Kernel Tuning Parameters

This document describes how to do boot-time configuration for an ultra-low latency, highly deterministic real-time system running on Linux.

The following parameters must be set at boot time by GRUB (or whatever bootloader floats your boat.

CPU Selection

Each application should be affine to a single physical NUMA node. That is to say that a given application should have all its pinned threads and crucial resources attached to a specific group of CPU cores that are on the same physical socket.

CPU cores are (almost always; there are no exceptions I know of) numbered from 0 to n-1, where n is the TOTAL NUMBER of CPU cores in a system.

isolcpus

Isolates the specified CPU cores. This is a comma-separated range list of CPUs to isolate.

For example, to isolate CPU cores 5-7, you can do something like:

```
isolcpus=5,6,7
```

nmi_watchdog

To eliminate spurious interrupts of a core, the NMI watchdog should be disabled.

To do this, specify the following in your kernel command line:

```
nmi_watchdog=0
```

nohz_full

When CONFIG*NO*HZ_FULL is enabled in your kernel, use this option to specify the CPU cores that should never have local timer events scheduled on them.

For example, to set nohz_full on cores 5-7, you can do something like:

```
nohz_full=5,6,7
```

Note that you should see that these CPU cores have been tagged for dynticks in dmesg at boot time as follows:

```
NO_HZ: Full dynticks CPUs: 5-7
```

This requires kernel 3.10 or later.

rcu_nocbs

To remove a specific CPU core from the RCU callback eligible CPU core list, specify a the `rcu_nocbs` item on the kernel command line.

NOTE: RCU is the kernel's read-copy-update synchronization mechanism.

For example, to set RCU subsystem to delegate RCU callbacks on cores 5-7 to a thread running on another core, you can do something like:

```
rcu_nocbs=5,6,7
```

This requires kernel 3.10 or later.

intel_idle.max_cstate

If Cstates might be enabled due to a buggy BIOS or similar, sets the Idle Loop for Intel CPUs to ensure that a given CPU core does not enter a low- power state. Not strictly needed, but can help.

```
intel_idle.max_cstate=0
```

processor.max_cstate

In case the generic idle loop takes over, sets the processor's maximum CState to full power. Not strictly needed, but can help.

```
processor.max_cstate=0
```

idle

Set the idle loop to poll rather than wait for an IPI to wake up. Not strictly needed.

```
idle=poll
```

Example GRUB Config Line for Debian

On Debian Linux, using the above assumptions for system topology, the following configuration could be added to `/etc/default/grub`. Be sure to re-run `update-grub` after making changes to this file.

```
GRUB_CMDLINE_LINUX="initrd=/install/initrd.gz isolcpus=5,6,7 nmi_watchdo
g=0 nohz_full=5,6,7 rcu_nocbs=5,6,7 intel_idle.max_cstate=0 processor.ma
x_cstate=0 idle=poll"
```