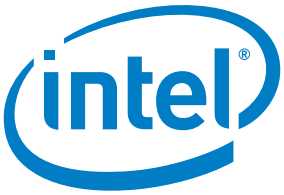


Intel® Xeon Phi™ Processor Software

User's Guide for Linux*

November 2017

US

Revision: 0.6

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

Disclaimer and legal information

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.

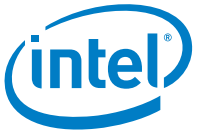
Intel does not warrant or guarantee the performance or compatibility of third party commercial products. Reference in this site to any specific commercial product, process, or service, is for the information and convenience of the public, and does not constitute endorsement, or recommendation by Intel.

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

Revision history

Revision Number	Description	Revision Date
0.6	Updated supported OS list, installation instructions, and added information about the *zonesort* module.	November 2017
0.5	Updated information on OS support.	October 2017
0.4	Updated brand names throughout the document.	September 2017
0.3	Document update for the release of Intel® Xeon Phi™ processor software version 2.0.	August 2017
0.2	Removed redundant and/or unnecessary sections.	June 2016
0.1	Initial draft of the user's guide.	June 2016

Table of contents



List of tables

§

1 Introduction

Intel® Xeon Phi™ processor software is a set of software and utilities that enable functionalities of the Intel® Xeon Phi™ processor. 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.

1.1 Notational conventions

This document uses the following 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
EDAC	Error Detection and Correction infrastructure in Linux* kernel which detects hardware problems.
gcc	The GNU C Compiler collection
gdb	The GNU Debugger
MCDRAM	High Bandwidth memory found in the processor package.
MCE	Machine Check Exception
MKL	Intel® Math Kernel Library
PMU	Performance Monitoring Unit is a set of counters used to understand events happening inside a CPU
RHEL*	Red Hat* Enterprise Linux*
SLES*	SUSE* Linux* Enterprise Server
Upstream kernel	The Linux* kernel source code from www.kernel.org

§

2 Intel® Xeon Phi™ processor software overview

The packages included in Intel® Xeon Phi™ processor software are meant to enable core functionalities of the Intel® Xeon Phi™ processor. If older versions of packages listed in this section are already installed on your host OS, they will be replaced.

2.1 The cpuid package

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

This component is delivered for SUSE* Linux* Enterprise Server 12 SP3.

2.2 The hwloc package

The *Portable Hardware Locality* (*hwloc*) provides hardware information, including NUMA memory nodes, shared caches, processor sockets, processor cores and processing units. It is mainly used by other applications so that they can utilize available hardware efficiently. For more information visit https://www.open-mpi.org/projects/hwloc/.

This component is delivered for Red Hat* Enterprise Linux* and CentOS*.

2.3 The memkind library

The *memkind* library is a user-extensible heap manager that provides an efficient allocation mechanism for multithreaded applications and supports 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 to the *usr/share/doc/memkind/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/.

2.4 The micperf package

Micperf incorporates a variety of benchmarks into a simple and unified user experience. 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 values that can be easily imported into a variety of other applications.

The *micprun* executable, the primary application in the *micperf* package, executes the following benchmarks: *MKL*[1] *SMP Linpack*[2] , *MKL HPLinpack*[2], *MKL HPCG, MKL SGEMM, MKL DGEMM, STREAM*[3], *Deepbench convolutions* (*libxsmm_layer*, *std_conv_bench*)[4] and *fio*[5]*.* These benchmarks were carefully chosen to demonstrate performance in all of the major bottlenecks in the system.

For more information please refer to the *micperf_users_guide.pdf* document included in the Intel® Xeon Phi™ processor software release package.

2.5 The sysdiag package

The *sysdiag* package contains the *SysDiag* processor diagnostics tool. *SysDiag* can monitor DDR, MCDRAM, PCI-E information, CPU temperature and CPU performance status data.

2.6 The zonesort package

The *zonesort* package contains the *zonesort* kernel module, which prevents cache performance degradation due to cache collisions caused by memory fragmentation. Refer to Appendix A.2 for more information.

§

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

[2] 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

[3] 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/.

[4] https://github.com/baidu-research/DeepBench

[5] https://github.com/axboe/fio

3 Intel® Xeon Phi™ processor software installation, upgrade and uninstallation

This chapter describes 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 and supported operating systems

The target system must contain an Intel® Xeon Phi™ processor.

Table 1 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* Linux* 7 (1708)	3.10.0-693.5.2.el7.x86_64
Red Hat* Enterprise Linux* 7.4	3.10.0-693.2.2.el7.x86_64
SUSE* Linux* Enterprise Server 12 SP3	4.4.73-5-default
Ubuntu* 17.10	4.13.0-16-generic

Note: If your host runs Red Hat* Enterprise Linux* Server 7.4 or CentOS* Linux* 7 (1708), ensure your OS kernel was updated to version 3.10.0-693.2.1.el7.x86_64 or later which contains critical errata: https://access.redhat.com/errata/RHBA-2017:2581.

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 software 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.

3.1.1 Root access

Many of the tasks described in this document require root access privileges. Verify you have such privileges on 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* may not retain the non-root

environment of the caller. This could, for example, result in use of an unexpected *PATH* variable leading to execution of the wrong code.

When *su* is used to become root, the non-root environment is mostly retained. To retain *HOME, SHELL, USER* and *LOGNAME* use *su* with the *-m* switch. See the *su* man page for details.

3.2 Installation

3.2.1 Intel® Xeon Phi™ processor software distribution

The latest Intel® Xeon Phi™ processor software distribution can be obtained from the Intel® Premier Support website https://premiersupport-scft.intel.com/. 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 xppsm-<release>-<os>.tar
```

3.2.2 Intel® Xeon Phi™ processor software installation

Change to the directory containing the extracted packages:

```
[host]$ cd xppsm-<release>/<os-version>/packages/x86_64/core
```

3.2.2.1 RHEL*:

Installing *memkind*:

```
[host]# yum install \
memkind-<version>-<release>.x86_64.rpm \
memkind-devel-<version>-<release>.x86_64.rpm
```

Installing *sysdiag*:

```
[host]# yum install \
sysdiag-<version>-<release>.x86_64.rpm
```

Installing *micperf*:

```
[host]# yum install \
micperf-<version>-<release>.x86_64.rpm
```

Installing *hwloc*:

Note: Enable *rhel-7-server-optional-rpms* repository to install the *hwloc-devel* package. For more information visit https://access.redhat.com/documentation/en-us/red_hat_subscription_management/1/html/rhsm/supplementary-repos.

```
[host]# yum install \
hwloc-<version>-<release>.x86_64.rpm \
hwloc-devel-<version>-<release>.x86_64.rpm \
hwloc-gui-<version>-<release>.x86_64.rpm \
hwloc-libs-<version>-<release>.x86_64.rpm \
hwloc-sbin-<version>-<release>.x86_64.rpm
```

Installing *zonesort*:

```
[host]# yum install \
kmod-zonesort-<version>-<release>.x86_64.rpm
```

3.2.2.2 CentOS*:

Installing *memkind*:

```
[host]# yum install \
memkind-<version>-<release>.x86_64.rpm \
memkind-devel-<version>-<release>.x86_64.rpm
```

Installing *sysdiag*:

```
[host]# yum install \
sysdiag-<version>-<release>.x86_64.rpm
```

Installing *micperf*:

```
[host]# yum install \
micperf-<version>-<release>.x86_64.rpm
```

Installing *hwloc*:

```
[host]# yum install \
hwloc-<version>-<release>.x86_64.rpm \
hwloc-devel-<version>-<release>.x86_64.rpm \
hwloc-gui-<version>-<release>.x86_64.rpm \
hwloc-libs-<version>-<release>.x86_64.rpm \
hwloc-sbin-<version>-<release>.x86_64.rpm
```

Installing *zonesort*:

```
[host]# yum install \
kmod-zonesort-<version>-<release>.x86_64.rpm
```

3.2.2.3 SLES*:

Installing *memkind*:

```
[host]# zypper install \
memkind-<version>-<release>.x86_64.rpm \
memkind-devel-<version>-<release>.x86_64.rpm
```

Installing *sysdiag*:

```
[host]# zypper install \
sysdiag-<version>-<release>.x86_64.rpm
```

Installing *micperf*:

```
[host]# zypper install \
micperf-<version>-<release>.x86_64.rpm
```

Installing *cpuid*:

```
[host]# zypper install \
cpuid-<version>-<release>.x86_64.rpm
```

Installing *zonesort*:

```
[host]# zypper install \
zonesort-kmp-default-<version><kernel_version>-
<release>.x86_64.rpm
```

3.2.2.4 Ubuntu*:

Installing *memkind*:

```
[host]# apt install memkind_<version>-<release>_amd64.deb
```

Installing *sysdiag*:

```
[host]# apt install sysdiag_<version>-<release>_amd64.deb
```

Installing *micperf*:

```
[host]# apt install micperf_<version>-<release>_amd64.deb
```

Installing *zonesort*:

```
[host]# apt install \
zonesort_<version>-<release>_amd64.deb
```

3.2.3 Intel® Xeon Phi™ processor software upgrade

Intel® Xeon Phi™ processor software supports automated updates. Use *yum, zypper* or *apt* to perform an update as described in Section 3.2.2.

3.2.4 Intel® Xeon Phi™ processor software uninstall

To check for a previous install of Intel® Xeon Phi™ processor software execute:

RHEL*/CentOS*/SLES*:

```
[host]$ rpm -qa | grep +xpps
```

Ubuntu*:

```
[host]$ apt list --installed | grep +xpps
```

The command above lists packages correlating to Intel® Xeon Phi™ processor software which need to be uninstalled.

RHEL*/CentOS*:

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

SLES*:

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

Ubuntu*:

```
[host]# apt remove <package-name>
```

3.3 Rebuilding Intel® Xeon Phi™ processor software packages

The Intel® Xeon Phi™ processor software source code is available in the xppsm-*<xppsm-version>/<os><os_version>/packages/source/core* directory.

Refer to your OS documentation for detailed instructions on rebuilding software from source.

§

4 Virtualization

QEMU version 2.9 or newer is required to fully support virtualization on the Intel® Xeon Phi™ processor. Older versions may work, however, they will only support up to 255 virtual CPUs.

4.1 QEMU installation remarks

Please note that some operating systems may provide old naming conventions for *QEMU* binary files in order to provide backward compatibility. Therefore, if *QEMU* needs to be used manually the file */usr/libexec/qemu-kvm* should be used instead of */usr/bin/qemu-system-x86_64*. All functionalities should remain unchanged.

Note: The convention mentioned above can be disregarded if *QEMU* is used through the *libvirt* API. If your installation of *QEMU* was upgraded to version 2.9 or later, you must restart the *libvirt* daemon with the following command:

```
[host]# systemctl restart libvirtd
```

RHEL*:

QEMU version 2.9 is provided by Red Hat* Virtualization packages. Installing *QEMU* from standard repositories may install an older version that lack pass-through support for all available virtual CPUs.

SLES*:

SLES* provides all components required for virtualization. Install *QEMU* using the command below.

```
[host]# zypper install qemu qemu-kvm libvirt
```

CentOS*:

Intel® Xeon Phi™ processor software provides *QEMU* in its release package. Extract the *xppsm-virt-<version>-<os>.tar* package and install the software.

4.2 Configuration

To fully support more than 255 virtual CPUs configure *QEMU* according to instructions below.

- The *–machine* option should be set to *q35 kernel_irqchip=split*.
- Intel® *iommu* must set with the *–device intel-iommu,intremap=on,eim=on* option.
- Intel® *iommu* driver must be present on the host OS.

4.2.1 QEMU raw command

The code snippet below presents an example *QEMU* command that creates a virtual machine with 288 virtual CPUs.

```
LC_ALL=C \
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin \
QEMU_AUDIO_DRV=none \
/usr/libexec/qemu-kvm \
-name guest=rh74_288vcpus,debug-threads=on \
-machine q35,accel=kvm,usb=off,dump-guest-core=off \
-cpu host -m 4096 \
-realtime mlock=off \
-smp 288,sockets=1,cores=72,threads=4 \
-drive \
file=/tmp/rh74_288vcpus/rhel74.qcow2,format=qcow2,if=none,id=driv
e-sata0-0-0 \
-device intel-iommu,intremap=on,eim=on \
-machine kernel_irqchip=split \
-msg timestamp=on \
-device \
ide-hd,bus=ide.0,drive=drive-sata0-0-0,id=sata0-0-0,bootindex=1 \
-nographic
```

4.2.2 Libvirt xml configuration file

Review the *example_vm_288vcpus.xml* file included in the Intel® Xeon Phi™ processor software release package.

4.2.3 Adding an Intel® OPA card to a virtual machine

To add a PCI card to a VM, you can either use the *hostdev* option to the virtual machine configuration xml file or add the *–device* parameter to *QEMU* command line. In both cases the device's parameters must correspond to the host's physical PCI bus. Alternatively, you can add a device using the *virt-manager* GUI application.

Adding a device in the *xml* file:

```
<hostdev mode='subsystem' type='pci' managed='yes'>
    <source>
        <address domain='0x0000' bus='0x01' slot='0x00'/>
    </source>
</hostdev>
```

Adding a device in the *QEMU* command:

```
-device \
pcie-root-port,port=0xa,chassis=3,id=pci.3,bus=pcie.0,addr=0x1.0x2 \
-device vfio-pci,host=01:00.0,id=hostdev0,bus=pci.2,addr=0x0
```

Note: Using a PCI device in parallel by the host OS and a virtual machine is not supported.

§

A Known issues

A.1 General issues

- The *hwloc* module requires the *hwloc-dump-hwdata* files to be present in */var/run/hwloc*. On SLES* 12 SP3 the *hwloc-dump-hwdata* application has be to run manually. On other systems Intel® Xeon Phi™ processor software provides a service which runs *hwloc-dump-hwdata* during system boot.

A.2 Performance issue in cache memory mode

PROBLEM:

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

SOLUTION:

Use the *zonesort* page sorting module provided in the Intel® Xeon Phi™ processor software.

INSTALLATION:

Follow installation instructions in Section 3.2.2.

USAGE:

The *zonesort* module sorts kernel free memory page lists to minimize cache misses when those pages are acquired by user processes. Since the module operates on free pages, enabling sorting before running user applications is recommended.

Due to high memory fragmentation, sorting pages alone may not be sufficient to restore initial performance. To decrease fragmentation and increase the amount of physically-contiguous pages, use memory compaction before sorting (see the example below).

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

1. Load the module:

```
[host]# modprobe zonesort
```

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
```

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

1. Load the module:

```
[host]# modprobe zonesort
```

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. Actively running sorts will finish regardless.
- Memory compaction must be handled by the system administrator. The *zonesort* module does not call it.
- 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 *zonesort* module can be written to only by root. To modify permissions we recommend using the *udev* manager, as presented in the example below.

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

```
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 sets the interfaces' access permissions each time the module is loaded.

DEBUGGING:

The *zonesort* module exposes additional interfaces, which may be useful for identifying the system's state:

- *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
```

- *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 can only be loaded if MCDRAM is configured as cache.

§

B References

1. Burger, T. W. (2011, March 5). *Intel® Virtualization Technology for Directed I/O.* Retrieved from Intel® Developer Zone: https://software.intel.com/en-us/articles/intel-virtualization-technology-for-directed-io-vt-d-enhancing-intel-platforms-for-efficient-virtualization-of-io-devices

2. Craig, L., & Bas, V. (2009). *Scaling Lean & Agile Development.* Boston: Addison-Wesley.

3. Danalis, A., Marin, G., McCurdy, C., Meredith, J. S., Roth, P. C., Spafford, K., . . . Vetter, J. S. (2010). The scalable heterogeneous computing (shoc) benchmark suite. In Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units. *GPGPU*, 63-74.

4. *https://github.com/axboe/fio*. (2017).

5. *https://github.com/baidu-research/DeepBench*. (2017).

6. Intel Corporation. (2017). *Intel® Math Kernel Library (Intel® MKL).* Retrieved from http://software.intel.com/en-us/intel-mkl

7. Jack, D., Luszczek, P., & Petitet, A. (2003). *The linpack benchmark: past, present and future. Concurrency and Computation: Practice and Experience.*

8. McCalpin, J. D. (1991-2007). *Stream: Sustainable memory bandwidth in high performance computers. A continually updated technical report.* Charlottesville: University of Virginia.

9. McCalpin, J. D. (1995). Memory bandwidth and machine balance in current high performance computers. *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, 19-25.

§