Hewlett Packard Enterprise

HPE Superdome Flex Server Performance Tuning Guide

Abstract

Performance tools, guidelines, and optimizations for Superdome Flex Server.

Part Number: XXXX-XXXX
Published: July 2019
Edition: 1

Notices

Acknowledgments

Revision history

Part number	Publication date	Edition	Summary of changes
XXXXX-XXXXX	TBD 2019	1	First edition.

Contents

HPE Superdome Flex Server performance tuning

AUTHOR NOTE: Topics, submap from SD Flex Administration Guide.

Managing system performance with HPE Foundation Software (HFS)

HPE Foundation Software (HFS) includes automatic boot-time optimization utilities, reliability features, and technical support tools. Designed for high performance computing, these tools help maximize HPE Superdome Flex Server system performance and availability.

NOTE: HFS is required on HPE Superdome Flex Server running Linux.

To install HFS, see **HPE Superdome Flex Server OS Installation Guide**.

Monitoring main memory

REVIEWERS: This topic needs updates to provide statements about using MEMlog in combination with DCPMMs (Optane persistent memory). Details are TBD, per engineering work. Also, please confirm if this information is still required with the RAS recommendation as memlog is disabled when RAS is enabled.

The MEMlog utility monitors the overall system health of each dual inline memory module (DIMM) on your system. The MEMlog utility is configured for your system when the HPE Foundation Software (HFS) package is installed. After the HFS installation, `memlog` starts automatically and uses BIOS settings to determine the operational memory mode. To determine the operational memory mode, enter the following command:

```
RMC cli> show npar verbose

This system is nPartition capable

Partitions: 3

Partition 1:
  Run State          : OS Boot
  Health Status      : OK
  Chassis OK/In      : 2/2
  CPUs OK/In         : 8/8
  Cores OK/In        : 224/224
  DIMMs OK/In        : 64/64
  IO Cards OK/In     : 3/3
  Hyper-Threading    : Off
  RAS                : On
  Boot Chassis       : r001i01b
  Boot Slots         : 3,5
  Secure Boot        : Off
  Secure Boot Next   : Off
  Volatile Memory    : 1535 GiB
  Persistent Memory  : 2016 GiB

Partition 11:
  Run State          : OS Boot
  Health Status      : OK
  Chassis OK/In      : 1/1
  CPUs OK/In         : 4/4
  Cores OK/In        : 112/112
  DIMMs OK/In        : 32/32
  IO Cards OK/In     : 0/0
  Hyper-Threading    : On
  RAS                : On
  Boot Chassis       : r001i11b
  Boot Slots         : 3,5
```

```
  Secure Boot       : Off
  Secure Boot Next  : Off
  Volatile Memory   : 767 GiB
  Persistent Memory : 1008 GiB

Partition 16:
  Run State         : OS Boot
  Health Status     : OK
  Chassis OK/In     : 1/1
  CPUs OK/In        : 4/4
  Cores OK/In       : 112/112
  DIMMs OK/In       : 32/32
  IO Cards OK/In    : 0/0
  Hyper-Threading   : On
  RAS               : On
  Boot Chassis      : r001i16b
  Boot Slots        : 3,5
  Secure Boot       : Off
  Secure Boot Next  : Off
  Volatile Memory   : 767 GiB
  Persistent Memory : 1008 GiB

* OK/In = OK/Installed
```

The preceding example shows the `RAS` entry set to `on`. On this system, reliability, availability, and serviceability (RAS) mode is enabled, which is the default. If the output shows the `RAS` entry set to `Hpc`, HPC mode is enabled and RAS is disabled.

The `ras=on` specification is the default. This setting enables ADDDC mode. This mode allows better error recovery but incurs a small performance penalty. When set, all memory error handling occurs in the BIOS, rather than `memlog`.

The `ras=hpc` specification enables maximum memory performance but incurs a small reliability penalty.

CAUTION: Hewlett Packard Enterprise strongly recommends using the `ras=on` specification to enable memory RAS features (ADDDC mode). HPE RAS features provide higher resiliency to DIMM faults versus standard memory error-correcting code (ECC).

The `ras=hpc` specification disables memory RAS features and therefore could result in compromised system resiliency and a potential server outage.

To check the memory RAS setting, enter the `show npar verbose` RMC command.

The memory modes affect `memlog` processing as follows:

- When RAS mode is enabled, `memlog` is disabled and exits.
- When HPC mode is enabled, `memlog` is enabled and monitors system health.

To determine whether `memlog` is running, enter the following command on the server:

```
remotehost% systemctl status memlog
```

For more information about the MEMlog utility, see the `memlog`(8) manpage.

For more information about the `modify` command, use the `help modify` RMC command.

Retrieving main memory health information

Hewlett Packard Enterprise recommends that you check your system periodically to determine whether the MEMlog utility has reported any hardware errors. The MEMlog utility verifies and diagnoses problems with the DIMMs. The utility messages appear in `/var/log/messages`.

The following are ways to use commands to retrieve information about memory problems and memory health:

- Use the `memlogd` command with the `–s` option to retrieve information about DIMMs tagged for repairs. If there are DIMMs to replace, the command returns a `CRITICAL` message along with detailed information for each failing DIMM. For example, the following message indicates a DIMM to be replaced:

```
uv:~ # memlogd -s
CRITICAL - one or more DIMM failling.
r001i11b04  P1-DIMM3A Size 16384MB Width 4 Rank 2 Row 16 Col 10 Bank 16
Serial 39d959c4 Part M393A2G40DB0-CPB     signed 2133
  Mon Dec 14 14:40:02 2015 Rank 1 Dram U14B Bank 3 Row 0x5fe5 Col 0x1e8
  single CB DQ47  Temp = 33C hits 26
```

  If there are no DIMMs to replace, the output is as follows:

```
Ok - all DIMMs within specification
```

- Scan the system log for entries that contain the string `MEMLOG`. If problems arise with any of the DIMMs on your system, the MEMlog utility writes a message to `/var/log/messages`. To retrieve these messages, enter the following command:

```
# grep MEMLOG /var/log/messages
```

  For example, the following messages indicate some DIMMs to replace:

```
# grep MEMLOG /var/log/messages

r1i0n0:Dec  9 07:29:45 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM1A Rank 0 DRAM U9 DQ4 Temp = 21C
r1i0n0:Dec  9 07:30:00 r1i0n0 MEMLOG[4595]:  P1-DIMM1A has a failed DRAM and must be replaced soon.
     Exposure to Uncorrected Error is high
r1i0n0:Dec  9 07:30:00 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM1A Rank 0 Bank 0 Row 0x0 Col 0x8 Temp = 21C
r1i0n0:Dec  9 07:30:00 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM1A Rank 0 DRAM U9 DQ4 Temp = 21C
r1i0n0:Dec  9 07:30:12 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM3A Rank 0 Temp = 22C
r1i0n0:Dec  9 07:30:12 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM3A Rank 0 DRAM U9 DQ4 Temp = 22C
r1i0n0:Dec  9 07:30:25 r1i0n0 MEMLOG[4595]:  P1-DIMM3A has a failed DRAM and must be replaced soon.
     Exposure to Uncorrected Error is high
r1i0n0:Dec  9 07:30:25 r1i0n0 MEMLOG[4595]: Read ECC  P1-DIMM3A Rank 0 Bank 0 Row 0x0 Col 0x8 Temp = 22C
```

NOTE: Some lines in the preceding output have been wrapped for inclusion in this documentation.

About page migration

If the main memory encounters correctable memory errors, system software analyzes the failure and determines the severity of the problem. The following events occur:

- If MEMlog determines that the correctable error is a repeatable failure, MEMlog completes the following actions:
  - MEMlog asks the kernel to migrate the 4k page that contains the corrected error address. If the page is mapped to user space and is not locked, the kernel copies the data to a new page. In addition, MEMlog remaps the old page to the new page.
  - The kernel retires the old page. It is no longer available for use.
- If the system software detects that the program data at that memory location is uncorrectable, the following events occur:
  - The kernel stops the application that was affected by the uncorrectable memory error.
  - The application exits.

  All other applications running continue to work. The faulty 4K page is retired and is not made available for use.

Example 1. The messages in this example indicate the following:

- The kernel received a request to migrate a page.
- The kernel migrated the page.
- The kernel received additional requests to migrate the page.

```
# grep "soft offline" /var/log/messages
[  +0.302854] soft offline: 0x171596ad8: page leaked
[  +0.417714] soft offline: 0x171596ad8: page already poisoned
```

Typically, you see the preceding messages when there is a delay in completing the migration. When a page is reported a second time, the system software checks the poisoned flag, detects the poisoned flag, and generates the `page already poisoned` message in `/var/log/messages`. When the kernel marks a page as poisoned, this action ensures that the page is not reused.

Example 2. The kernel writes messages such as the following to its log file when a system encounters an uncorrectable memory error:

```
[  139.475068] {1}[Hardware Error]: Hardware error from APEI Generic Hardware Error Source: 2
[  139.475069] {1}[Hardware Error]: It has been corrected by h/w and requires no further action
[  139.475070] {1}[Hardware Error]: event severity: corrected
[  139.475070] {1}[Hardware Error]:  Error 0, type: corrected
[  139.475071] {1}[Hardware Error]:  fru_text:
[  139.475071] {1}[Hardware Error]:   section_type: memory error
[  139.475072] {1}[Hardware Error]:   error_status: 0x0000000000040400
[  139.475072] {1}[Hardware Error]:   physical_address: 0x0000000fba0e9000
[  139.475073] {1}[Hardware Error]:   node: 0
[  139.541602] Disabling lock debugging due to kernel taint
[  139.547595] mce: [Hardware Error]: Machine check events logged
[  139.554109] mce: Uncorrected hardware memory error in user-access at fba0e9000
[  139.562114] MCE 0xfba0e9: Killing einj_tool:4730 due to hardware memory corruption
[  139.570782] MCE 0xfba0e9: dirty LRU page recovery: Recovered
```

Monitoring system performance

You can use Linux utilities, HPE Foundation Software (HFS) utilities, and open-source utilities to monitor system performance.

The Linux utilities include `w`, `ps`, `top`, `vmstat`, `iostat`, and `sar`.

The HFS utilities are **`gr_systat`**, **`hubstats`**, **`nodeinfo`**, and **`topology`**.

Configuring CPU frequency scaling

CPU frequency scaling allows the operating system to scale the processor frequency automatically and dynamically. Hewlett Packard Enterprise configures the CPU frequency scaling setting on all HPE Superdome Flex Server systems before they leave the factory. The default setting is assumed to be correct for most implementations. The CPU frequency scaling setting lets your system take advantage of the Intel Turbo Boost technology that is built into each processor.

The Intel Turbo Boost Technology allows processor cores to run faster than the base operating frequency as long as they are operating within the limits set for power, current, and temperature. The CPU frequency scaling setting also affects power consumption and enables you to manage power consumption. For example, theoretically, you can cut power consumption if you clock the processors from 2 GHz down to 1 GHz.

Changing the CPU governor setting and frequency setting

The default CPU frequency governor setting can inhibit system performance. The `hpe-auto-config` utility automatically sets the CPU frequency setting to `performance` mode and sets CPUs to maximum frequency. This procedure explains how to override the automatic CPU frequency setting.

Procedure

1. Enter the following command to view the available CPU governor settings, and study the output to determine which governor setting is appropriate for your site:

   ```
   # cpupower frequency-info -g
   ```

   Hewlett Packard Enterprise recommends that you verify that the CPU governor setting is `performance` and if some other setting is shown, change it to `performance`.

2. Enter the following command to display the available CPU frequencies:

   ```
   # cpupower frequency-info
   ```

   Inspect the `frequency steps` field, and choose a minimum and/or maximum frequency.

3. Enter one or more of the following commands to change the governor and/or frequency settings:

   - Enter the following command to change the governor setting:

     ```
     # cpupower frequency-set -g GOVERNOR
     ```

     For *GOVERNOR*, specify the setting you want.

   - Use the `cpupower frequency-set` command to change one of the following:

     - Both the minimum frequency and the maximum frequency
     - The maximum frequency
     - The minimum frequency

     ```
     # cpupower frequency-set -u MAX -d min
     # cpupower frequency-set -u MAX
     # cpupower frequency-set -d MIN
     ```

     For *MAX* and *MIN*, specify a value in the following format: *VALUE*[*UNIT*]

     The default *UNIT* is KHz. You can also specify a *UNIT* of Hz, MHz, GHz, or THz.

4. Enter the following command and verify that the *GOVERNOR* setting you specified appears in the `cpupower` command output in the `current policy` field:

   ```
   # cpupower frequency-info
   ```

5. Create a configuration file that includes the settings you configured in this procedure.

   Your goal is to create a file that includes the command you ran in this procedure. Make sure that the file has execute permission.

   ```
   # echo "cpupower frequency-set -g performance -u 3000MHz -d 2000MHz" > \
   /etc/hpe-auto-config/90_cpu_frequency.sh
   # chmod 744 /etc/hpe-auto-config/90_cpu_frequency.sh
   ```

   When the system boots, the settings in this file override the default `hpe-auto-config` settings to ensure that the settings you configured in this procedure are included after the boot.

Configuring turbo mode

Procedure

1. Make sure that you configured a governor setting.

   For information about how to configuring a governor setting, see **Changing the CPU governor setting and frequency setting** on page 8.

2. Use the `cat` command to retrieve the list of available frequencies.

```
# cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
3301000 3300000 3200000 3100000 3000000 2900000 2800000 2700000 2600000
2500000 2400000 2300000 2200000 2100000 2000000 1900000 1800000 1700000
1600000 1500000 1400000 1300000 1200000
```

   The preceding output shows the available frequencies. The output lists frequencies in order from the highest, 3,301,000 KHz, to the lowest, 1,200,000 KHz.

   The second frequency listed is always the processor nominal frequency. This processor is a 3.3 GHz processor, so 3,300,000 KHz is the nominal frequency.

   You can also obtain the nominal frequency by entering the following command and examining the information in the model name field:

```
# cat /proc/cpuinfo
```

3. Use the `cpupower` command to set the frequency to the nominal frequency of 3.3 GHz plus 1 MHz.

   That is, specify a frequency of 3,301 MHz.

```
# cpupower frequency-set -u 3301MHz
```

   Later, if you want to disable turbo mode, enter the following command to set the maximum frequency back to the nominal frequency:

```
# cpupower frequency-set -u 3300MHz
```

Activating an extended tuning script

You can use HPE Foundation Software to select and activate an extended tuning script for your system. These scripts optimize performance for applications such as SAP HANA OS.

This document describes a general procedure for using tuning scripts. For more detailed information, see *Configuration Guide for HPE Superdome Flex Solutions for SAP HANA with 3PAR All Flash Storage*.

Procedure

1. Verify boot parameters needed for your system configuration, and make changes as necessary.
2. Create and set your parameters in the configuration script file.
   - For SLES, use the configuration file *HPE-Recommended_OS_settings.conf* located in the */etc/saptune/extra* directory. If the file does not exist, create the file.
   - For RHEL, use the configuration file *tuned.conf* located in the */etc/tuned/sap-hpe-hana* directory. If the file does not exist, create the file.
3. Verify that tuning settings are applied by using the `sysctl` command.

Additional HPE Foundation Software (HFS) utilities

There are additional HFS commands and utilities available that typically require no user involvement. Hewlett Packard Enterprise technical support staff members might guide you in the use of these commands when troubleshooting or tuning.

- **`hpe-auto-config`**
- **`hpe_irqbalance`**

Performance tuning overrides

AUTHOR NOTE: This content covers SH_CR0011 ability to modify directory compression mode.

```
0x0 is new default (extra compressed)
0x1 is the old default (comressed0 – so this what customer might choose to
use.
0x2 is not trusted (uncompressed) -  the 0x2 (1/16th, uncompressed) setting
should not be advertised or listed except indirectly as "not recommended".
```

AUTHOR NOTE: Future updates should cover other features of SH_CR0011.

```
CR#11 (SH_CR0011) -- In all there are four changes being requested to improve system performance:
- The ability to modify directory compression mode
- The ability to RRQ/IRQ settings
- Enable Sub-NUMA clustering
- Enable appropriate package and core c-states to allow Turbo-boosting
```

Memory directory compression options

The Superdome Flex system tracks the sharing state of system main memory cache lines by utilizing a portion of main memory as a directory.

The size of this directory can be 1/64 or 1/32 the size of main memory.

Depending on the size of the computer system and the nature of the customer workload more optimal performance may be achieved by overriding the value automatically determined by the system firmware.

Performance experiments across systems of various sizes have shown that the most compressed setting results in the best performance.

The default setting is the "extra-compressed" setting (1/64 or 1/32 the size of main memory), as of firmware release 2.1.

Table 1: EFI shell DirectoryFormat variable values

Value	Directory size	Description
0x0	1/64 of main memory	Extra compressed
0x1	1/32 of main memory	Compressed

Checking the current memory directory format

The current directory format of a booted system can be determined by inspecting the value of the EFI variable `DirectoryFormat` using GUID `FDD70221-655E-48E8-B77D-EA392EC5F60C`.

Procedure

1. Access the EFI shell for a partition.
2. Use the `sysconfig` or `setvar` command to list he `DirectoryFormat` variable value.

```
Shell> sysconfig
....
XNC Directory Format: 1/32 (0x1)
....
```

```
Shell> setvar DirectoryFormat -guid FDD70221-655E-48E8-B77D-EA392EC5F60C
Variable - NV+RT+BS - 'FDD70221-655E-48E8-B77D-EA392EC5F60C:DirectoryFormat' - DataSize = 0x01
00000000: 01 *.*
```

Modifying the memory directory size

The directory format value can be overridden to take effect after the next system reset by setting an EFI variable of the name `DirectoryFormatNext` utilizing EFI variable GUID `FDD70221-655E-48E8-B77D-EA392EC5F60C`.

Any mechanism which can set this value in the EFI variable store (e.g. Redfish, EFI shell `setvar` command, or operating system EFI variable access) can configure this EFI variable.

Procedure

1. Access the EFI shell for the partition.
2. Set the `DirectoryFormatNext` EFI variable to the desired value.

   From the EFI shell, the `DirectoryFormatNext` variable can be set using the `setvar` command

   ```
   Shell> setvar DirectoryFormatNext -guid FDD70221-655E-48e8-B77D-EA392EC5F60C -bs -rt -nv =0x0
   Creating or updating a variable:
   GUID FDD70221-655E-48E8-B77D-EA392EC5F60C, Name DirectoryFormatNext: Success
   ```

   The specified is used after the next reset.

   If the `DirectoryFormatNext` EFI variable is set to an invalid value, the specified value will be ignored.

Performance guidelines for Superdome Flex Server

AUTHOR NOTE: Topics from "Linux Performance and Tuning on HPE Servers" (Yann Allandit, HPE Oracle presales consultant)

HPE Superdome Flex system architecture

This is new content requiring SME review and editing.

The HPE Superdome Flex Server uses a modular system architecture designed around a 5U 4-socket building block. This modular approach allows the system to be easily scaled from 4 to 32 CPU sockets in 4-socket increments. See **Superdome Flex system architecture**.

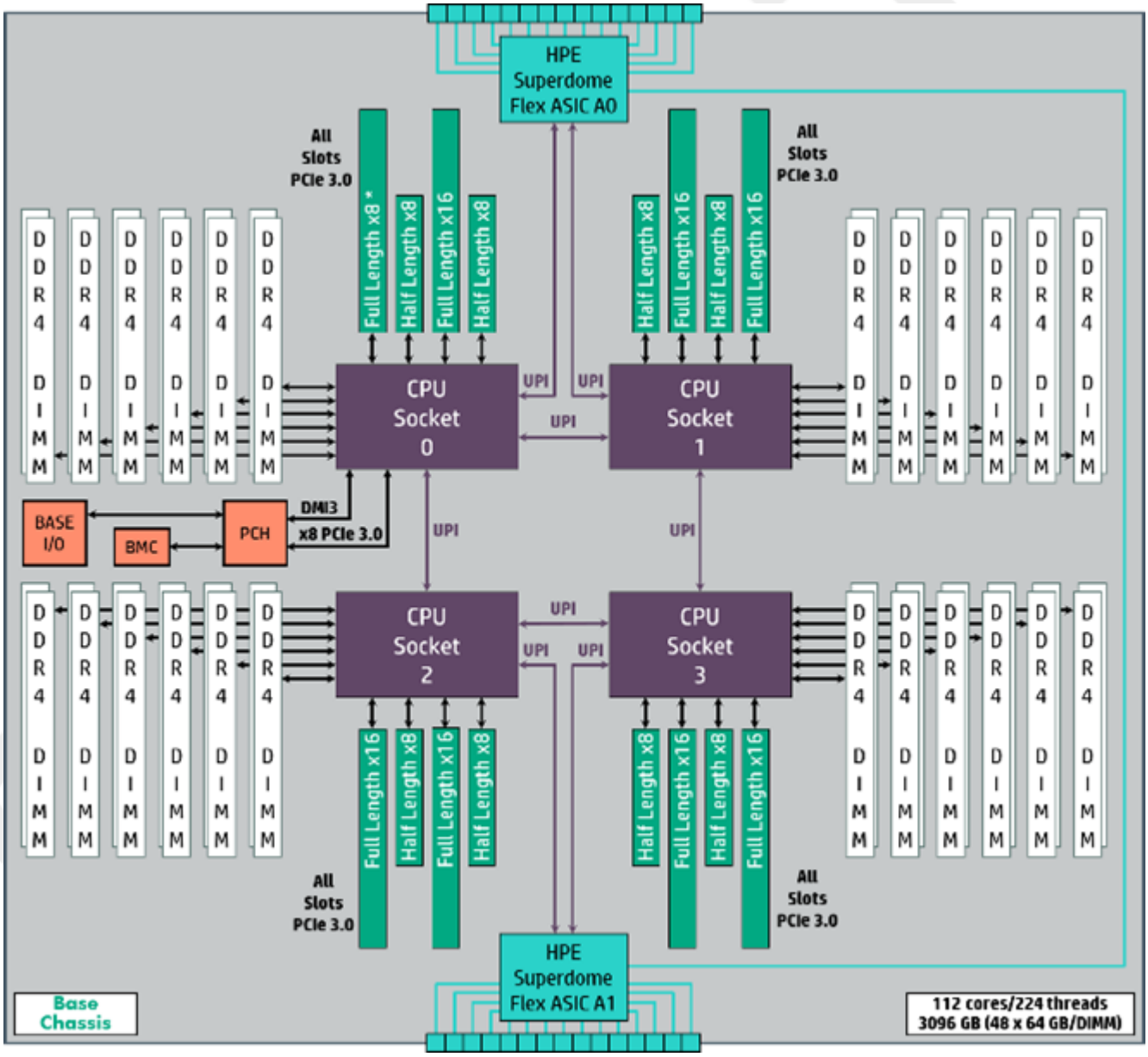


Figure 1: Superdome Flex system architecture

Flex Grid

External Flex Grid interconnect cables provide a point-to-point interconnect fabric between all chassis to guarantee low latency and high bandwidth data transfer.

- ~213.33 GB/s bisection crossbar bandwidth for an 8-socket system (2 chassis)
- ~426.67 GB/s bisection crossbar bandwidth for a 16-socket system (4 chassis)
- ~853.33 GB/s bisection crossbar bandwidth for a 32-socket system (8 chassis)

Future Flex Grid enhancements will include fault tolerance and online repair capabilities.

PCIe interface cards

Flexible stand-up PCIe Gen3 card format is supported in two different options. This will allow the use of all low-profile cards or up to four full-height/double-wide cards.

System maximum configuration

A maximum system configuration consists of these components:

Compute	Memory	I/O
32 CPUs	384 DIMMs	72 (x8) slots
896 cores	48TB capacity	56 (x16) slots
1792 threads		boot storage

Memory latency

This is new content requiring SME review and editing.

Due to the unique Superdome Flex architecture, memory access latency is extremely small. Each chassis contains four interconnected CPUs, each with it's own dedicated memory. All memory can be accessed by any CPU in the same chassis or in other interconnected chassis in the same system or by other connected systems.

There are five levels of memory access. Each of these levels result in different access delays due to the increased number of "hops" required reach the memory.

1. Local access--direct access by local CPU connected to the memory
2. Remote on-board 1-hop access--indirect access by another CPU connected directly to the local CPU
3. Remote on-board 2-hop access--indirect access by another CPU not directly connected to the local CPU, requiring a hop through a directly connected CPU
4. Remote off-board access--indirect access by a CPU in another chassis in the same system
5. Remote off-enclosure access--remote access by another system

Processor support: Platinum and Gold SKUs

This is new content requiring SME review and editing.

HPE Superdome Flex system architecture supports two interprocessor communication modes. The four CPUs are arranged in "clumps" depending on the type of processor.

- 4S mode--four CPU sockets in a clump

- Requires Platinum level processors
- All CPUs communicate with each other with UPI links through the processor uncore.
- All remote traffic (above 4S) is made through cable-connected Superdome Flex ASICs.
- Supports four levels of latency:
  - Local access
  - Remote on-board 1-hop access
  - Remote on-board 2-hop access
  - Remote off-board access
- Highest performance.

- 2S+2S mode--two CPU sockets in a clump with two separate clumps
  - Supports Gold level processors
  - Two CPUs, 0 - 1 (or 2 - 3), communicate with each other with UPI links through the processor uncore. Accessing the other two CPUs (2 - 3 or 0 - 1) requires the on-board path through the Superdome Flex Grid.
  - All remote traffic (above 4S) is made through the Superdome Flex ASICs.
  - Supports three levels of latency:
    - Local access
    - Remote on-board 1-hop access
    - Remote off-board access
  - Perfect for price/performance optimized solutions.

Intel® Turbo Boost Technology

This is new content requiring SME review and editing.

Intel® Turbo Boost Technology manages the CPU operation to enhance performance (P-states) and power consumption (C-states). See **P-States: processor performance power states** on page 16 and **C-States: processor idle states** on page 17.

P-States: processor performance power states

This is new content requiring SME review and editing.

P-States reduce CPU power consumption without preventing the processor from executing code.

A P-State is an operational state, meaning that the core/processor can be doing useful work in any P-state. The most obvious example is when your laptop is using a low power profile and operating on battery. The OS will lower the C0 operating frequency and voltage (enter a higher P-state). Reducing the operating frequency reduces the speed at which the processor operates, reducing the power consumption. Reducing the voltage decreases the leakage current from the CPU transistors, making the processor more energy-efficient resulting in further gains. The net result is a significant reduction in the power usage of the processor. With reduced CPU operating frequencies, an application will take longer to run which may or may not be a problem from a power perspective.

Intel Turbo Boost Technology increases performance by increasing processor frequency and enabling faster speeds when conditions allow.

We need more information about using Turbo Boost to increase performance.

C-States: processor idle states

This is new content requiring SME review and editing.

C-States reduce power consumption by idling the processor when it no code to execute.

With the exception of C0, where the CPU is active and busy doing something, a C-state is an idle state. Since an idle processor is not doing any useful work, the CPU can be shut down with no detrimental affect in the overall system processing.

Tuned-adm utility

This is new content requiring SME review and editing.

The tuned-adm Oracle utility provides predefined profiles for typical use cases. To list the available profiles, including the active profile, use the `tuned-adm list` command.

Example:

```
[root@rsfotest1 oracle-rsfo]# tuned-adm list
Available profiles:
- balanced
- desktop
- latency-performance
- network-latency
- oracle-rsfo
- powersave
- throughput-performance
- vitual-guest
- virtual-host
Current active profile: oracle-rsfo
```

You can also create your own profiles in `/usr/lib/tuned/oracl-rsfo`.

System semaphores

This is new content requiring SME review and editing.

Semaphores are a system resource that Oracle utilizes for interprocess communication and they occupy a relatively small memory space. Since Oracle 10g, the number of semaphores required by an instance is equal to twice the setting of the processes parameter.

The `SEMOPM` kernel parameter is used to control the number of semaphore operations that can be performed per `semop` system call. The `semop` system call (function) provides the ability to do operations for multiple semaphores with one `semop` system call.

To check the SEMOPM setting, view the `/proc/sys/kernel/sem` file. This file lists, in order, the following settings:

```
semmsl: maximum number of semaphore per array
semmns: maximum number of semaphore systemwide
semopm: maximum ops per semop call
semmni: maximum arrays
```

```
# cat /proc/sys/kernel/sem
250  32000  100   256
```

For more information about semaphores and guidance for configuring an Oracle database, see the "Oracle Processes and System Semaphore" article at **https://bigironoracle.blogspot.com/**.

Kernel and boot parameters

This is new content requiring SME review and editing.

The `sysctl.comf` Linux configuration file is used to override default kernel parameter values. This file contains values to be read in and set by `sysctl`.

For details about supported Linux kernel and boot parameters, see `sysctl.conf(5)`.

To list all possible parameters, use `/sbin/sysctl -a` or see `sysctl(8)`.

Need to confirm recommendations and provide more guidance, possible in other topics.

`sysctl.conf` kernel parameters

The `sysctl.conf` kernel parameters, with sample values, include:

- kernel.sched_autogroup_enabled = 0
- kernel.sched_migration_cost_ns = 5000000 (default is 50000)
- kernel.numa_balancing = 1 (default)
- vm.min_free_kbytes = 512000

kernel.sched_autogroup_enabled

This is a relatively new patch which Linux lauded back in late 2010. It basically groups tasks by TTY so perceived responsiveness is improved. But on server systems, large daemons like PostgreSQL are going to be launched from the same pseudo-TTY, and be effectively choked out of CPU cycles in favor of less important tasks.

The default setting is 1 (enabled) on some platforms. By setting this to 0 (disabled), we saw an outright 30% performance boost on the same pgbench test. A fully cached scale 3500 database on a 72GB system went from 67k TPS to 82k TPS with 900 client connections.

kernel.sched_migration_cost

The migration cost is the total time the scheduler will consider a migrated process "cache hot" and thus less likely to be re-migrated. By default, this is 0.5ms (500000 ns), and as the size of the process table increases, eventually causes the scheduler to break down. On our systems, after a smooth degradation with increasing connection count, system CPU spiked from 20 to 70% sustained and TPS was cut by 5-10x once we crossed some invisible connection count threshold. For us, that was a pgbench with 900 or more clients.

The migration cost should be increased, almost universally on server systems with many processes. This means systems like PostgreSQL or Apache would benefit from having higher migration costs. We've had good luck with a setting of 5ms (5000000 ns) instead.

When the breakdown occurs, system CPU (as obtained from sar) increases from 20% on a heavy pgbench (scale 3500 on a 72GB system) to over 70%, and %nice/%user is cut by half or more. A higher migration cost essentially eliminates this artificial throttle.

`sysctl.conf` boot parameters

The `sysctl.conf` boot parameters, with sample values, include:

- scsi_mod.use_blk_mq = 1
- dm_mode.use_blk_mq = 1
- Cgroup_disable = cpu

HPE Foundation Software kernel parameters

Need to confirm HFS values and settings.

HPE Foundation Software sets these kernel parameters

- intel_idle.max_cstate = 1
- transparent_hugepage = never
- numa_balancing = disable

Oracle HugePages memory setting

This is new content requiring SME review and editing.

Size the Oracle System Global Area (SGA) big enough. The starting point is sga_target value of 70% of the RAM size.

Using Hugepages:

- Linux introduced the HugePages feature with the 2.6.x kernel.
- Provides an alternative to the 4K page size (2MB).
- Not swappable.
- Decrease page table overhead (page table will be 800MB if trying to handle 50GB of RAM)
- AMM (Automatic Memory Management) and HugePages are not compatible.

Steps after disabling AMM:

```
SQL> alter system reset memory_target;
SQL> alter system reset MEMORY_MAX_TARGET;
```

For more information, see these related Oracle notes:

Need links to these Oracle papers

- HugePages on Linux: What It Is... and What It Is Not [ID361323.1]
- Shell Script to Calculate Values Recommended Linux HugePages / HugeTLB Configuration [ID 401749.1]
- HugePages and Oracle Database 11g Automatic Memory Management (AMM) on Linux [ID 749851.1]
- HugePages on Oracle Linux 64-bit [ID 361468.1]

Checking and setting the Oracle HugePages setting

You can check and set the size of the Oracle HugePage memory setting on a system by using this procedure.

Prerequisites

This is new content requiring SME review and editing.

- Disable AMM:

  ```
  SQL> alter system reset memory target;
  SQL> alter system reset MEMORY_MAX_TARGET;
  ```

  - memory_target=0
  - memory_max_target=0

- Oracle database is up and running

Procedure

1. Run the `hugepages_setting` script.

```
root@oracle52 #  ./hugepages_settings.sh
```

2. Update the `/etc/security/limits.conf` file.

```
root@oracle52 # more /etc/security/limits.conf
__
*  soft  memlock    30397977
*  hard  memlock    30397977
```

   The value (in KB) should be slightly smaller than the installed RAM.

3. Update the `/etc/sysctl.conf` file.

   - `vm.nr_hugepages = 2770`
   - `vm.hugetlb_shm_group = 'id -g oracle'`

4. Update Oracle init parameters.

```
use_large_pages = "ONLY"
db_cache_siZe = 100G
db_16k_cache_size = 32G
```

5. Restart the instances.

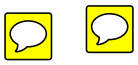
6. Check the HugePages usage.

```
root@oracle52 # grep Huge /proc/meminfo
HugePages_Total:  2770
HugePages_Free:   2770
HugePages_Rsve:     0
```

```
root@oracle52 # grep Huge /proc/meminfo
HugePages_Total:  2770
HugePages_Free:   2555
HugePages_Rsve:   1865
```

Performance commands and tools

List of command references to include.



```
COMMANDS TO INCLUDE IN FIRST EDITION:
lscpu
cpupower frequency-set --governor=performance
cpupower idle-set --enable-all
Tool (sudo):  x86_energy_perf_policy -v performance
tuned-adm
ethtool  (network config)
lscpu
Application Tuner Express (ATX) utility (HPE)
linuxki (HPE)
RFSO (rapid setting for Oracle)

FUTURE:
./mlc --latency_matrix
./mlc --bandwidth_matrix
hubstats
```

Application Tuner Express (ATX)

Syntax

Description

Parameters

Subcommands

Options

Permissions

Restrictions

Usage

Example input

Example output

cpupower

Syntax

Description

Parameters

Subcommands

Options

Permissions

Restrictions

Usage

Example input

Example output

ethtool

Syntax

Description

Parameters

Subcommands

Options

Permissions

Restrictions

Usage

Example input

Example output

linuxki

Syntax

Description

Parameters

Subcommands

Options

Permissions

Restrictions

Usage

Example input

Example output

lscpu

Syntax

Description

Parameters

Subcommands

Options

Permissions

Restrictions

Usage

Example input

Example output

Rapid Setting for Oracle (RSFO)

Syntax

Description

Parameters

Subcommands

Options

Permissions

Restrictions

Usage

Example input

Example output

tuned-adm

Syntax

Description

Parameters

Subcommands

Options

Permissions

Restrictions

Usage

Example input

Example output

x86_energy_perf_policy

Syntax

Description

Parameters

Subcommands

Options

Permissions

Restrictions

Usage

Example input

Example output

Linux application tuning

REVIEWERS: These topics are originally from *Linux Application Tuning Guide for SGI X86-64 Based Systems*

About performance tuning

After you analyze your code to determine where performance bottlenecks are occurring, you can turn your attention to making your programs run their fastest. One way to do this is to use multiple CPUs in parallel processing mode. However, this should be the last step. The first step is to make your program run as efficiently as possible on a single processor system and then consider ways to use parallel processing.

Intel provides tuning information, including information about the Intel processors, at the following website:

http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html .

Performance tuning addresses the process of tuning your application for a single processor system and then tuning it for parallel processing. It also addresses how to improve the performance of floating-point programs and MPI applications.

Memory use strategies

The following are some general memory use goals and guidelines:

- Register reuse. Do a lot of work on the same data before working on new data.
- Cache reuse. The program is much more efficient if all of the data and instructions fit in cache. If the data and instructions do not fit in the cache, try to use what is in cache before using anything that is not in cache.
- Data locality. Try to access data that is nearby in memory before attempting to access data that is far away in memory.
- I/O efficiency. Perform a large amount of I/O operations all at once, rather than a little bit at a time. Do not mix calculations and I/O.

Memory hierarchy latencies

Memory is not arranged as a flat, random access storage device. It is critical to understand that memory is a hierarchy to get good performance. Memory latency differs within the hierarchy. Performance is affected by where the data resides.

CPUs that are waiting for memory are not doing useful work. Software should be hierarchy-aware to achieve best performance, so observe the following guidelines:

- Perform as many operations as possible on data in registers.
- Perform as many operations as possible on data in the cache(s).
- Keep data uses spatially and temporally local.
- Consider temporal locality and spatial locality.

  Memory hierarchies take advantage of temporal locality by keeping more recently accessed data items closer to the processor. Memory hierarchies take advantage of spatial locality by moving contiguous words in memory to upper levels of the hierarchy.

Non-uniform Memory Access (NUMA)

REVIEWERS: This topic needs updates (e.g. "chassis") for Superdome Flex Server. Also, what of value can be added here?

In DSM systems, memory is physically located at various distances from the processors. As a result, memory access times (latencies) are different or **nonuniform**. For example, it takes less time for a processor blade to reference its locally installed memory than to reference remote memory.

ccNUMA architecture

As the name implies, the cache-coherent non-uniform memory access (ccNUMA) architecture has two parts: cache coherency and nonuniform memory access.

Cache coherency

HPE Superdome Flex Server uses caches to reduce memory latency. Although data exists in local or remote memory, copies of the data can exist in various processor caches throughout the system. Cache coherency keeps the cached copies consistent.

To keep the copies consistent, the ccNUMA architecture uses directory-based coherence protocol. In directory-based coherence protocol, each 64-byte block of memory has an entry in a table that is referred to as a **directory**. Like the blocks of memory that they represent, the directories are distributed among the compute and memory blade nodes. A block of memory is also referred to as a **cache line**.

Each directory entry indicates the state of the memory block that it represents. For example, when the block is not cached, it is in an **unowned state**. When only one processor has a copy of the memory block, it is in an **exclusive state**. When more than one processor has a copy of the block, it is in a **shared state**. A bit vector indicates the caches that may contain a copy.

When a processor modifies a block of data, the processors that have the same block of data in their caches must be notified of the modification. Superdome Flex Server uses an invalidation method to maintain cache coherence. The invalidation method purges all unmodified copies of the block of data, and the processor that wants to modify the block receives exclusive ownership of the block.

Determining Process Placement

This topic describes methods that you can use to determine where different processes are running. This can help you understand your application structure and help you decide if there are obvious placement issues. Note that all examples use the C shell.

The following procedure explains how to set up the computing environment.

To create the computing environment

Procedure

1. Set up an alias as in this example, changing *guest* to your username:

   ```
   % alias pu "ps -edaf|grep guest" % pu
   ```

   The `pu` command alias shows current processes.

2. Create the `.toprc` preferences file in your login directory to set the appropriate `top (1)` options.

   If you prefer to use the `top(1)` defaults, delete the `.toprc` file.

   ```
   % cat <<EOF>> $HOME/.toprc
   YEAbcDgHIjklMnoTP|qrsuzV{FWX
   2mlt
   EOF
   ```

3. Inspect all processes, determine which CPU is in use, and create an alias file for this procedure.

The CPU number appears in the first column of the `top(1)` output:

```
% top -b -n 1 | sort -n | more
% alias top1 "top -b -n 1 | sort -n "
```

Use the following variation to produce output with column headings:

```
% alias top1 "top -b -n 1 | head -4 | tail -1;top -b -n 1 | sort -n"
```

4. View your files, replacing *guest* with your username:

   ```
   % top -b -n 1 | sort -n | grep guest
   ```

   Use the following variation to produce output with column headings:

   ```
   % top -b -n 1 | head -4 | tail -1;top -b -n 1 | sort -n grep guest
   ```

The following topics present examples:

- **#unique_54**
- **#unique_55**

Determining system configuration

One of the first steps in application tuning is to determine the details of the system that you are running. Depending on your system configuration, different options might or might not provide good results.

The `topology(1)` command displays general information about systems, with a focus on node information. This can include node counts for racks, chassis, node IDs, NASIDs, memory per node, system serial number, partition number, Hub versions, CPU to node mappings, and general CPU information.

The `topology(1)` command is part of the SGI Foundation Software package.

```
h2-615:~ # topology
System type: Superdome Flex
System name: h2-615
Serial number: 5UF7110001
Partition number: 0
       1 Rack
       2 Chassis
     224 CPUs (online: 0-223)
       8 Nodes
     486 GB Memory Total
       1 BASE I/O Card
       1 PCIe Card
       1 Co-processor
       2 Fibre Channel Controllers
       4 Network Controllers
       1 SATA Storage Controller
       1 USB Controller
       1 VGA GPU
       1 RAID Controller
h2-615:~ # topology --summary --nodes --cpus
System type: Superdome Flex
System name: h2-615
Serial number: 5UF7110001
Partition number: 0
       1 Rack
       2 Chassis
     224 CPUs (online: 0-223)
       8 Nodes
     486 GB Memory Total
```

```
      1 BASE I/O Card
      1 PCIe Card
      1 Co-processor
      2 Fibre Channel Controllers
      4 Network Controllers
      1 SATA Storage Controller
      1 USB Controller
      1 VGA GPU
      1 RAID Controller
 Node   Location    NASID  CPUS    Memory
----------------------------------------
   0  r001i01b00h1   0002   28      60 GB
   1  r001i01b00h0   0000   28      60 GB
   2  r001i01b01h1   0006   28      60 GB
   3  r001i01b01h0   0004   28      60 GB
   4  r001i06b00h1   000a   28      60 GB
   5  r001i06b00h0   0008   28      60 GB
   6  r001i06b01h1   000e   28      60 GB
   7  r001i06b01h0   000c   28      60 GB
CPU     Location PhysID CoreID APIC-ID Family Model Speed L1(KiB) L2(KiB) L3(KiB)
-------------------------------------------------------------------------------
  0 r001i01b00h1     00     00       0      6    85  2600 32d/32i    1024   19712
  1 r001i01b00h1     00     01       2      6    85  2600 32d/32i    1024   19712
  2 r001i01b00h1     00     02       4      6    85  2600 32d/32i    1024   19712
  3 r001i01b00h1     00     03       6      6    85  2600 32d/32i    1024   19712
  4 r001i01b00h1     00     04       8      6    85  2600 32d/32i    1024   19712
  5 r001i01b00h1     00     05      10      6    85  2600 32d/32i    1024   19712
  6 r001i01b00h1     00     06      12      6    85  2600 32d/32i    1024   19712
  7 r001i01b00h1     00     08      16      6    85  2600 32d/32i    1024   19712
  8 r001i01b00h1     00     09      18      6    85  2600 32d/32i    1024   19712
  9 r001i01b00h1     00     10      20      6    85  2600 32d/32i    1024   19712
 10 r001i01b00h1     00     11      22      6    85  2600 32d/32i    1024   19712
 11 r001i01b00h1     00     12      24      6    85  2600 32d/32i    1024   19712
 12 r001i01b00h1     00     13      26      6    85  2600 32d/32i    1024   19712
 13 r001i01b00h1     00     14      28      6    85  2600 32d/32i    1024   19712
 14 r001i01b00h0     01     00      32      6    85  2600 32d/32i    1024   19712
 15 r001i01b00h0     01     01      34      6    85  2600 32d/32i    1024   19712
 16 r001i01b00h0     01     02      36      6    85  2600 32d/32i    1024   19712
 ...
```

The `cpumap(1)` command displays logical CPUs and shows relationships between them in a human-readable format. Aspects displayed include hyperthread relationships, last level cache sharing, and topological placement. The `cpumap(1)` command gets its information from `/proc/cpuinfo`, the `/sys/devices/system` directory structure, and `/proc/sgi_uv/topology`. When creating cpusets, the numbers reported in the output section called `Processor Numbering on Node(s)` correspond to the `mems` argument you use to define a cpuset. The cpuset `mems` argument is the list of memory nodes that tasks in the cpuset are allowed to use.

REVIEWERS -- need to replace this reference. Also check the above topology directory path, etc.

For more information, see the *SGI Cpuset Software Guide*.

```
h2-615:~ # cpumap
Fri May 24 09:47:07 MDT 2019
h2-615.fchst.rdlabs.hpecorp.net
This is an SGI UV
model name           : Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz
Architecture         : x86_64
cpu MHz              : 2600.000
cache size           : 19712 KB (Last Level)
Total Number of Sockets                 : 8
Total Number of Cores                   : 112    (14 per socket)
Hyperthreading                          : ON
Total Number of Physical Processors     : 112
Total Number of Logical Processors      : 224    (2 per Phys Processor)
UV Information
```

```
 HUB Version:                    UVHub  5.0
 Number of Hubs:                8
 Number of connected Hubs:           8
 Number of connected NUMAlink ports:       62
================================================================================
Hub-Processor Mapping
  Hub Location      Processor Numbers -- HyperThreads in ()
  --- ----------    ---------------------------------------
    0 r001i01b00h1     0    1    2    3    4    5    6    7    8    9   10
11   12   13
                  (  112  113  114  115  116  117  118  119  120  121  122
123  124  125 )
    1 r001i01b00h0    14   15   16   17   18   19   20   21   22   23   24
25   26   27
                  (  126  127  128  129  130  131  132  133  134  135  136
137  138  139 )
    2 r001i01b01h1    28   29   30   31   32   33   34   35   36   37   38
39   40   41
                  (  140  141  142  143  144  145  146  147  148  149  150
151  152  153 )
    3 r001i01b01h0    42   43   44   45   46   47   48   49   50   51   52
53   54   55
                  (  154  155  156  157  158  159  160  161  162  163  164
165  166  167 )
    4 r001i06b00h1    56   57   58   59   60   61   62   63   64   65   66
67   68   69
                  (  168  169  170  171  172  173  174  175  176  177  178
179  180  181 )
    5 r001i06b00h0    70   71   72   73   74   75   76   77   78   79   80
81   82   83
                  (  182  183  184  185  186  187  188  189  190  191  192
193  194  195 )
    6 r001i06b01h1    84   85   86   87   88   89   90   91   92   93   94
95   96   97
                  (  196  197  198  199  200  201  202  203  204  205  206
207  208  209 )
    7 r001i06b01h0    98   99  100  101  102  103  104  105  106  107  108
109  110  111
                  (  210  211  212  213  214  215  216  217  218  219  220
221  222  223 )
================================================================================
Processor Numbering on Node(s)
   Node    (Logical) Processors
  ------    ------------------------
     0      0    1    2    3    4    5    6    7    8    9   10   11   12
13  112  113  114  115  116  117  118  119  120  121  122  123  124  125
     1     14   15   16   17   18   19   20   21   22   23   24   25   26
27  126  127  128  129  130  131  132  133  134  135  136  137  138  139
     2     28   29   30   31   32   33   34   35   36   37   38   39   40
41  140  141  142  143  144  145  146  147  148  149  150  151  152  153
     3     42   43   44   45   46   47   48   49   50   51   52   53   54
55  154  155  156  157  158  159  160  161  162  163  164  165  166  167
     4     56   57   58   59   60   61   62   63   64   65   66   67   68
69  168  169  170  171  172  173  174  175  176  177  178  179  180  181
     5     70   71   72   73   74   75   76   77   78   79   80   81   82
83  182  183  184  185  186  187  188  189  190  191  192  193  194  195
     6     84   85   86   87   88   89   90   91   92   93   94   95   96
97  196  197  198  199  200  201  202  203  204  205  206  207  208  209
     7     98   99  100  101  102  103  104  105  106  107  108  109  110
111  210  211  212  213  214  215  216  217  218  219  220  221  222  223
================================================================================
Sharing of Last Level (3) Caches
  Socket    (Logical) Processors
  ------    ------------------------
```

```
     0      0    1    2    3    4    5    6    7    8    9   10   11   12
13  112  113  114  115  116  117  118  119  120  121  122  123  124  125
     1     14   15   16   17   18   19   20   21   22   23   24   25   26
27  126  127  128  129  130  131  132  133  134  135  136  137  138  139
     2     28   29   30   31   32   33   34   35   36   37   38   39   40
41  140  141  142  143  144  145  146  147  148  149  150  151  152  153
     3     42   43   44   45   46   47   48   49   50   51   52   53   54
55  154  155  156  157  158  159  160  161  162  163  164  165  166  167
     4     56   57   58   59   60   61   62   63   64   65   66   67   68
69  168  169  170  171  172  173  174  175  176  177  178  179  180  181
     5     70   71   72   73   74   75   76   77   78   79   80   81   82
83  182  183  184  185  186  187  188  189  190  191  192  193  194  195
     6     84   85   86   87   88   89   90   91   92   93   94   95   96
97  196  197  198  199  200  201  202  203  204  205  206  207  208  209
     7     98   99  100  101  102  103  104  105  106  107  108  109  110
111  210  211  212  213  214  215  216  217  218  219  220  221  222  223
================================================================================
HyperThreading
  Shared Processors
  -----------------
 (    0,  112) (    1,  113) (    2,  114) (    3,  115)
 (    4,  116) (    5,  117) (    6,  118) (    7,  119)
 (    8,  120) (    9,  121) (   10,  122) (   11,  123)
 (   12,  124) (   13,  125) (   14,  126) (   15,  127)
 (   16,  128) (   17,  129) (   18,  130) (   19,  131)
 (   20,  132) (   21,  133) (   22,  134) (   23,  135)
 (   24,  136) (   25,  137) (   26,  138) (   27,  139)
 (   28,  140) (   29,  141) (   30,  142) (   31,  143)
 (   32,  144) (   33,  145) (   34,  146) (   35,  147)
 (   36,  148) (   37,  149) (   38,  150) (   39,  151)
 (   40,  152) (   41,  153) (   42,  154) (   43,  155)
 (   44,  156) (   45,  157) (   46,  158) (   47,  159)
 (   48,  160) (   49,  161) (   50,  162) (   51,  163)
 (   52,  164) (   53,  165) (   54,  166) (   55,  167)
 (   56,  168) (   57,  169) (   58,  170) (   59,  171)
 (   60,  172) (   61,  173) (   62,  174) (   63,  175)
 (   64,  176) (   65,  177) (   66,  178) (   67,  179)
 (   68,  180) (   69,  181) (   70,  182) (   71,  183)
 (   72,  184) (   73,  185) (   74,  186) (   75,  187)
 (   76,  188) (   77,  189) (   78,  190) (   79,  191)
 (   80,  192) (   81,  193) (   82,  194) (   83,  195)
 (   84,  196) (   85,  197) (   86,  198) (   87,  199)
 (   88,  200) (   89,  201) (   90,  202) (   91,  203)
 (   92,  204) (   93,  205) (   94,  206) (   95,  207)
 (   96,  208) (   97,  209) (   98,  210) (   99,  211)
 (  100,  212) (  101,  213) (  102,  214) (  103,  215)
 (  104,  216) (  105,  217) (  106,  218) (  107,  219)
 (  108,  220) (  109,  221) (  110,  222) (  111,  223)
```

AUTHOR NOTE -- The following should be a separate topic

The `x86info(1)` command displays x86 CPU diagnostics information. Type one of the following commands to load the `x86info(1)` command if the command is not already installed:

- On Red Hat Enterprise Linux (RHEL) systems, type the following:

  ```
  # yum install x86info.x86_64
  ```

- On SLES systems, type the following:

  ```
  # zypper install x86info
  ```

```
h2-033:~ # lscpu
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              448
On-line CPU(s) list: 0-447
Thread(s) per core:  2
Core(s) per socket:  28
Socket(s):           8
NUMA node(s):        8
Vendor ID:           GenuineIntel
CPU family:          6
Model:               85
Model name:          Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz
Stepping:            4
CPU MHz:             2500.000
CPU max MHz:         3800.0000
CPU min MHz:         1000.0000
BogoMIPS:            5000.00
Virtualization:      VT-x
L1d cache:           32K
L1i cache:           32K
L2 cache:            1024K
L3 cache:            39424K
NUMA node0 CPU(s):   0-27,224-251
NUMA node1 CPU(s):   28-55,252-279
NUMA node2 CPU(s):   56-83,280-307
NUMA node3 CPU(s):   84-111,308-335
NUMA node4 CPU(s):   112-139,336-363
NUMA node5 CPU(s):   140-167,364-391
NUMA node6 CPU(s):   168-195,392-419
NUMA node7 CPU(s):   196-223,420-447
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge
mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall
nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl
xtopology nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl
vmx smx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid dca sse4_1 sse4_2 x2apic
movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm
3dnowprefetch cpuid_fault epb cat_l3 cdp_l3 invpcid_single pti intel_ppin
ssbd mba ibrs ibpb stibp tpr_shadow vnmi flexpriority ept vpid fsgsbase
tsc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm cqm mpx rdt_a avx512f
avx512dq rdseed adx smap clflushopt clwb intel_pt avx512cd avx512bw avx512vl
xsaveopt xsavec xgetbv1 xsaves cqm_llc cqm_occup_llc cqm_mbm_total
cqm_mbm_local dtherm ida arat pln pts hwp hwp_act_window hwp_pkg_req pku
ospke md_clear flush_l1d
```

You can also use the `uname` command, which returns the kernel version and other machine information.

```
h2-615:~ # uname -a >uname.1
Linux h2-615 4.12.14-150.17-default #1 SMP Thu May 2 15:15:46 UTC 2019 (bf13fb8) x86_64 x86_64 x86_64 GNU/Linux
```

For more system information, change to the `/sys/devices/system/node/node0/cpu0/cache` directory and list the contents. For example:

```
h2-615:/sys/devices/system/node/node0/cpu0/cache # ls
index0  index1  index2  index3  power  uevent
```

Change directory to `index0` and list the contents, as follows:

```
h2-615:/sys/devices/system/node/node0/cpu0/cache/index0 # ls
coherency_line_size  physical_line_partition  size
id                   power                    type
level                shared_cpu_list          uevent
number_of_sets       shared_cpu_map           ways_of_associativity
```

Resetting the file limit resource default

Several large user applications use the value set in the `limit.h` file as a hard limit on file descriptors, and that value is noted at compile time. Therefore, some applications might need to be recompiled in order to take advantage of the SGI system hardware.

To regulate these limits on a per-user basis for applications that do not rely on `limit.h`, you can modify the `limits.conf` file. This allows the administrator to set the allowed number of open files per user and per group. This also requires a one-line change to the `/etc/pam.d/login` file.

The following procedure explains how to change the `/etc/pam.d/login` file.

To change the file limist resource default

Procedure

1. Add the following line to `/etc/pam.d/login`:

   ```
   session  required  /lib/security/pam_limits.so
   ```

2. Add the following line to `/etc/security/limits.conf`, where *username* is the user's login and *limit* is the new value for the file limit resource:

   ```
   [username]  hard  nofile  [limit]
   ```

The following command shows the new limit:

```
ulimit -H -n
```

Because of the large number of file descriptors that some applications require, such as MPI jobs, you might need to increase the system-wide limit on the number of open files on your SGI system. The default value for the file limit resource is 1024. The default of 1024 file descriptors allows for approximately 199 MPI processes per host. You can increase the file descriptor value to 8196 to allow for more than 512 MPI processes per host by adding adding the following lines to the `/etc/security/limits.conf` file:

```
*     soft    nofile      8196
*     hard    nofile      8196
```

The `ulimit -a` command displays all limits, as follows:

```
h2-033:~ # ulimit -a
core file size          (blocks, -c) unlimited
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 24007968
max locked memory       (kbytes, -l) 64
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 24007968
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

About cpusets and control groups (cgroups)

AUTHOR COMMENT -- Is this topic necessary or sufficient/complete?

SGI systems support both cgroups and cpusets. The cpusets are a subsystem of cgroups.

Using SGI MPI

REVIEWERS: Section needs MPI team review.

The SGI Performance Suite includes the SGI Message Passing Toolkit (SGI MPT). SGI MPT includes both the SGI Message Passing Interface (SGI MPI) and SGI SHMEM. SGI MPI is optimized and more scalable for SGI UV series systems than the generic MPI libraries. SGI MPI takes advantage of the SGI UV architecture and SGI nonuniform memory access (NUMA) features.

Use the `-lmpi` compiler option to use MPI. For a list of environment variables that are supported, see the `mpi(1)` man page.

The `MPIO_DIRECT_READ` and `MPIO_DIRECT_WRITE` environment variables are supported under Linux for local XFS filesystems in SGI MPT version 1.6.1 and beyond.

MPI provides the MPI-2 standard MPI I/O functions that provide file read and write capabilities. A number of environment variables are available to tune MPI I/O performance. The `mpi_io(3)` man page describes these environment variables.

For information about performance tuning for MPI applications, see the following:

- *SGI MPI and SGI SHMEM User Guide*
- *MPInside Reference Guide*

Other Performance Analysis Tools

The following tools might be useful to you when you try to optimize your code:

- The Intel® VTune™ Amplifier XE, which is a performance and thread profiler. This tool can perform both local sampling and remote sampling experiments. In the remote samplling case, the VTune data collector runs on the Linux system, and an accompanying graphical user interface (GUI) runs on a Windows system, which is used for analyzing the results. The VTune profiler allows you to perform interactive experiments while connected to the host through its GUI.

  For information about Intel VTune Amplifier XE, see the following URL:
  http://software.intel.com/en-us/intel-vtune-amplifier-xe#pid-3773-760

- Intel Inspector XE, which is a memory and thread debugger. For information about Intel Inspector XE, see the following:

  http://software.intel.com/en-us/intel-inspector-xe/

- Intel Advisor XE, which is a threading design and prototyping tool. For information about Intel Advisor XE, see the following:

  http://software.intel.com/en-us/intel-advisor-xe

Profiling with `perf`

Linux Perf Events provides a performance analysis framework. It includes hardware-level CPU performance monitoring unit (PMU) features, software counters, and tracepoints. The `perf` RPM comes with the operating system, includes man pages, and is not an SGI product.

For more information, see the following man pages:

- `perf(1)`
- `perf-stat(1)`
- `perf-top(1)`
- `perf-record(1)`

- `perf-report(1)`
- `perf-list(1)`

`numactl` Command

The `numactl(8)` command runs processes with a specific NUMA scheduling or memory placement policy. The policy is set for an executable command and inherited by all of its children. In addition, `numactl(8)` can set a persistent policy for shared memory segments or files. For more information, see the `numactl(8)` man page.

Using the `iostat(1)` command

The `iostat(1)` command monitors system input/output device loading by observing the time the devices are active, relative to their average transfer rates. You can use information from the `iostat` command to change system configuration information to better balance the input/output load between physical disks. For more information, see the `iostat(1)` man page.

In the following `iostat(1)` command, the `10` specifies a 10-second interval between updates:

```
h2-615:~ # iostat 10
Linux 4.12.14-150.17-default (h2-615)     05/24/2019     _x86_64_    (224
CPU)
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.42    0.00    0.01    0.00    0.00   99.57
Device             tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               2.13        44.76        11.50     668101     171568
sdc               0.01         0.45         0.00       6700          0
sdb               0.01         0.45         0.00       6700          0
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           5.37    0.00    0.00    0.00    0.00   94.63
Device             tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               0.00         0.00         0.00          0          0
sdc               0.00         0.00         0.00          0          0
sdb               0.00         0.00         0.00          0          0
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           5.36    0.00    0.00    0.00    0.00   94.64
Device             tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               0.30         0.00         1.60          0         16
sdc               0.00         0.00         0.00          0          0
sdb               0.00         0.00         0.00          0          0
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           5.36    0.00    0.00    0.00    0.00   94.64
Device             tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               1.40         0.00         9.20          0         92
sdc               0.00         0.00         0.00          0          0
sdb               0.00         0.00         0.00          0          0
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           5.42    0.00    0.00    0.00    0.00   94.57
Device             tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               0.20         0.80         0.40          8          4
sdc               0.00         0.00         0.00          0          0
sdb               0.00         0.00         0.00          0          0
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           5.36    0.00    0.01    0.00    0.00   94.63
Device             tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               0.40        15.60         0.00        156          0
sdc               0.00         0.00         0.00          0          0
sdb               0.00         0.00         0.00          0          0
```

Using the `ps(1)` command

To determine active processes, use the `ps(1)` command, which displays a snapshot of the process table.

The `ps -A r` command returns all the processes currently running on a system.

```
guest@h2-615:~/app1> ps -eF r
UID          PID   PPID  C    SZ   RSS PSR STIME TTY      STAT   TIME CMD
guest      66470  66029 99  1037   772  98 13:24 pts/0    R      0:05 ./app1
guest      66471  66029 99  1037   816  70 13:24 pts/0    R      0:05 ./app1
guest      66472  66029 99  1037   676  56 13:24 pts/0    R      0:05 ./app1
guest      66473  66029 99  1037   692   0 13:24 pts/0    R      0:05 ./app1
guest      66474  66029 99  1037   700 208 13:24 pts/0    R      0:05 ./app1
guest      66475  66029 99  1037   676 209 13:24 pts/0    R      0:05 ./app1
guest      66476  66029 99  1037   692  29 13:24 pts/0    R      0:05 ./app1
guest      66477  66029 99  1037   816   1 13:24 pts/0    R      0:05 ./app1
guest      66478  66029 99  1037   768  57 13:24 pts/0    R      0:05 ./app1
guest      66479  66029 99  1037   772  28 13:24 pts/0    R      0:05 ./app1
guest      66480  66029 99  1037   696  42 13:24 pts/0    R      0:05 ./app1
guest      66481  66029 99  1037   816  14 13:24 pts/0    R      0:05 ./app1
guest      66494  66029  0  9184  3424  92 13:24 pts/0    R+     0:00 ps -eF r
guest@h2-615:~/app1> ps -A r
   PID TTY      STAT   TIME COMMAND
 66470 pts/0    R      1:10 ./app1
 66471 pts/0    R      1:10 ./app1
 66472 pts/0    R      1:10 ./app1
 66473 pts/0    R      1:09 ./app1
 66474 pts/0    R      1:10 ./app1
 66475 pts/0    R      1:10 ./app1
 66476 pts/0    R      1:10 ./app1
 66477 pts/0    R      1:10 ./app1
 66478 pts/0    R      1:10 ./app1
 66479 pts/0    R      1:10 ./app1
 66480 pts/0    R      1:10 ./app1
 66481 pts/0    R      1:10 ./app1
 66707 pts/0    R+     0:00 ps -A r
```

Using the `sar(1)` command

The `sar(1)` command returns the content of selected, cumulative activity counters in the operating system. Based on the values in the *count* and *interval* parameters, the command writes information *count* times spaced at the specified *interval*, which is in seconds. For more information, see the `sar(1)` man page. The following example shows the `sar(1)` command with a request for information about CPU 1, a count of 10, and an interval of 10:

```
h2-615:~ # sar -P 1 10 10 | tee sar.1
Linux 4.12.14-150.17-default (h2-615)     05/24/2019     _x86_64_     (224 CPU)
01:41:06 PM     CPU     %user     %nice   %system   %iowait    %steal     %idle
01:41:16 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
01:41:26 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
01:41:36 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
01:41:46 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
01:41:56 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
01:42:06 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
01:42:16 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
01:42:26 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
01:42:36 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
01:42:46 PM       1    100.00      0.00      0.00      0.00      0.00      0.00
Average:          1    100.00      0.00      0.00      0.00      0.00      0.00
```

Using the top(1) command

To monitor running processes, use the top (1) command. This command displays a sorted list of top CPU utilization processes.

Using the vmstat(8) command

The vmstat(8) command reports virtual memory statistics. It reports information about processes, memory, paging, block IO, traps, and CPU activity. For more information, see the vmstat(8) man page.

In the following vmstat(8) command, the 10 specifies a 10-second delay between updates.

```
uv44-sys:~ # vmstat 10
procs -----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 2  0      0 235984032 418748 8649568    0    0     0     0    0    0  0  0 100  0  0
 1  0      0 236054400 418748 8645216    0    0     0  4809 256729 3401  0  0 100  0  0
 1  0      0 236188016 418748 8649904    0    0     0   448 256200  631  0  0 100  0  0
 2  0      0 236202976 418748 8645104    0    0     0   341 256201 1117  0  0 100  0  0
 1  0      0 236088720 418748 8592616    0    0     0   847 257104 6152  0  0 100  0  0
 1  0      0 235990944 418748 8648460    0    0     0   240 257085 5960  0  0 100  0  0
 1  0      0 236049568 418748 8645100    0    0     0  4849 256749 3604  0  0 100  0  0
```

Without the *delay* parameter, which is 10 in this example, the output returns averages since the last reboot. Additional reports give information on a sampling period of length *delay*. The process and memory reports are instantaneous in either case.

Using the w(1) command

To obtain a high-level view of system usage that includes information about who is logged into the system, use the w(1) command, as follows:

```
uv44-sys:~ # w
 15:47:48 up  2:49,  5 users,  load average: 0.04, 0.27, 0.42
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
root     pts/0     13:10    1:41m  0.07s  0.07s -bash
root     pts/2     13:31    0.00s  0.14s  0.02s w
boetcher pts/4     14:30    2:13   0.73s  0.73s -csh
root     pts/5     14:32    1:14m  0.04s  0.04s -bash
root     pts/6     15:09   31:25   0.08s  0.08s -bash
```

The w command's output shows who is on the system, the duration of user sessions, processor usage by user, and currently executing user commands. The output consists of two parts:

- The first output line shows the current time, the length of time the system has been up, the number of users on the system, and the average number of jobs in the run queue in the last one, five, and 15 minutes.
- The rest of the output from the w command shows who is logged into the system, the duration of each user session, processor usage by user, and each user's current process command line.

Websites

HPE Superdome Flex Server websites

HPE Superdome Flex Server product page

www.hpe.com/support/superdome-flex-product

HPE Superdome Flex Server customer documentation

www.hpe.com/support/superdome-flex-docs

HPE Superdome Flex Server software

www.hpe.com/support/superdome-flex-software

Server operating systems and virtualization software

www.hpe.com/us/en/servers/server-operating-systems.html

HPE Superdome Flex Server QuickSpecs

www.hpe.com/support/superdome-flex-quickspecs

HPE Superdome Flex Server spare parts list

www.hpe.com/support/superdome-flex-spareparts

HPE Superdome Flex Server support documentation

HPE Superdome Flex Server documentation for support specialists is available at **www.hpe.com/support/superdome-flex-docs-restricted** by signing in to the **Hewlett Packard Enterprise Support Center** with an entitled account.

Related product websites

HPE 9361-4i RAID Controller (Q2N11A)

1. Go to the Broadcom MegaRAID SAS 9361-4i product page.

   https://www.broadcom.com/products/storage/raid-controllers/megaraid-sas-9361-4i

2. Select the **Documentation** tab.
3. Under **User Guide** select:
   - *12Gb/s MegaRAID SAS RAID Controllers User Guide*
   - *12Gb/s MegaRAID SAS Software User Guide*

HPE 3154-8e RAID Controller (Q6M15A)

- **HPE Smart Storage Administrator User Guide**
- *Microsemi Adaptec Smart HBA & RAID - Installation And User's Guide*
  1. Go to the Microsemi Adaptec SmartRAID 3154-8e product page.

     https://storage.microsemi.com/en-us/support/raid/sas_raid/asr-3154-8e/

  2. Select the **Documentation** tab and download the guide.

General websites

Hewlett Packard Enterprise Information Library

www.hpe.com/info/EIL

For additional websites, see **Support and other resources**.

Support and other resources

Accessing Hewlett Packard Enterprise Support

- For live assistance, go to the Contact Hewlett Packard Enterprise Worldwide website:

  http://www.hpe.com/info/assistance

- To access documentation and support services, go to the Hewlett Packard Enterprise Support Center website:

  http://www.hpe.com/support/hpesc

Information to collect

- Technical support registration number (if applicable)
- Product name, model or version, and serial number
- Operating system name and version
- Firmware version
- Error messages
- Product-specific reports and logs
- Add-on products or components
- Third-party products or components

Accessing updates

- Some software products provide a mechanism for accessing software updates through the product interface. Review your product documentation to identify the recommended software update method.
- To download product updates:

  Hewlett Packard Enterprise Support Center

  www.hpe.com/support/hpesc

  Hewlett Packard Enterprise Support Center: Software downloads

  www.hpe.com/support/downloads

  Software Depot

  www.hpe.com/support/softwaredepot

- To subscribe to eNewsletters and alerts:

  www.hpe.com/support/e-updates

- To view and update your entitlements, and to link your contracts and warranties with your profile, go to the Hewlett Packard Enterprise Support Center **More Information on Access to Support Materials** page:

  www.hpe.com/support/AccessToSupportMaterials

IMPORTANT: Access to some updates might require product entitlement when accessed through the Hewlett Packard Enterprise Support Center. You must have an HPE Passport set up with relevant entitlements.

Customer self repair

Hewlett Packard Enterprise customer self repair (CSR) programs allow you to repair your product. If a CSR part needs to be replaced, it will be shipped directly to you so that you can install it at your convenience. Some parts do not qualify for CSR. Your Hewlett Packard Enterprise authorized service provider will determine whether a repair can be accomplished by CSR.

For more information about CSR, contact your local service provider or go to the CSR website:

http://www.hpe.com/support/selfrepair

Remote support

Remote support is available with supported devices as part of your warranty or contractual support agreement. It provides intelligent event diagnosis, and automatic, secure submission of hardware event notifications to Hewlett Packard Enterprise, which will initiate a fast and accurate resolution based on your product's service level. Hewlett Packard Enterprise strongly recommends that you register your device for remote support.

If your product includes additional remote support details, use search to locate that information.

Remote support and Proactive Care information

HPE Get Connected

www.hpe.com/services/getconnected

HPE Proactive Care services

www.hpe.com/services/proactivecare

HPE Proactive Care service: Supported products list

www.hpe.com/services/proactivecaresupportedproducts

HPE Proactive Care advanced service: Supported products list

www.hpe.com/services/proactivecareadvancedsupportedproducts

Proactive Care customer information

Proactive Care central

www.hpe.com/services/proactivecarecentral

Proactive Care service activation

www.hpe.com/services/proactivecarecentralgetstarted

Warranty information

To view the warranty information for your product, see the links provided below:

HPE ProLiant and IA-32 Servers and Options

www.hpe.com/support/ProLiantServers-Warranties

HPE Enterprise and Cloudline Servers

www.hpe.com/support/EnterpriseServers-Warranties

HPE Storage Products

www.hpe.com/support/Storage-Warranties

HPE Networking Products

www.hpe.com/support/Networking-Warranties

Regulatory information

To view the regulatory information for your product, view the *Safety and Compliance Information for Server, Storage, Power, Networking, and Rack Products*, available at the Hewlett Packard Enterprise Support Center:

www.hpe.com/support/Safety-Compliance-EnterpriseProducts

Additional regulatory information

Hewlett Packard Enterprise is committed to providing our customers with information about the chemical substances in our products as needed to comply with legal requirements such as REACH (Regulation EC No 1907/2006 of the European Parliament and the Council). A chemical information report for this product can be found at:

www.hpe.com/info/reach

For Hewlett Packard Enterprise product environmental and safety information and compliance data, including RoHS and REACH, see:

www.hpe.com/info/ecodata

For Hewlett Packard Enterprise environmental information, including company programs, product recycling, and energy efficiency, see:

www.hpe.com/info/environment

Documentation feedback

Hewlett Packard Enterprise is committed to providing documentation that meets your needs. To help us improve the documentation, send any errors, suggestions, or comments to Documentation Feedback (**docsfeedback@hpe.com**). When submitting your feedback, include the document title, part number, edition, and publication date located on the front cover of the document. For online help content, include the product name, product version, help edition, and publication date located on the legal notices page.