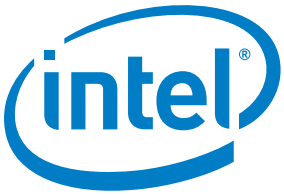

# Intel® Xeon Phi™ Processor Software

## User's Guide

## November 2016



US

Revision: 1.8

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.

# Table of Contents

# List of Figures



# List of Tables

# Revision History

| Date | Revision | Description |
|---|---|---|
| November 2016 | 1.8 | "Workarounds" section has been modified to reflect latest information about provided software.<br>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.

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

## 1.1 Terminology

| | |
|---|---|
| **DTS** | Developer Tool Set ( https://access.redhat.com/documentation/en-US/Red_Hat_Developer_Toolset/3/index.html) |
| **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. |

## 1.2 Operating Systems Support

Intel® Xeon Phi™ processor software supports two types of operating systems:

- Linux* Kernel-based operating systems, refer to Section 3.1 for more information.
- Microsoft Windows* Server 2016 TP5

§

# 2 Linux* Intel® Xeon Phi™ Processor Software Content

## 2.1 Kernel

Linux* Kernel delivered with Intel® Xeon Phi™ processor software is based on an OS distribution kernel. Intel® Xeon Phi™ processor software specific additions are 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* 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 Cpupower package

Please note that the *cpupower* package is only delivered for SUSE* Linux* distribution. It consists of the following tools:

*cpupower* - shows and sets processor power related values

*turbostat* - reports processor frequency and idle statistics

## 2.4 The cpuid Package

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

## 2.5 The hwloc Package

The *Portable Hardware Locality* (*hwloc*) software package provides a portable abstraction (across OS, versions, architectures, etc) of the hierarchical topology of modern architectures, including NUMA memory nodes, shared caches, processor sockets, processor cores and processing units (logical processors or "threads"). It also gathers various system attributes such as cache and memory information. It primarily aims at helping applications with gathering information about modern computing hardware so as to utilize it accordingly and efficiently. *Hwloc* may display the topology in multiple convenient formats. It also offers a powerful programming interface (C API) to gather information about the hardware, bind processes, and much more.

## 2.6 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/

http://www.linux-kongress.org/2010/slides/lk2010-mcelog-kleen.pdf

*Mcelog* coexists in system with EDAC driver and both mechanism work independently although their functionalities may overlap. Both have been enabled for KNL platform and their output has been validated. The choice which system should be used depends on the needs and expectations of 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 kernel ring buffer (dmesg) while *mcelog* appends them to sysylog (/var/log/messages).

Status of each component can be checked using below commands:

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

- for edac (both edac_core and sb_edac modules should be loaded):
  ```
  $ lsmod | grep edac
  ```

## 2.7 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 *memkind* library is built on top of *jemalloc* and enables partitioning of the heap between kinds of memory in NUMA-capable systems.

There are several strategies (*memkind kinds*) of heap management provided out-of-the-box by the library, such as allocating from standard or high bandwidth memory, as well as using standard or huge pages (both 2 MB and 1 GB sizes).

Heap management strategy can be adjusted either by using one of the predefined *kinds* exampled above, or user-created ones, which address application specific needs. More information about the predefined *kinds* and creating custom ones can be found in the *memkind* manual (sections KINDS and MEMKIND OPERATIONS) and *memkind* examples ("*new_kind_example.c*").

The *memkind* library provides full compatibility with ISO C standard APIs.

The high bandwidth memory interface (*hbwmalloc* API) is a set of standard heap management functions such as *malloc, calloc, realloc* and *free*, prefixed by *hbw_ **. This API also provides *hbwmalloc_allocator* class compatible with the C++ standard library allocator concepts, and features policy that determines behavior when there is not enough free high bandwidth memory to satisfy a user's request. To find out more about the *hbwmalloc* API please refer to its man page.

The standard *memkind* API provides a set of standard heap management functions, each one prefixed by *memkind_** and with additional parameter to specify the *kind*. The standard API also includes functionality for *kind* management, error handling and debugging. To find out more about the *memkind* API please refer to its man page.

More information about installing and using the *memkind* library can be found in its README file.

The source code repositories, and additional information can be found at http://memkind.github.io/memkind/

## 2.8 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 [3] SMP Linpack [4], MKL SGEMM, MKL DGEMM, SHOC [5] download, SHOC readback, and STREAM [6], [7]. These benchmarks were carefully chosen to demonstrate performance in all of the major bottlenecks in the system.

## 2.9 The systools-sb Package

*Systools-sb* package contains *SysDiag* tool which provide a variety of information and diagnostics for the processor.

*SysDiag* tool provides DDR memory information, MCDRAM information, and PCI-E information. It also provides temperature and performance state information of the CPU.

For detailed information execute *SysDiag* tool help.

§

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

This chapter describes how Intel® Xeon Phi™ processor software can be installed and configured.

***Note:*** It is strongly recommended to read through this chapter before actually proceeding with installation to ensure that all required components and facilities are available. It is also strongly recommended that these installation steps be performed in the order they are presented.

***Note:*** All software packages provided for the Intel® Xeon Phi™ processor x200 are marked with the **xppsl** label. This document assumes that the system does not contain an early ship version of the software, which might have been labelled differently. It is necessary to remove any early ship packages from your system before following the steps below. Instructions on how to remove those packages are provided in the early ship software user's guide.

## 3.1 Prerequisites

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

Intel® Xeon Phi™ processor software has been validated against specific versions of CentOS* and SUSE* Linux* Enterprise Server (SLES*) as the main operating system. Table 1 lists the versions of these operating systems.

**Table 1 Validated Host Operating Systems (Linux*)**

| Supported OS Versions | Kernel Version |
|---|---|
| CentOS* 7.2 | kernel-3.10.0-327.36.3.el7.xppsl_1.4.3 |
| SUSE* Linux* Enterprise Server 12 | kernel-default-3.12.61-52.54.31.gaec1a363.12.5.xppsl_1.4.3 |
| SUSE* Linux* Enterprise Server 12 SP1 | kernel-default-3.12.66-60.64.8.273.g9e1b23.xppsl_1.4.3 |
| SUSE* Linux* Enterprise Server 12 SP2 | kernel-default-4.4.21-69.xppsl_1.4.3 |

To obtain the version of the kernel running on the host, execute:

```
$ uname –r
```

***Note:*** Some packages that will be installed require access to the standard distribution packages and repositories. If you disabled any of standard repositories please consider

re-enabling them to prevent *failed dependencies* issues. To get more information please check the information provided by your operating system documentation – for Red Hat* Enterprise Linux*, and for SUSE* Linux* Enterprise Server.

## 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* environment 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 will replace some of pre-installed packages that come with your OS distribution. Packages that will be replaced are listed below:

- *cpuid*
- *cpupower*
- *hwloc*
- *mcelog*
- *memkind*
- *perf-3.12 (SLES 12.0 only)*

## 3.4 Installation

The following process will **not** replace your current Linux* kernel. Installation will add new kernel to grub, so it will be possible to choose the Intel® Xeon Phi™ processor software kernel on startup. 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 software.intel.com. The software package releases are available in separate tar files for each supported OS. It is important to download a package for your operating system.

After downloading, un-tar the package:

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

### 3.4.2 Intel® Xeon Phi™ Processor Software Upgrade

*Yum* and *zypper* both support software upgrades and downgrades. Intel® Xeon Phi™ processor software from version 1.1.2 also supports updates. If you are on version 1.1.2 or above to install newer version of Intel® Xeon Phi™ processor software please follow steps described in Section 3.4.4.

### 3.4.3 Intel® Xeon Phi™ Processor Software Uninstall

To check for a previously installed version of Intel® Xeon Phi™ processor software package execute:

```
$ rpm -qa | grep xppsl
```

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

- Red Hat* Enterprise Linux*/CentOS*:

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

- SUSE* Linux* Enterprise Server

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

### 3.4.4 Intel® Xeon Phi™ Processor Software Installation

**Red Hat* Enterprise Linux*/CentOS*:**

```
$ cd rhel<os-version>/
```

Install RPMs:

```
$ yum install x86_64/*rpm
```

**CentOS*:**

```
$ cd centos<os-version>/
```

Install RPMs:

```
$ yum install x86_64/*rpm
```

**SUSE* Linux* Enterprise Server:**

```
$ cd suse<os-version>/
```

Install RPMs:

```
$ zypper install noarch/*rpm x86_64/*rpm
```

***Note:*** In rare cases *zypper* might not be able to find all dependencies returning a *Failed dependencies* error message. The solution to this issue is manual installation of the missing software:

```
$ cd suse<os-version>
$ zypper install noarch/kernel-macros-3.12.28-4.6.xppsl_ \
<xppsl-version>.noarch.rpm noarch/ \
kernel-devel-3.12.28-4.6.xppsl_<xppsl-version>.noarch.rpm
$ zypper install noarch/*rpm x86_64/*rpm
```

If the following error occurs:

*"The selected package 'kernel-devel-3.12.28-4.6.*xppsl_<xppsl-*version>.noarch' from repository 'Plain RPM files cache' has lower version than the installed one."*

Please use the command below to force install the package.

```
$ zypper install --oldpackage kernel-devel-3.12.28- \
4.6.xppsl_<xppsl-version>.noarch
```

***Note:*** Update the additional *devel* and *debuginfo* packages if they were installed with the previous version of the software. Not updating these packages will result in dependency conflicts when running the commands above.

# 4 Rebuilding Linux* Intel® Xeon Phi™ Processor Software based Package Locally

Typically an RPM is pre-compiled and ready for direct installation. The corresponding source code can also be distributed. This is done in an SRPM package, which also includes the *SPEC* file describing the software and how it is built. The SRPM also allows the user to compile and modify the code.

The source code for user space tools is included in Intel® Xeon Phi™ processor software with 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:

Go to your Intel® Xeon Phi™ processor software directory.

CentOS*:

```
$ cd centos*/srpms/
```

Red Hat* Enterprise Linux*:

```
$ cd rhel*/srpms/
```

SUSE* Linux* Enterprise Server:

```
$ cd suse*/srpms/
```

To build the RPM package, use the following command:

```
$ rpmbuild --rebuild <source_rpm_file>
```

§

# 5 Microsoft Windows* Intel® Xeon Phi™ Processor Software content

## 5.1 The hbmalloc Library

The *hbmalloc* library is designed to provide a programmer with access to the high bandwidth memory (MCDRAM) by allocating pages of MCDRAM memory. Provided API allows users to select allocation strategies, such as allocating with different guarantee levels of returned memory (preferred MCDRAM, guaranteed MCDRAM while allocation time, or locked on MCDRAM node memory) or page type (default or large pages).

§

# 6 Microsoft Windows* Intel® Xeon Phi™ Processor Software Installation

## 6.1 Intel® Xeon Phi™ Processor Software Installation

1) Unzip the *xppsl-<version>-Intel(R) Xeon Phi(TM) server-windows.zip* file.

2) Double-click the *xppsl-<version>-Intel(R) Xeon Phi(TM) server-windows.exe* file.

3) Follow the instructions on screen to complete the installation.

   a. Select *Enable locking pages in memory* checkbox to add privilege *SeLockMemoryPrivilege* for the user (required to allocate large pages).

4) After the installation reboot of the system for the new settings to take effect.

5) If the Windows* security window appears select the **Always trust software from Intel®** checkbox.

6) The default installation path is *C:\Program Files\Intel*\XPPSL\ it can be changed during installation.

## 6.2 Intel® Xeon Phi™ Processor Software Unattended Installation

1) In a command window, navigate to the directory that contains the setup files (For example: *C:\Users\<username>\Downloads\xppsl-<version>- Intel(R) Xeon Phi(TM) server-windows*)

```
User> cd "C:\Users\<username>\Downloads\xppsl-<version>-
Intel(R) Xeon Phi(TM) server-windows"
```

2) Enter the following command:

```
User>"xppsl-<version>-Intel(R) Xeon Phi(TM) server-windows.exe"
/s /v /qn /V"/quiet"
```

When using the unattended installation process, after entering the command it may take a few minutes to complete the installation.

To select *Enable locking pages* checkbox with unattended installation, *LOCKPAGES=1* parameter should be passed. Using */norestart* parameter allows to postpone the system reboot.

```
User>"xppsl-<version>-Intel(R) Xeon Phi(TM) server-windows.exe"
/s /v /qn /V"/quiet /norestart LOCKPAGES=1"
```

## 6.3 Intel® Xeon Phi™ Processor Software Upgrade

Upgrading the Intel® Xeon Phi™ processor software can be achieved by following instructions in the previous section. Users may choose to manually uninstall the previous version or let the installer automatically search and remove previous release prior to installing the latest one.

## 6.4 Intel® Xeon Phi™ Processor Software Uninstall

To uninstall the software stack open the **Control Panel**, choose **Programs and features** and remove the *"Intel(R) Xeon Phi(TM) server"* application.

§

# 7 Known Issues

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

2. Package *xppsl-hwloc-devel* requires enabling RHEL* 7 subscription.

3. Package *xppsl-hwloc-devel* can break Intel® Xeon Phi™ processor software installation due to missing dependency *"pkgconfig(libpciaccess)"*.

   This dependency cannot be satisfied by using RHEL* 7 DVD. Registering and enabling RHEL* subscription is required. To enable subscription please refer to RHEL* 7 user guide. This behavior occurs also for distribution *hwloc-devel* package. Missing package name for RHEL* 7.1 is *"libpciaccess-devel-0.13.1-4.1.el7.x86_64"*.

4. Package *xppsl-hwloc* does not update and needs to be reinstalled manually.

5. The *hbmalloc* library does not guarantee to allocate memory from the closest MCDRAM node when SNC-2 or SNC-4 Memory Mode is used.

6. 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:

   ```
   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:

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

   Run the following command to install the module:

   ```
   semodule -i ./hwloc.pp
   ```

7. Performance comparisons between RHEL* 7.2 and SLES* 12 SP1 based on the STREAM benchmark revealed that memory transfers to/from MCDRAM in SLES* are ~4% faster:

   - SLES* 490 GB/s
   - RHEL* 470 GB/s

Booting RHEL* 7.2 in the tickless mode will rectify this difference. For more information please see *RedHat_tickless_xppsl.pdf*.

8. The *xppsl-micperf-1.4.1* package cannot be upgraded to *xppsl-micperf-1.4.2* or above using *yum* or *zypper*. It is necessary to remove the package completely prior to installing a new version (refer to Section 3.4.4 for installation instructions).

   **RHEL*/CentOS*:**

   ```
   # yum remove xppsl-micperf
   ```

   **SLES*:**

   ```
   # zypper rm xppsl-micperf
   ```

§

# 8 Workarounds

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.

## 8.1 Performance issue in cache memory mode

**PROBLEM:** The cache mode design places MCDRAM as a direct mapped cache. It was observed, however, that on Linux* systems this design causes cache performance degradation over time. It is caused by increased number of cache collisions caused by memory fragmentation, which in turn decreases performance of the system.

**SOLUTION:**

**Page sorting module**

**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 the Intel® Xeon Phi™ Processor Software.

   *# cd xppsl-<xppsl-version>/<os-version>/rpms/x86_64/*

2. Install the kernel module package:

   RHEL*/CENTOS*:
   *# yum install kmod-xppsl-addons-*.x86_64.rpm*

   SUSE*:
   *# zypper install xppsl-addons-kmp-default-*.x86_64.rpm*

**USAGE:**
The module exposes a special *sysfs* interface to the user, allowing them to trigger page sorting on demand. Page sorting can be used during runtime when performance drop has been observed. Follow the instructions below to trigger page sorting.

1. Load the module:
   *# modprobe zonesort_module*

2. Trigger sorting (the call returns once sorting completes):
   *# echo <numa_node> > /sys/kernel/zone_sort_free_pages/nodeid*

**DEBUGGING:**

Standalone kernel does not provide many interfaces that enable debugging of direct mapped cache issues. Moreover, existing interfaces provide information that may be difficult to interpret in context of problems presented in this chapter.

For that reason the described module exposes additional debug helpers, 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:

   *# 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:

   *# 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 along with the Intel® Xeon Phi™ Processor Software.

§

# 9 Linux* 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.

## 9.1 Overview

Linux* vendors, such as Red Hat* and SUSE*, take the power of open source software and make it available for the enterprise through *distributions* like Red Hat* Enterprise Linux* (RHEL*) [1] or SUSE* Linux* Enterprise Server (SLES*) [2]. 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.

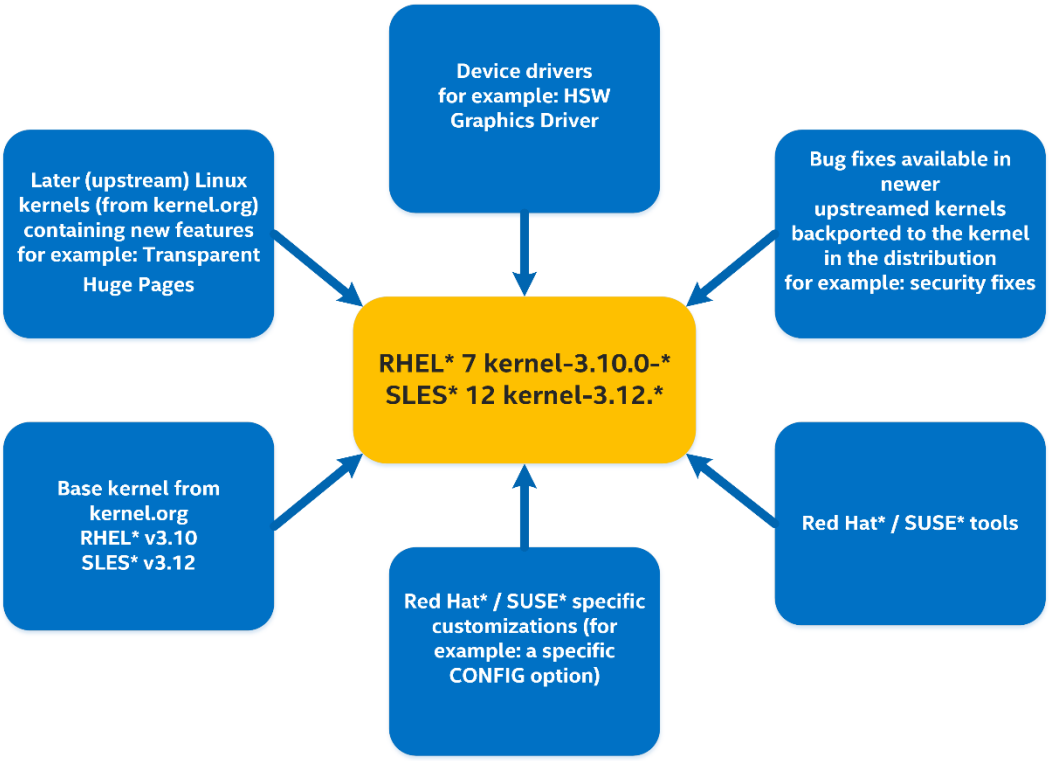


**Figure 1 Components of a Linux* Distribution**

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/

## 9.2 Huge pages

### 9.2.1 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.

### 9.2.2 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 4 1 GB huge pages and 1024 2 MB 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:

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

### 9.2.3 Huge Pages on SUSE* Linux* Enterprise Server

**Boot time mode:**

The default size of Huge Page in SLES* is 2 MB. To enable Huge Pages bigger than default size additional configuration is required. Boot time mode distribute 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 4 1 GB huge pages and 1024 2 MB 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 is not supporting 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:

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

### 9.2.4 Allocate all MCDRAM for 1G Pages

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

- Set *Treat MCDRAM as Hotplug* node to *enabled*. This can be enabled by setting a corresponding option in BIOS.
- Add kernel command line "*movable_node*" – 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). After 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.

## 9.3 Red Hat* Enterprise Linux* Distribution Options

Linux* support for the processor can be accomplished through the selection of different options. The following sections elaborate on each option in detail with their pros and cons, sorted from easiest to hardest from the end user perspective.

Each section contains a table that describes whether a particular feature is usable (noted as "*Enabled*" column) by relying on architectural approach or has been optimized with the processor specific parameters. The "*Vendor support*" row is listed for referencing if the option will likely void a support contract with a Linux* vendor; final word on contract validity is up to the vendor.

### 9.3.1 Intel® Xeon Phi™ Processor x200 Optimized RHEL* Distribution

**Definition:** By general availability of the processor, most if not all patches will be included in the RHEL* 7.3 distribution. (Additional patches that are

not part of RHEL* 7.3 will be available on http://www.intel.com. These patches are not required, but if used will provide optimal processor support. Note that applying those patches may void the support contract with the OS vendor

**Pros:** All the patches required to optimally support the processor features are part of the default installation. The customer gets support from the Linux* vendor and receives qualifications needed from that vendor.

**Cons:** Additional work may be required to port applications, scripts, etc. from an older RHEL* version to RHEL* 7.3. Earlier versions of RHEL* 7.X may contain some of the patches i.e. RHEL* 7.2 contains AVX-512 patch.

| Feature | Status |
|---|---|
| AVX-512 | Supported |
| Power Management | Supported |
| Performance Measurement Unit (PMU) | Supported |
| EDAC | Supported |
| Turbostat | Supported |
| CPU enumeration | Supported |
| Coretemp | Supported |
| Memory management | Supported |
| memkind | Supported |
| mcelog | Supported |
| hwloc | Supported |
| rasdaemon | Supported |
| cpuid | Supported |

## 9.4 SUSE* Linux* Enterprise Server Distribution Options

### 9.4.1 Intel® Xeon Phi™ Processor x200 Optimized SLES* Distribution

**Definition:** By general availability of the processor, most if not all patches will be included in the SLES* 12 SP2 distribution. (Additional patches that are not part of SLES* 12 SP2 will be available on http://www.intel.com. These patches are not required, but if used will provide optimal processor support, but may void the support contract with the OS vendor)

**Pros:** All the patches required to optimally support the processor features are part of the default installation. The customer gets support from the Linux* vendor and receives qualifications needed from that

vendor.

**Cons:** Additional work may be required to port applications, scripts, etc. from an older SLES* version to SLES* 12 or newer. SLES* 12 SP1 will contain only AVX-512 patch.

| Feature | Status |
|---|---|
| AVX-512 | Supported |
| Power Management | Supported |
| Performance Measurement Unit (PMU) | Supported |
| EDAC | Supported |
| Turbostat | Supported |
| CPU enumeration | Supported |
| Coretemp | Supported |
| Memory management | Supported |
| memkind | Supported |
| mcelog | Supported |
| hwloc | Supported |
| rasdaemon | Supported |
| cpuid | Supported |

*** SLES 12.2 is still in BETA phase and some features may not be integrated**

### 9.4.2 SLES* 12 with an Intel®-Provided Kernel.

**Definition:** The customer installs an Intel® provided kernel, based on SLES* 12, which contains all the processor kernel patches.

**Pros:** Allows customers to do early work to utilize all the new features of the processor before their Linux* vendor releases a processor enabled distribution. Also allows customers who are locked into using a version of SLES* with no processor enabling, to utilize the processor's full potential.

**Cons:** The use of a kernel different from the one provided by the Linux* vendor may void the support contract with the OS vendor.

| Feature | Status |
|---|---|
| AVX-512 | Supported |
| Power Management | Supported |
| Performance Measurement Unit (PMU) | Supported |
| EDAC | Supported |
| Turbostat | Supported |
| CPU enumeration | Supported |

| | |
|---|---|
| Coretemp | Supported |
| Memory management | Supported |
| memkind | Supported |
| mcelog | Supported |
| hwloc | Supported |
| rasdaemon | Supported |
| cpuid | Supported |

§

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

## 10.1 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 is selected by the Linux* vendor and might not include the updated versions.

For such components, the following options are available:

### 10.1.1 Intel® Xeon Phi™ Processor Enabled OS Distribution Versions

These versions of 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.

#### 10.1.1.1 Red Hat Developer Toolset (DTS) Version 3

For customers using Red Hat*, DTS is available at

https://access.redhat.com/documentation/en-US/Red_Hat_Developer_Toolset/3/index.html

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. DTS is available for RHEL* 6 and RHEL* 7

### 10.1.2 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.

§

# 11 Conclusions

The addition of new hardware support provided by an enterprise Linux* distribution or Microsoft Windows* is a staged process, where a number of variables come into play. The options provided in this document are not definitive and are meant to serve only as a guide; ultimately the customer needs to decide if any of the options described in this paper fits their needs.

§

# 12 References

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

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

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

[4] 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.

[5] Anthony Danalis, Gabriel Marin, Collin McCurdy, Jeremy S. Meredith, Philip C. Roth, Kyle Spafford, Vinod Tipparaju, and Jeffrey S. Vetter. The scalable heterogeneous computing (shoc) benchmark suite. In Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units, GPGPU '10, pages 63–74, New York, NY, USA, 2010. ACM.

[6] 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/.

[7] 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.

§