Unbreakable Enterprise Kernel

Release Notes for Unbreakable Enterprise Kernel Release 3

E48380-10
June 2020

Abstract

This document contains information on the Unbreakable Enterprise Kernel Release 3. This document may be updated after it is released. To check for updates to this document, and to view other related Oracle documentation, refer to:

Unbreakable Enterprise Kernel Documentation

This document is intended for users and administrators of Oracle Linux. It describes potential issues and the corresponding workarounds you may encounter while using the Unbreakable Enterprise Kernel Release 3 with Oracle Linux. Oracle recommends that you read this document before installing or upgrading Unbreakable Enterprise Kernel Release 3.

Document generated on: 2020-06-04 (revision: 442)

Table of Contents

Preface

Unbreakable Enterprise Kernel: Release Notes for Unbreakable Enterprise Kernel Release 3 provides a summary of the new features, changes, and known issues in the Unbreakable Enterprise Kernel Release 3.

Audience

This document is written for system administrators who want to use the Unbreakable Enterprise Kernel with Oracle Linux. It is assumed that readers have a general understanding of the Linux operating system.

Related Documents

The documentation for this product is available at:

Unbreakable Enterprise Kernel Documentation

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1 New Features and Changes

Table of Contents

The Unbreakable Enterprise Kernel Release 3 (UEK R3) is Oracle's third major release of its heavily tested and optimized operating system kernel for Oracle Linux 6 on the x86-64 architecture. It is based on the mainline Linux kernel version 3.8.13.

The 3.8.13-16 release also updates drivers and includes bug and security fixes.

Oracle actively monitors upstream checkins and applies critical bug and security fixes to UEK3.

UEK R3 uses the same versioning model as the mainline Linux kernel version. It is possible that some applications might not understand the 3.x versioning scheme. If an application does require a 2.6 context, you can use the `uname26` wrapper command to start it. However, regular Linux applications are usually neither aware of nor affected by Linux kernel version numbers.

1.1 Notable Changes

The following sections describe the major new features of Unbreakable Enterprise Kernel Release 3 (UEK R3) relative to UEK R2. If applicable, the mainline version in which a feature was introduced is noted in parentheses.

For brief summaries of other changes, see Appendix A, *Other Changes*.

1.1.1 Architecture

- Support for the Intel IVB processor family has been added.

- The `efivars` module provides an area of firmware-managed, nonvolatile storage, which can be used as a persistent storage backend to maintain copies of kernel oopses and aid the diagnosis of problems. (3.1)

1.1.2 Control Groups and Linux Containers

Control groups (*cgroups*) and Linux Containers (LXC) are now supported features. LXC is supported for 64-bit hosts, but not 32-bit hosts (in any case, UEK R3 is not available for the 32-bit x86 architecture). Both 32-bit and 64-bit guest containers can be configured. However, some applications might not be supported for use with these features.

- The cgroups feature allows you to manage access to system resources by processes. For more information, see Control Groups in *Oracle® Linux 6: Administrator's Solutions Guide*.
- LXC is based on the cgroups and namespaces functionality. Containers allow you to safely and securely run multiple applications or instances of an operating system on a single host without risking them interfering with each other. Containers are lightweight and resource-friendly, which saves both rack space and power. For more information, see *Oracle® Linux 6: Administrator's Solutions Guide*.

  The `lxc-attach` command is supported by UEK R3 with the `lxc-0.9.0-2.0.4` package. `lxc-attach` allows you to execute an arbitrary command inside a running container from outside the container. For more information, see the `lxc-attach(1)` manual page.

Note

To access this feature, use `yum update` to install the `lxc-0.9.0-2.0.4` package (or later version of this package).

1.1.3 Core Kernel Functionality

- To avoid binary incompatibility in applications that do not understand the 3.x versioning scheme, the `UNAME26` personality patch can be used to report the kernel version as 2.6.*x* where *x* is derived from the real kernel version. The `uname26` program is provided to activate the `UNAME26` personality patch for 3.x kernels. `uname26` does not replace the `uname` command. Instead, it acts as a wrapper that modifies the return value of the `uname()` system call to return a 2.6.*x* version number. If an application fails due to the 3.8.x version number, you can use the following command to start it in a 2.6 context:

```
# uname26 application
```

  The following example demonstrates the effect of using `uname26` as a wrapper program:

```
# uname -r
3.8.13-16.el6uek.x86_64
# uname26 uname -r
2.6.48-16.el6uek.x86_64
```

  The `uname26` program is available in the `uname26` package. (3.1)

- Structured logging in `/dev/kmsg` uses `printk()` to attach arbitrary key/value pairs to logged messages, which carry machine-readable data that describes the context of the message when it was created. The key/value pairs allow you to reliably identify messages according to device, driver, subsystem, class, and type. The addition of a facility number to the `syslog` prefix allows continuation records to be merged. (3.5)
- PCI Express runtime D3cold power state is supported. This deepest power saving state for PCIe devices removes all main power. (3.6)

- Virtual Function I/O (VFIO) allows safe, non-privileged access to bare-metal devices from user-space drivers by virtual machines that use direct device access (*device assignment*) to obtain high I/O performance. From perspective of the device and the host, the VM appears as a user-space driver, which provides the benefits of reduced latency, higher bandwidth, and the direct use of bare-metal device drivers. This feature could potentially be used by high-performance computing and similar applications. (3.6)
- Huge pages support a zero page as a performance optimization. This feature was previously available only for normal sized pages (4 KB). When a process references a new memory page, the kernel assigns a pointer to the zero page rather than allocating a real page of memory and filling this with zeroes. When the process does attempt to write to the zero page, a write-protection fault is generated and the kernel allocates a real page of memory to the process's address space. (3.8)
- A new foundation for the NUMA implementation will be used as the basis for future enhancements. (3.8)
- The `memory` control group now supports both stack and slab kernel usage parameters with the following additional memory usage parameters (specified relative to `memory.kmem`):

`failcnt`	Kernel memory usage hits (display only).
`limit_in_bytes`	Kernel memory hard limit (set or display).
`max_usage_in_bytes`	Maximum recorded kernel memory usage (display only).
`usage_in_bytes`	Current kernel memory allocation (display only).

  `memory.kmem.limit_in_bytes` is intended to help limit the effect of fork bombs. (3.8)
- Automatic balancing of memory allocation for NUMA nodes. (3.8)
- The value of the SCSI error-handling timeout is now tunable. If a SCSI device times out while processing file system I/O, the kernel attempts to bring the device back online by resetting the device, followed by resetting the bus, and finally by resetting the controller. The error-handling timeout defines how many seconds the kernel should wait for a response after each recovery attempt before performing the next step in the process. For some fast-fail scenarios, it is useful to be able to adjust this value as the kernel might need additional time to try several combinations of bus device, target, bus, and controller. You can read and set the timeout via `/sys/class/scsi_device/*/device/eh_timeout`. The default timeout value is 10 seconds. (3.8)
- Variable-sized huge pages via the `flags` argument to `mmap()` or the `shmflg` argument to `shmget()`. Bits 26-31 of these arguments specify the base-2 logarithm of the page size. For example, values of `21 << 26` and `30 << 26` represent page sizes of 2 MB (2^21) and 1 GB (2^30) respectively. A value of zero selects the default huge page size. (3.8)
- The watchdog timer device (displayed in `/proc/devices`) provides a framework for all watchdog timer drivers, `/dev/watchdog`, and the `sysfs` interface for hardware-specific watchdog code. (3.8)
- The Precision Time Protocol (PTP), defined in IEEE 1588, is enabled. PTP can be used to achieve synchronization of systems to within a few tens of microseconds. If hardware time-stamping units are used, synchronization to within a few hundred nanoseconds can be achieved. (3.8)

1.1.4 Cryptography

- An Extended Verification Module (EVM) includes a digital signature that allows file metadata to be protected by using digital signatures instead of Hashed Message Authentication Control (HMAC). (3.3)
- Kernel modules can now be signed using X.509 certificates. (3.7)

1.1.5 Device Mapper

The device mapper supports an external, read-only device as the origin for a thinly-provisioned volume. Any reads to the unprovisioned area of the thin device are passed through to this device. For example, a host could run its guest VMs on thinly provisioned volumes where the base image for all of the VMs resides on a single device. (3.4)

1.1.6 Diagnostics

- The `cpupowerutils` feature extends the capabilities of `cpufrequtils`, and provides statistics for CPU idle and turbo/boost modes. On AMD systems, it also displays information about boost states and their frequencies. For more information, see http://lwn.net/Articles/433002/. (3.1)
- `zcache` version 3 supports multiple clients and in-kernel transcendent memory (`tmem`) code, and adds `tmem` callbacks to support RAMster and corresponding no-op stubs in the `zcache` driver. New `sysfs` parameters provide additional information and allow policy control. (3.1)

1.1.7 DTrace

DTrace is a comprehensive dynamic tracing framework that was initially developed for the Oracle Solaris operating system. DTrace provides a powerful infrastructure to permit administrators, developers, and service personnel to concisely answer arbitrary questions about the behavior of the operating system and user programs in real time.

Note

The DTrace utility packages (`dtrace-utils*`) are available only on the Unbreakable Linux Network (ULN).

DTrace 0.4 in UEK R3 has the following additional features compared with DTrace 0.3.2 in UEK R2:

- In UEK R2, you had to install separately available packages that contained a DTrace-enabled version of the kernel, and you had to boot the system with this kernel to be able to use DTrace. In UEK R3, DTrace support is integrated with the kernel. To use DTrace, you still need to install the `dtrace-utils` and `dtrace-modules` packages, which are available on the `ol6_x86_64_UEKR3_latest` and `ol6_x86_64_Dtrace_userspace_latest` channels. If you use `yum` to install the `dtrace-utils` package, it automatically pulls in the other packages, such as `dtrace-modules`, that are required.
- The `libdtrace` headers, which required for implementing a `libdtrace` consumer, are now located in the separate `dtrace-utils-devel` package. The headers for provider development are located in the `dtrace-modules-provider-headers` package. If you require these packages, you must install them separately from the `dtrace-modules` or `dtrace-utils` packages.
- Meta-provider support has been implemented, which allows DTrace to instantiate providers dynamically on demand. An example of a meta-provider is the `fasttrap` provider that is used for user-space tracing.
- User-space statically defined tracing (USDT) supports SDT-like probes in user-space executable and libraries. To ensure that your program computes the arguments to a DTrace probe only when required, you can use an *is-enabled probe* test to verify whether the probe is currently enabled.
- USDT requires programs to be modified to include embedded static probe points. The `sys/sdt.h` header file is provided to support USDT, but you can also use the `-h` option to `dtrace` to generate a suitable header file from a provider description file.

  The `-G` option to the `dtrace` command processes the provider description file and the compiled object files for the code that contains the probe points to generate a DOF ELF object file (which is a Extensible

Linking Format (ELF) object file with a DTrace Object Format (DOF) section). You can then create a DTrace-enabled executable or shared library by linking this DOF ELF object file with the object files.

For more information, refer to *Oracle® Linux: DTrace Guide*.

- To enable the use of USDT probes in DTrace-enabled programs, you must load the new `fasttrap` module:

```
# modprobe fasttrap
```

  Currently, the `fasttrap` provider supports the use of USDT probes. It is not used to implement the `pid` provider.

- DTrace-enabled versions of user-space applications are planned to be made available via the playground repository of Oracle Yum Server (https://yum.oracle.com/repo/OracleLinux/OL6/playground/latest/x86_64/). The packages that are provided in the playground repository are intended for experimentation only and you should not use them with production systems. Oracle does not offer support for these packages and does not accept any liability for their use.

  PHP 5.4.20, PHP 5.5.4, and later versions can be built with DTrace support on Oracle Linux. See https://blogs.oracle.com/opal/entry/using_php_dtrace_on_oracle.

  PostgreSQL 9.2.4 includes support for DTrace as described in http://www.postgresql.org/docs/9.2/static/dynamic-trace.html. You can build a DTrace-enabled version of `pgsql` by specifying the `--enable-dtrace` option to `configure` as described in http://www.postgresql.org/docs/9.2/static/install-procedure.html. For information about obtaining the PostgreSQL packages, see http://www.postgresql.org/download/linux/redhat/.

- The DTrace header files in the kernel, kernel modules, and DTrace user-space utility have been restructured to provide better support for custom consumers and DTrace-related utilities.

- The `systrace` provider has been updated to account for changes in the 3.8.13 kernel.

- Symbol lookup can now be performed by the `&` operator. `ustack()` output contains symbolic names instead of addresses provided that the symbols are present in the `DT_NEEDED` section of the ELF objects or in libraries that have been loaded with `dlopen()` or `dlmopen()`. Symbol lookup of global symbols in user-space processes respects symbol interposition and similar methods of symbol-ordering. Symbol lookup works correctly with programs that you compiled against the version of the GNU C Library (`glibc`) that ships with Oracle Linux 6.4 or later. With other versions of `glibc`, symbol lookup might fall back to using a simpler approach that does not support symbol interposition or `dlmopen()`. As symbol lookup depends on new machinery in the kernel that uses `waitfd()` and `PTRACE_GETMAPFD`, it does not work with earlier DTrace kernels.

- The `-x evaltime=`{`exec` | `main` | `preinit` | `postinit`} option to `dtrace` is now available with the following limitations:

  - `postinit` (the default behavior) is equivalent to `main`.

  - For statically linked binaries, `preinit` is equivalent to `exec`, and it might not skip `ld.so` initialization, which can happen after `main()`.

  - For stripped, statically linked binaries, both `postinit` and `main` are equivalent to `preinit`, because the `main` symbol cannot be looked up if there is no symbol table.

  In previous versions of DTrace, the default behavior was equivalent to `evaltime=exec` being set.

- You can now set DTrace options by using environment variables named `DTRACE_OPT_NAME`, where *`NAME`* is the name of the option in upper case. For example, the variable name corresponding to `incdir`, which adds a `#include` directory to the preprocessor search path, is `DTRACE_OPT_INCDIR`:

```
# export DTRACE_OPT_INCDIR=/usr/lib64/dtrace:/usr/include/sys
```

- The following changes have been made to user-visible internals:
  - The name of the ELF section in which CTF data is stored has been changed from `.dtrace_ctf` to `.ctf`.
  - The storage representation of internal kernel symbols has been improved, which reduces DTrace memory usage at start up by approximately one megabyte.
  - The `libdtrace` public API header now names its arguments.
  - The prototypes for several `libdtrace` functions have changed.
  - Two undocumented `libproc` environment variables (`_LIBPROC_INCORE_ELF` and `_LIBPROC_NO_QSORT`) from Oracle Solaris have been removed because the code, whose behaviour they adjusted, no longer exists.
  - New low-overhead debugging machinery has been implemented. If you export the `DTRACE_DEBUG=signal` environment variable, DTrace will emit debugging output only when it receives a `SIGUSR1`, avoiding the overhead due to `printf()` locking affecting any timings. The mechanism uses a ring buffer with a default size of 100 (in units of megabytes), which you can adjust by setting the value of the `DTRACE_DEBUG_BUF_SIZE` variable.
- Negative values specified to `dtrace` options that take only positive integers are now correctly diagnosed as errors.
- It is now possible to obtain correct value for the `ERR` registers.
- For more information about DTrace, refer to *Oracle® Linux: DTrace Guide*.

1.1.8 File Systems

btrfs

In UEK R3, btrfs is based on version 3.8, whereas btrfs in the latest update to UEK R2 is based on version 3.0 with some additional backported features, such as support for large metadata blocks and device statistics.

The following notable features are implemented for the btrfs file system in UEK R3 in addition to those features that are already provided in UEK R2:

- Support for changing the RAID profile without unmounting the file system. (3.3)
- The `btrfs-restore` data recovery tool attempts to extract files from a damaged file system and copy them to a safe location. (3.4)
- `fsck` in btrfs can now repair extent-allocation trees. (3.4)
- Support in `mkfs` for metadata blocks of up to 64 KB (either 16 or 32 KB is recommended). (3.4)
- Performance improvements to page cache and CPU usage, and the copy-on-write mechanisms. (3.4)

- Improved auditing to handle unexpected conditions more effectively. When unexpected errors occur, current transactions abort, errors are returned to user-space callers, and the file system enters read-only mode. (3.4)
- The `btrfs device stats` command reports I/O failure statistics, including I/O errors, CRC errors, and generation checks of metadata blocks for each drive. (3.5)
- Performance improvements to memory reclamation and synchronous I/O latency. (3.5)
- Subvolume-aware quota groups (*qgroups*) allow you to set different size limits for a volume and its subvolumes. For more information, see https://btrfs.wiki.kernel.org/index.php/UseCases. (3.6)
- The `send` and `receive` subcommands of `btrfs` allow you to record the differences between two subvolumes, which can either be snapshots of the same subvolume or parent and child subvolumes. For an example of using the send/receive feature to implement an efficient incremental backup mechanism, see https://btrfs.wiki.kernel.org/index.php/Incremental_Backup. (3.6)
- Cross-subvolume reflinks allow you to clone files across different subvolumes within a single mounted btrfs file system. However, you cannot clone files between subvolumes that are mounted separately. (3.6)
- The copy-on-write mechanism can be disabled for an empty file by using the `chattr +C` command to add the NOCOW file attribute to the file, or by creating the file in a directory on which you have set NOCOW. For some applications this feature can reduce fragmentation and improve performance. (3.7)
- File hole punching, which allows you to mark a portion of a file as unused, so freeing up the associated storage. The `FALLOC_FL_PUNCH_HOLE` flag to the `fallocate()` system call removes the specified data range from a file. The call does not change the size of the file even if you remove blocks from the end of the file. A typical use case for hole punching is to deallocate unused storage previously allocated to virtual machine images. (3.7)
- The `fsync()` system call writes the modified data of a file to the hard disk. (3.7)
- Replacing devices without unmounting or otherwise disrupting access to the file system by using the `replace` subcommand to `btrfs`, for example:

  ```
  # btrfs replace failed_device replacement_device mountpoint
  ```

  You do not need to unmount the file system or to stop active tasks. If the power fails during replacment, the process resumes when the file system is next mounted. (3.8)

For more information, see https://btrfs.wiki.kernel.org/index.php/Changelog.

cifs

The Common Internet File System (CIFS) now provides experimental support for SMB v2, which is the successor to the CIFS and SMB network file sharing protocols. (3.7)

ext3 and ext4

File system barriers are now enabled by default. If you experience a performance regression, you can disable the feature by specifying the `barrier=0` option to `mount`. (3.1)

ext4

- Store checksums of various metadata fields. Each time that a metadata field is read, the checksum of the read data is compared with the stored checksum to detect metadata corruption. (3.5)

- Quota files are now stored in hidden inodes as file system metadata instead of as separate files in the file system director hierarchy. Quotas are enabled as soon as the file system is mounted. (3.6)

f2fs

f2fs is an experimental file system that is optimized for flash memory storage devices and solid state drives (SSDs). (3.8)

FUSE

The `numa` mount option has been added to select code paths that improve performance on NUMA systems.

NFS

The NFS version 4.1 client supports Sessions, Directory Delegations, and parallel NFS (pNFS) as defined in RFC 5661. pNFS can take advantage of cluster systems by providing scalable parallel access, either to a file system or to individual files that are distributed on multiple servers. (3.7)

XFS

Journals now implement checksums for verifying log integrity. (3.8)

1.1.9 Memory Management

- The *frontswap* feature can store swap data is stored in transcendent memory, which is neither directly accessible to nor addressable by the kernel. Using transcendent memory in this way can significantly reduce swap I/O. Frontswap is so named because it can be thought of as being the opposite of a backing store for a swap device. A suitable storage medium is a synchronous, concurrency-safe, page-oriented, pseudo-RAM device such as Xen Transcendent Memory (*tmem*) or in-kernel compressed memory (*zmem*). (3.5)

- Safe swapping is supported using network block devices (NBDs) or NFS. (3.6)

1.1.10 Networking

- TCP controlled delay management (*CoDel*) is a new active queue management algorithm that is designed to handle excessive buffering across a network connection (*bufferbloat*). The algorithm is based on for how long packets are buffered in the queue rather than the size of the queue. If the minimum queuing time rises above a threshold value, the algorithm discards packets and reduces the transmission rate of TCP. (3.5)

- TCP connection repair implements process checkpointing and restart, which allows a TCP connection to be stopped on one host and restarted on another host. Container virtualization can use this feature to move a network connection between hosts. (3.5)

- TCP and STCP early retransmit allows fast retransmission (under certain conditions) to reduce the number of duplicate acknowledgements. (3.5)

- TCP fast open (TFO) can speed up the opening of successive TCP connections between two endpoints by eliminating one round time trip (RTT) from some TCP transactions. A performance improvement of between 4 and 41% has been measured for web page loading.

  TFO is not enabled by default. To enable it, use the following command:

```
# sysctl -w net.ipv4.tcp_fastopen=1
```

To make the change persist across system reboots, add the following entry to `/etc/sysctl.conf`:

```
net.ipv4.tcp_fastopen = 1
```

Applications that want to use TFO must notify the system using appropriate API calls, such as the `TCP_FASTOPEN` option to `setsockopt()` on the server side or the `MSG_FASTOPEN` flag with `sendto()` on the client side. (client side 3.6, server side 3.7)

- The TCP small queue algorithm is another mechanism intended to help deal with bufferbloat. The algorithm limits the amount of data that can be queued for transmission by a socket. The limit is set by `/proc/sys/net/ipv4/tcp_limit_output_bytes`, where the default value is 128 KB. To reduce network latency, specify a lower value for this limit. (3.6)

1.1.11 Performance

- The *slub* slab allocator now implements wider lockless operations for most paths on CPU architectures that support `CMPXCHG` (compare and exchange) instructions. This change can improve the performance of slab intensive workloads. (3.1)
- The `perf report --gtk` command launches a simple GTK2-based performance report browser. (3.4)
- The `perf annotate` command now allows you to use the `Enter` key to trace recursively through function calls in the TUI interface. (3.4)
- The `perf record -b` command supports a new hardware-based, branch-profiling feature on some CPUs that allows you to examine branch execution. (3.4)
- *Uprobes* allow you to place a performance probe at any memory address in a user application so that you can collect debugging and performance information non-disruptively. (3.5)
- The `perf trace` command can be used to record a workload according to a specified script, and to display a detailed trace of a workload that was previously recorded. This command provides an alternative interface to `strace`. (3.7)

1.1.12 Security

- The secure computing mode feature (*seccomp*) is a simple sandbox mechanism that, in strict mode, allows a thread to transition to a state where it cannot make any system calls except from a very restricted set (`_exit()`, `read()`, `sigreturn()`, and `write()`) and it can only use file descriptors that were already open. In filter mode, a thread can specify an arbitrary filter of permitted systems calls that would be forbidden in strict mode. Access to this feature is by using the `prctl()` system call. For more information, see the `prctl(2)` manual page. (3.5)
- Supervisor mode access prevention (SMAP) is a new security feature that will be supported by future Intel processors. SMAP forbids kernel access to user-space memory pages, which should help eliminate some forms of exploit. If the SMAP bit has been set in CR4, an attempt is made to access user-space memory from privileged mode causes a page-fault exception. For more information, refer to the *Intel® Architecture Instruction Set Extensions Programming Reference*. (3.7)

1.1.13 Storage

- The LSI MPT3SAS driver has been added to support LSI MPT Fusion based SAS3 (SAS 12.0 Gb/s) controllers.

- The OpenFabrics Enterprise Distribution (OFED) 2.0 stack has been integrated, which supports the following InfiniBand (IB) hardware on systems with an x86-64 architecture:
  - Mellanox ConnectX-2 InfiniBand Host Channel Adapters
  - Mellanox ConnectX-3 InfiniBand Host Channel Adapters are supported for Oracle X4-2, X4-2L, and Netra X3-2 servers
  - Sun InfiniBand QDR Host Channel Adapter PCIe #375-3696

  OFED 2.0 supports the following protocols:
  - SCSI RDMA Protocol (SRP) enables access to remote SCSI devices via remote direct memory access (RDMA)
  - iSCSI Extensions for remote direct memory access (iSER) provide access to iSCSI storage devices
  - Reliable Datagram Sockets (RDS) is a high-performance, low-latency, reliable connectionless protocol for datagram delivery
  - Sockets Direct Protocol (SDP) supports stream sockets for RDMA network fabrics
  - Ethernet over InfiniBand (EoIB)
  - Internet Protocol over InfiniBand (IPoIB)
  - Ethernet tunneling over IPoIB (eIPoIB)

  and the following RDS features:
  - Async Send (AS)
  - Quality of Service (QoS)
  - Automatic Path Migration (APM)
  - Active Bonding (AB)
  - Shared Request Queue (SRQ)
  - Netfilter (NF)
- Support for IB, OFED, and RDS is integrated into the kernel. The OFED user-space RPMs continue to be provided, but the `kernel-ib` and `ofa-kernel` RPMs are not required.
- A new iSCSI implementation raises the supported iSCSI target framework to LIO version 4.1. (3.1)

1.1.14 Virtualization

- Paravirtualization support has been enabled for Oracle Linux guests on Windows Server 2008 Hyper-V or Windows Server 2008 R2 Hyper-V.
- VFS scalability improvements:
  - The `inode_sta.nr_unused` counter has been converted to a per-CPU counter.
  - The global LRU list of unused inodes has been converted to a per-superblock LRU list.

- The ipruce_sem semaphore has been removed because of changes to the LRU lists.
- The `i_alloc_sem` functionality has been replaced with a simplified scheme.
- The scalability of mount locks has been improved for file systems that do not have mount points.
- The use of `inode_hash_lock` is avoided for pipes and sockets.

(3.1)

- `privcmd` is a new character device driver that handles access to arbitrary hypercalls through XenFS. (3.3)
- `xenbus_backend` is a new device driver for `xenbus` used by XenFS. (3.3)
- The `xenbus` device driver adds a new character device featuring `nmap` for the pre-allocated ring and an `ioctl()` for the event channel via XenFS. (3.3)
- The Virtual Extensible LAN (VXLAN) tunneling protocol overlays a virtual network on an existing Layer 3 infrastructure to allow the transfer of Layer 2 Ethernet packets over UDP. This feature is intended for use by a virtual network infrastructure in a virtualized environment. Use cases include virtual machine migration and software-defined networking (SDN). (3.7)

1.2 Xen Improvements

Relative to Unbreakable Enterprise Kernel Release 2 Quarterly Update 4, numerous bug fixes and performance improvements have been incorporated into the Unbreakable Enterprise Kernel to support Xen usage, including:

- Fixes for EDD, x2apic, XenBus, and PVHVM vCPU hotplug issues.
- The indirect-descriptor feature, which increases throughput and reduces latency for block I/O.

1.3 Driver Updates

The Unbreakable Enterprise Kernel supports a large number of hardware and devices. In close cooperation with hardware and storage vendors, Oracle has updated several device drivers. The list given below indicates the drivers whose versions differ from the versions in mainline Linux 3.8.13.

1.3.1 Storage Adapter Drivers

Broadcom

- NetXtreme II Fibre Channel over Ethernet driver (`bnx2fc`) version 2.3.4.
- NetXtreme II iSCSI driver (`bnx2i`) version 2.7.6.1d.

Cisco

- Cisco FCoE HBA Driver (`fnic`) version 1.5.0.45.

Emulex

- Blade Engine 2 Open-iSCSI driver (`be2iscsi`) version 10.0.467.0o.
- Fibre Channel HBA driver (`lpfc`) version 0:8.3.7.26.2p.

LSI

- LSI Fusion-MPT base driver (`mptbase`) version 4.28.20.03.
- LSI Fusion-MPT `ioctl` driver (`mptctl`) version 4.28.20.03.
- LSI Fusion-MPT Fibre Channel host driver (`mptfc`) version 4.28.20.03.
- LSI Fusion-MPT IP Over Fibre Channel driver (`mptlan`) version 4.28.20.03.
- LSI Fusion-MPT SAS driver (`mptsas`) version 4.28.20.03.
- LSI Fusion-MPT SCSI host driver (`mptscsih`) version 4.28.20.03.
- LSI Fusion-MPT SPI host driver (`mptspi`) version 4.28.20.03.
- LSI Fusion-MPT SAS 2.0 driver (`mpt2sas`) version 17.00.00.00.
- LSI Fusion-MPT SAS 3.0 driver (`mpt3sas`) version 03.00.00.00.

MegaRAID

- MegaRAID SAS driver (`megaraid_sas`) version 06.600.18.00.

Mellanox

- ConnectX Ethernet driver (`mlx4_en`) version 2.1.4.

  Handles Ethernet-specific functions and plugs into the netdev mid-layer.

QLogic

- Fibre Channel HBA driver (`qla2xxx`) version 8.05.00.03.39.0-k.
- iSCSI driver (`qla4xxx`) version 5.03.00.03.06.02-uek3.

  Supports Open-iSCSI.

1.3.2 Network Adapter Drivers

Broadcom

- NetXtreme II network adapter driver (`bnx2`) version 2.2.3n.
- NetXtreme II 10Gbps network adapter driver (`bnx2x`) version 1.76.54.
- Converged Network Interface Card core driver (`cnic`) version 2.5.16g.
- Tigon3 Ethernet adapter driver (`tg3`) version 3.131d.

Emulex

- Blade Engine 2 10Gbps adapter driver (`be2net`) version 4.6.63.0u.

Intel

- Legacy (PCI and PCI-X*) Gigabit network adapter driver (`e1000`) version 7.3.21-k8-NAPI.

The `e1000` driver in UEK R3 is taken from the driver for the mainline Linux kernel. The version number for this driver appears to be lower than the Intel version (8.0.35-NAPI), but it incorporates fixes that have been made since Intel ceased supporting the driver.

- PRO/1000 PCI-Express Gigabit network adapter driver (`e1000e`) version 2.4.14-NAPI.
- Gigabit Ethernet network adapter driver (`igb`) version 4.3.0.
- Base driver for Intel Ethernet Network Connection (`igbvf`) version 2.3.2.
- 10 Gigabit PCI-Express network adapter driver (`ixgbe`) version 3.15.1.
- 10 Gigabit Server Adapter virtual function driver (`ixgbevf`) version 2.8.7.

QLogic

- 1/10 GbE Converged/Intelligent Ethernet Adapter driver (`qlcnic`) version 5.2.43.
- QLE81xx network adapter driver (`qlge`) version v1.00.00.32.

Realtek PCI Express Gigabit Ethernet controller

- Realtek PCI Express Gigabit Ethernet controller (`r8169`) version 2.3LK-NAPI.

Oracle

- Sun Blade 40/10Gigabit Ethernet network driver (`sxge`) version 0.06202013.

VMware

- VMware VMXNET3 virtual ethernet driver (`vmxnet3`) version 1.1.30.0-k.

1.3.3 Miscellaneous Drivers

InfiniBand

- iSCSI Extensions for RDMA (iSER) Protocol over InfiniBand (`ib_iser`) version 1.1.
- InfiniBand SCSI RDMA Protocol initiator (`ib_srp`) version 1.2.

Oracle

- Reliable Datagram Sockets driver (`rds`) version 4.1.

  RDS provides in-order, non-duplicated, highly-available, low-overhead, reliable delivery of datagrams between hundreds of thousands of non-connected endpoints.

1.4 New and Updated Packages

To support the newly added functionality that the Unbreakable Enterprise Kernel Release 3 provides, the following RPM packages have been added or updated from the ones included in the base distribution.

- `bfa-firmware` (Brocade Fibre Channel HBA firmware)
- `crash` (`crash`, kernel analysis utility)

`crash-devel`

- `device-mapper-multipath` (device mapper)
- `device-mapper-multipath-libs`
- `dracut` (event-driven `initramfs` infrastructure)

  `dracut-caps`

  `dracut-fips`

  `dracut-fips-aesni`

  `dracut-generic`

  `dracut-kernel`

  `dracut-network`

  `dracut-tools`
- `drbd84-utils` (HA utilities for MySQL and Oracle Linux 6)
- `dtrace-modules` (DTrace modules)

  `dtrace-modules-headers`

  `dtrace-modules-provider-headers`

  `dtrace-utils` (DTrace utilities)

  `dtrace-utils-devel`
- `e2fsprogs` (`ext*` file-system utilities)

  `e2fsprogs-devel`

  `e2fsprogs-libs`
- `fuse` (FUSE file system)

  `fuse-devel`

  `fuse-libs`
- `ib-bonding` (`ip-bond`, IPoIB bonding-interface utility)
- `ibacm` (`ib_acm` daemon for InfiniBand fabrics)

  `ibacm-devel`
- `ibutils` (OpenIB Mellanox InfiniBand diagnostic utilities)
- `infiniband-diags` (OpenFabrics Alliance InfiniBand diagnostic utilities)

  `infiniband-diags-compat`
- `iscsi-initiator-utils` (iSCSI daemon and utilities)

`iscsi-initiator-utils-devel`

- `kernel-uek` (UEK R3 kernel)

  `kernel-uek-debug`

  `kernel-uek-debug-devel`

  `kernel-uek-devel`

  `kernel-uek-doc`

  `kernel-uek-firmware`

  `kernel-uek-headers`

- `kexec-tools` (`kexec` and `kdump` user-space components)
- `kpartx` (`kpartx`, partition manager)
- `libcom_err` (common error description library)

  `libcom_err-devel`

- `libdtrace-ctf` (DTrace CTF library)

  `libdtrace-ctf-devel`

- `libibcm` (user-space InfiniBand connection manager)

  `libibcm-devel`

- `libibmad` (OpenFabrics Alliance InfiniBand management datagram library)

  `libibmad-devel`

  `libibmad-static`

- `libibumad` (OpenFabrics Alliance InfiniBand user MAD library)

  `ibibumad-devel`

  `libibumad-static`

- `libibverbs` (user-space RDMA (InfiniBand/iWARP) hardware library)

  `libibverbs-devel`

  `libibverbs-devel-static`

  `libibverbs-utils`

- `libmlx4` (Mellanox ConnectX InfiniBand HCA user-space driver)

  `libmlx4-devel`

- `librdmacm` (user-space RDMA connection manager)

  `librdmacm-devel`

`librdmacm-utils`

- `libsdp` (user-space Sockets Direct Protocol library)

  `libsdp-devel`

- `libss` (command-line interface parsing library)

  `libss-devel`

- `lxc` (Linux Containers)

  `lxc-devel`

  `lxc-libs`

- `mstflint` (Mellanox firmware-burning utility)
- `netxen-firmware` (QLogic Linux Intelligent Ethernet (3000 and 3100 Series) adapter firmware)
- `ofed-docs` (OpenFabrics Enterprise Distribution documentation)
- `ofed-scripts`
- `opensm` (OpenIB InfiniBand subnet manager and management utilities)

  `opensm-devel`

  `opensm-libs`

  `opensm-static`

- `perftest` (InfiniBand performance tests for RDMA networks)
- `ql2400-firmware` (firmware for QLogic 2400 series mass storage adapter devices)
- `ql2500-firmware` (firmware for QLogic 2500 series mass storage adapter devices)
- `qperf` (`qperf`, utility for measuring socket and RDMA performance)
- `rdma` (InfiniBand/iWARP kernel-module initialization scripts)
- `rds-tools` (RDS utilities)
- `sdpnetstat` (`sdpnetstat`, InfiniBand SDP diagnostic utility)
- `srptools` (InfiniBand SDP utilities)
- `uname26` (`uname26`, wrapper utility for the `UNAME26` personality patch)
- `xfsdump` (administrative utilities for the XFS file system)
- `xfsprogs` (XFS file-system utilities)

  `xfsprogs-devel`

  `xfsprogs-qa-devel`

For details of the channels on which these packages are available, see Chapter 3, *Installation and Availability*.

1.5 Technology Preview

The following features included in the Unbreakable Enterprise Kernel Release 3 are still under development, but are made available for testing and evaluation purposes.

- **DRBD (Distributed Replicated Block Device)**

  A shared-nothing, synchronously replicated block device (*RAID1 over network*), designed to serve as a building block for high availability (HA) clusters. It requires a cluster manager (for example, pacemaker) for automatic failover.

- **Kernel module signing facility**

  Applies cryptographic signature checking to modules on module load, checking the signature against a ring of public keys compiled into the kernel. GPG is used to do the cryptographic work and determines the format of the signature and key data.

- **Transcendent memory**

  Transcendent Memory (*tmem*) provides a new approach for improving the utilization of physical memory in a virtualized environment by claiming underutilized memory in a system and making it available where it is most needed. From the perspective of an operating system, tmem is fast pseudo-RAM of indeterminate and varying size that is useful primarily when real RAM is in short supply. To learn more about this technology and its use cases, see the Transcendent Memory project page at https://oss.oracle.com/projects/tmem/.

1.6 Compatibility

Oracle Linux maintains user-space compatibility with Red Hat Enterprise Linux, which is independent of the kernel version running underneath the operating system. Existing applications in user space will continue to run unmodified on the Unbreakable Enterprise Kernel Release 3 and no re-certifications are needed for RHEL certified applications.

To minimize impact on interoperability during releases, the Oracle Linux team works closely with third-party vendors whose hardware and software have dependencies on kernel modules. The kernel ABI for UEK R3 will remain unchanged in all subsequent updates to the initial release. In this release, there are changes to the kernel ABI relative to UEK R2 that require recompilation of third-party kernel modules on the system. Before installing UEK R3, verify its support status with your application vendor.

Chapter 2 Known Issues

This chapter describes the known issues for the Unbreakable Enterprise Kernel Release 3.

Updating Oracle Linux Fails if the kernel-uek Package Cannot Be Updated

By default, the installation of Oracle Linux includes the `dtrace-modules` package for UEK R3. This package requires a specific `kernel-uek` version. However, a `yum update` fails if it cannot update the `kernel-uek` package when the `installonly_limit` of three updates is reached as the `dtrace-modules` package does not allow the existing `kernel-uek` packages to be removed. `yum` displays an error message similar to the following example:

```
--> Finished Dependency Resolution
Error: Package: kernel-uek-debug-3.8.13-55.1.1.el6uek.x86_64 (public_ol6_UEKR3_latest)
          Requires: kernel-firmware = 3.8.13-55.1.1.el6uek
```

The workaround is to remove any existing `dtrace-modules` packages before updating the `kernel-uek` package, for example:

```
# for package in `rpm -qa | grep dtrace-modules`; do yum remove -y $package; done
```

When you have removed all `dtrace-modules` packages, you can update Oracle Linux, including the UEK R3 kernel. If you want to use DTrace with UEK R3, reinstall the `dtrace-modules` package for the current kernel:

```
# yum install dtrace-modules-`uname -r`
```

(Bug ID 21669543)

ACFS

Oracle ASM Cluster File System (ACFS) is currently not supported for use with UEK R3. (Bug ID 16318126)

ACPI

- On some systems you might see ACPI-related error messages in `dmesg` similar to the following:

  ```
  ACPI Error: [CDW1] Namespace lookup failure, AE_NOT_FOUND
  ACPI Error: Method parse/execution failed [_SB_._OSC||\||]
  ACPI Error: Field [CDW3] at 96 exceeds Buffer [NULL] size 64 (bits)]]>
  ```

  These messages, which are not fatal, are caused by bugs in the BIOS. Contact your system vendor for a BIOS update. (Bug ID 13100702)

- The following messages indicate that the BIOS does not present a suitable interface, such as `_PSS` or `_PPC`, that the `acpi-cpufreq` module requires:

  ```
  kernel: powernow-k8: this CPU is not supported anymore, using acpi-cpufreq instead.
  modprobe: FATAL: Error inserting acpi_cpufreq
  ```

  There is no known workaround for this error. (Bug ID 17034535)

ASM

Calling the `oracleasm init` script, `/etc/init.d/oracleasm`, with the parameter `scandisks` can lead to error messages about missing devices similar to the following:

```
oracleasm-read-label: Unable to open device "device": No such file or directory
```

However, the device actually exists. You can ignore this error message, which is triggered by a timing issue. Only use the `init` script to start and stop the `oracleasm` service. All other options, such as scandisks, `listdisk`, and `createdisk`, are deprecated. For these and other administrative tasks, use `/usr/sbin/oracleasm` instead. (Bug ID 13639337)

bnx2x driver

When using the `bnx2x` driver in a bridge, disable Transparent Packet Aggregation (TPA) by including the statement `options bnx2x disable_tpa=1` in `/etc/modprobe.conf`. (Bug ID 14626070)

btrfs

- If you use the `--alloc-start` option with `mkfs.btrfs` to specify an offset for the start of the file system, the size of the file system should be smaller but this is not the case. It is also possible to specify an offset that is higher than the device size. (Bug ID 16946255)
- The usage information for `mkfs.btrfs` reports `raid5` and `raid6` as possible profiles for both data and metadata. However, the kernel does not support these features and cannot mount file systems that use them. (Bug ID 16946303)
- The `btrfs filesystem balance` command does not warn that the RAID level can be changed under certain circumstances, and does not provide the choice of cancelling the operation. (Bug ID 16472824)
- Converting an existing ext2, ext3, or ext4 root file system to btrfs does not carry over the associated security contexts that are stored as part of a file's extended attributes. With SELinux enabled and set to enforcing mode, you might experience many permission denied errors after reboot, and the system might be unbootable. To avoid this problem, enforce automatic file system relabeling to run at bootup time. To trigger automatic relabeling, create an empty file named `.autorelabel` (for example, by using `touch`) in the file system's `root` directory before rebooting the system after the initial conversion. The presence of this file instruct SELinux to recreate the security attributes for all files on the file system. If you forget to do this and rebooting fails, either temporarily disable SELinux completely by adding `selinux=0` to the kernel boot parameters, or disable enforcing of the SELinux policy by adding `enforcing=0`. (Bug ID 13806043)
- Commands such as `du` can show inconsistent results for file sizes in a btrfs file system when the number of bytes that is under delayed allocation is changing. (Bug ID 13096268)
- The copy-on-write nature of btrfs means that every operation on the file system initially requires disk space. It is possible that you cannot execute any operation on a disk that has no space left; even removing a file might not be possible. The workaround is to run `sync` before retrying the operation. If this does not help, remount the file system with the `-o nodatacow` option and delete some files to free up space. See https://btrfs.wiki.kernel.org/index.php/ENOSPC.
- Btrfs has a limit of 237 or fewer hard links to a file from a single directory. The exact limit depends on the number of characters in the file name. The limit is 237 for a file with up to eight characters in its file name; the limit is lower for longer file names. Attempting to create more than this number of links results in the error `Too many links`. You can create more hard links to the same file from another directory. Although the limitation of the number of hard links in a single directory has been increased to 65535, the version of `mkfs.btrfs` that is provided in the `btrfs-progs` package does not yet support the compatibility flag for this feature. (Bug ID 16278563)
- The `-c` option to the `btrfs qgroup limit` command is redundant as the quota limit is always enforced after compression. (Bug ID 16557528)

- If you run the `btrfs quota enable` command on a non-empty file system, any existing files do not count toward space usage. Removing these files can cause usage reports to display negative numbers and the file system to be inaccessible. The workaround is to enable quotas immediately after creating the file system. If you have already written data to the file system, it is too late to enable quotas. (Bug ID 16569350)
- The `btrfs quota rescan` command is not currently implemented. The command does not perform a rescan and returns without displaying any message. (Bug ID 16569350)
- When you overwrite data in a file, starting somewhere in the middle of the file, the overwritten space is counted twice in the space usage numbers that `btrfs qgroup show` displays. (Bug ID 16609467)
- If you run `btrfsck --init-csum-tree` on a file system and then run a simple `btrfsck` on the same file system, the command displays a Backref mismatch error that was not previously present. (Bug ID 16972799)
- Btrfs tracks the devices on which you create btrfs file systems. If you subsequently reuse these devices in a file system other than btrfs, you might see error messages such as the following when performing a device scan or creating a RAID-1 file system, for example:

  ```
  ERROR: device scan failed '/dev/cciss/c0d0p1' - Invalid argument
  ```

  You can safely ignore these errors. (Bug ID 17087097)
- If you use the `-s` option to specify a sector size to `mkfs.btrfs` that is different from the page size, the created file system cannot be mounted. By default, the sector size is set to be the same as the page size. (Bug ID 17087232)

CPU microcode update failures on PVM or PVHVM guests

When running Oracle Linux 6 with UEK R3, you might see error messages in `dmesg` or `/var/log/messages` similar to this one:

```
microcode: CPU0 update to revision 0x6b failed.
```

You can ignore this warning. You do not need to upgrade the microcode for virtual CPUs as presented to the guest. (Bug ID 12576264, 13782843)

DHCP lease is not obtained at boot time

If DHCP lease negotiation takes more than 5 seconds at boot time, the following message is displayed:

```
ethX: failed. No link present. Check cable?
```

If the `ethtool ethX` command confirms that the interface is present, edit `/etc/sysconfig/network-scripts/ifcfg-ethX` and set `LINKDELAY=N`, where *N* is a value greater than 5 seconds (for example, 30 seconds). Alternatively, use NetworkManager to configure the interface. (Bug ID 16620177)

dm-nfs obsoleted

In UEK R2, the `dm-nfs` module provided the ability to create a loopback device for a mounted NFS file or file system. For example, the feature allowed you to create the shared storage for an Oracle 3 VM cluster on an NFS file system. The `dm-nfs` module provided direct I/O to the server and bypassed the `loop` driver to avoid an additional level of page caching. The `dm-nfs` module is not provided with UEK R3. The `loop` driver can now provide the same I/O functionality as `dm-nfs` by extending the AIO interface to perform direct I/O. To create the loopback device, use the `losetup` command instead of `dmsetup`.

DTrace

- Using `kill -9` to terminate `dtrace` can leave breakpoints outstanding in processes being traced, which might sooner or later kill them.
- Argument declarations for probe definitions cannot be declared with derived types such as `enum`, `struct`, or `union`.
- The following compiler warning can be ignored for probe definition arguments of type `string` (which is a D type but not a C type):

```
provider_def.h:line#: warning: parameter names (without types) in function declaration
```

ERST message

You can safely ignore the following message that might be displayed in `syslog` or `dmesg`:

```
ERST: Failed to get Error Log Address Range.
```

The message indicates that the system BIOS does not support an Error Record Serialization Table (ERST). (Bug ID 17034576)

ext4 inline data

The inline data feature that allows the data of small files to be stored inside their inodes is not yet available. The `-O inline_data` option to the `mkfs.ext4` and `tune2fs` commands is not supported. (Bug ID 17210654)

Firmware warning message

You can safely ignore the following firmware warning message that might be displayed on some Sun hardware:

```
[Firmware Warn]: GHES: Poll interval is 0 for generic hardware error source:
1, disabled.
```

(Bug ID 13696512)

Huge pages

One-gigabyte (1 GB) huge pages are not currently supported for the following configurations:

- HVM guests
- PV guests
- Oracle Database

Two-megabyte (2 MB) huge pages have been tested and work with these configurations.

(Bug ID 17299364, 17299871, 17271305)

I/O scheduler

The Unbreakable Enterprise Kernel uses the `deadline` scheduler as the default I/O scheduler. For the Red Hat Compatible Kernel, the default I/O scheduler is the `cfq` scheduler.

ioapic failure messages

You can safely ignore messages such as `ioapic: probe of 0000:00:05.4 failed with error -22`. Such messages are the result of the `ioapic` driver attempting to re-register I/O APIC PCI devices that were already registered at boot time. (Bug ID 17034993)

InfiniBand warning messages when disabling a switch port

You might see the following warning messages if you use the `ibportstate disable` command to disable a switch port:

```
ibwarn: [2696] _do_madrpc: recv failed: Connection timed out
ibwarn: [2696] mad_rpc: _do_madrpc failed; dport (Lid 38)
ibportstate: iberror: failed: smp set portinfo failed
```

You can safely ignore these warnings. (Bug ID 16248314)

IPoIB mode switching

The Internet Protocol over InfiniBand (IPoIB) driver supports the use of either connected mode or datagram mode with an interface, where datagram mode is the default mode. Changing the mode of an InfiniBand interface by echoing either `connected` or `datagram` to `/sys/class/net/ibN/mode` is not supported. It is also not possible to change the mode of an InfiniBand interface while it is enabled.

To change the IPoIB mode of an InfiniBand interface:

1. Edit the `/etc/sysconfig/network-scripts/ifcfg-ibN` configuration file, where *N* is the number of the interface:
   - To configure connected mode, specify `CONNECTED_MODE=yes` in the file.
   - To configure datagram mode, either specify `CONNECTED_MODE=no` in the file or do not specify this setting at all (datagram mode is enabled by default).

   Note

   Before saving your changes, make sure that you have not specified more than one setting for `CONNECTED_MODE` in the file.

2. To enable the specified mode on the interface, use the following commands to take down the interface and bring it back up:

   ```
   # ifdown ibN
   # ifup ibN
   ```

(Bug ID 17479833)

libfprint

The following message might appear in `dmesg` or `/var/log/messages`:

```
WARNING! power/level is deprecated; use power/control instead.
```

The USB subsystem in UEK R3 deprecates the `power/level sysfs` attribute in favor of the `power/control` attribute. The `libfprint` fingerprinting library triggers this warning via `udev` rules that try to

use the old attribute first. You can safely ignore this warning. The setting of the appropriate power level still succeeds. (Bug ID 13523418)

Large memory system fails to boot

If a large memory system fails to start, boot it using an alternate kernel to UEK R3 and disable the kdump service before booting into the UEK R3 kernel:

```
# chkconfig kdump off
```

(Bug ID 16765434)

Linux Containers (LXC)

- The correct operation of containers might require that you completely disable SELinux on the host system. For example, SELinux can interfere with container operation under the following conditions:
  - Running the `halt` or `shutdown` command from inside the container hangs the container or results in a `permission denied` error. (An alternate workaround is to use the `init 0` command from inside the container to shut it down.)
  - Setting a password inside the container results in a `permission denied` error, even when run as `root`.
  - You want to allow `ssh` logins to the container.

  To disable SELinux on the host:

  1. Edit the configuration file for SELinux, `/etc/selinux/config` and set the value of the `SELINUX` directive to `disabled`.
  2. Shut down and reboot the host system.
- The `root` user in a container can affect the configuration of the host system by setting some `/proc` entries. (Bug ID 17190287)
- Using `yum` to update packages inside the container that use `init` scripts can undo changes made by the Oracle template.
- Migrating live containers (`lxc-checkpoint`) is not yet supported.
- Oracle Database is not yet supported for use with Linux Containers. The following information is intended for those who want to experiment with such a configuration.

  The following `/proc` parameter files may only be set on the host and not for individual containers:

  - `/proc/sys/fs/aio-max-nr`
  - `/proc/sys/net/core/rmem_default`
  - `/proc/sys/net/core/rmem_max`
  - `/proc/sys/net/core/wmem_default`
  - `/proc/sys/net/core/wmem_max`
  - `/proc/sys/net/ipv4/ip_local_port_range`

Setting the parameters in the host to the Oracle recommended values sets them for all containers and allows the Oracle database to run in a container. For more information, see Configuring Kernel Parameters. (Bug ID 17217854)

NUMA warning messages on a non-NUMA system

You can safely ignore the following warning messages in `dmesg` and `/var/log messages` if you see them on a non-NUMA system:

```
kernel: NUMA: Warning: node ids are out of bound, from=-1 to=-1 distance=10
hcid[4293]: Register path:/org/bluez fallback:1
kernel: No NUMA configuration found
```

(Bug ID 13711370)

pcspkr driver error message

You can safely ignore the following error message:

```
Error: Driver 'pcspkr' is already registered, aborting...
```

The message arises from an alias conflict between `snd-pcsp` and `pcspkr`. To prevent the message from being displayed, add the following line to `/etc/modprobe.d/blacklist.conf`:

```
blacklist snd-pcsp
```

(Bug ID 10355937)

sched_yield() settings for CFS

For the Unbreakable Enterprise Kernel, `kernel.sched_compat_yield=1` is set by default. For the Red Hat Compatible Kernel, `kernel.sched_compat_yield=0` is used by default.

Soft lockup errors when booting

When upgrading or installing the UEK R3 kernel on fast hardware, usually with SAN storage attached, the kernel can fail to boot and `BUG: soft lockup` messages are displayed in the console log. The workaround is to increase the baud rate from the default value of 9600 by amending the kernel boot line in `/boot/grub/grub.conf` to include an appropriate console setting, for example:

```
console=ttyS0,115200n8
```

A value of 115200 is recommended as smaller values such as 19200 are known to be insufficient for some systems (for example, see https://docs.oracle.com/cd/E19045-01/blade.x6220/820-0048-18/sp.html#0_pgfId-1002490). If the host implements an integrated system management infrastructure, such as ILOM on Sun and Oracle systems or iLO on HP systems, configure the integrated console baud rate to match the setting for the host system. Otherwise, the integrated console is likely to display garbage characters. (Bug ID 17064059, 17252160)

Transparent Huge Pages

This release removes the Transparent Huge Pages (THP) feature. Following extensive benchmarking and testing, Oracle found that THP caused a performance degradation of between 5 and 10% for some

workloads. This performance degradation was a result of a slower memory allocator code path being used even when the applications were not using THP. When the fact that huge pages are not swappable was taken into account, the positive effect that THP should provide was outweighed by its negative effects.

After installing this UEK release, you cannot enable THP (for example, by specifying kernel boot parameters). The THP settings under `/sys/kernel/mm/transparent_hugepage` have also been removed. A future update might contain an updated THP implementation which resolves the performance issue.

Note

This change does not affect support for applications that use explicit huge pages (for example, Oracle Database).

(Bug ID 16823432)

User Namespaces

The kernel functionality (`CONFIG_USER_NS`) that allows unprivileged processes to create namespaces for users inside which they have root privileges is not currently implemented because of a clash with the implementation of XFS. This functionality is primarily intended for use with Linux Containers. As a result, the `lxc-checkconfig` command displays `User namespace: missing`. (Bug ID 16656850)

Virtualization

- When booting UEK R3 as a PVHVM guest, you can safely ignore the following kernel message:

```
register_vcpu_info failed:
          err=-38
```

  (Bug ID 13713774)

- Under Oracle VM Server 3.1.1, migrating a PVHVM guest that is running the UEK R3 kernel causes a disparity between the date and time as displayed by `date` and `hwclock`. The workaround post migration is either to run the command `hwclock --hctosys` on the guest or to reboot the guest. (Bug ID 16861041)

- On virtualized systems that are built on Xen version 3, including all releases of Oracle VM 2 including 2.2.2 and 2.2.3, disk synchronization requests for ext3 and ext4 file systems result in journal corruption with kernel messages similar to the following being logged:

```
blkfront: barrier: empty write xvda op failed
blkfront: xvda: barrier or flush: disabled
```

  In addition, journal failures such as the following might be reported:

```
Aborting journal on device xvda1
```

  The workaround is to add the mount option `barrier=0` to all ext3 and ext4 file systems in the guest VM before upgrading to UEK R3. For example, you would change a mount entry such as:

```
UUID=4e4287b1-87dc-47a8-b69a-075c7579eaf1  /  ext3  defaults  1 1
```

  so that it reads:

```
UUID=4e4287b1-87dc-47a8-b69a-075c7579eaf1  /  ext3  defaults,barrier=0  1 1
```

  This issue does not apply to Xen 4 based systems, such as Oracle VM 3. (Bug ID 17310816)

X.509 Certificates for module verification

The system reports a message similar to the following if there is a problem loading an in-kernel X.509 module verification certificate at boot time:

```
Loading module verification certificates
X.509: Cert 0c21da3d73dcdbaffc799e3d26f3c846a3afdc43 is not yet valid
MODSIGN: Problem loading in-kernel X.509 certificate (-129)
```

This error occurs because the hardware clock lags behind the system time as shown by `hwclock`, for example:

```
# hwclock
Tue 20 Aug 2013 01:41:40 PM EDT -0.767004 seconds
```

The solution is to set the hardware clock from the system time by running the following command:

```
# hwclock --systohc
```

After correcting the hardware clock, no error should be seen at boot time, for example:

```
Loading module verification certificates
MODSIGN: Loaded cert 'Slarti: Josteldalsbreen signing key:
0c21da3d73dcdbaffc799e3d26f3c846a3afdc43'
```

(Bug ID 17346862)

Chapter 3 Installation and Availability

Table of Contents

You can install Unbreakable Enterprise Kernel Release 3 on Oracle Linux 6 Update 4 or newer, running either the Red Hat compatible kernel or a previous version of the Unbreakable Enterprise Kernel. If you are still running an older version of Oracle Linux, first update your system to the latest available update release.

The Unbreakable Enterprise Kernel Release 3 is supported on the x86-64 architecture but not on x86.

3.1 Installation Overview

If you have a subscription to Oracle Unbreakable Linux support, you can obtain the packages for Unbreakable Enterprise Kernel Release 3 by registering your system with the Unbreakable Linux Network (ULN) and subscribing it to additional channels. See Section 3.2, “Subscribing to ULN Channels”.

If your system is not registered with ULN, you can obtain most of the packages from Oracle Yum Server. See Section 3.3, “Enabling Access to Oracle Yum Server Channels”.

If you have previously installed any OFED packages on your system, and you want to replace these with the latest packages that are provided on the `ol6_x86_64_ofed_UEK` channel, you must manually remove some of the existing packages. See Section 3.4, “Upgrading OFED Packages”.

Having subscribed your system to the appropriate channels on ULN or Oracle Yum Server, upgrade your system. See Section 3.5, “Upgrading Your System”.

3.2 Subscribing to ULN Channels

The kernel image and user-space packages are available on the following ULN channels:

- `ol6_x86_64_latest` (latest user-space packages for Oracle Linux 6 other than DTrace, OFED, and DRBD packages)
- `ol6_x86_64_UEKR3_latest` (`kernel-uek*`, `dtrace-modules-*`, `libdtrace-*`, and `uname26`)
- `ol6_x86_64_Dtrace_userspace_latest` (`dtrace-utils*`)
- `ol6_x86_64_ofed_UEK` (latest OFED tools packages)
- `ol6_x86_64_mysql-ha-utils` (`drbd84-utils`)

The following procedure assumes that you have already registered your system with ULN.

To subscribe your system to a channel on ULN:

1. Log in to https://linux.oracle.com with your ULN user name and password.
2. On the Systems tab, click the link named for the system in the list of registered machines.

3. On the System Details page, click **Manage Subscriptions**.
4. On the System Summary page, select each required channel from the list of available channels and click the right arrow to move the channel to the list of subscribed channels.

   Subscribe the system to the `ol6_x86_64_latest` and `ol6_x86_64_UEKR3_latest` channels. If required, you can also add the channels for the DTrace, OFED, and DRBD packages. You do not need to subscribe the system to the `ol6_x86_64_UEK_latest` channel.
5. Click **Save Subscriptions**.

For information about using ULN, see *Oracle® Linux: Unbreakable Linux Network User's Guide for Oracle Linux 6 and Oracle Linux 7*.

3.3 Enabling Access to Oracle Yum Server Channels

At the Oracle Yum Server repository at https://yum.oracle.com/, the kernel image and user-space packages are available on the following channels:

- `ol6_latest` (latest user-space packages for Oracle Linux 6 other than the OFED tool packages)
- `ol6_UEKR3_latest` (`kernel-uek*`, `dtrace-modules-*`, `libdtrace-*`, and `uname26`)
- `ol6_ofed_UEK` (latest OFED tools packages)

Note

The DTrace utility and DRBD packages are not available on Oracle Yum Server.

To enable access to the channels on Oracle Yum Server, create entries such as the following in `/etc/yum.conf` or in a repository file in the `/etc/yum.repos.d` directory:

```
[ol6_latest]
name=Oracle Linux $releasever Latest ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL6/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol6_UEK_latest]
name=Latest Unbreakable Enterprise Kernel for Oracle Linux $releasever ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL6/UEK/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=0

[ol6_UEKR3_latest]
name=Latest Unbreakable Enterprise Kernel Release 3 for Oracle Linux $releasever ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL6/UEKR3/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=1

[ol6_playground_latest]
name=Latest mainline stable kernel for Oracle Linux 6 ($basearch) - Unsupported
baseurl=https://yum.oracle.com/repo/OracleLinux/OL6/playground/latest/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=0

[ol6_ofed_UEK]
```

```
name=OFED supporting tool packages for Unbreakable Enterprise Kernel on Oracle Linux 6 ($basearch)
baseurl=https://yum.oracle.com/repo/OracleLinux/OL6/ofed_UEK/$basearch/
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-oracle
gpgcheck=1
enabled=0
```

To enable a channel, set the value of the `enabled` parameter for the channel to 1.

To disable a channel, set the value of the `enabled` parameter for the channel to 0.

In this example, access is enabled to the `ol6_latest` and `ol6_UEKR3_latest` channels but not to the `ol6_UEK_latest`, `ol6_playground_latest` and `ol6_ofed_UEK` channels.

You can find more information about installing the software at https://yum.oracle.com/, from where you download a copy of a suitable repository file (https://yum.oracle.com/public-yum-ol6.repo).

Note

By default, the `ol6_x86_64_UEK_latest` channel is enabled and the `ol6_UEKR3_latest` channel is disabled in the `public-yum-ol6.repo` file. To be able to install the kernel packages for UEK R3, you must enable the `ol6_UEKR3_latest` channel. You can disable the `ol6_UEK_latest` channel.

3.4 Upgrading OFED Packages

If you have enabled the `ol6_ofed_UEK` channel, you must remove any existing OFED packages for the 32-bit x86 architecture before you can upgrade the remaining OFED packages on your system. You must also completely remove and reinstall the `ibutils` packages. The latest version of the `ibutils` package no longer depends on an `ibutils-libs` package as the libraries are now included in `ibutils` itself.

1. Use the following command to remove any non-upgradable packages for the x86 architecture:

```
# rpm -e infiniband-diags-1.5.12-5.el6.i686 \
libibcm-1.0.5-3.el6.i686 \
libibcm-devel-1.0.5-3.el6.i686 \
libibmad-1.3.9-1.el6.i686 \
libibmad-devel-1.3.9-1.el6.i686 \
libibumad-1.3.8-1.el6.i686 \
libibumad-devel-1.3.8-1.el6.i686 \
libibverbs-1.1.6-5.el6.i686 \
libibverbs-devel-1.1.6-5.el6.i686 \
libmlx4-1.0.4-1.el6.i686 \
librdmacm-1.0.17-0.git4b5c1aa.el6.i686 \
librdmacm-devel-1.0.17-0.git4b5c1aa.el6.i686 \
opensm-devel-3.3.15-1.el6.i686 \
opensm-libs-3.3.15-1.el6.i686 \
ibacm-devel-1.0.8-0.git7a3adb7.el6.i686
```

2. Enter the following commands to remove the existing `ibutils` and `ibutils-libs` packages and install the new `ibutils` package:

```
# rpm -e ibutils-1.5.7-7.el6.x86_64 \
ibutils-libs-1.5.7-7.el6.x86_64
# yum install ibutils
```

3.5 Upgrading Your System

After enabling access to the appropriate channels, including `ol6_UEKR3_latest`, in the Oracle Yum Server repository or `ol6_x86_64_UEKR3_latest` on ULN, run the following command to upgrade the system to UEK R3:

```
# yum update
```

If you have questions regarding configuring or using `yum` to install updates, refer to *Oracle® Linux 6: Administrator's Solutions Guide*.

The kernel's source code is available via a public git source code repository at https://oss.oracle.com/git/?p=linux-uek3-3.8.git.

Appendix A Other Changes

Table of Contents

The following sections describe other features of Unbreakable Enterprise Kernel Release 3 (UEK R3). The mainline version in which a feature was introduced is noted in parentheses.

A.1 Architecture

- `vsysscall` emulation and `vsyscall` parameter. (3.1)
- `INTEL_MID` configuration. (3.1)
- `mrst_pmu` driver for Intel Moorestown Power Management Unit. (3.1)
- Hardware memory error recovery support for ACPI, APEI, and GHES. (3.1)
- `printk()` support for recoverable error via NMI for ACPI, APEI, and GHES. (3.1)

A.2 Block Devices

- Strict CPU affinity can be enabled by setting the value of `/sys/block/blkdev/queue/rq_affinity` to 2. Performance on some systems benefits from being directed to the strict requester CPU rather than using per-socket steering. (3.1)
- CFQ I/O scheduler performance tuning adds think time check for a group, which makes bandwidth usage more efficient by not leaving queues active when there are no further requests for the group. (3.1)
- Flakey target support in the device mapper adds the `corrupt_bio_byte` parameter to simulate corruption by overwriting a byte at a specified position with a specified value while the device is down. The `drop_writes` option parameter drops writes silently while the device is down. (3.1)
- The device mapper supports MD RAID-1 personality through the `dm-raid` target. (3.1)
- The device mapper supports the ability to parse and use metadata devices with `dm-raid`. Without the metadata devices, many RAID features would be unavailable. (3.1)
- Experimental support for thin provisioning in the device mapper allows the creation of multiple thinly provisioned volumes from a storage pool and recursive snapshots to an arbitrary depth. (3.2)

- I/O-less dirty throttling and reduced file-system writeback from page reclamation greatly reduces I/O seeks and CPU contention. (3.2)
- The `cfq_target_latency` parameter under `sysfs` allows throughput and read latency to be tuned. (3.4)
- The device mapper supports adding and removing space at the end of the devices when resizing RAID-10 arrays with `near` and `offset` layouts. (3.4)
- Thin target in the device mapper supports discards. When non-discard I/O completes and the associated mappings are quiesced, any discards that were deferred (via `ds_add_work()` in `process_discard()`) are queued for processing by the worker thread. (3.4)
- Thin target in the device mapper provides user-space access to pool metadata. Two new messages can be sent to the thin pool target allowing it to take a snapshot of the metadata. This read-only snapshot can be accessed from user space concurrently with the live target. (3.5)
- Thin target in the device mapper uses dedicated slab caches (whose names are prefixed with `dm_`) rather than relying on `kmalloc` memory pools backed by generic slab caches. This allows independent accounting of memory usage and any associated memory leakage by thin provisioning. (3.5)
- RAID-5 XOR checksumming is optimized by taking advantage of the 256-bit YMM registers introduced by Advanced Vector Extensions (AVX). (3.5)
- RAID-6 includes Supplemental Streaming SIMD Extensions 3 (SSSE3) optimized recovery functions and a new algorithm for selecting the most appropriate function to use for recovery. (3.5)
- MD allows a reshape operation to be reversed by implementing a new `reshape_direction` attribute that can be set when `delta_disks` is zero, and which can take one of the values `forward` or `backwards`. (3.5)
- A RAID-10 array can be reshaped to a different `near` or `offset` layout, a different chunk size, and a different number of devices. The number of copies cannot be changed. (3.5)
- An existing partition can be resized, even if currently in use, by using the operation code `BLKPG_RESIZE_PARTITION` with the `BLKPG ioctl()`. (3.6)
- Add MD support for `RAID10` (striped mirrors) and `RAID1E` (integrated adjacent stripe mirroring). (3.6)
- Thin target in the device mapper adds `read-only` and `fail-io` modes to thin provisioning. If a transaction commit fails, a pool's metadata device transitions to `read-only` mode. If a commit fails when the device is in `read-only` mode, a transition to `fail-io` mode occurs. In `fail-io` mode, the pool and all associated thin devices report a status of `fail` if a commit fails. (3.6)
- The persistent data debug space map checker has been removed from the device mapper. The feature consumed a lot of memory and caused other issues when enabled on large pools. (3.6)
- RAID-1 in MD now prevents the merging of large requests to enhance the performance of SSD devices that function more efficiently with large request transfers. (3.6)
- Support for the `WRITE SAME` request implemented on some SCSI devices to allow a block to be efficiently replicated throughout a block range. Only a single logical block need be transferred from the host. The storage device writes the same data to all blocks specified by the request. (3.7)
- The `BLKZEROOUT ioctl()` can be used to zero out block ranges via `blkdev_issue_zerooout()`. (3.7)
- Fastmap support provides a method for attaching an unsorted block image (UBI) device in real-time. Rather than scanning the entire device, Fastmap locates a checkpoint. (3.7)

- MD adds `TRIM` discard support for linear RAID-0, RAID-1, RAID-5, and RAID-10. (3.7)
- DM adds rebuild capacity and replacement slot validation for RAID-10 arrays. (3.7)
- RAID-6 recovery is optimized by taking advantage of the 256-bit YMM registers introduced by Advanced Vector Extensions 2 (AVX2). (3.8)

A.3 Core Kernel Functionality

- Add a lock-less NULL-terminated single list. (3.1)
- Add a library function implementing a `crc8` algorithm to support the `brcm80211` driver. (3.1)
- Make the `gen_pool` memory allocator lockless. This change makes it safe to use the memory allocator in NMI handlers and other special unblockable contexts where deadlocks might occur. (3.1)
- Implement the `PTRACE_INTERRUPT`, `PTRACE_LISTEN`, `PTRACE_SEIZE`, and `TRAP_NOTIFY` `ptrace()` requests. (3.1)
- Adds `/sys/module/`*`module_name`*`/uevent` files to all module entries to provide a method for managing built-in modules from user space. (3.1)
- Add support for the implementation of `SEEK_HOLE` and `SEEK_DATA` in `lseek()`. (3.1)
- Add the `!` escape character to `/` in `hostname` and `comm` strings in core dumps. (3.1)
- If the value of the `sysctl` parameter `shm_rmid_forced` is set to 11, all shared memory objects are marked for removal with `IPC_RMID`. As this change breaks POSIX compliance, you need to ensure that no threads are using the orphaned memory. (3.1)
- Add support for generic I/O power management domains (v8) by introducing common headers, helper functions, and callbacks to allow platforms to use simple, generic power domains for runtime power management. (3.1)
- Add system-wide power transitions (system suspend and hibernation) support for generic domains (v5). Add `suspend`, `resume`, `freeze`, `thaw`, `poweroff`, and `restore` callbacks that are associated with `struct generic_pm_domain` objects and have `pm_genpd_init()` interpret them as appropriate. (3.1)
- Add wakeup device support for system-sleep transitions. Introduce a new generic power management domain callback routine, `.active_wakeup()`. This routine is used during the `noirq` phase of system suspend and hibernation to decide how to handle wakeup devices. (3.1)
- Add the ability to set a maximum limit for allowable CPU bandwidth to the process bandwidth controller. The limit is specified as a quota and a period for a group of processes. (3.2)
- To reduce the performance impact from using `i_mutex` lock with `generic_file_llseek()`, an almost lockless `generic_file_llseek()` is added to VFS that allows the maximum file size of the file system to be passed in, instead of always using `maxbytes` from the superblock. (3.2)
- A boot parameter of the form `root=PARTUUID=`*`uuid`*`,PARTNROFF=`*`partition_number_offset`* extends the `root=PARTUUID=`*`uuid`* syntax to select the root partition by specifying an integer offset from a known, unique partition. (3.2)
- Add a fault reporting mechanism to the input/output memory management unit (IOMMU) API. (3.2)
- Allow partition creation from user space and add discard support for loop devices. (3.2)

- When performing AIO, allocate `kiocb` structures in batches to reduce the CPU overhead of a process taking and releasing the context lock. (3.2)
- Add support for the tagged files ease-of-use feature in `sysfs`. (3.2)
- Add a `comm` change event to the process connector. (3.2)
- Add architecture-independent support for `highmem` page poisoning and verification to `debug-pagealloc`. (3.2)
- Add support for `poll()` in `sysctl` so that user-space applications can be notified of changes to `sysctl` entries. (3.2)
- The x32 kernel ABI (kABI) allows programs to take advantage of x86-64 features such as a larger number of CPU registers, better floating-point performance, faster position-independent code shared libraries, function parameters passed via registers, and faster system-call instructions. The kABI uses 32-bit pointers and avoids the overhead of 64-bit pointers. The program is limited to a 4-GB virtual address space. However, reducing the memory footprint can also allow a program to run faster. (3.4)
- The `nomodule` kernel parameter can be used to disable module loading as an alternative to using `sysctl`.
- The `prctl()` PR`_GET_CHILD_SUBREAPER` and `PR_SET_CHILD_SUBREAPER` options implement simple process supervision of orphaned processes. (3.4)
- Thread stacks are now marked correctly for `proc/`*`pid`*`/maps` under `procfs`. (3.4)
- Restore the `sysctl` setting `kernel.pty.max` as the global limit of pseudo terminals (by default, 4096). (3.4)
- Add abilities to turn the reboot notifier on or off, and to enter the debugger and stop kernel execution before rebooting. (3.4)
- To improve performance, VFS now uses `unsigned long` accesses for `dcache` name comparison and hashing. (3.4)
- `/proc/`*`pid`*`/task/`*`tid`*`/children` entries provide information about task children and can be useful for process checkpoint and restore operations. (3.5)
- `/proc/`*`pid`*`/pagemap` now reports whether file pages are `shared-anon` or `file-page`. (3.5)
- The `skew_tick` boot option mitigates `xtime_lock` contention on larger systems or read-copy-update (RCU) lock contention on all systems when `CONFIG_MAXSMP` is set. This option increases power consumption and should only be enabled if the system runs jitter-sensitive workloads (typically, HPC or RT). (3.5)
- Inode `stat` information is moved closer together to increase the likelihood of cache hits. (3.5)
- The `fallocate()` file-system operation allows preallocation space for a file. (3.5)
- Stale power-aware scheduling remnants and dysfunctional knobs have been removed from the process scheduler. (3.5)
- The `EPOLLWAKEUP` flag prevents system suspension while `epoll` events are ready. (3.5)
- `ramoops` uses the `pstore` interface instead of `/dev/mem`. (3.5)
- Add ECC support to `pstore/ram`. (3.5)

- `make tools` is now integrated with the kernel build system. (3.5)
- The kernel parameter `RCU_FANOUT_LEAF` can be used to control leaf-level fanout for RCU locking to reduce cache-miss initialization latencies on large systems. (3.5)
- RCU locking now implements a direct algorithmic sleepable RCU (SRCU) implementation to prevent OS jitter and performance degredation. (3.5)
- Add `rbtree` node caching support to IPC `mqueue` for the case where the queue is empty, improve performance of `send/recv`, and update maximums for the `mqueue` subsystem. (3.5)
- Add symbolic and hard link restrictions to VFS to address security issues. (3.6)
- Improvements to the IOMMU group implementation. (3.6)
- Remove the non-working x86 power estimation feature from the process scheduler. (3.6)
- Add hysteresis attributes (used by most thermal sensors) on a per-trip-point basis to the thermal framework. (3.6)
- Add support for states that affect multiple CPUs. This is potentially useful in implementations where CPUs leverage a shared, coupled power state. (3.6)
- The `rcutree.rcu_fanout_leaf` boot parameter allows the value of `RCU_FANOUT_LEAF` to be increased but not decreased. (3.6)
- Firmware files can be loaded directly from the file system rather than from `udev`. (3.7)
- `xattr` support in cgroups allow run-time metadata to be attached to cgroups. (3.7)
- The `disable_nmi` command in `kdb` disables NMI-entry and releases the port. (3.7)
- Add a special serial console driver to allow the temporary use of an NMI debugger port as a normal console via the `nmi_console` command. (3.7)
- RCU locking changes:
  - Control grace period duration from `sysfs`.
  - Make `rcutree` module parameters visible in `sysfs`.
  - Allow an RCU lock to be placed in an extended quiescent state when the CPU runs in user space.

  (3.7)
- Add system call to enforce that kernel modules are loaded only from a read-only cryptographically verified root file system. (3.8)
- Applications can choose between using 1-GB and 2-MB huge pages. Typically, this feature is used in conjunction with a NUMA policy. (3.8)
- Add option to allow assignment of a memory node as movable memory, which allows an entire node to be hot-pluggable. (3.8)
- Add `sysctl` variables to tune checkpoint/restart in user space (CRIU) including specifying the ID of the next IPC object to be allocated. (3.8)
- Introduce CRIU message queue copy feature so that all pending IPC messages can be retrieved without deleting them from the queue. (3.8)

- Correct the implementation of hierarchy support for the freezer cgroup. If a cgroup is frozen, all its descendants are also frozen. (3.8)
- Implement the `PTRACE_O_EXITKILL` `ptrace()` request. (3.8)
- Add the `VmFlags` field to `/proc/PID/smaps` output. Required by CRIU. (3.8)
- Add `TIOCGPKT`, `TIOCGPTLCK` and `TIOCGEXCL` `ioctl()` calls to obtain the package mode and locking state of a pseudo terminal, and to obtain exclusive mode on a tty. (3.8)
- Add a module parameter to force the use of expedited RCU primitives, which can benefit some embedded applications. (3.8)
- Allow selected CPUs to have RCU callbacks offloaded to kthreads to prevent or minimize OS jitter. (3.8)
- Provide support in `sysfs` to determine the maximum number of virtual functions (VFs) and Single Root I/O Virtualization (SR-IOV) capable PCIe devices that are supported, and the methods that are available for enabling and disabling VFs on a per-device basis. (3.8)
- Add a `sysfs` node to present the available frequencies for power management. (3.8)
- Add the `PM_QOS_FLAG_NO_POWER_OFF` and `PM_QOS_FLAG_REMOTE_WAKEUP` power management QoS device flags. (3.8)
- Add a `sysfs` node to present frequency transition information for power management. (3.8)

A.4 Cryptography

- Ablkcipher now support encryption and decryption for AES, DES, and 3DES. (3.1)
- Add an eCryptfs mount option to check that the UID of the device being mounted is the same as the expected UID. (3.1).
- The `encrypted` key type has been extended with the introduction of the `ecryptfs` format, intended for use with the eCryptfs file system. The `ecryptfs` format stores an authentication token structure inside an encrypted key payload, containing a randomly generated symmetric key. (3.1)
- An new user-space configuration API enables the instantiation, removal, and display of cryptographic algorithms from user space. (3.2)
- An x86-64 implementation of Blowfish provides two sets of assembler functions:
  - Regular one-block-at-a-time (1-way) encryption and decryption functions
  - Four-blocks-at-a-time (4-way) functions that provide improved performance on out-of-order CPUs

  On in-order CPUs, the performance of 4-way functions should be equal to that of 1-way functions. (3.2)
- An x86-64 assembler implementation of the SHA1 algorithm uses Supplemental Streaming SIMD Extensions 3 (SSSE3) instructions or Advanced Vector Extensions (AVX) if available. Testing with the `tcrypt` module demonstrates that raw hash performance is up to 2.3 times faster than the C implementation. (3.2)
- A 3-way parallel x86-64 assembler implementation of Twofish encrypts data in three-block chunks, which improves cipher performance on out-of-order CPUs. (3.2)
- Add support for MD5 algorithms to CAAM. (3.3)

- RSA digital-signature verification is implemented using the multiprecision math library from GnuPG, and is used by the IMA/EVM digital signature extension. (3.3)
- A 4-way parallel i586/SSE2 assembler implementation of Serpent encrypts data in 4-block chunks. (3.3)
- An 8-way parallel x86-64/SSE2 assembler implementation of Serpent encrypts data in 8-block chunks (two 4-block chunk SSE2 operations are performed in parallel to improve performance on out-of-order CPUs). (3.3)
- LRW and XTS support added to Serpent-sse2. (3.3)
- HMAC algorithms added to Talitos. (3.3)
- XTS support added to `twofish-x86_64-3way`. (3.3)
- Add sha224 and sha384 variants to existing AEAD algorithms in CAAM. (3.4)
- Add x86-64 assembler implementation of the Camellia block cipher. Two sets of functions are provided:
  - Regular one-block-at-a-time (1-way) encryption and decryption functions
  - Two-blocks-at-a-time (2-way) functions that provide improved performance on out-of-order CPUs

  On in-order CPUs, the performance of 2-way functions should be equal to that of 1-way functions. (3.4)
- Add Tegra AES hardware driver supporting `ecb`, `cbc`, `ofb`, and `ansi_x9.31rng` modes, and 128, 192 and 256-bit key sizes. (3.4)
- Add a slice-by-8 algorithm to the existing slice-by-4 algorithm in `crc32`. The BITS size is expanded from 32 to 64, tables are extended from `tab[4][256]` to `tab[8][256]`, and inner-loop code is added. (3.4)
- Improve performance of `aesni_intel` by using parallel LRW and XTS encryption with AES-NI hardware pipelines. (3.7)
- Add IPSec extended sequence number (ESN) support to CAAM and Talitos. (3.7)
- A x86-64/AVX assembler implementation of the Cast5 block cipher allows 16 blocks to be processed in parallel. (3.7)
- Implement signature verification algorithms for RSA public key cryptography. At present, only the signature verification algorithm is supported (PKCS# | RFC3447). (3.7)
- Add a crypto key parser for binary (DER) X.509 certifications, an ASN.1 decoder, and a simple ASN.1 grammar compiler. (3.7)
- Add HASH-HMAC with SHA algorithms and MD5 to CAAM. (3.6)
- Add hardware random number generator support to CAAM. (3.6)
- Add a x86-64/AVX assembler implementation of the Serpent block cipher. (3.6)
- Add x86-64/AVX assembler implementation of the Twofish block cipher. (3.6)
- Add sha224, sha384, and sha512 to the existing AEAD algorithms in Talitos so that it supports all combinations of CBC (AES, 3DES-EDE) and HMAC (SHA-1, 224, 256, 384, and 512). (3.6)

A.5 Device Mapper

- The always writable feature indicates that a target does not support read-only mode. (3.2)

- The immutable feature indicates that a target type cannot be mixed with any other target type. Once loaded into a device, it cannot be replaced with a table that contains a different type. (3.2)
- Add a singleton table that can contain only one target. (3.2)
- Log device dependency allows registration of a log device so that it is included in the list of device dependencies. (3.2)
- A verity target allows a device to store cryptographic hashes of file system blocks. The device can be used to check every read of the file system. If the hash of the block does not match that of the file system, the read fails. (3.4)

A.6 Driver Support

- Broadcom NetXtreme II 10Gbps network adapter driver (`bnx2x`): Add AutogrEEEn support for BCM84833 and 5418se, and multiple concurrent I2 traffic classes. (3.1)
- Broadcom NetXtreme II iSCSI driver (`bnx2i`): Add support for 57800, 57810, and 57840. (3.1)
- Brocade BFA FC SCSI driver (`bfa`):
  - FAA support
  - HBA diagnostic support
  - CEE information and statistics query
  - Flash configuration
  - Collect and reset `fcport` statistics
  - Configure LUN masking
  - Configure QoS and collect statistics
  - Support for obtaining SFP information
  - Support for FC-transport based Asynchronous Event Notification
  - Support for I/O profiling
  - Collect or reset fabric statistics
  - Configure and query flash boot partition
  - Configure trunking on Brocade adapter ports
  - store driver configuration in flash memory
  - Brocade-1860 Fabric Adapter 16Gbs support and flash controller fixes
  - Brocade-1860 Fabric Adapter Hardware enablement
  - Brocade-1860 Fabric Adapter vHBA support
  - Initiator-based LUN masking

  (3.1)

- Emulex Blade Engine 2 10Gbps adapter driver (`be2net`): Add support for multiple Tx queues. (3.1)
- Emulex FC/FCoE driver (`lpfc`): Add FCF priority failover functionality. (3.1)
- Intel PRO/1000 PCI-Express Gigabit network adapter driver (`e1000e`): Add Jumbo Frame support for the 82583 Gigabit Ethernet Controller. (3.1)
- QLogic 1/10 GbE Converged/Intelligent Ethernet Adapter driver (`qlcnic`): Add multi-protocol internal loopback support. Driver can now generate loopback traffic, conduct tests, and return the results to an application. (3.1)
- `coretemp`: Add core and package threshold support. The thresholds are configured using the `tempX_max` and `tempX_max_hyst` interfaces in `sysfs`. An interrupt is generated if the CPU temperature reaches or crosses above `tempX_max` or if it drops below `tempX_max_hyst`. To allow the hysteresis mechanism to work, the value of `tempX_max` should be configured to be several degrees higher than the value of `tempX_max_hyst`. (3.1)

A.7 File Systems

btrfs

- Add a `DCACHE_NEED_LOOKUP` flag to `d_flags` to improve the performance of `ls` and `readdir()`. (3.1)
- Switching from tree locks to reader/writer locks improves the performance of read and write-intensive workloads. (3.1)
- Performance improvements in several areas, particularly for random write workloads. (3.2)
- Allowing overcommit of `ENOSPC` reservations to improve performance. (3.2)
- Add automatic backup of superblock information about tree roots for the previous 4 commits. Add the `-o recovery` mount option to enable use the root history log if required. (3.2)
- Add code to follow back references, replacing the manual process for walking those references, and including more detailed corruption messages. (3.2)
- Allow user-space utilities to inspect metadata. (3.2)
- Improve performance of checksum verification of read-aheads. (3.2)
- Add the `nospace_cache` mount option to disable cache loading without clearing the cache. (3.2)
- Improve performance of committing transactions. (3.2)
- When mounting a subvolume, allow a path relative to the tree root to be specified to `-o subvol`. (3.2)
- Rework the logic for cluster allocation. (3.3)
- Rewrite the block group trimming code. (3.3)
- Increase the size of system chunks. (3.3)
- Remove caching code that caused unnecessary fragmentation and complexity. (3.4)
- Remove the code to silently switching single chunks to RAID-0 when balancing a file system. The restriper now allows a choice of RAID-0 or concatenation. (3.4)

- Support metadata blocks that are larger than 4 KB. (3.5)
- The `thread_pool` size can be changed at remount time. (3.5)
- Add the `DEVICE_READY ioctl()` to be used in conjunction with `btrfs device ready device`, providing a lightweight method of telling if all the devices required for a file system are currently in the cache. (3.6)
- Allow compression to be disabled by specifying the `compress=no` mount option. (3.6)
- Improve multithread buffer reads. (3.6)
- Support UUIDs for subvolumes, and introduce `ctime`, `otime`, `stime`, and `rtime` for subvolumes, including a `transid` for each time. (3.6)
- Rework the `DEV_STATS ioctl()` to allow it to either get or reset device statistics depending on the argument specified. (3.6)
- Make the `compress` and `nodatacow` mount options mutually exclusive. To improve `O_SYNC` performance, asynchronous metadata checksumming is not performed under some circumstances. (3.7)

For more information, see https://btrfs.wiki.kernel.org/index.php/Changelog.

cifs

- Add UID/GID to SID mapping. (3.2)
- Add `backup` mount option. (3.2)
- Allow larger `rsize` (up to 16 MB) and change the default to 1 MB. (3.2)
- Introduce credit-based flow control. (3.4)
- Add the `cache=strict|none` mount option to specify the cache type instead of the `strictcache` and `forcedirectio` options. The legacy options are now mutually exclusive. (3.5)
- The `vers=2.1` mount option forces an SMB2 mount. By default, `vers=1` (CIFS) is used. (3.5)
- The `vers=2.0` mount option forces an SMB2.02 mount. (3.8)

ext4

- Reduce CPU overhead when appending files preallocated using `fallocate()` with mode `FALLOC_FL_KEEP_SIZE` via direct I/O. (3.2)
- Reduce CPU overhead by optimizing `memmove()` lengths in extent and index insertions. (3.2)
- Support block sizes of up to 1 MB using the `-C` option to `mkfs.ext4`. This change is not backwards compatible with older kernels. (3.2)
- Remove the `resize` and `journal=update` mount option. (3.4)
- Improve performance of `truncate` and `unlink`. (3.7)
- Support online resizing of metablock group (`META_BG`) and 64-bit file systems. (3.7)
- Add `max_dir_size_kb` mount option to specify a maximum directory size. (3.7)
- Re-enable `-o discard` functionality in no-journal mode. (3.7)

- Remove support for disabling extended attributes. (3.8)
- Implement support for `SEEK_DATA` and `SEEK_HOLE`. (3.8)

NFS

- Add support for the RAID-5 read-4-write interface. (3.2)
- Add `v4.0` and `v4.1` mount options. (3.4)
- The kernel can deduce the value of `clientaddr` if this mount option is not specified for NFS v4. (3.4)
- Add the `migration` mount option that specifies whether a server supports Transparent State Migration (TSM). (3.7)
- Handle IPv6 remote addresses from GETDEVICEINFO (required for pNFS). (3.8)
- Remove the deprecated `nfsctl()` system call and all related code. (3.8)

pstore

- Add runtime logging support for kernel messages to allow debugging of hangs caused by hardware issues. (3.6)
- Add console message handling. The log size is configurable by using the `ramoops.console_size` module option, and the log is accessible at *`pstore-mountpoint`*`/console-ramoops`. (3.6)
- Add persistent function tracing. The kernel can save the function call chain log to a persistent RAM buffer, which can be decoded and dumped after a reboot. You can use the log to determine the function that was called immediately prior to a reset or panic. (3.6)

tmpfs

- Increase the file size limit for tmpfs. (3.1)
- Support `fallocate()` `FALLOC_FL_PUNCH_HOLE` and preallocation. (3.5)

XFS

- Improve performance of the inode cache. (3.1)
- Improve scalability of per-file-system quotas. (3.4)
- Implement support for `SEEK_DATA` and `SEEK_HOLE`. (3.5)
- Make the `inode32` and `inode64` mount options work with remounts. (3.7)
- Make `inode64` the default allocation mode. (3.7)
- Add the `XFS_IOC_FREE_EOFBLOCKS` `ioctl()` to enable `EOFBLOCKS` scanning. (3.8)

A.8 Memory Management

- Add `memory.vmscan_stat` memory control group that displays numbers of scanned, rotated, and freed pages, and elapsed times for direct reclaim and soft reclaim. (3.1)
- Extend the memory hotplug API to allow memory hotplug in virtual machines. Also required for the Xen balloon driver. (3.1)

- Fix significant stalls in the page allocator when copying large amounts of data on NUMA machines. (3.1)
- Add `slub_debug` method to the `slub` slab allocator to check if memory is not freed and help diagnose memory usage. (3.1)
- Reduce CPU overhead of `slub_debug`. (3.1)
- The cross memory attach feature adds the system calls `process_vm_readv` and `process_vm_writev()`, which allow data to be transferred between the address spaces of the two processes without passing through kernel space. (3.2)
- Add a block plug for page reclaim to `vmscan` that reduces CPU overhead by reducing lock contention and merging requests. (3.2)
- Implement per-CPU cache in slub for partial pages. (3.2)
- Restrict access to slab files under `procfs` and `sysfs`, hiding `slabinfo` and `/sys/kernel/slab/*`. (3.2)
- Add the `slab_max_order` kernel parameter that determines the maximum allowed order for slabs. High settings can cause OOMs due to memory fragmentation. The default value is 1 for systems with more than 32 MB of RAM. Otherwise, the default value is 0. (3.3)
- To increase the probability of detecting memory corruption, change the buddy allocator to retain more free, protected pages and to interlace free, protected pages and allocated pages. (3.3)
- Charge the pages dirtied by an exited process to random dirtying tasks. (3.3)
- Allow the poll time and call intervals to balance dirty pages to be controlled by the value of the `max_pause` parameter. (3.3)
- Fix dirtied pages accounting on sub-page writes. (3.3)
- Introduce the dirty rate limit to compensate a task's think time when computing the final pause time. (3.3)
- Reduce dirty throttling polls and CPU overhead. (3.3)
- Avoid tiny dirty poll intervals. (3.3)
- Make swap-in read-ahead skip over holes, allowing the system to swap back in at several MB/s, instead of a few hundred kB/s. (3.4)
- Introduce bit-optimized iterator and radix tree cleanup in the core page cache. (3.4)
- Improve allocation of contiguous memory chunks by adding DMA mapping helper functions. (3.5)
- Remove swap token code and lumpy reclaim. (3.5)
- Improve throughput and reduce CPU overhead by allowing swap read-ahead to be merged. (3.6)
- Add cgroup controller that allows HugeTLB usage per control group to be limited and enforces the limit during page faults. (3.6)

A.9 Networking

- Add CPU fanout policies for hashing to the packet interface based on mapping socket buffers to Rx hashes, and a pure round-robin scheme. (3.1)

- Improve the client announcement mechanism in the Better Approach To Mobile Adhoc Networking (B.A.T.M.A.N.) routing protocol. The change resolves performance and latency issues with the previous implementation by appending client changes (new client joined or client left) to the OGM. System overhead is reduced by allowing nodes to modify their global tables by means of updates. The new `ROAMING_ADVERTISEMENT` packet type eliminates latency and packet drop issues seen with OGM broadcasting. (3.1)
- Add support for zero-copy socket buffers. Adds user-space buffer support in the socket buffer shared information. (3.1)
- Use MD5 to compute protocol sequence numbers and fragment IDs per RFC1948. Update code to take into account current CPU speeds and to use a full 32-bit sequence number. (3.1)
- Add a multicast group for DCB to provide a clean method for disseminating kernel DCB link attributes to user space. (3.1)
- Add SELinux context support to the `AUDIT` target of `netfilter`. (3.1)
- Add range support for IPv4 to `netfilter`. (3.1)
- Lower the default init retransmission timeout (RTO) from 3 seconds to 1 second per RFC2988bis. The RTO falls back to 3 seconds if a `SYN` or `SYN-ACK` packet has been retransmitted and the TCP time stamp option is not on. (3.1)
- Implement support for Auto-ASCONF (see RFC5061) in the Stream Control Transmission Protocol (SCTP) stack. The change includes features for enabling and configuring settings. (3.1)
- Reduce the false sharing effect. (3.1)
- Reduce CPU overhead of `check_leaf()` with the route cache disabled. (3.1)
- Add support to the `virtio_net` driver to obtain Rx and Tx ring parameter information from an Ethernet device. Used by the `ethtool -g ethX` command. (3.2)
- Implement AP isolation on the receiver and sender side for B.A.T.M.A.N. When a node receives a unicast packet, it checks whether the source and destination client can communicate due to the AP isolation. (3.2)
- Remove the IPv4 `gc_interval` from `sysctl`. (3.2)
- Add `TPACKET_V3` support including a flexible buffer implementation. (3.2)
- Allow forwarding of some link-local frames by network bridges. You can use `/sys/class/net/brX/bridge/group_fwd_mask` in `sysfs` to control frame forwarding. (3.2)
- Implement TCP proportional rate reduction. (3.2)
- Add `netlink`-based Content Addressable Network (CAN) routing. (3.2)
- Add support for the socket monitoring interface used by the `ss` tool. (3.3)
- Add support for the SCSI RDMP Protocol (SRP) target driver. The SRP protocol allows an initiator to access a block storage device on another host (target) over a network that supports the RDMA protocol. Currently, the RDMA protocol is supported by InfiniBand. (3.3)
- Add unresolved queue limits to `neigh`. Deprecate `/proc/sys/net/ipv4/neigh/default/unres_qlen`, and replace it with `unres_qlen_bytes`. (3.3)
- Add CAIF USB support. (3.3)

- Add an extended accounting infrastructure for `netfilter` over `nfnetlink`, which allows the display of real-time traffic accounting without requiring a complicated and resource-consuming implementation in user space. (3.3)
- Add `nfacct match` to `netfilter`, which supports extended accounting. (3.3)
- Add reverse patch filter (`rpfilter`) to `netfilter`, which allows matching of packets where replies use the same interface on which the packet arrived. (3.3)
- Add adaptive random early detection (RED) active queue management (AQM) to the packet scheduler. (3.3)
- Add an optional RED on top of stochastic fairness queueing (SFQ) to the packet scheduler, enabling SFQ features such as specifying a smaller per flow limit for in-flight packets, up to 65408 active flows (as compared to 127 previously), head drops instead of tail drops, and optional RED on each SFQ flow queue. (3.3)
- Add 802.1q `netpoll` support to `vlan`. (3.3)
- Add `NTF_USE` bridge support plus other changes to allow the control of forwarding database via `netlink`. (3.3)
- New plug-queuing discipline allows a user space application to plug or unplug a network output queue via the Netlink interface. (3.4)
- Add the ability to change the routing algorithm at runtime to B.A.T.M.A.N. (3.4)
- RCU conversion in TCP allows access to MD5 keys without locking the listener socket. (3.4)
- For some workloads, allowing `splice()` to build full TSO packets can reduce number of logical packets sent by an order of magnitude, making zero-copy TCP faster than one-copy. (3.4)
- Add the `SO_PEEK_OFF` socket option. (3.4)
- Support peeking offset for datagram sockets, seqpacket sockets, and stream sockets. (3.4)
- Add `MSG_TRUNC` support for datagram sockets so that `recv()` returns the real length of the packet, even if it is longer than the passed buffer. (3.4)
- Add missing `SO_NOFCS` socket option. (3.4)
- Add timeout extension to `netfilter`, which allows timeout policies to be attached to the flow via the connection tracking target. Add the `cttimeout` infrastructure for fine timeout tuning. (3.4)
- Add NAT support for expectation classes in `netfilter`. (3.4)
- Add exceptions support to `netfilter`. (3.4)
- Merge `ipt_LOG` and `ip6_LOG` into `xt_log` in `netfilter`. (3.4)
- Add hardware-independent IEEE 802.15.4 networking stack for softMAC devices. (3.5)
- Tune performance of `sk_add_backlog`. (3.5)
- Add binary option type, a load-balancer module, a per-port option for enabling or disabling ports, and support for per-port options to the `team` device. (3.5)
- Add raw packet `QP` type `IB_QPT_RAW_PACKET` to InfiniBand core. This allows applications to build a complete packet, including L2 headers, when sending. On the receive side, the hardware does not strip any headers. This feature is designed for user-space direct access to Ethernet. (3.5)

- Treat ND option 31 as user land (DNSSL support) in IPv6 per RFC6106. The 8-bit identifier of the DNSSL option type assigned by the IANA has the value 31. (3.5)
- Replace basic bridge loop avoidance code in the `batman-adv` module. (3.5)
- Set traffic class for CAIF packets based on socket priority, CAIF protocol type, or type of message. (3.5)
- Add generic `PF_BRIDGE:RTM_FDB` hooks and two new flags: `NTF_MASTER` and `NTF_SELF`. (3.5)
- Add Explicit Congestion Notification (ECN) capability to `pktsched`. Instead of dropping packets, attempt to mark them as ECN. (3.5)
- Remove support for token ring. (3.5)
- Remove support for Econet protocol. (3.5)
- Add an optional QoS attribute to DCB netlink to allow the setting of a rate limit for an ETS TC. 3.5
- Add CEE notify calls when an APP change or `setall` command is made from user space. (3.5)
- Add HMARK target support to `netfilter`. (3.5)
- If `net.bridge.bridge-nf-filter-vlan-tagged` is enabled in `sysctl`, bridge `netfilter` removes the `vlan` header temporarily and feeds the packet to `iptables` or `ip6tables`. Add `bridge-nf-pass-vlan-input-device`, which if set to `on` (default is `off`), `netfilter` also sets the `in` interface to the `vlan` interface if this interface exists. This change allows the `iptables REDIRECT` target work with *vlan-on-top-of-bridge* configurations and the use of `iptables -i`" to match the vlan device name. (3.5)
- Allow byte-based limit mode can be used with `netfilter`, for example, to support ingress-traffic policing or to detect when a host or port consumes more bandwidth than expected. (3.5)
- Add support for sync threads to `netfilter`. (3.5)
- Remove `ip_queue` support from `netfilter`. (3.5)
- Add support for Layer 2 Tunneling Protocol (L2TP) over UDP in IPv6. (3.5)
- Add L2TPv3 IP encapsulation support for IPv6. (3.5)
- Add `netlink` API for L2TPv3 unmanaged tunnels over IPv6. (3.5)
- Remove IPv4 routing cache that was vulnerable to denial of service attacks. (3.6)
- Implement RFC 5691 3.2 and RFC 5961 4.2 (Mitigation against Blind Reset attack using RST bit and SYN bit). (3.6)
- Add VTI support. (3.6)
- Add an interface option `route_localnet` that enables the routing of the 127/8 address block and processing of ARP requests on a specific interface (for example, to address a pool of virtual guests behind a load balancer). (3.6)
- Add `multiqueue` and `netpoll` support to `team`. (3.6)
- Add experimental zero-copy Tx support to `tun`. (3.6)
- Add support for 40GbE. (3.6)

- Add fail-open support to `netfilter`, where the queue-full condition does not drop packets. (3.6)
- Add user-space connection tracking helper infrastructure to `netfilter`. (3.6)
- Extends the `ethtool` interface to add support for the EEE commands: `get_eee`'and `set_eee`. (3.6)
- Add Generic Routing Encapsulation (GRE) over IPv6, generic segmentation offload (GSO), and GRO capability. (3.7)
- Set default MTU for `loopback` devices to 64 KB. Allows TCP stacks to build large frames and significantly reduces stack overhead. (3.7)
- Add an extended attribute to store data for the mapping between inode numbers in `sockfs` and protocol types for use by `lsof`. 3.7
- Implement a per-task fragmentation allocator, which can improve TCP stream performance by 20% on `loopback` devices. (3.7)
- Various `netfilter` changes:
  - Add a protocol-independent NAT core.
  - Add IPv6 `MASQUERADE` target.
  - Add IPv6 `NETMAP` target.
  - Add IPv6 `REDIRECT` target.
  - Add IPv6 `AT` support.
  - Support IPv6 FTP NAT helper.
  - Support IPv6 IRC NAT helper.
  - Support IPv6 SIP NAT helper.
  - Support IPv6 in the amanda NAT helper.
  - Add stateless IPv6-to-IPv6 Network Prefix Translation target.
  - Remove `xt_NOTRACK`.

  (3.7)
- Add link layer control (LLC) core layer to HCI 2, add an SHDLC `llc` module to the `lic` core, and add LLCP raw socket support to NFC. (3.7)
- Support IPv6 transmit hashing (and TCP or UDP over IPv6) in the bonding driver. (3.7)
- Add support for dumping diagnostic core and basic socket information (family, type and protocol) at socket creation time. (3.7)
- Add support to `ethtool` for setting the MDI/MDI-x state for twisted-pair wiring. (3.7)
- Add 64-bit statistics support to PPP, including `tx_bytes`, `rx_bytes`, `tx_packets`, and `rx_packets`. 3.7
- Add generic `netlink` support for `tcp_metrics` that allows unlinking and deletion of entries after a grace period. (3.7)

- Add bridge port parameters over `netlink` to permit dumping, monitoring, and changing the bridge multicast database. (3.8)
- Add support for RFC 5961 5.2 Blind Data Injection Attack Mitigation. (3.8)
- Change default TCP hash size, and add support for hardware-offloaded encapsulation and offloading of encapsulated packets for VXLAN and IP GRE. (3.8)
- Add vlan tag access to `netfilter`. (3.8)
- Add extensions to VXLAN to support Distributed Overlay Virtual Ethernet (DOVE) networks. (3.8)
- Add IPv6 `set` action functionality to `openswitch`. (3.8)
- Add GSO support to IPIP tunnels, increasing the performance of a single TCP flow. (3.8)
- Implement IPv6 fragment handling for IPVS (3.8)
- Add support in `netfilter` for querying the destination address of a redirected connection. (3.8)
- Add `NOTRACK` target recovery to `netfilter`. (3.8)
- Implement QFQ+ in `sched`. (3.8)
- Add support for RTM_GETNETCONF to routing `netlink`. (3.8)
- Add support for per-association statistics by implementing the `SCTP_GET_ASSOC_STATS` call for the Stream Control Transmission Protocol (SCTP). (3.8)
- Add a `sysctl` that allows the selection of the HMAC algorithm (static or dynamic) used by SCTP. (3.8)
- Add support for `SO_ATTACH_FILTER` required to save the full state of a socket. (3.8)
- Convert tun/tap into a multiqueue device and expose the queues as file descriptors in user space. (3.8)

A.10 perf Utility

- Add the `--symfs` option to `perf annotate`. (3.2)
- Add the `drop monitor` script. (3.2)
- Add the `-o` and `--append` options to `perf stat`. (3.2)
- Add the `-M` option. (3.2)
- Add annotation output controls to all `perf` tools that have integrated annotation. (3.2)
- Include information about the host environment in `perf.data`:

`HEADER_HOSTNAME`	Host name.
`HEADER_OSRELEASE`	Kernel release number.
`HEADER_ARCH`	Hardware architecture.
`HEADER_CPUDESC`	Generic CPU description.
`HEADER_NRCPUS`	Number of online, available CPUs.

`HEADER_CMDLINE`	`perf` command line.
`HEADER_VERSION`	`perf` version.
`HEADER_TOPOLOGY`	CPU topology.
`HEADER_EVENT_DESC`	Full event description (`attrs`).
`HEADER_CPUID`	Easy-to-parse, low-level CPU identification.

(3.2)

- Accept FIFOs as input files. (3.3)
- Add `-a` option for system-wide profiling. (3.3)
- Implement printing snapshots to files. (3.6)
- Add sort by source line number. (3.6)
- Add PMU event alias support. (3.6)
- Add support for `perf kvm stat` to analyze `kvm` `vmexit`, `mmio`, and `ioport`. (3.7)
- Add union member access. (3.7)
- Add `--list-opts` option to print long option names for use with bash. (3.7)
- Add script browser. (3.8)
- Add new display options (`-F`, `-p`, and `-P`) to `perf diff`. (3.8)
- `perf inject` now supports input from a file. 3.8
- Add `--pre` and `--post` options to `perf stat`. (3.8)
- Add `gtk.`*`command`* `config` option to launch the GTK browser. This is equivalent to specifying `--gtk` option on command line (3.8)
- Add new features to `perf trace`. (3.8)
- Expose hardware events translations in `sysfs`. (3.8)
- Add `trace_options` boot parameter to set trace options at boot time, such as enabling event stack dumps. (3.8)

A.11 Power Management

- Add a generic DVFS framework with device-specific (non-CPU) OPPs. (3.2)
- Improve performance of LZO/plain hibernation. (3.2)
- Implement per-device power management QoS contraints. (3.2)

A.12 Security

- Add `/sys/kernel/security/tomoyo/audit_interface`, which generates audit logs in the form of domain policy so they can be reused and appended to domain_policy interface by the TOMOYO auditing

daemon (`tomoyo-auditd`). TOMOYO is a kernel security module which implements mandatory access control (MAC). (3.1)

- Add ACL group support for TOMOYO, which allows permissions to be globally granted. (3.1)
- Add policy namespace support for LXC (Linux containers). The policy namespace has its own set of domain policy, exception policy and profiles, independent of other namespaces. (3.1)
- Add built-in policy support needed to support enforcing mode from early in the boot sequence. (3.1)
- Make several TOMOYO options configurable to support activating access controls without calling an external policy loader program. (3.1)
- Permit the use of the following properties as conditions with TOMOYO: `argv[]`, `envp[]`, `execve()`, executable's real path and symlink target, owner or group of file objects, and the UID or GID of the current thread. (3.1)
- Implement Extended Verification Module (EVM), which protects a file's security extended attributes (`xattrs`) against integrity attacks. (3.2)
- Implement Smack protections for domain transition: BPRM unsafe flags, secure exec, clear unsafe personality bits, and clear parent death signal. (3.2)
- Enhance performance of Smack rule list lookups. (3.2)
- Allow user access to `/smack/access`, removing the requirement for `CAP_MAC_ADMIN`. (3.2)
- Add environment variable name restriction to TOMOYO. (3.2)
- Add socket operation restriction to TOMOYO. (3.2)
- Add control for generation of access granted logs in TOMOYO. (3.2)
- Allow domain transition without `execve()` in TOMOYO. (3.2)
- Allow audit matching on inode `gid`. (3.3)
- Allow inter-field comparison in audit rules between the `gid` of a running task and the `gid` of an inode. (3.3)
- Add a new audit filter type `AUDIT_FIELD_COMPARE` to indicate which fields should be compared. (3.3)
- Allow system call exit filter matching based on the `uid` of the owner of an inode used in the call. (3.3)
- Add support for digital signature verification in EVM. File metadata can be protected using digital signatures instead of HMAC. (3.3)
- Add a Yama Linux security module to collect DAC security improvements. (3.4)
- Add AppArmor security module file tracking to `securityfs`. (3.4)
- Add AppArmor security module initial features directory to `securityfs` for displaying boolean features flags and the known capability mask. (3.4)
- Add `default_type` statements to SELinux. (3.5)
- Add default source and target selectors for the user, role, and range of new objects in SELinux. (3.5)
- Allow seek operations on the file-exposing policy used by the `sesearch` SELinux policy query tool. (3.5)

- Add auditing of failed attempts to set invalid labels in SELinux. (3.5)
- Add checking for the open permission on truncate calls to SELinux. (3.5)
- Support long Smack labels. (3.5)
- Set recursive transmute attribute for Smack in all cases. (3.5)
- Allow manager programs which do not start with `/` in TOMOYO to handle differences between distributions. (3.5)
- Add two modes to the Yama `ptrace` restrictions. (3.5)
- Add support for invalidating a key. (3.5)
- Implement revoking of all rules for a subject label in Smack. (3.7)
- Allow Yama to be unconditionally stacked, regardless of which LSM module is primary. (3.7)
- Add the Integrity Measurement Architecture, which supports audit log hashes, digital signature verification, and the integrity appraisal extension. (3.7)

A.13 Storage

Block management in the software RAID MD layer now adds bad blocks to a bad-block list so that the system does not use them. (3.1)

A.14 Virtualization

- Add memory hotplug support for the Xen balloon driver. (3.1)
- Add Xen PCI backend driver. (3.1)
- Implement discard requests and support old-style BARRIER. (3.2)
- Increase recommended maximum number of VCPU from 64 to 160. (3.4)
- Allow host IRQ sharing for assigned PCI 2.3 devices. (3.4)
- Add infrastructure for software and hardware-based TSC rate. (3.4)
- Move the Hyper-V storage driver out of the staging area. (3.4)
- Add support for VLAN trunking to Hyper-V. Linux guests can now configure multiple VLANs using a single synthetic NIC on a Windows 8 Hyper-V host. (3.4)
- Support new KVP message types. (3.4)
- Support new KVP verbs for Hyper-V in the user level daemon. (3.4)
- Implements multiconsole support for Hyper-V. 3.4
- Support enumeration from all available pools for Hyper-V. (3.4)
- Update Xen ACPI processor to implement C and P state driver that uploads ACPI data to the hypervisor. (3.4)
- Add netconsole support to Xen. (3.4)

- Use the S4 code to provide S3 support for `virtio` devices. (3.4)
- Add a `virtio`-based remote processor messaging bus to allow message-based communication with the remote processor (if supported by the firmware). (3.4)
- Add direct MSI message injection for in-kernel IRQ chips. (3.5)
- Unregister from the `hwrng` interface and remove the `virtio` queue before entering the S3 or S4 states. On restore, add the `virtio` queue and re-register with `hwrng`. (3.6)
- Add `mcelog` support to Xen. (3.6)
- Reduce the I/O path in the guest kernel to achieve high IOPS and lower latency. (3.7)
- Add Xen EFI video mode support. (3.7)
- Implement backend support for paged out grant targets (retry loop and hooks). (3.7)
- Implement Xen ACPI processor aggregator driver (`pad`). (3.8)
- Remove support for i386 processors. (3.8)