CN_VNC_有关VNC连接异常断开后导致系统运行负荷高的解决方法

Exported from Confluence on 2024 January 26

We reserve the right to change the content of this document without prior notice. The information contained herein is believed to be accurate as of the date of export, however, B&R makes no warranty, expressed or implied, with regards to the information contained within this document. B&R shall not be liable in the event if incidental or consequential damages in connection with or arising from the use of this information. The software names, hardware names and trademarks used in this document are registered by the respective companies.

BnR Generic pdf Template Version 2

Table of Contents

问题表现 3

· 问题表现

问题

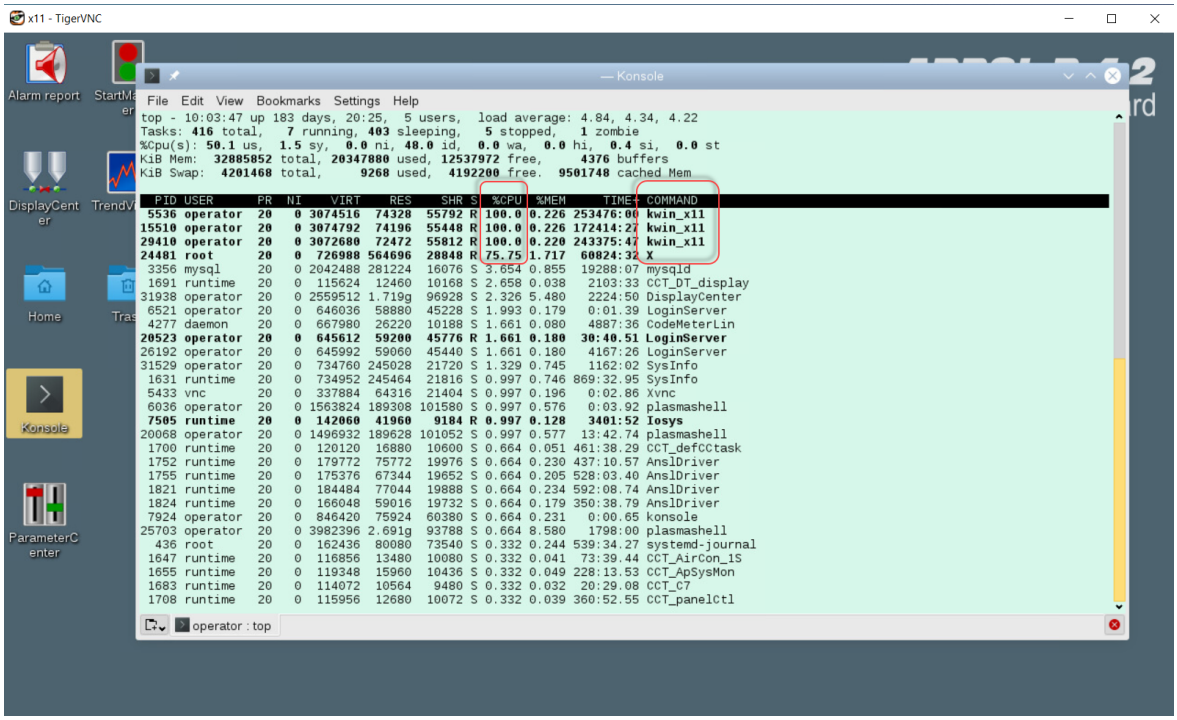
在APROL R4.2 版本，如果有经常使用 VNC 方式连接 APROL 系统的项目情况下，VNC 连接异常退出时，如由于网络突然中断，后续恢复 VNC 连接后，可能会遇到系统负荷一直非常高的情况，个别进程甚至高达系统负荷的100%。目前收集的信息看这些占负荷非常高的进程主要有 kwin_x11、LoginServer 之间的其中之一或者两者都有。当然，并不是说 VNC 异常中断后一定会出现个别进程负荷异常高的情况，只是当出现这种情况时，占用过高的系统负荷会导致系统运行的不稳定或者卡顿，VNC 连接自动断开等，因此需要进行针对性的处理。

问题的诊断以及解决应对方法

对于出现的问题，首先需要通过相关命令查看症状是否一致，然后再根据具体的情况采用不同的解决方案。下面就具体的排查和解决方案逐一描述。

1. 诊断方式

对于 VNC 异常中断后，如发现系统存在明显的卡顿现象时，需要通过使用 top 命令去确认一下是否存在有 kwin_x11 或者 LoginServer 进程占用负荷异常高的情况。方法就是在 VNC 连接的当前系统，打开一个 konsole ，然后输入命令 top 并回车来查看相应系统负荷占用的情况。如下图所示，主要关注 %CPU 和 COMMAND 两列的值，通过这个可以看到相应的进程名称以及所占的负荷。从下图我们可以看到有三个 kwin_x11 的进程占 CPU 负荷都高达100%，这个是不正常的。通过 Ctrl + Z 可以结束 top 命令的执行。



2. 通用解决方法

在通过 **top** 命令发现相关负荷异常的进程后，就需要通过相应的命令去把异常的进程给处理掉，如采用相应的 **kill** 命令去杀死这些异常的进程，然后重启系统能解决这个问题。

但是，由于这种现象具有的偶发性，用户可能并不具备对相应的故障的诊断能力，因此还是建议采用定期检测这些异常进程然后进行相应的处理。

3. kwin_x11异常处理

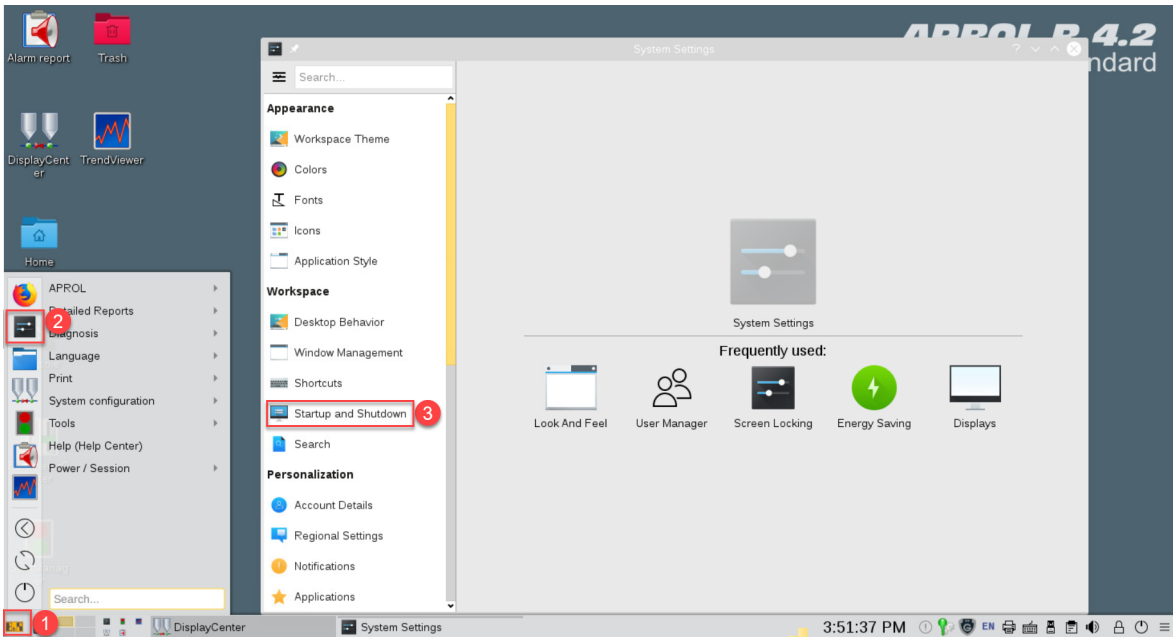
对于异常的 **kwin_x11** 进程，可以通过创建一个 **sh** 脚本来检测并处理，具体的脚本代码如下图所示。

```
# Global inputs
cpu_limit="10"
proc_name="kwin_x11"

# Reading out the kwin_x11 PIDs
Procs=$(ps -C "$proc_name" -o pid=,pcpu= | awk --assign maxcpu="$cpu_limit" '$2>maxcpu {print $1 ";"}')

# Killing processes
IFS=';'
for Proc in $Procs; do
    echo $Proc
    kill -9 $Proc
done
unset IFS
```

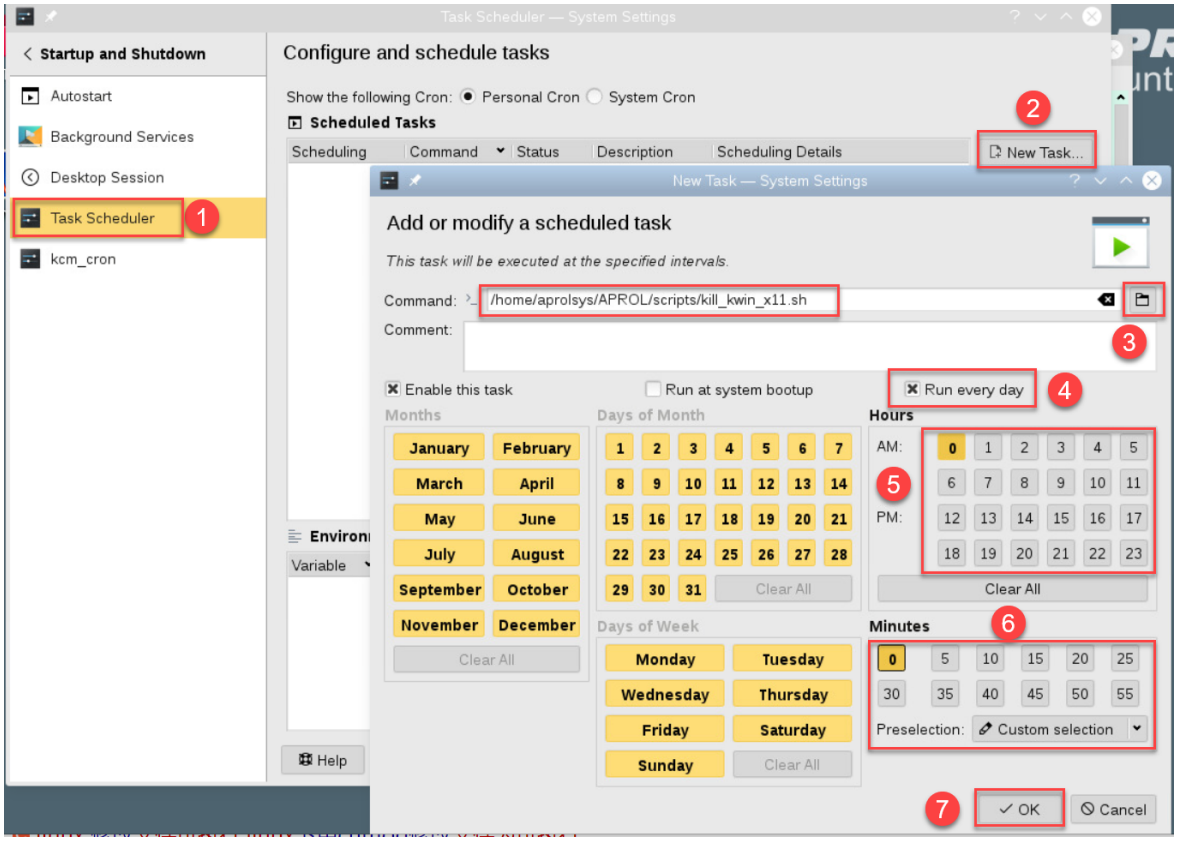
上图代码中 cpu_limit 的值可以设置为一个合适的限值，现在代码里设置为10，即检测 **kwin_x11** 的进程所占系统负荷是否大于 10%，大于这个限值后就去杀死这个超过限值的 **kwin_x11** 进程。因此这个值可以根据实际的项目应用加以适当的调整。另外，这个创建的脚本请保存在 **VNC** 连接的计算机的 **/home/aprolsys/APROL/scripts** 里面，脚本名称可以命名为 **kill_kwin_x11.sh** 。相应的 **kill_kwin_x11.sh** 可咨询贝加莱工程师获取。

对于上一步创建的 **kill_kwin_x11.sh** 脚本，需要设置定时去执行，一般建议每天或者设置更短的时间去定期执行。具体的设置步骤如下，下面以连接的 **runtime** 系统为例，其余同理。

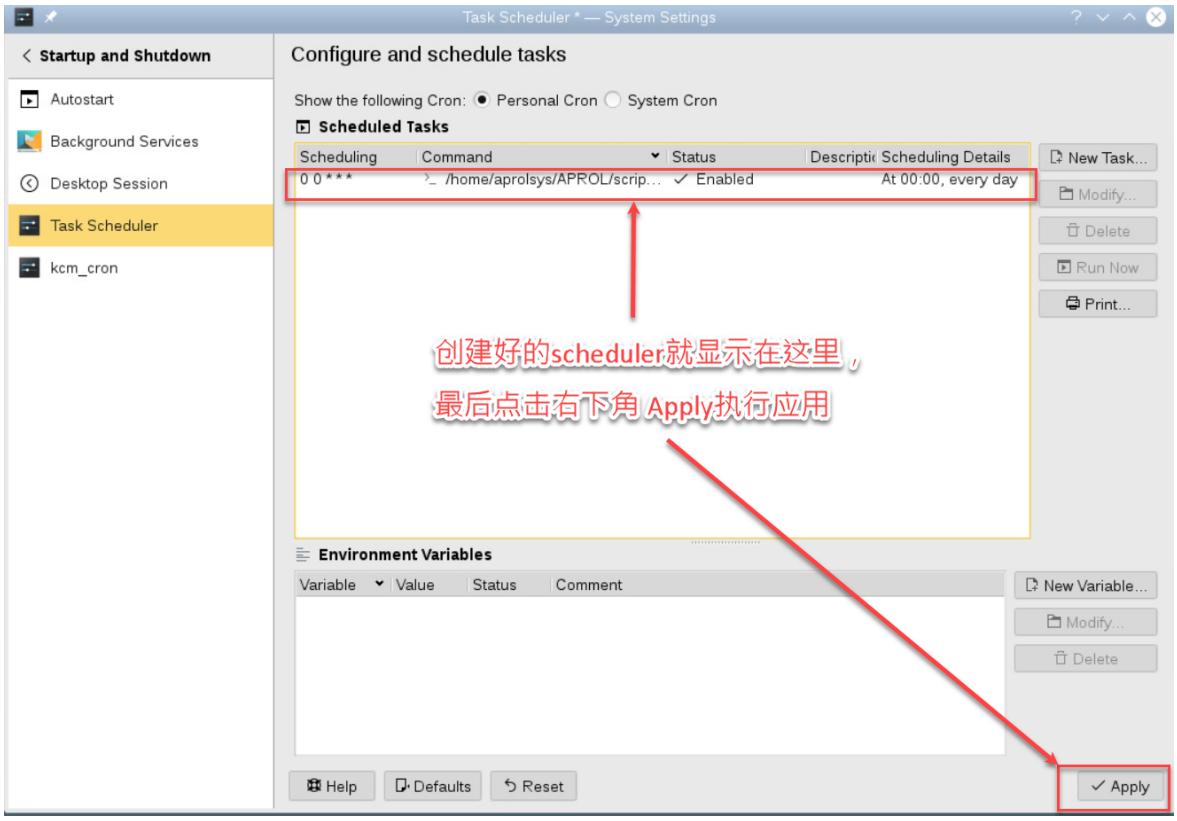
首先如下图去打开 **systemsettings** 设置，选择 **Startup and Shutdown** 。



再如图所示在 **Task Scheduler** 里面去创建一个新的任务，选择上面创建的 **kill_kwin_x11.sh** 脚本，并进行相应的执行时间设置，注意需要确保此处设定的时间系统是在运行的。



最后如图把新创建的任务进行应用即可。



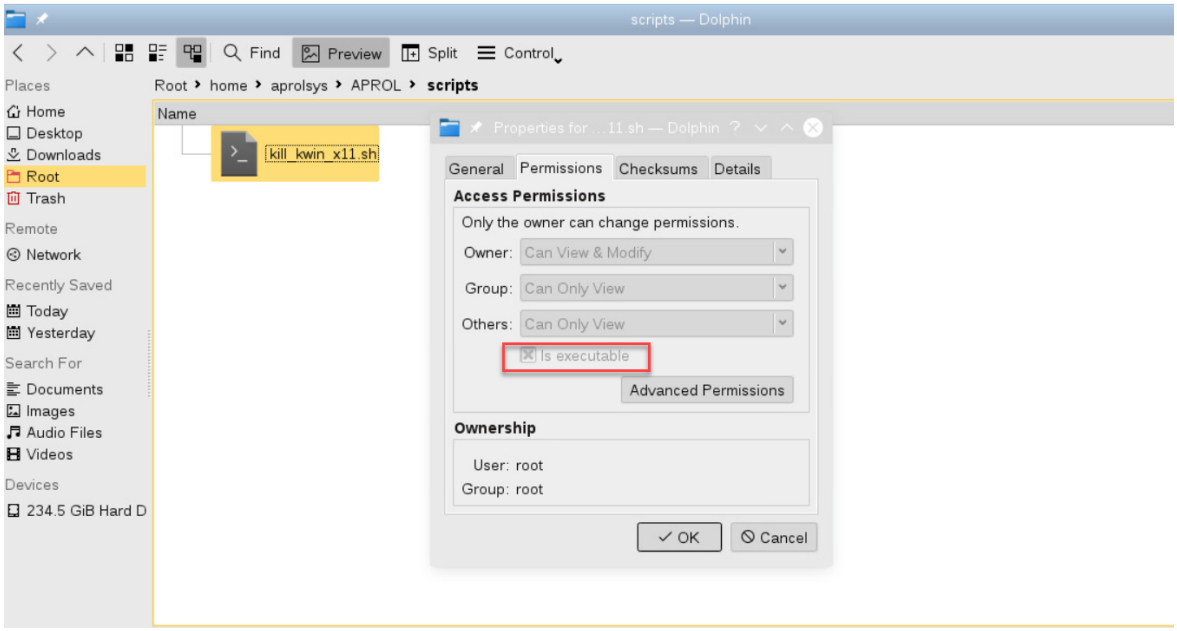
4. LoginServer异常处理

如果 VNC 异常断开后，通过 top 命令查询发现是 LoginServer 进程所占系统负荷过高，那么就要对 LoginServer 这个进程做相应的处理。请注意，LoginServer 这个进程在系统运行有操作用户登陆情况时是一直运行的，因此 LoginServer 进程在系统正常运行情况下的通常负荷是多少需要正确评估，以便设置合适的负荷的限值。考虑到 VNC 异常断开可能有 kwin_x11 或者 LoginServer 这两个进程占负荷异常的情况，因此建议在一个脚本里面把两种情况都进行检测并处理，当然也可以分别创建不用的 sh 脚本来处理。

下图的代码实例同时考虑了 kwin_x11 和 LoginServer 这两个进程的处理，这里主要需要设置合适的 cpu_limit 的限值。

```
#!/bin/bash
cpu_limit=15

# kwin_x11 search
pids=$(ps -C kwin_x11 -o pid=,pcpu= | awk --assign maxcpu="$cpu_limit" '$2>maxcpu {print $1}')

for pid in $pids
do
echo "killing $pid ..."
kill -9 $pid
done

# LoginServer search
pids=$(ps -C LoginServer -o pid=,pcpu= | awk –assign  maxcpu="$cpu_limit" '$2>maxcpu {print $1}')

for pid in $pids
do
echo "killing $pid ..."
kill -9 $pid
done
```

如果处理 LoginServer 是创建了单独的 sh 脚本，那么也同样需要对这个 sh 脚本去设置定时执行，具体步骤如上一步对 kwin_x11 的处理时相同的步骤，这里就不再赘述了。

5. 注意事项

对于前面章节里创建的 **sh** 脚本，需要检查并确定这个脚本文件是可执行的，如下图所示，这个脚本的属性里面 **“Is executable”** 应该是勾选的。如果没有勾选，可以通过命令 **chmod +x** 去增加执行权限或者直接登陆 **root** 系统去直接修改。