ALERT: Disable Transparent HugePages on SLES11, RHEL6, OEL6 and UEK2 Kernels (Doc ID 1557478.1)

Modified: Jun 8, 2013 Type: ALERT Status: PUBLISHED Priority: 3

In this Document

APPLIES TO:

Oracle Database - Enterprise Edition
Linux x86
Linux x86-64

DESCRIPTION

Starting with RedHat 6, OEL 6, SLES 11 and UEK2 kernels, Transparent HugePages are implemented and enabled (default) in an attempt to improve the memory management. Transparent HugePages are similar to the HugePages that have been available in previous Linux releases. The main difference is that the Transparent HugePages are set up dynamically at run time by the khugepaged thread in kernel while the regular HugePages had to be preallocated at the boot up time.

Because Transparent HugePages are known to cause unexpected node reboots and performance problems with RAC, Oracle strongly advises to disable the use of Transparent HugePages. In addition, Transparent Hugepages may cause problems even in a single-instance database environment with unexpected performance problems or delays. As such, Oracle recommends disabling Transparent HugePages on all Database servers running Oracle.

Note: This ONLY applies to the new feature Transparent HugePages, Oracle highly recommends the use of standard HugePages that were recommended for previous releases of Linux. See Document 361323.1 for additional information on HugePages.

OCCURRENCE

Starting with SLES11, RHEL 6, OEL 6 and UEK2 Kernels, Transparent HugePages are introduced and enabled by default, and this can cause node reboots and performance problems.

SYMPTOMS

Node reboots.

The ocssd.log may show some of the threads are blocked (but this does not show all the time):

```
2013-05-01 14:30:45.255: [    CSSD][224204544]clssscMonitorThreads clssnmvKillBlockThread not scheduled for 7500 msecs
2013-05-01 14:30:46.945: [    CSSD][224204544]clssscMonitorThreads clssnmvWorkerThread not scheduled for 8030 msecs
```

WORKAROUND

To check if the Transparent HugePages are enabled in your server execute the following:

Default/Enabled setting is [always]:

```
# cat /sys/kernel/mm/transparent_hugepage/enabled
[always] never
```

Disabled setting is [never]:

```
# cat /sys/kernel/mm/transparent_hugepage/enabled
always [never]
```

If "enabled" is **NOT** set to "[never]", the Transparent HugePages are being used.

You can also issue:

```
# grep AnonHugePages /proc/meminfo
```

If the output contains a line like "AnonHugepages: xxxx kB", with a value > 0kB the kernel is using Transparent HugePages.

Because the kernel currently uses Transparent HugePages only for the anonymous memory blocks like stack and heap, the value of AnonHugepages in /proc/meminfo is the current amount of Transparent HugePages that the kernel is using.

To disable Transparent HugePages boot time either one of the following 2 methods may be used:

Add the following to the kernel boot line in /etc/grub.conf (this is the preferred method) and reboot the server:

```
transparent_hugepage=never
```

Once modified the line will read similar to the following example:

```
title Oracle Linux Server (2.6.32-300.25.1.el6uek.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-300.25.1.el6uek.x86_64 ro root=LABEL=/ transparent_hugepage=never
        initrd /initramfs-2.6.32-300.25.1.el6uek.x86_64.img
```

OR

Add the following lines in /etc/rc.local and reboot the server:

```
if test -f /sys/kernel/mm/transparent_hugepage/enabled; then
   echo never > /sys/kernel/mm/transparent_hugepage/enabled
fi
if test -f /sys/kernel/mm/transparent_hugepage/defrag; then
   echo never > /sys/kernel/mm/transparent_hugepage/defrag
fi
```

PATCHES

none

HISTORY

06/05/2013 the alert is written
06/05/2013 the alert is published
06/06/2013 removed "cat /sys/kernel/mm/transparent_hugepage/enabled" from the section where we check the transparent hugepages are disabled.
06/06/2013 changed "Starting SLES11..." to 'Starting with SLES11"
06/07/2013 Changed title to include "ALERT", modified to cover SIDB, corrected typo in 2nd solution, expanded the solution for disable via kernel boot line and validation via AnonHugePages.