ORACLE VM

An Oracle White Paper
October 2013

Tuning the SPARC CPU to Optimize Workload Performance

Tuning the SPARC CPU to Optimize Workload Performance

Important note: the tuning method described in this whitepaper is supported in Oracle VM Server for SPARC but is deprecated, as described in Oracle VM Server for SPARC 3.1 Release Notes. By default, domains are created for maximum throughput, and the Oracle Solaris OS automatically uses the critical thread API to optimize for single-thread workloads. Oracle recommends using the critical thread API only if explicit control is needed. This document should be used primarily for historical reference.

Starting with Oracle VM Server for SPARC 2.1 patch release (147507-xx), you can use dynamic CPU threading controls to optimize workload performance on SPARC T4 and later systems that support Oracle VM Server for SPARC. These threading controls enable you to specify the number of hardware threads to be activated per core, thereby ensuring exclusive access to hardware resources like CPU cache for application software threads running on those cores. In some cases this can provide performance benefits for selected applications. Existing applications can take advantage of the SPARC dynamic threading performance benefits without having to be rewritten or recompiled. It should be noted that Oracle Solaris will normally tune the system for highest performance automatically without administrator intervention, and provides a flexible, effective way to designate which application processes should be given preferential access to CPU resources

This paper describes how to use the Oracle VM Server for SPARC CPU threading controls to optimize CPU performance on SPARC T4 and subsequent platforms. CPU performance can be optimized for CPU-bound workloads by tuning CPU cores to maximize the number of instructions per cycle (IPC). Or, CPU performance can be optimized for maximum throughput by tuning CPU cores to use a maximum number of CPU threads. By default, the CPU is tuned for maximum throughput.

This paper covers the following topics:

- “CPU Threading Modes and Workloads”
- “Setting the CPU Threading Mode”
- “Threading Control Limitations”

CPU Threading Modes and Workloads

On SPARC T4 and later systems, you can optimize CPU performance by specifying the CPU threading mode. In most cases Solaris automatically makes the right resource assignments without any need for administrative action. Additionally, if a particular CPU-intensive application needs to be given preferential access to a CPU core's resources, the Solaris resource manager can be told to attempt to do so by giving the application threads a priority higher than 59.

If that is not sufficient for critical applications, based on performance measurements of metrics like cache misses, the CPU threading model can be used to ensure dedicated access to core resources. The use of this method should be based on measured performance and understanding application behavior. Depending on the workload, and the resource optimizations performed automatically by Solaris, there may or may not be a measurable performance benefit. The threading mode can be set dynamically and independently for each domain on the system. A reboot is not required to change the threading mode, and the set mode is persistent across domain reboots or platform power cycles. Note that you *must* perform an initial reboot to properly configure the system.

By selecting the appropriate CPU threading mode, you can improve the performance of applications and workloads that are running on a domain. You can select a threading mode that either maximizes throughput or maximizes the number of instructions per cycle, as follows:

- **Maximizing for throughput (`max-throughput`).** Workloads that benefit most from high throughput run multiple concurrent software threads. These workloads may be CPU-intensive, but spread their CPU processing over multiple executable threads. As a result, the "instructions per clock" of an individual thread has less importance than the total number of instructions executed. When you optimize for maximum throughput, you enable CPU cores to concurrently run a maximum number of hardware threads. This mode is best for running heavily threaded workloads, such as those performed by Java application servers, web servers, database servers, and file servers, or multiple independent applications in the same Solaris instance. This mode is used by default and is also used on older SPARC T-series platforms, such as SPARC T3 platforms.
- **Maximizing for IPC (`max-ipc`).** Workloads that benefit most from high IPC are CPU intensive, with a small number of dispatchable threads or processes. This is typical in systems that run intensive arithmetic computations. When you optimize for maximum IPC, you enable a CPU thread to execute more instructions per CPU cycle because it has exclusive ownership of per-core resources like level-1 cache. This optimization is achieved by reducing the number of CPU threads that are concurrently active on the same CPU core.

Selecting the CPU Threading Mode

Select the CPU threading mode of a domain by using the `ldm add-domain` or `ldm set-domain` command to set the `threading` property.

```
ldm add-domain [mac-addr=num] [hostid=num]
 [failure-policy=ignore|panic|reset|stop]
 [extended-mapin-space=on] [master=master-ldom1,...,master-ldom4]
 [threading=max-throughput|max-ipc] ldom


ldm set-domain [mac-addr=num] [hostid=num]
 [failure-policy=ignore|panic|reset|stop]
 [extended-mapin-space=[on|off]]
 [master=[master-ldom1,...,master-ldom4]]
 [threading=max-throughput|max-ipc] ldom
```

The `threading` property is used to dynamically change the threading mode by specifying one of the following values:

- **`max-throughput`**. Use this value to select the threading mode that maximizes throughput. This mode activates all threads that are assigned to the domain. This mode is used by default and is also selected if you do not specify any mode (`threading=`).
- **`max-ipc`**. Use this value to select the threading mode that maximizes the number of instructions per cycle (IPC). When you use this mode on the SPARC T4 platform, only one thread is active for each CPU core that is assigned to the domain. Selecting this mode requires that the domain is configured with the whole-core constraint.

Use the `ldm add-core` or `ldm set-core` command to configure the whole-core constraint. See the `ldm` (1M) man page.

Note that changing the threading mode dynamically activates or deactivates CPU threads. So, the number of virtual CPUs that are available in the domain also dynamically changes.

The `max-ipc` threading mode leverages the whole-core constraint, so you must abide by the whole-core constraint requirements and restrictions to do the following:

- Change the number of cores that are allocated to a domain.
- Enable or disable the whole-core constraint.

Thus, to dynamically change the threading mode of a running domain to `max-ipc` mode, you must configure the domain with the whole-core constraint.

For information about the restrictions, see "Threading Control Limitations" on page 7. For more information about the `add-domain` and `set-domain` subcommands, see the `ldm`(1M) man page.

Viewing the `threading` Property Value

You can use the following commands to view the `threading` property value:

- The `ldm list -o resmgmt` command shows the constraints. The following example output shows that the `threading` property is set to `max-ipc`:

```
# ldm list -o resmgmt ldg1
NAME
ldg1
CONSTRAINT
whole-core
max-cores=3
threading=max-ipc
```

- The `ldm list -o cpu` command shows the deactivated virtual CPUs by specifying a value of `0` in the `UTIL` column. The bold text in the following `max-ipc` example shows that only one thread is activated per CPU:

```
# ldm list -o cpu ldg1
NAME
ldg1
VCPU
VID PID CID UTIL STRAND
0   8   1   0.3% 100%
1   9   1   0    100%
2   10  1   0    100%
3   11  1   0    100%
4   12  1   0    100%
5   13  1   0    100%
6   14  1   0    100%
7   15  1   0    100%
8   24  2   0.4% 100%
...
```

- The `ldm list -l` command includes all the information about the specified domain. The bold text in the following example shows that the `threading` property is set to `max-ipc`:

```
# ldm list -l ldg1
...
VID PID CID UTIL STRAND
0   8   1   0.6% 100%
1   9   1   0    100%
2   10  1   0    100%
3   11  1   0    100%
4   12  1   0    100%
5   13  1   0    100%
6   14  1   0    100%
...
CONSTRAINT
whole-core
max-cores=3
threading=max-ipc
...
```

Threading Control Limitations

The threading controls feature has the following restrictions:

- The whole-core constraint restrictions apply. See "Allocating CPUs" in *Oracle VM Server for SPARC Administration Guide*.
- The `threading` property value does not persist across a domain migration.
- The `threading` property cannot be set to `max-ipc` while Power Management (PM) is enabled. When PM runs, all domains must have the `threading` property set to `max-throughput`.

Conclusion

The Oracle VM Server for SPARC software enables you to use dynamic CPU threading controls to optimize workload performance on SPARC systems. You can optimize CPU performance for CPU-bound workloads or for high throughput workloads. By default, the CPU is tuned for maximum throughput. The method described in this whitepaper can be used to control this setting, but in most cases Oracle recommends using Solaris priority settings instead.

For more information about Oracle's virtualization, visit www.oracle.com/virtualization.

Tuning the SPARC CPU to Optimize Workload Performance
October 2013
Author: Cathleen Reiher
Contributing Authors: Alexandre Chartre, Rory Foster, Menno Lageman, Jorge Osario, Jeff Savit

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com