Oracle Linux

Simultaneous Multithreading Notice

F24364-06

October 2022

Notice Description

Simultaneous multithreading (SMT) is a technique for improving the efficiency of processor workload on modern processors by allowing multiple independent threads to be executed on a single physical core at the same time. SMT has various implementations, including Intel's Hyper-threading feature, but all implementations are effectively similar.

SMT is generally positive for performance for many applications and can improve the efficiency of most systems. For this reason, SMT is often enabled in the kernel by default, where hardware allows for it. In some cases, particularly where an application executes threads that become a bottleneck and the architecture of the CPU does not handle this particularly well, performance can be diminished and you may wish to disable SMT.

More importantly, recent security issues specific to microprocessor architecture and functionality such as speculative processing may be better mitigated against by disabling SMT as well as applying the appropriate patching. The following CVEs are typical of the types of issues where disabling SMT may be desirable:

- CVE-2018-3646
- CVE-2018-3620
- CVE-2018-12130
- CVE-2018-12126
- CVE-2018-12127
- CVE-2019-11091

You should consider disabling SMT if your system may run untrusted code or allows user input for code that may not adequately check and sanitize input values.

Action Items

SMT can be disabled in a variety of ways. In general, the options that are available depend on the hardware or the running kernel that you are using. Recommended approaches to disabling SMT are described in each of the sections below. You should consider disabling SMT if your system may run untrusted code or allows user input for code that may not adequately check and sanitize input values.

Disable SMT in system firmware

Since SMT is a hardware feature, the most practical way to disable it for any system is to disable it in the system firmware. This is usually done at boot by accessing the system's BIOS or UEFI to set an option there. You should refer to your system vendor documentation for more information on how to do this.

If SMT is disabled in the firmware, the operating system is unable to enable this function.

It is not always practical to disable options within the firmware and doing so requires manual intervention at boot, that may not offer flexibility in a large data center. Furthermore, if someone enables this option again in the firmware you may not detect the change.

Although disabling SMT in the firmware is the recommended approach, you may wish to configure your kernel boot options to disable SMT even if it is re-enabled in the firmware, or simply to avoid having to hard set this option within the firmware. See Set kernel boot parameter to disable SMT.

Set kernel boot parameter to disable SMT

Most current kernel releases support the `nosmt` boot option that can be applied as a boot argument at boot. This option disables SMT within the kernel. It is not possible to override this option in a running system by attempting to switch SMT on as described in Disable SMT at Runtime.

You can apply the `nosmt` option to the kernel boot configuration for the default kernel using the `grubby` command.

```
# grubby --args=nosmt --update-kernel=DEFAULT
```

This change only updates the default kernel, but the change is usually also applied to subsequent kernel updates on Oracle Linux 6 and Oracle Linux 7. On Oracle Linux 8 the change only updates the current default kernel and is not applied to subsequent kernel updates.

The use of `grubby` to set this parameter is preferred, but if there is a possibility that the `grub2-mkconfig` might be used on the system, any parameters set by `grubby` may be overwritten. To prevent this, ensure that the `nosmt` argument is applied to the `GRUB_CMDLINE_LINUX` line in `/etc/sysconfig/grub`.

Note that the system must be rebooted for these changes to take effect.

Disable SMT at Runtime

The kernel may provide options to control SMT at runtime, so that you can disable SMT without requiring a reboot. Note that doing this does not disable SMT permanently and it will be enabled again when you reboot the system. If you use this option, ensure that you also choose to disable the option in your firmware at next boot, or you configure your kernel boot parameters to disable SMT so that this option is triggered at boot.

It is also worth bearing in mind that re-enabling SMT on a live system once it has been disabled is not recommended. If you re-enable SMT, consider a reboot as mandatory for safe and predictable behavior on your system.

You can check the current runtime status for SMT by running:

```
# cat /sys/devices/system/cpu/smt/control
```

The command may return any of the following possible states:

- **on**

  SMT is supported by the CPU and enabled. All logical CPUs can be onlined and offlined without restrictions. You can set this as an option by writing this value, but it is ignored if the CPU does not support SMT; SMT is disabled in the system firmware; or the `nosmt` kernel boot option is set.

- **off**

  SMT is supported by the CPU and disabled. Only the so called primary SMT threads can be onlined and offlined without restrictions. An attempt to online a non-primary sibling is rejected. You can set this as an option by writing this value.

- **forceoff**

  Same as 'off' but the state cannot be controlled. Attempts to write to the control file are rejected once this state is set. You can set this status by writing this value to the control file, but it is typically set to this value when SMT is disabled in the kernel at boot. To unset this value, a reboot is required and you may need to check that the `nosmt` kernel boot argument is unset.

- **notsupported**

  The processor does not support SMT. It's therefore not affected by the SMT implications of L1TF. Attempts to write to the control file are rejected. It is not possible to set this as an option.

To set one of these options at runtime, enter the following command:

```
# echo off > /sys/devices/system/cpu/smt/control
```