreSQL database dump complete

```

巡检分析：

<1>Timing Task：数据库定时备份任务，检查配置是否正确；

<2>Files of The Last: 检查目录下时间最近的数据库备份文件，检查文件日期及大小；

<3>Check Backup File:

version: B3332 表示数据库对应交付件是 B3332 版本的；

PostgreSQL database dump complete 表示数据库备份完整。

7.6 pagserver 服务中 IPCNum 检测

巡检示例：

```
----- checklist : pagserver IPCNum -----
pagserver IPCNum config:
IPCNum_max_config( A ) = 1000
IPCNum_in_reality( B ) = 1001
difference value( A-B ) = -1

A < B , need increase MaxIPCNum in pagsyscfg.ini
***** RESULT(pagserver IPCNum) : [ wrong ] *****
```

```
----- checklist : pagserver IPCNum -----
pagserver IPCNum config:
IPCNum_max_config( A ) = 1000
IPCNum_in_reality( B ) = 0
difference value( A-B ) = 1000

( A-B ) > 500 , maybe need decrease MaxIPCNum in pagsyscfg.ini
***** RESULT(pagserver IPCNum) : [ wrong ] *****
```

巡检分析：

<1> IPCNum_max_config: pagsyscfg.ini 配置文件中设置的最大值

MaxIPCNum；

<2> IPCNum_in_reality: 从数据库中查询到的实际的 IPCNum；

<3>如果实际值大于配置值，则需要增大配置值；如果实际值远小于配置值（差值大于 500），则可以适当减小配置值，降低控制块个数，节省内存。

☆自动化巡检----超过标准值会提示，并列出超标项

8. 系统日志收集

8.1 last 信息

巡检示例：

```
----- checklist : boot record -----
Boot Record (last 20) :
reboot system boot 2.6.32-220.el6.x Sat Jun 22 14:17 - 17:17 (26+02:59)
reboot system boot 2.6.32-220.el6.x Wed May 22 13:57 - 17:17 (57+03:19)
reboot system boot 2.6.32-220.el6.x Tue May 14 16:28 - 13:55 (7+21:26)
reboot system boot 2.6.32-220.el6.x Fri May 10 10:21 - 16:26 (4+06:05)
reboot system boot 2.6.32-220.el6.x Wed May 8 17:17 - 10:19 (1+17:01)
reboot system boot 2.6.32-220.el6.x Sat May 4 19:47 - 17:16 (3+21:29)
reboot system boot 2.6.32-220.el6.x Sat May 4 03:08 - 17:16 (4+14:08)
reboot system boot 2.6.32-220.el6.x Sat May 4 01:58 - 03:06 (01:08)
reboot system boot 2.6.32-220.el6.x Thu May 2 20:47 - 01:56 (1+05:08)
reboot system boot 2.6.32-220.el6.x Sun Apr 28 01:43 - 01:56 (6+00:12)
reboot system boot 2.6.32-220.el6.x Sun Apr 28 01:13 - 01:56 (6+00:43)
reboot system boot 2.6.32-220.el6.x Sun Apr 28 01:08 - 01:09 (00:01)
reboot system boot 2.6.32-220.el6.x Tue Jul 2 04:27 - 00:19 (-65+-4:-8)
reboot system boot 2.6.32-220.el6.x Tue Jul 2 04:03 - 00:19 (-65+-3:-44)
reboot system boot 2.6.32-220.el6.x Thu Jun 27 21:42 - 00:19 (-60+-21:-23)
reboot system boot 2.6.32-220.el6.x Sat May 25 00:19 - 00:19 (-27+00:00)
reboot system boot 2.6.32-220.el6.x Fri Mar 22 14:30 - 00:19 (63+09:48)
```

巡检分析：

<1>查看系统重启记录信息。

8.2 Messages 日志

(1) 重启信息

巡检示例：

```
-----
messages日志：重启记录
Jan 23 17:14:04 vm135 kernel: BIOS-provided physical RAM map:
Jan 23 17:14:19 vm135 Monitor: system start type: [HOT] start!
Jan 23 17:19:41 vm135 kernel: BIOS-provided physical RAM map:
Jan 23 17:19:58 vm135 Monitor: system start type: [HOT] start!
Feb 13 08:09:48 vm135 kernel: BIOS-provided physical RAM map:
Feb 13 08:10:00 vm135 Monitor: system start type: [COLD] start!
Feb 18 14:55:28 vm135 kernel: BIOS-provided physical RAM map:
Feb 18 14:55:42 vm135 Monitor: system start type: [COLD] start!
Feb 20 15:43:40 vm135 kernel: BIOS-provided physical RAM map:
Feb 20 15:43:48 vm135 Monitor: system start type: [HOT] start!
-----
```

巡检分析：

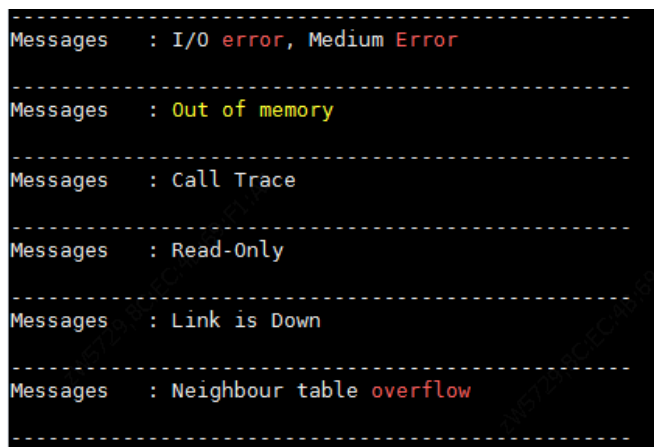
<1>查看系统重启时间和重启模式。

注：[HOT] start：服务器不断电并进行系统重启；

[COLD] start：服务器断电，然后重新开机。

(2) 报错关键信息

巡检示例：



```
-----
Messages : I/O error, Medium Error
-----
Messages : Out of memory
-----
Messages : Call Trace
-----
Messages : Read-Only
-----
Messages : Link is Down
-----
Messages : Neighbour table overflow
-----
```

巡检分析：

<1>I/O Error 或 Medium Error

常见问题：硬本地盘故障导致服务异常

建议措施：更换服务器硬盘

注：I/O error 或 medium error 后面跟的设备号必须是本机磁盘才行

（BM 会挂载 IPSAN 盘符，例如 sdaj 之类，当网络异常也会打印 I/O error，要区分开）

<2>Out of Memory

常见问题：内存不足引发 oom killer 导致服务被杀掉

建议措施：部署 pidstat，检查是否存在内存泄露

<3>Call Trace

常见问题：进程崩溃

建议措施：开启 coredump，收集 core 文件分析进程崩溃原因

<4>Read-Only

常见问题：异常断电、异常重启、硬盘故障导致文件系统只读，引发问题

建议措施：本地硬盘如无 I/O Error 或 Medium Error，重装系统

<5>Link is Down

常见问题：网卡或网线闪断导致无法登录

建议措施：更换网线或网口

<6>Neighbour Table Overflow

常见问题：ARP 邻表溢出

建议措施：使用命令 `ip -4 -s ntable show` 查看实际参数值，检查子网掩

码是否设置不合理，导致二层网络 ip 数量过多，调整 `gc_thresh` 相关参数

8.3 Sar 日志

巡检示例：

```

sar日志
Linux 3.10.0-514.el7.x86_64 (vm22) 03/11/2019 _x86_64_ (8 CPU)

```

		%usr	%nice	%sys	%iowait	%steal	%irq	%soft	%guest	%gnice	%idle
12:00:01 AM	CPU	14.98	0.00	5.23	9.36	0.00	0.00	0.18	0.00	0.00	70.26
12:10:01 AM	all	17.24	0.00	6.25	12.93	0.00	0.00	0.06	0.00	0.00	63.52
12:10:01 AM	0	16.42	0.00	5.96	11.33	0.00	0.00	0.07	0.00	0.00	66.22
12:10:01 AM	1	15.52	0.00	6.13	8.17	0.00	0.00	0.06	0.00	0.00	70.11
12:10:01 AM	2	16.47	0.00	6.70	19.01	0.00	0.00	0.22	0.00	0.00	57.60
12:10:01 AM	3	11.93	0.00	4.22	6.05	0.00	0.00	0.04	0.00	0.00	77.76
12:10:01 AM	4	12.21	0.00	4.21	4.98	0.00	0.00	0.04	0.00	0.00	78.57
12:10:01 AM	5	18.75	0.00	3.85	6.25	0.00	0.00	0.90	0.00	0.00	70.25
12:10:01 AM	6	11.32	0.00	4.48	6.18	0.00	0.00	0.04	0.00	0.00	77.98

CPU 状态监测

12:00:01 AM	kbmemfree	kbmemused	%memused	kbbuffers	kbcached	kbcommit	%commit	kbactive	kbinact	kbdirty
12:10:01 AM	179632	15954708	98.89	9480	2898280	45530388	91.63	12561992	2224012	12056
12:20:02 AM	169224	15965116	98.95	5944	2953700	45522124	91.61	12556560	2260140	12220
12:30:01 AM	210068	15924272	98.70	8988	2911516	45509120	91.59	12532496	2239132	9116
12:40:02 AM	230424	15903916	98.57	7276	2904624	45525484	91.62	12511696	2243432	6156
12:50:01 AM	183828	15950512	98.86	5240	2974148	45500392	91.57	12538560	2269208	23036
01:00:01 AM	241252	15893088	98.50	9172	2917556	45490548	91.55	12493076	2254404	12572
01:10:01 AM	210160	15924180	98.70	7236	2968752	45486428	91.54	12518036	2266764	11860
01:20:02 AM	178408	15955932	98.89	4800	3005192	45524860	91.62	12526868	2287524	13632
01:30:01 AM	194592	15939748	98.79	10192	2974864	45520416	91.61	12518624	2270368	13120

内存状态监测

巡检分析：

<1>%idle 列，即 CPU 空闲时间百分比。数值越高代表压力越小；

<2>kbcommit：保证当前系统运行所需要的内存（RAM+swap）；

<3>%commit 列，即所需内存百分比。数值持续高于 100%则需要加大

内存；

<4>pswpin/s：每秒系统换入的交换页面（swap page）数量；

pswpout/s：每秒系统换出的交换页面（swap page）数量。如果交互页面非常频繁(大于 1000)，则说明内存不足；

<5>pgsteal/s：每秒钟从 cache 中被清除来满足内存需要的页个数。如果该列数据较大（大于 1000），则说明内存不足。

9. 服务器补丁及工具部署检查

9.1 通用看门狗补丁

巡检示例：

```
----- checklist : watchdog -----
Watchdog Patch : installed
```

巡检分析：

<1>检查是否安装看门狗补丁

9.2 Coredump

巡检示例：

```
----- checklist : coredump -----
Coredump Status : on
Core File Path : /root/core_dump-%e-%p-%t-%s-%h
Coredump config :
# /etc/security/limits.conf

#<domain>      <type>  <item>      <value>
#
#*              soft    core        0
#*              hard    rss         10000
#@student       hard    nproc       20
#@faculty       soft    nproc       20
#@faculty       hard    nproc       50
#ftp            hard    nproc       0
#@student       -        maxlogins   4

# End of file
```

巡检分析：

<1>检查 coredump 是否开启，和 core 文件生成路径和相关配置

9.3 Tcpdump

巡检示例：

```
-----  
Tcpdump Install : installed  
version: 2018-04-04  
/var/vm_msg : 1 M  
[ running ]  
tcpdump 15040 S 8800 /var/vm_msg/2019-04-10-16-57-14_snmp.cap  
root 15760 15739 vm_msg.sh defunct  
-----
```

Tcpdump 运行状态

```
-----  
Tcpdump Install : installed  
version: 2018-04-04  
/var/vm_msg : 1 M  
[ stopped ]  
-----
```

Tcpdump 停止状态

巡检分析：

<1>检查背景抓包是否安装，是否开启，检查抓包文件的占用空间

注：若背景抓包一直开启，则有可能耗尽磁盘分区空间



视无界 智以恒