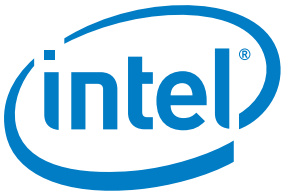

Intel® Xeon Phi™ Processor Software

User's Guide

September 2017

Copyright © 2017 Intel Corporation

All Rights Reserved

US

Revision: 2.3

World Wide Web: http://www.intel.com

Legal Disclaimer

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting: http://www.intel.com/design/literature.htm

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at http://www.intel.com/ or from the OEM or retailer.

No computer system can be absolutely secure.

Intel, Xeon, Xeon Phi and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

Intel does not warrant or guarantee the performance or compatibility of third party commercial products. Reference in this site to any specific commercial product, process, or service, is for the information and convenience of the public, and does not constitute endorsement, or recommendation by Intel.

*Other names and brands may be claimed as the property of others.

Copyright© 2017, Intel Corporation. All rights reserved.

Table of Contents

1 Introduction 6

1.1 Notational conventions 6
1.2 Terminology 6

2 Intel® Xeon Phi™ Processor Software Overview 7

2.1 Kernel 7
2.2 Kernel tools 7
2.3 The cpuid Package 7
2.4 The hwloc Package 7
2.5 The mcelog Package 7
2.6 The memkind Library 8
2.7 The micperf Package 8
2.8 The systools-sb Package 8

3 Intel® Xeon Phi™ Processor Software installation, upgrade and uninstallation ... 9

3.1 Prerequisites 9
3.2 Root Access 10
3.3 Distribution packages replacement 10
3.4 Installation 10
3.4.1 Get the Intel® Xeon Phi™ Processor Software distribution 10
3.4.2 Intel® Xeon Phi™ Processor Software Uninstall 11
3.4.3 Intel® Xeon Phi™ Processor Software Installation 11
3.5 Rebuilding Intel® Xeon Phi™ Processor Software based Package Locally 11

4 Kernel Support for Intel® Xeon Phi™ Processor x200 Product Family 13

4.1 Overview 13
4.2 Huge pages 14
4.2.1 Overview 14
4.2.2 Huge Pages on Red Hat* Enterprise Linux* 14
4.2.3 Huge Pages on SUSE* Linux* Enterprise Server 15
4.2.4 Allocate all MCDRAM for 1G Pages 16

5 User Space Components not delivered with Intel® Xeon Phi™ Processor Software 17

5.1 Development Tools 17
5.1.1 Intel® Xeon Phi™ Processor Enabled OS Distribution Versions 17
5.1.2 Processor Enabled Versions of the User Space Components 17

6 Known Issues 18

6.1 General issues 18
6.2 Performance issue in cache memory mode 19

7 References 22

List of Figures

Figure 1 Components of a Linux* Distribution 14

List of Tables

Table 1 Validated Host Operating Systems (Linux*) 9

Revision History

Date	Revision	Description
August 2017	2.3	Removed deprecated sections.
May 2017	2.2	Minor document revision, corrected formatting and wording in several sections.
January 2016	2.1	Removed deprecated content from former section 7, merged former section 6 into section 5. Updated the supported OS list.
December 2016	2.0	"Workarounds" section has been updated. Removed Microsoft Windows* from supported OS list
November 2016	1.9	"Workarounds" section has been updated. Clarified information DTS tools.
November 2016	1.8	"Workarounds" section has been modified to reflect latest information about provided software. Updated supported kernels table.
August 2016	1.7	Expanded mcelog section.
August 2016	1.6	Added "Workarounds" chapter.
July 2016	1.5	Updated install section to contain information how to handle early ship software versions.
June 2016	1.4	Corrected trademarks.
April 2016	1.3	Updated known-issues section.
February 2016	1.2	Added Microsoft Windows* support information.
December 2015	1.1	Fixed OS table support.
December 2015	1.0	Initial official version.
September 2015	0.5	Draft revision for review.

§

1 Introduction

Intel® Xeon Phi™ Processor Software is a set of software and utilities that enable functionalities of the Intel® Xeon Phi™ processor x200. This document will allow its readers to understand and utilize those features.

This paper is meant to serve as a guide; usage and options are subjective to the customer's needs.

Please note that this document pertains only to systems containing at least one Intel® Xeon Phi™ processor x200.

1.1 Notational conventions

zypper rm <package>	Commands and their arguments in prose sections are *italicized*.
packages/x86_64/core/	Files and directories in prose sections are *italicized.*
`COURIER text`	Code and commands entered by the user. A backslash symbol: \ indicates that command is continued in the next line.
`Italic COURIER text`	Terminal output by the computer.
`[host]$`	Commands that do not require root privileges
`[host]#`	Commands that require root privileges.

1.2 Terminology

DTS	Developer Tool Set
Upstream kernel	The Linux* kernel source code from www.kernel.org
gcc	The GNU C Compiler collection
gdb	The GNU Debugger
EDAC	Error Detection and Correction infrastructure in Linux kernel, Its purpose is to detect problems with the hardware in a system running Linux*.
PM	Power Management
PMU	Performance Monitoring Unit, is a set of counters used to understand events happening inside a CPU
MCDRAM	High Bandwidth memory found in the processor package.
MCE	Machine Check Exception
memkind	Helper library allows direct memory allocations in the MCDRAM.

§

2 Intel® Xeon Phi™ Processor Software Overview

2.1 Kernel

Linux* Kernel delivered with Intel® Xeon Phi™ Processor Software is based on an OS distribution kernel. Intel® Xeon Phi™ Processor Software also contains specific additions in form of patches which enable different core functionalities of the Intel® Xeon Phi™ processor x200. These functionalities are described further in this document.

2.2 Kernel tools

Please note that the *kernel-tools* package is only delivered for Red Hat* Linux* 7.2 distribution. It consists of the following tools:

- *cpupower* - shows and sets processor power related values
- *turbostat* - reports processor frequency and idle statistics
- *x86_energy_perf_policy* - read or write MSR_IA32_ENERGY_PERF_BIAS

2.3 The cpuid Package

Cpuid is a user space tool that provides an interface for querying information about the x86 CPU.

2.4 The hwloc Package

The *hwloc* (Hardware Locality) package provides an abstraction of the system's architecture and topology, including CPU's, caches, memory, processing threads, and NUMA modes. It also offers a C API to gather information about hardware, bind processes, and much more [1].

2.5 The mcelog Package

mcelog is a utility that collects and decodes Machine Check Exception data. It can be run either as a daemon, or by *cron*. More detailed information about *mcelog* can be found at http://www.mcelog.org/ and http://www.linux-kongress.org/2010/slides/lk2010-mcelog-kleen.pdf

Mcelog coexists in system with the EDAC driver. Both mechanisms work independently, although their functionalities may overlap. Both tools were enabled for the KNL platform and their output was validated. The choice which system should be used depends on the needs and expectations of the system administrator. While *mcelog* is more flexible by giving the user possibility to configure some options, EDAC, as a part of kernel, can be considered more reliable. Having both components activated at the same time is also possible. If the system has both components up, configured and running, each memory error should be reported by both *mcelog* and

EDAC. By default EDAC outputs errors to the kernel ring buffer (*dmesg*) while *mcelog* appends them to the syslog (*/var/log/messages*).

Status of each component can be checked using below commands:

- for mcelog (status of the mcelog service should be "active (running)"):

```
[host]$ systemctl status mcelog
```

- for EDAC (both edac_core and sb_edac modules should be loaded):

```
[host]$ lsmod | grep edac
```

2.6 The memkind Library

The *memkind* library is a user-extensible heap manager, designed to provide efficient allocation mechanism for multithreaded applications and support for high bandwidth memory (MCDRAM).

The standard *memkind* API provides a set of standard heap management functions, each one prefixed by *memkind_**. Additional parameters specify the heap management *"kind"*. The standard API also includes functionality for managing *kinds*, error handling and debugging. To find out more about the *memkind* API please refer to its man page or the *README* file.

Further reference is available in the *Intel® Xeon Phi™ Processor Programming and Leveraging High Bandwidth Memory whitepaper rev. 0.5*. This document can be downloaded from Intel® IPS or CDI Doc #570827.

The source code repositories can be found at http://memkind.github.io/memkind/.

2.7 The micperf Package

Micperf is designed to incorporate a variety of benchmarks into a simple user experience with a single interface for execution and a unified means of data inspection. The user interface consists of five executables: one for execution of benchmarks (*micprun*), and four that interpret the output of the first one. The results can be displayed as professional quality plots, human readable text or comma separated value output that can be easily imported into a variety of other applications.

The *micprun* executable, the primary application in the *micperf* package, executes six benchmarks: MKL [2] SMP Linpack, MKL HPLinpack [3], MKL HPCG [4], MKL SGEMM, MKL DGEMM, and STREAM [5], [6]. These benchmarks were carefully chosen to demonstrate performance in all of the major bottlenecks in the system.

2.8 The systools-sb Package

The *Systools-sb* package contains the *SysDiag* tool which provides a variety of information and diagnostics for the processor. *SysDiag* also monitors DDR, MCDRAM and PCI-E information. It also monitors temperature and performance state data of the CPU. For detailed information execute the *SysDiag* tool help.

§

3 Intel® Xeon Phi™ Processor Software installation, upgrade and uninstallation

This chapter describes how Intel® Xeon Phi™ Processor Software installation and configuration.

Note: Before proceeding with installation, please read through this chapter to ensure all required components and facilities are available. Following these installation steps in the presented order is strongly recommended.

3.1 Prerequisites

It is necessary that your system contains at least one Intel® Xeon Phi™ processor x200.

The table below lists major Linux* distributions Intel® Xeon Phi™ Processor Software was validated against.

Table 1 Validated Host Operating Systems (Linux*)

Supported OS Versions	Kernel Version
CentOS* 7.2	kernel-3.10.0-327.36.3.el7.xppsl_1.5.3
CentOS* 7.3	kernel-3.10.0-514.el7
Red Hat* Enterprise Linux* 7.3	kernel-3.10.0-514.el7
Red Hat* Enterprise Linux* 7.4	kernel-3.10.0-693.el7
SUSE* Linux* Enterprise Server 12 SP2	kernel-default-4.4.21-69.1
SUSE* Linux* Enterprise Server 12 SP3	kernel-default-4.4.73-5

To obtain the host's running kernel version execute:

```
[host]$ uname –r
```

Note: Some packages that will be installed require access to the standard distribution packages and repositories. If you disabled any of the standard repositories, please consider re-enabling them to prevent *failed dependency* issues. For more information, refer to documentation provided by your operating system vendor.

3.2 Root Access

Many of the tasks described in this document require administrative access privileges (i.e. root access). Verify that you have such privileges to the machines you will configure.

The use of *sudo* to acquire root privileges should be done carefully because its use may cause subtle and undesirable side effects. *Sudo* might not retain the non-root environment of the caller. This could, for example, result in use of a different *PATH* variable than expected, ending up with execution of the wrong code.

When *su* is used to become root, the non-root environment is (mostly) retained. (*HOME, SHELL, USER, LOGNAME* are reset unless the *-m* switch is given. See the *su* man page for details).

3.3 Distribution packages replacement

Please note, that installing Intel® Xeon Phi™ Processor Software may replace some of pre-installed packages that come with your OS distribution. Packages that may be replaced are listed below:

- *cpuid*
- *cpupower*
- *hwloc*
- *mcelog*
- *memkind*

3.4 Installation

The following process will **not** replace your current Linux* kernel. In some Linux* distributions the installation may add a new kernel to grub, so it is be possible to choose the Intel® Xeon Phi™ Processor Software kernel on startup. In such cases, the newly installed kernel contains information about Intel® Xeon Phi™ Processor Software version, possible kernel names are described in Table 1.

3.4.1 Get the Intel® Xeon Phi™ Processor Software distribution

The latest Intel® Xeon Phi™ Processor Software distribution can be obtained from the http://www.software.intel.com website. Software releases are available in separate tar files for each supported OS. Download the appropriate package for your operating system.

After downloading, extract the release package:

```
[host]$ tar xvf xppsl-<version>-<os>.tar
[host]$ cd xppsl-<xppsl-version>/
```

3.4.2 Intel® Xeon Phi™ Processor Software Uninstall

To check for a previously installed version of Intel® Xeon Phi™ Processor Software execute:

```
[host]$ rpm -qa | grep xppsl
```

Packages that correlate with Intel® Xeon Phi™ Processor Software will be listed and have to be uninstalled:

RHEL*/CentOS*:

```
[host]# yum remove [package-name]
```

SLES*:

```
[host]# zypper rm [package-name]
```

3.4.3 Intel® Xeon Phi™ Processor Software Installation

It is necessary to remove the package completely prior to installation.

RHEL*/CentOS*:

```
[host]$ cd rhel<os-version>/
```

Install RPMs:

```
[host]$ yum install x86_64/*rpm
```

SLES*:

```
[host]$ cd suse<os-version>/
```

Install RPMs:

```
[host]$ zypper install x86_64/*rpm
```

3.5 Rebuilding Intel® Xeon Phi™ Processor Software based Package Locally

The source code for user space tools is included in Intel® Xeon Phi™ Processor Software for both Red Hat* Enterprise Linux* and SUSE* Linux* Enterprise Server. The quickest way to handle the *.*src.rpm* files is to use the *rpmbuild* command. Please follow steps described below for instructions on rebuilding RPM files.

Go to your Intel® Xeon Phi™ Processor Software directory:

CentOS*:

```
[host]$ cd centos*/srpms/
```

RHEL*:

```
[host]$ cd rhel*/srpms/
```

SLES*:

```
[host]$ cd suse*/srpms/
```

Build the RPM package with the following command:

```
[host]$ rpmbuild --rebuild <source_rpm_file>
```

§

4 Kernel Support for Intel® Xeon Phi™ Processor x200 Product Family

The Intel® Xeon Phi™ processor x200 product family requires changes to various pieces of the current Linux* distribution; these changes are being released as patches and RPM source/binary packages, providing a specific version of the Linux* kernel, user space libraries and other utilities.

These changes are planned to be released as part of the associated open source projects. In addition, Intel® is working with Linux* vendors to provide support for the processor.

4.1 Overview

Linux* vendors, such as Red Hat* and SUSE*, take the power of open source software and make it available for enterprises through *distributions* like Red Hat* Enterprise Linux* (RHEL*) [7] or SUSE* Linux* Enterprise Server (SLES*) [8]. In addition to collecting a set of components, Linux* vendors also test and certify their entire distribution and provide support.

A Linux* distribution includes a Linux* kernel, and several other important pieces of open source software such as GNU shell utilities, compilers (*gcc, binutils,* etc) and tools/libraries (*mcelog, hwloc*, etc), daemons, the graphical desktop (X server) and bootloaders like GRUB. Individual vendors also include software built in-house by that company. All of these pieces come together as a single product we think of as the operating system (OS). Additionally, companies like Red Hat* and SUSE* patch the source code in their distributions by picking up bug fixes (for functional, performance or security related issues), perform extensive testing to certify the entire distribution, and provide support (assurance) in case their customers encounter problems.

The Linux* upstream kernel from http://www.kernel.org undergoes many changes between the day the base version is selected by a vendor for inclusion in a particular distribution release and the day that release is shipped. Figure 1 tries to depict how a Linux* kernel for a release of a distribution such as RHEL*/SLES* is created.

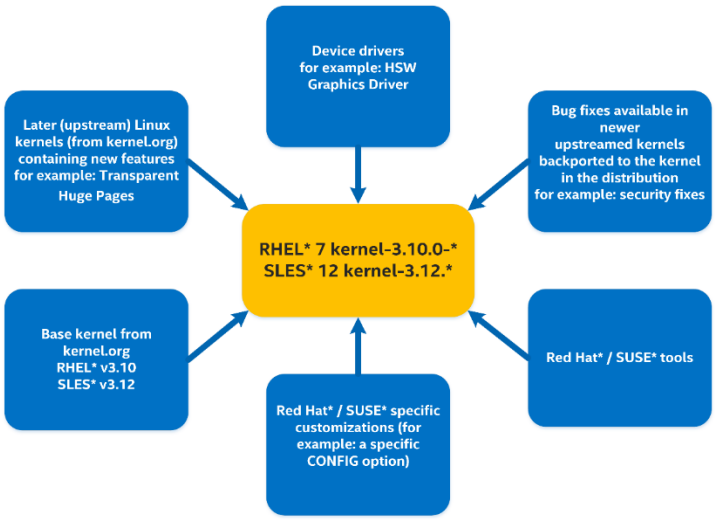

Figure 1 Components of a Linux* Distribution

The URL below captures current and planned RHEL releases along with the specific base Linux* kernel version for each release: https://access.redhat.com/articles/3078

An article discussing how different Linux* vendors construct their distributions can be found at the following URL http://lwn.net/Articles/486304/

4.2 Huge pages

4.2.1 Overview

Linux* systems support 2 MB and 1 GB huge pages, which can be allocated at boot or at runtime. Huge pages can significantly increase performance, particularly for large memory and memory-intensive workloads.

When huge pages are allocated during boot time, they are distributed equally between nodes. Runtime allocation allows the system administrator to choose which NUMA node to allocate those pages from. However, runtime page allocation can be more prone to allocation failure than boot time allocation due to memory fragmentation.

4.2.2 Huge Pages on Red Hat* Enterprise Linux*

Boot time mode:

1G huge pages on boot-time mode are enabled by default in Red Hat* Enterprise Linux* kernel. To allocate different sizes of huge pages at boot time, use the following command, specifying the number of huge pages. This example allocates four 1GB huge pages and 1024 2MB huge pages:

```
'default_hugepagesz=1G hugepagesz=1G hugepages=4 hugepagesz=2M
hugepages=1024'
```

Change this command line to specify a different number of huge pages to be allocated at boot.

Runtime mode:

Huge pages could be also allocated in the runtime mode on RHEL*/CentOS* systems. To allocate them use following command:

```
[host]# echo <number_of_pages> > \
sys/devices/system/node/node[0-9]*\
/hugepages/hugepages-<size_in_bytes>/nr_hugepages
```

4.2.3 Huge Pages on SUSE* Linux* Enterprise Server

Boot time mode:

The default size of Huge Page in SLES* is 2 MB. Additional configuration is required to enable huge Pages bigger than the default size. Boot time mode distributes huge pages equally between the nodes.

To allocate different sizes of huge pages at boot time, use the following kernel boot parameters, specifying the number of huge pages. This example allocates four 1GB huge pages and 1024 2MB huge pages:

```
'hugepagesz=1G hugepagesz=1G hugepages=4 hugepagesz=2M
hugepages=1024'
```

Change this command line to specify a different number of huge pages to be allocated at boot.

Runtime mode:

Be advised, that default SLES* kernel does not support huge pages in real-time mode. To enable this feature it is necessary to install additional kernel patches, and rebuild kernel with following lines in the kernel *config*:

```
CONFIG_CMA=y
CONFIG_CMA_DEBUG=n
```

Patches to apply:

Kernel Commit SHA	Patch name
bae7f4a	*hugetlb*: add *hstate_is_gigantic()*
a7407a2	*hugetlb:update_and_free_page()*: don't clear *PG_reserved* bit
1cac6f2	*hugetlb:move* helpers up in the file
944d9fe	*hugetlb:add* support for gigantic page allocation at runtime

To allocate huge pages use following command:

```
[host]# echo <number_of_pages> > \
sys/devices/system/node/node\
[0-9]*/hugepages/hugepages-<size_in_bytes>/nr_hugepages
```

4.2.4 Allocate all MCDRAM for 1G Pages

To allocate all MCDRAM for 1G pages is necessary to execute the following commands:

- Enter your platform's BIOS and set the *Treat MCDRAM as Hotplug* node option to *enabled*.
- Add the "*movable_node*" kernel command line– it allows a node to have only movable memory. This option allows the following two things: when the system is booting, node full of *hotpluggable* memory can be arranged to have only movable memory so that the whole node can be hot-removed (specifying the *movable_node* boot option is required). Once the system is up, the option allows users to online all the memory of a node as movable memory so that the whole node can be hot-removed. Users who do not use the memory *hotplug* feature can leave this option on since they do not specify *movable_node* boot option, or they do not online memory as movable.

§

5 User Space Components not delivered with Intel® Xeon Phi™ Processor Software

5.1 Development Tools

User space components like *gcc, binutils* and *gdb* have been updated to include support for AVX-512 code. However, the versions of these components shipped in a Linux* distribution are selected by the Linux* vendor and might not include the updated versions. Consult sections below for further assistance.

5.1.1 Intel® Xeon Phi™ Processor Enabled OS Distribution Versions

RHEL* will have full user space support for AVX-512 processor features. The customer will get support from the Linux* vendor and receive any qualifications required from that vendor.

5.1.1.1 Red Hat* Developer Toolset (DTS) Version 3 or later

For customers using Red Hat*, DTS is available at:

https://developers.redhat.com/products/developertoolset/overview/

DTS 3 (and later) provides optional versions of *gcc, gdb* and *binutils*. These optional versions are not replacements for the main tools in the distribution, but provide alternate versions of *gcc* 4.9, *binutils* 2.24 and *gdb* 7.8, which are enabled for AVX-512.

5.1.2 Processor Enabled Versions of the User Space Components

The customer can build the open source versions of *gcc*, *binutils* and *gdb* which support AVX-512 and install them as an optional tool chain. By using upstreamed versions, customers can get support for those components from the developer community.

§

6 Known Issues

The Intel® Xeon Phi™ Processor x200 platform-specific features have been enabled in both Linux* upstream kernel and vendor kernels, therefore, provided the system was set up in accordance to this guide, user should be able to fully utilize the hardware. However, some issues cannot be directly addressed in kernel, or the solution cannot be upstreamed for some reason. This chapter describes such problems and shows possible ways to eliminate or mitigate their consequences.

6.1 General issues

1. Package *debuginfo* type conflicts with distribution/upstream packages

2. The *hwloc* memory side cache discovery might fail when SELinux MLS policy is enforced. Install the *hwloc* policy module to mitigate this issue. Please note, that this module requires the *hwloc-dump-hwdata* files to be present in */var/run/hwloc.*

   Prerequisites:

   - *policycoreutils* with SELinux scripts
   - *selinux-devel* to build policy.

   Use the following command to check if the *hwloc* module is installed:

   ```
   [host]$ semodule -l | grep hwloc
   ```

   Build it manually in case it is missing from your system. It is required to obtain the policy from the SELinux repo:

   ```
   [host]$ git clone \
   https://github.com/TresysTechnology/refpolicy-contrib
   [host]$ cd refpolicy-contrib
   [host]$ make -f /usr/share/selinux/devel/Makefile hwloc.pp
   ```

   Run the following command to install the module:

   ```
   [host]$ semodule -i ./hwloc.pp
   ```

3. In RHEL* 7.2, to achieve the best possible MCDRAM bandwidth performance the OS must be booted in tickles mode. For more information please see the *tickless_xppsl.pdf* document.

4. The *hwloc* service requires the *dmi-sysfs* Linux kernel module to be loaded. Create an appropriate entry in */etc/modules-load.d/* to load it automatically. Use the command below.

   ```
   [host]# echo "dmi-sysfs" > /etc/modules-load.d/dmi_sysfs.conf
   ```

6.2 Performance issue in cache memory mode

PROBLEM:

The cache mode design places MCDRAM as a direct mapped cache. On Linux* systems this design causes cache performance degradation over time due to increased number of cache collisions caused by memory fragmentation.

SOLUTION:

Use the page sorting module provided in Intel® Xeon Phi™ Processor Software.

INSTALLATION:

If the Intel® Xeon Phi™ Processor Software is installed and running on your system, the correct module is already installed and can be used; proceed to the "Usage" section.

If your machine is running one of the supported vendor kernels, install the correct kernel module package by following the steps below.

1. Navigate to the directory containing binary packages for Intel® Xeon Phi™ Processor Software.

   ```
   [host]# cd xppsl-<xppsl-version>/<os-version>/rpms/x86_64/
   ```

2. Install the kernel module package:

RHEL*/CENTOS*:

```
[host]# yum install kmod-xppsl-addons-*.x86_64.rpm
```

SLES*:

```
[host]# zypper install xppsl-addons-kmp-default-*.x86_64.rpm
```

USAGE:

The module sorts kernel free memory pages lists in a way that further minimizes cache misses when those pages are acquired by user processes. Since the module operates on free pages, it is suggested to employ sorting before running each user application.

Furthermore, due to high memory fragmentation, sorting pages alone may not be sufficient to restore initial performance. That is why it is mandatory to use memory compaction beforehand, which increases the amount of groups of physically-contiguous pages. To achieve best efficiency compaction ought to be used before sorting (see example).

Sorting can be called on-demand similar to the example below:

1. Load the module:

   ```
   [host]# modprobe zonesort_module
   ```

2. Trigger memory compaction:

   ```
   [host]# echo 1 > /proc/sys/vm/compact_memory
   ```

3. Trigger sorting (the call returns once sorting completes):

```
[host]# echo <numa_node*> > \
/sys/kernel/zone_sort_free_pages/nodeid
```

*- currently numa_node can only be set to 0, for details please refer to section "remarks" below

Alternatively, you can configure sorting to trigger automatically with an interval:

1. Load the module:

```
[host]# modprobe zonesort_module
```

2. Set the interval of periodic sorting:

```
[host]# echo <interval_in_sec> > \
/sys/kernel/zone_sort_free_pages/sort_interval
```

Note that in case of periodic sorting:

- The action will always be taken on all online nodes. Unlike using *zone_sort_free_pages/nodeid* interface, the node to be sorted cannot be chosen.
- Writing value 0 (zero) disables periodic sorting and cancels all pending activities (if the sorting is ongoing it will finish nonetheless).
- Memory compaction has to be handled by THE system administrator. The module does not call it internally.
- On-demand sorting is disabled. Writing to *zone_sort_free_pages/nodeid* while *zone_sort_free_pages/sort_interval* is set to non-zero value will return *EBUSY*.

ADMINISTRATION:

By default, due to security reasons, all interfaces exposed by the module can be written to only by superuser. If the permissions are to be modified it is recommended to do that through the *udev* manager, as in the example below:

1. Create the file */etc/udev/rules.d/99-zonesort.rules* with the contents:

```
ACTION=="add", DEVPATH=="/module/zonesort_module",
SUBSYSTEM=="module", RUN+="/bin/chmod 0666
/sys/kernel/zone_sort_free_pages/sort_interval
/sys/kernel/zone_sort_free_pages/nodeid"
```

2. Reload the *udev* rules to apply changes:

```
[host]# udevadm control --reload-rules
```

The inserted rule changes access permissions to the interfaces every time the module is being loaded.

DEBUGGING:

The module exposes additional interfaces, which may be useful for identifying the state of the running system:

A. *buddy_lists*

   Provides details of the current state of the kernel buddy allocator. In order to use it, dump its contents to a file:

   ```
   [host]# cat /sys/kernel/debug/buddy_lists  > output_file
   ```

B. *directmappedcache_state*

   Provides information similar to */proc/pagetypeinfo* but extended for the purpose of direct mapped cache debugging. The data can be obtained by printing the entry to standard output:

   ```
   [host]# cat /sys/kernel/debug/directmappedcache_state
   ```

For further details on how to interpret the results please refer to the source code of the module, which is delivered with the Intel® Xeon Phi™ Processor Software.

REMARKS:

- The module does not support explicitly allocated huge pages.
- The module has been validated for stock kernels of supported OS distributions (see Table 1). There is no guarantee the module will be functional when used with other kernels.
- The module does not support hybrid memory mode in any of the cluster modes.
- SNC4 and SNC2 cluster modes in the hybrid and cache memory modes are not supported.

§

7 References

[1] https://www.open-mpi.org/projects/hwloc/

[2] Intel Math Kernel Library (Intel MKL) 11.0. http://software.intel.com/en-us/intel-mkl

[3] Jack Dongarra, Piotr Luszczek, and Antoine Petitet. The linpack benchmark: past, present and future. Concurrency and Computation: Practice and Experience, 15(9):803–820, 2003.

[4] http://www.hpcg-benchmark.org/

[5] John D. McCalpin. Stream: Sustainable memory bandwidth in high performance computers. Technical report, University of Virginia, Charlottesville, Virginia, 1991-2007. A continually updated technical report. http://www.cs.virginia.edu/stream/.

[6] John D. McCalpin. Memory bandwidth and machine balance in current high performance computers. IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter, pages 19–25, December 1995.

[7] http://www.redhat.com/en/technologies/linux-platforms

[8] https://www.suse.com/products/server/

§