Isolate CPUs from Kernel Scheduler

Disable CPU(s)

Sysfs

```
# echo 0 > /sys/devices/system/cpu/cpu4/online
```

> When disabling a CPU this way any processes already assigned to this core will keep working but no new work will be assigned.

Kernel Parameter

> `isolcpus` — Isolate CPUs from the kernel scheduler.
>
> Synopsis `isolcpus= cpu_number [, cpu_number ,...]`
>
> Description Remove the specified CPUs, as defined by the cpu_number values, from the general kernel SMP balancing and scheduler algroithms. The only way to move a process onto or off an "isolated" CPU is via the CPU affinity syscalls. cpu_number begins at 0, so the maximum value is 1 less than the number of CPUs on the system.
>
> This option is the preferred way to isolate CPUs. The alternative, manually setting the CPU mask of all tasks in the system, can cause problems and suboptimal load balancer performance.

Use Isolated CPU(s)

`taskset [options] -p [mask] pid`

> taskset is used to set or retrieve the CPU affinity of a running process given its PID or to launch a new COMMAND with a given CPU affinity. CPU affinity is a scheduler property that "bonds" a process to a given set of CPUs on the system. The Linux scheduler will honor the given CPU affinity and the process will not run on any other CPUs. Note that the Linux scheduler also supports natural CPU affinity: the scheduler attempts to keep processes on the same CPU as long as practical for performance reasons. Therefore, forcing a specific CPU affinity is useful only in certain applications.

Revision #1
Created 2 April 2021 13:37:32 by dustin@swigg.net
Updated 2 April 2021 14:03:32 by dustin@swigg.net