Table of Contents

Document Control / Version Documentation ................................................................. viii
Overview ......................................................................................................................... 1
OSDL Carrier Grade Linux ............................................................................................... 1
Open Source Development Lab (OSDL) ........................................................................ 1
Carrier Grade Linux Working Group ............................................................................. 2
Selection of Requirements ............................................................................................. 2
Terms Used in this Document ......................................................................................... 2
   Category ....................................................................................................................... 3
   Requirement IDs .......................................................................................................... 3
   Version Assignment .................................................................................................... 3
   Priority ......................................................................................................................... 4
   Application Type ......................................................................................................... 5
OSDL CGL Requirements, Priority 1 ............................................................................. 6
   Category: Standards .................................................................................................... 6
      Requirement: 1.1 Linux Standard Base Compliance ................................................. 6
   POSIX Interface Compliance ....................................................................................... 6
      Requirement: 1.2.1 POSIX Timer Interface Standard Compliant ......................... 6
      Requirement: 1.2.2 POSIX Signal Interface Standard Compliant ....................... 7
      Requirement: 1.2.3 POSIX Message Queue Interface Standard Compliant ......... 7
      Requirement: 1.2.4 POSIX Semaphore Interface Standard Compliant ............... 7
      Requirement: 1.3 Event Logging POSIX IEEE 1003.25 ........................................... 7
   IPv6, IPSECv6, MIPv6 .................................................................................................. 7
      Requirement: 1.4.1 IPv6 RFCs ............................................................................... 7
      Requirement: 1.4.2 IPSECv6 RFCs .................................................................... 8
      Requirement: 1.4.3 MIPv6 RFCs ......................................................................... 9
   Category: Platform ...................................................................................................... 9
      Hotswap Support ..................................................................................................... 9
# Table of Contents

OSDL CGL Requirements Definition Version 1.1

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement: 2.1.1 Hot Insert</td>
<td>10</td>
</tr>
<tr>
<td>Requirement: 2.1.2 Hot Remove</td>
<td>10</td>
</tr>
<tr>
<td>Requirement: 2.1.3 Hot Device Identity</td>
<td>10</td>
</tr>
<tr>
<td>Requirement: 2.2 Remote Boot Support</td>
<td>10</td>
</tr>
<tr>
<td>Requirement: 2.3 Boot Cycle Detection</td>
<td>11</td>
</tr>
<tr>
<td>Requirement: 2.4 Loading Proprietary Modules</td>
<td>11</td>
</tr>
<tr>
<td>Requirement: 2.5 Diskless Systems</td>
<td>11</td>
</tr>
<tr>
<td><strong>No Console Operation</strong></td>
<td></td>
</tr>
<tr>
<td>Requirement: 2.6.1 Serial Console Connection</td>
<td>11</td>
</tr>
<tr>
<td><strong>Category: Availability</strong></td>
<td>12</td>
</tr>
<tr>
<td><strong>Hardened Driver Support</strong></td>
<td>12</td>
</tr>
<tr>
<td>Requirement: 3.1.1 Device Driver Hardening Spec</td>
<td>12</td>
</tr>
<tr>
<td>Requirement: 3.1.2 Sample Hardened Device Driver</td>
<td>12</td>
</tr>
<tr>
<td><strong>Watchdog Timer</strong></td>
<td>12</td>
</tr>
<tr>
<td>Requirement: 3.2.1 Watchdog Timer Interface Req</td>
<td>12</td>
</tr>
<tr>
<td>Requirement: 3.3 Application Heartbeat Monitor</td>
<td>13</td>
</tr>
<tr>
<td><strong>Ethernet Multiple NIC Bonding</strong></td>
<td>13</td>
</tr>
<tr>
<td>Requirement: 3.4.1 Ethernet Link Aggregation</td>
<td>13</td>
</tr>
<tr>
<td>Requirement: 3.4.2 Ethernet Link Failover</td>
<td>14</td>
</tr>
<tr>
<td><strong>RAID Support</strong></td>
<td>14</td>
</tr>
<tr>
<td>Requirement: 3.5.1 RAID 1 Support</td>
<td>14</td>
</tr>
<tr>
<td>Requirement: 3.6 Resilient Filesystem Support</td>
<td>14</td>
</tr>
<tr>
<td>Requirement: 3.7 Disk and Volume Management</td>
<td>15</td>
</tr>
<tr>
<td><strong>Category: Serviceability</strong></td>
<td>15</td>
</tr>
<tr>
<td><strong>Resource Monitor</strong></td>
<td>15</td>
</tr>
<tr>
<td>Requirement: 4.1.1 Resource Monitor Specification</td>
<td>15</td>
</tr>
<tr>
<td>Requirement: 4.1.2 Resource Monitor Framework</td>
<td>15</td>
</tr>
<tr>
<td>Requirement: 4.1.3 Resource Monitor Subsystems</td>
<td>16</td>
</tr>
<tr>
<td>Requirement: 4.1.4 Resource Monitor Data Persistence</td>
<td>16</td>
</tr>
<tr>
<td>Requirement: 4.1.5 Resource Monitor Performance</td>
<td>16</td>
</tr>
<tr>
<td><strong>Kernel Dumps</strong></td>
<td>16</td>
</tr>
<tr>
<td>Requirement: 4.2.1 Kernel Dump Targets</td>
<td>16</td>
</tr>
<tr>
<td>Requirement: 4.2.2 Kernel Summary Dump</td>
<td>17</td>
</tr>
</tbody>
</table>
Table of Contents

Requirement: 4.3 Kernel Message Structuring ................................................................. 17
Requirement: 4.4 Dynamic Debug/Probe Insertion .......................................................... 17
Requirement: 4.5 Platform Signal Handler .................................................................... 18
Requirement: 4.6 Remote Access to Event Log .............................................................. 18
Category: Tools ............................................................................................................. 18

Requirement: 5.1 User-Level (gdb) Debug Support for Threads .................................... 18
Requirement: 5.2 Kernel Dump Analysis ....................................................................... 18

Kernel Debugging ........................................................................................................... 19

Requirement: 5.3.1 Kernel Debugger ............................................................................ 19
Category: Performance .................................................................................................. 19

Soft Real-Time Support ................................................................................................. 19

Requirement: 6.1.1 Soft Real-Time Performance ......................................................... 19
Requirement: 6.1.2 Kernel Preemption ......................................................................... 20
Requirement: 6.2 RAID 0 Support ............................................................................... 20
Requirement: 6.3 Application Loading ......................................................................... 20

Scaling Analysis ............................................................................................................. 20

Requirement: 6.4.1 Concurrent Timers Scaling Behavior and Report ............................ 21

OSDL CGL Requirements, Priority 2 ............................................................................ 22
Category: Standards ....................................................................................................... 22

IPv6, IPSECv6, MIPv6 .................................................................................................. 22

Requirement: 1.7.1 IPv6 RFCs .................................................................................... 22
Requirement: 1.7.2 IPSECv6 RFCs ............................................................................ 23
Requirement: 1.7.3 MIPv6 RFCs ............................................................................... 23
Requirement: 1.8 SCTP RFCs ..................................................................................... 23
Category: Platform ......................................................................................................... 24

No Console Operation .................................................................................................. 24

Requirement: 2.7 Automatic Alternate Boot Selection .............................................. 24
Requirement: 2.8 Hyperthreading of CPUs .................................................................. 24
Category: Availability ................................................................................................... 25

Watchdog Timer ............................................................................................................ 25

Requirement: 3.2.2 Watchdog Timer Pre-Timeout Interrupt ........................................ 25

Software Live Upgrade ................................................................................................. 25

Requirement: 3.8.1 Software Live Upgrade Minimal Reboot ........................................ 25
Requirement: 3.8.2 Software Live Upgrade Version Check ................................................ 26
Requirement: 3.8.3 Software Live Upgrade Log ................................................................. 26
Requirement: 3.8.4 Software Live Upgrade Rollback .......................................................... 26

Category: Serviceability ....................................................................................................... . 26
Requirement: 4.7 Fast Boot-Up Time Enabling ................................................................. 26

Online Diagnostics .............................................................................................................. 27
Requirement: 4.9.1 Online Diagnostics Specification .......................................................... 27
Requirement: 4.9.2 Online Diagnostic Framework .............................................................. 27
Requirement: 4.9.3 Diagnostics Support to CIM Provider and CIMOM ............................... 27
Requirement: 4.10 Force Unmount ....................................................................................... 27
Requirement: 4.13 Fast Reboot ............................................................................................. 28

Category: Tools ................................................................................................................ ..... 28

Kernel Debugging ............................................................................................................... 28
Requirement: 5.3.2 Kernel Debugger Authentication .......................................................... 28

Fault Injection ..................................................................................................................... 28
Requirement: 5.4.1 Fault Injection Specification ................................................................. 29
Requirement: 5.4.2 Fault Injection Framework ................................................................. 29
Requirement: 5.4.3 Fault Injection Points ............................................................................ 29

Kernel ................................................................................................................................... 29
Requirement: 5.5.1 Kernel Flat/Graph Execution Profiling ................................................. 29
Requirement: 5.5.2 Kernel Sampling for Profiling ............................................................... 30
Requirement: 5.6 System Tools to Analyze Execution Profiles ........................................... 30
Requirement: 5.8 Multithreaded Core Dump Support for Threaded Applications ............... 30

Category: Performance .......................................................................................................... 30

Scaling Analysis .................................................................................................................. 30
Requirement: 6.4.2 Concurrent Thread Scaling Behavior and Report ................................... 31
Requirement: 6.4.3 Concurrent Process Scaling Behavior .................................................. 31

Enhancements for Scaling ................................................................................................. 31
Requirement: 6.5.1 Enhancements for High Concurrent Timer Scaling ............................. 31
Requirement: 6.5.2 Enhancements for High Concurrent Thread Scaling ............................. 32
Requirement: 6.5.3 Concurrent Process Scaling ............................................................... 32
Requirement: 6.5.4 SMP and Lock Contention Scaling ..................................................... 32
Requirement: 6.6 Kernel I/O Performance Analysis ............................................................ 32
Table of Contents

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirement: 6.8 Page Flushing</td>
<td>32</td>
</tr>
<tr>
<td>Requirement: 6.9 Process Affinity</td>
<td>33</td>
</tr>
<tr>
<td>Requirement: 6.10 Selectable Scheduler Policy Framework</td>
<td>33</td>
</tr>
</tbody>
</table>

OSDL CGL Requirements, Priority 3 ................................................................. 34

Category: Standards .......................................................................................... 34

Requirement: 1.9 Service Availability Forum Proposed Standards Compliance .... 34

Category: Availability ....................................................................................... 35

Requirement: 3.9 Application Fail-Over Enabling .............................................. 35

Requirement: 3.10 Data Checkpointing ............................................................... 35

Requirement: 3.11 Multi-Node Volume Management ......................................... 35

Requirement: 3.12 Cluster Filesystem ................................................................. 36

Category: Serviceability .................................................................................... 36

Requirement: 4.11 Linux Panic Handler Enhancement ......................................... 36

Requirement: 4.12 Kernel Dump of Live System .................................................. 36

Category: Tools .................................................................................................. 37

Requirement: 5.7 Debugging Support for Fork ..................................................... 37

Category: Performance ....................................................................................... 37

Requirement: 6.11 Managing Transient Data ....................................................... 37
## Document Control / Version Documentation

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description of Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>8/7/02</td>
<td>Initial ratification of the document.</td>
</tr>
<tr>
<td>1.1</td>
<td>10/31/02</td>
<td>Ratification of version 1.1 of this document.</td>
</tr>
</tbody>
</table>

Between this document and the 1.0 version there are a small number of new requirements identified, and a descriptive clarifications on a number of existing requirements. In addition, all requirements have been assigned to an OSDL CGL 'release version' that reflects an expected availability of the function described.

- Hyperthreading
- Fast reboot
- Kernel debugger authentication
- Core dumps for multi-threaded applications
- SMP and Lock Contention Improvements

These new requirements are complementary to existing requirements. They reflect immediate feedback on the work done to this point and help to provide a more rounded definition of OSDL CGL.
Overview

**OSDL Carrier Grade Linux**

The OSDL Carrier Grade Linux (CGL) definition outlines an enhanced form of Linux that is tailored for the needs of the emerging communications market. With the convergence of datacom, telecom, and Internet domains, the most striking characteristics of today's communications market can be summarized as follows:

- Networks are converging for multimedia communications services.
- Future multimedia-type data services will require substantially greater bandwidth, requiring new architectures to reduce the delivery cost of these services.
- Commercial off-the-shelf software components cannot be utilized because of a lack of a common standard in carrier-grade platform architectures.
- Time to market for new services needs to be shortened.
- Proprietary platforms (which are the mainstay of the current market) have higher development costs than do open-standards-based platforms.

To address these needs, OSDL established the Carrier Grade Linux Working Group. This group is an industry forum that focuses on collecting requirements and specifying an architecture to guide the development of Carrier Grade Linux, as well as promoting the development of commercial and open source components to implement the required functionality of the platform.

**Open Source Development Lab (OSDL)**

Founded in August 2000 and supported by a global consortium of industry leaders, the Open Source Development Lab (OSDL) is an independent, vendor-neutral, non-profit organization dedicated to enabling and guiding Linux and Linux-based development for enterprise and carrier-grade functionality worldwide. Headquartered outside Portland, Oregon, OSDL fulfills a critical need in the open source development community by providing access to high-end hardware for programming and testing. There are over 20 industry leaders who sponsor OSDL. More information on OSDL or the Carrier Grade Linux and Data Center Linux Working Groups is available at [http://www.osdl.org/projects/cgl](http://www.osdl.org/projects/cgl).
Overview

The Carrier Grade Linux Working Group is a forum of industry leaders working to guide the development of Carrier Grade Linux and encourage development of commercial and open standard components. This group oversees the following functions:

- Develop the roadmap for a Linux platform that supports commercial off-the-shelf-based software building blocks to create Linux-based products for carrier-grade markets.
- Achieve consistency across Linux distributions.
- Develop architecture, requirements, and validation specifications.

The CGL Working Group uses the following process for achieving these functions:

- Collect requirements from the telecommunications market.
- Specify the architectural framework and develop a roadmap.
- Align with the existing industry forums and communities like Linux Standard Base (LSB) and Service Availability™ Forum.
- Support the launch of Open Source Software projects to implement the new functionality.
- Work closely with Linux distributors to enable timely support in standard products.
- Work closely with standards and certification bodies to support development of a validation process for the OSDL definition of Carrier Grade Linux.

Selection of Requirements

The requirements selected for this document reflect primarily the needs for functionality, hardening, and conformance to standards. Initially acceptable levels of many functions, such as security and availability, while limited in comparison to the strictest existing telecommunications platforms, can be achieved through use of best-practices documentation. Therefore, there are few requirements for security included in the current list. Requirements for security will be addressed in a future version.

Terms Used in this Document

The terms “may,” “must,” and “shall” are used as described in RFC 2119, found at http://www.faqs.org/rfcs/rfc2119.html.

In this document you will also see the following terms:

- Category
- Priority
- Requirement ID
- Version Assignment
- Application Type

The following sections explain what these terms mean.
Category
Requirements for the OSDL CGL definition have been grouped into the following categories:

- Standards
- Platform
- Availability
- Serviceability (including online debugging)
- Tools
- Performance
- Security
- Scalability

Requirement IDs
The Requirement ID is the number associated with a requirement (e.g.; “2.2 Remote Boot Support”). The first digit of the Requirement ID number identifies the category to which a requirement belongs, as shown in the following list:

- 1.x.x Standards
- 2.x.x Platform
- 3.x.x Availability
- 4.x.x Serviceability
- 5.x.x Tools
- 6.x.x Performance
- 7.x.x Security
- 8.x.x Scalability

Version Assignment
Every requirement needs to be satisfied through some means. Most will result in code that will satisfy the functional request. However, some requirements included here require that written specifications and performance analyses be produced to provide a basis for later code. Where requirements here indicate functional enhancements, we view that as a feature and assign it to an expected resolution version.

The first indicator provides a guidance specifying the nature of inclusion of the feature satisfying the requirement in delivered versions of OSDL CGL:

- **Core**  A ‘core’ feature is specified by OSDL CGL to be required to be present and functional in OSDL CGL. These are features that significantly impact the reliability of OSDL CGL, and are expected to be adopted into the mainstream code base of Linux.

- **Configurable** A ‘configurable’ feature is specified by OSDL CGL to be required to be present in OSDL CGL, but may be configured to be non-functional if an installation so chooses. These features provide useful function in specialized environments, are useful tools not needed for normal function, or are code enhancements not likely to be adopted into the mainstream Linux code base.

- **NA (Not Applicable)** A feature not requiring code to be satisfied.
The second, optional, indicator is an explanation for why features are not assigned as Core to the current, 1.1, release. We believe that those features assigned as Core in this initial specification are adequately justified here and in the related Architecture specification.

**Priority**
Requirements are given any of three priorities:

- Priority 1  Required for the initial release.
- Priority 2  Desirable for the initial release; required for the second release.
- Priority 3  Item for a future release but not required immediately.
Application Type
To successfully drive requirements and the initial roadmaps for the OSDL CGL definition, three application types have been identified. Their characteristics represent the first key applications to be implemented using the OSDL CGL definition.

- Gateways (G)
- Signaling Servers (S)
- Management (M)

The following paragraphs define these application types. For more detailed information, see the Carrier Grade Linux Technical Scope white paper.

Gateways
Gateways are bridges between two different technologies or administration domains. For example, a media gateway performs the critical function of converting voice messages from a native telecommunications time-division-multiplexed (TDM) network to an Internet protocol (IP) packet-switched network. A gateway maintains a large number of connections in real time over a large number of interfaces without losing a frame or packet. They are implemented on dedicated platforms with replicated (rather than clustered) systems used for redundancy.

Signaling Servers
Signaling servers handle call control, session control, and radio recourse control. A signaling server handles the routing and maintains the status of calls over the network. It takes the requests of user agents who want to connect to other user agents and routes these requests to the appropriate signaling. Signaling servers require soft real time response capabilities of less than 80 milliseconds and may manage tens of thousands of simultaneous connections. Due to requirements for quick switching and a capacity to manage large numbers of connections, a signaling server application is context-switch and memory intensive.

Management Servers
Management servers handle traditional network management operations as well as service management and customer management. These servers provide services such as a Home Location Register and Visitor Location Register (for wireless networks) or customer information (such as personal preferences including features the customer is authorized to use). Typically, management applications are data and communication intensive. Their response time requirements are less stringent by several orders of magnitude compared to those of signaling and gateway applications.
OSDL CGL Requirements, Priority 1

Category: Standards

Requirement: 1.1 Linux Standard Base Compliance
Version Assignment: Core
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall comply with Linux Standard Base (LSB) version 1.2. OSDL CGL will use the version 1.1 test suites to test for compliance.

- It is possible and reasonable that the variety of carrier platforms and requirements will need to allow OSDL CGL to have defined and limited exceptions to the LSB.
- These considerations shall be done through the Proof of Concept and Validation subgroups working with the LSB.

POSIX Interface Compliance
OSDL CGL specifies that carrier grade Linux shall provide POSIX-compatible interfaces to support direct porting of common carrier grade applications. OSDL CGL shall follow the Austin Group specifications (aka IEEE Std 1003.1-2001 / The Open Group Base Specifications Issue 6), also known as the IEEE Std 1003.1-2001 or POSIX 2001 standard. POSIX 2001 subsumes the previous contents of IEEE Std 1003.1 editions prior to 2001 so includes all of the base contents and amendments of the previous versions.

Requirement: 1.2.1 POSIX Timer Interface Standard Compliant
Version Assignment: Core
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide Linux timer support that fully complies with the POSIX 2001 standard.
Requirement: 1.2.2 POSIX Signal Interface Standard Compliant
Version Assignment: Core
Application Type: G,S,M
Description: OSDL CGL specifies that carrier grade Linux shall provide support for signals to fully comply with the POSIX 2001 standard.

Requirement: 1.2.3 POSIX Message Queue Interface Standard Compliant
Version Assignment: Core
Application Type: G,S,M
Description: OSDL CGL specifies that carrier grade Linux shall provide Message queue support that fully complies with the POSIX 2001 standard.

Requirement: 1.2.4 POSIX Semaphore Interface Standard Compliant
Version Assignment: Core
Application Type: G,S,M
Description: OSDL CGL specifies that carrier grade Linux shall provide semaphore support that fully complies with the POSIX 2001 standard.

Requirement: 1.3 Event Logging POSIX IEEE 1003.25
Version Assignment: Core
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide event logging support that complies with draft POSIX Standard 1003.25, “System API - Services for Reliable, Available, and Serviceable Systems” (not yet adopted). Pointers to the draft can be found at http://evlog.sf.net

IPv6, IPSECv6, MIPv6
IPv6 support is considered essential for OSDL CGL for delivering next generation networks support. IPSECv6 is a core part and must also be delivered.

Mobile IPv6 (MIPv6) is also a core technology of IPv6 like IPSec for IPv6. Major industry players such as Nokia, Ericsson, Cisco and Juniper consider mobile IPv6 very important for next generation mobile networks for delivering mobile multi-media solutions to the mobile embedded devices.

Full IPv6 support is split between priority 1 and priority 2 items, due to the size of the implementation effort.

Requirement: 1.4.1 IPv6 RFCs
Version Assignment: Core
Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the following core IPv6 IETF RFCs in the initial release:

- RFC 2461: Neighbor Discovery for IP Version 6 (IPv6)
- RFC 2462: IPv6 Stateless Address Autoconfiguration
- RFC 2463: Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification
- RFC 1981: Path MTU Discovery for IP version 6
- RFC 2373: IP Version 6 Addressing Architecture
- RFC 2526: Reserved IPv6 Subnet Anycast Addresses
- RFC 2374: An IPv6 Aggregatable Global Unicast Address Format
- RFC 2375: IPv6 Multicast Address Assignments
- RFC 2473: Generic Packet Tunneling in IPv6 Specification
- RFC 2464: Transmission of IPv6 Packets over Ethernet Networks
- RFC 2472: IP Version 6 over PPP
- RFC 2292: Advanced Sockets API for IPv6
- RFC 2553: Basic Socket interface extensions for IPv6
- RFC 2893: Transition mechanisms for IPv6 hosts and routers
- RFC 2784: Generic Routing encapsulation
- RFC 3041: Privacy Extensions for Stateless Address Autoconfiguration in IPv6
- Default Address selection for IPv6

Requirement: 1.4.2 IPSECv6 RFCs

Version Assignment: Configurable

Full functionality requires cryptographic support. Due to differing regulations support will differ based on locality and distribution.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the following core IPSECv6 IETF RFCs in the initial release:

- RFC 2401: Security Architecture for the Internet Protocol (Supporting transport mode only)
- RFC 2402: IP Authentication Header (AH)
- RFC 2406: IP Encapsulating Security Payload (ESP)
- RFC 2403: The Use of HMAC-MD5-96 within ESP and AH
- RFC 2404: The Use of HMAC-SHA-1-96 within ESP and AH
- RFC 2405: The ESP DES-CBC Cipher Algorithm With Explicit IV
- RFC 2367: PF_KEY Key Management API, Version 2
Requirement: **1.4.3 MIPv6 RFCs**
Version Assignment: Configurable

- The IETF RFCs are in final editing stages before ratification. The code is implemented to these RFCs but minor changes may still take place.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the following core MIPv6 IETF RFCs in the initial release:

- Mobility Support in IPv6
- User level:
  - RFC 2428: FTP Extensions for IPv6 and NATs
  - RFC 1886: DNS Extensions to support IP version 6
  - RFC 2874: DNS Extensions to Support IPv6 Address Aggregation and Renumbering

Requirement: **1.5 SNMP Support**
Version Assignment: Core

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide Simple Network Management Protocol (SNMP) support. All three versions of SNMP agent (SNMPv1, SNMPv2, and SNMPv3) shall be supported.

Requirement: **1.6 POSIX Threads Standards Compliance**
Version Assignment: Core

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for threads that are full compliant with the POSIX 2001 standard definitions for signal handling, mutexes, message queues, and process primitives related to threads.

**Category: Platform**

**Hotswap Support**

OSDL CGL shall support standard based hardware that provides device and controller hot swap operations. Where specifications for software operations of the platform exist, OSDL CGL should support these specifications.

Version Assignment: Configurable

- There are a number of different implementations for supporting hotswap of devices. The Linux community has not converged on a specific implementation, and that is quite certain for the ‘Hot Identity’ requirement. While it is vital that CGL support these
functions, a converged method of doing so remains a work in progress.

Requirement: 2.1.1 Hot Insert
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support that continues to provide service when a device, capable of hot insertion, has been added to the system.

Requirement: 2.1.2 Hot Remove
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support to gracefully remove software access to a device in the system so it can be physically removed.

Requirement: 2.1.3 Hot Device Identity
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support that ensures a single device will retain its identity across hot insert and removals and across system boots.

Requirement: 2.1.3.1 System Device Enumeration Specification
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide a “CGL System Device Enumeration Framework Specification.” This specification shall define a framework that will contain the status of all available system devices. This specification shall define a set of common interfaces/APIs so that an application can enumerate, via this framework, and acquire all available system devices and their status.

Requirement: 2.1.3.2 System Device Enumeration Framework
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide a framework for an application to enumerate all available system devices and their status.

Requirement: 2.2 Remote Boot Support
Version Assignment: Core
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support for remote booting across common LAN and WAN communication media to support diskless systems.

Requirement: **2.3 Boot Cycle Detection**
Version Assignment: Configurable

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t yet consider this to be a core function.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for detecting a frequent reboot cycle due to recurring failures and will go offline if this occurs.

Requirement: **2.4 Loading Proprietary Modules**
Version Assignment: Core

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall not prevent the loading of proprietary modules.

Requirement: **2.5 Diskless Systems**
Version Assignment: Core

Application Type: S

Description: OSDL CGL specifies that carrier grade Linux shall provide support for Linux on diskless systems of various types, including but not limited to:

- No disk installed, boot remotely and load all applications remotely.
- Disk installed, can be used for booting and loading applications but not operations (no swap or temp space).
- These options with and without a ramdisk.

**No Console Operation**

OSDL CGL must provide the capability to boot and administer a system that has no mouse, keyboard nor VGA monitor physically attached.

Requirement: **2.6.1 Serial Console Connection**
Version Assignment: Core

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for a
connection to a system console via a serial port on the system where a serial port exists. All output that would appear on a local console must appear on the remote console.

**Category:** Availability

**Hardened Driver Support**
OSDL CGL will provide hardened device drivers.

Requirement: 3.1.1 Device Driver Hardening Specification
Version Assignment: NA
Application Type: G, S, M
Description: OSDL CGL shall introduce a specification proposing Hardening of Device Drivers. The purpose of this specification is to provide common interfaces and APIs and coding practices for existing and new device drivers. This shall include three levels of device driver hardening requirements listed in Req ID 3.1.2 below.

Requirement: 3.1.2 Sample Hardened Device Driver
Version Assignment: NA
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide a sample hardened device driver to demonstrate the methods being described in the Device Driver Hardening Specification, at least one sample device driver will be provided.

**Watchdog Timer**
OSDL CGL shall utilize a watchdog timer and have one configured. Existing implementations should be used to work with various offered hardware cards and drivers to enable this support.

Requirement: 3.2.1 Watchdog Timer Interface Requirements
Version Assignment: Core
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide the ability to use the Linux /dev/watchdog interface to reset the hardware watchdog timer. This timeout value shall be a configurable item. A configurable action can be performed when a timeout occurs.
Requirement: **3.3 Application Heartbeat Monitor**

Version Assignment: Configurable

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t yet consider this to be a core function.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide an application heartbeat service that allows applications to register to be monitored via specified APIs, as currently being proposed through the Open Cluster Framework (OCF) and allows a registered application to be monitored by the service. This mechanism shall use periodic synchronized events (heartbeats) between an application and the monitor. If a registered application fails to provide a heartbeat, the monitor shall report the event via the system event log.

Application heartbeat time shall be granular to at least 10 msec resolution (at least 100 heartbeats/second).

The application heartbeat service shall be available to any process or sub-process (thread) entity on the system. A process or thread may register for multiple heartbeats. Each heartbeat request can have its own parameters to specify heartbeat granularity.

This requirement does not specify a maximum number of concurrent heartbeat registrants that the monitor can handle. However, if the monitor cannot handle any additional registrants, the request will return a specific error so the registrant will know this.

The application heartbeat service requires a registrant to specify a unique identifier upon registration. If the given identifier is not unique, an error will be returned. The registrant may use its PID or choose some other system-unique value. The latter necessary if a single process wishes to register for multiple heartbeats.

**Ethernet Multiple NIC Bonding**

OSDL CGL shall support bonding of multiple ethernet NICs within a single node. Specific capabilities can be configured separately.

Version Assignment: Configurable

- Direct support for aggregated and bonded Ethernet drivers requires use of specific switches and cabling which may not be desirable for all installations and so use of these feature is left up to the installation.

Requirement: **3.4.1 Ethernet Link Aggregation**

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall support multiple
ethernet cards to be bonded for bandwidth aggregation.

- Support can be configured out if desired.

Requirement: **3.4.2 Ethernet Link Failover**
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall support automatic failover of an IP address from one ethernet NIC to another within a single node using the ethernet bonding support.

- Support can be configured out if desired.

**RAID Support**
OSDL CGL shall provide RAID support.

Version Assignment: Configurable

- RAID support is up to the installation to choose.

NOTE: The requirement number for RAID 0 support appears in the section called Performance.

Requirement: **3.5.1 RAID 1 Support**
Application Type: M
Description: OSDL CGL specifies that carrier grade Linux shall provide RAID 1 (Mirroring) support so that the OS maintains duplicate sets of all data on separate disk drives. OSDL CGL RAID 1 support shall allow booting off of selected mirror disk drive even if the other drive is failed. OSDL CGL RAID 1 implementation shall provide a user-controllable parameter to throttle the syncing operation.

- Support can be configured out if desired.

Requirement: **3.6 Resilient Filesystem Support**
Version Assignment: Configurable

- Specific use of filesystem modes is up to the installation.

Application Type: M
Description: OSDL CGL specifies that carrier grade Linux shall provide support for the installation of a filesystem that is resilient against system failures in terms of recovering rapidly upon reboot without requiring a full, traditional fsck. This is normally achieved using logging or journaling techniques.

- Support can be configured out if desired.
Requirement: **3.7 Disk and Volume Management**
Version Assignment: Configurable

- Specific use of such management capabilities is up to the installation.

Application Type: M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the installation of a subsystem that supports hard disks to be managed without incurring downtime:

- Physical disks can be grouped into volumes and the volume definitions can be modified without downtime.
- Filesystems that are defined within volumes can be enlarged without requiring unmounting.
- Support can be configured out if desired.

**Category: Serviceability**

**Resource Monitor**

OSDL CGL shall provide the capability to monitor a machine's resources and their health.

Version Assignment: Configurable

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t yet consider this to be a core function.

Requirement: **4.1.1 Resource Monitor Specification**
Application Type: G, S, M

Description: OSDL CGL shall introduce a CGL Resource Monitor Framework Specification. This specification shall define a framework so that a set of resource subsystems/devices can be “plug-in” to the framework dynamically and perform the monitoring function. This specification shall define a set of common interfaces/APIs so an application can enumerate all available subsystems/devices and their status.

Requirement: **4.1.2 Resource Monitor Framework**
Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide a resource monitor framework that complies with the CGL Resource Monitor Framework Specification. As per this specification, this framework will allow a new subsystem to be created and plugged in to perform monitoring functions. This framework shall provide threshold settings and notification so that an event will be sent to the event log when a particular
resource/threshold is reached.

Requirement: **4.1.3 Resource Monitor Subsystems**  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux shall provide the following resource monitor subsystems: Disk Space Usage, Kernel Resources (CPU utilization, Memory Usage), Network Usage, and /proc status.

Requirement: **4.1.4 Resource Monitor Data Persistence**  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux resource monitor framework shall provide the ability to capture and persist time stamped data. A set of APIs must be provided for an application to retrieve persisted historical data for analysis and display purposes.

Requirement: **4.1.5 Resource Monitor Performance**  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux resource monitor framework shall meet stringent CPU- and memory-usage limits. Note that based on the number of resources being monitored, cpu and memory usage can climb dramatically. A guideline should be that the framework overhead be limited to 1% of the CPU over and above that needed to perform the specified level of monitoring.

**Kernel Dumps**  
OSDL CGL shall provide enhancements for producing and storing kernel dumps.  
Version Assignment: Core

Requirement: **4.2.1 Kernel Dump Targets**  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux shall provide support for producing and storing kernel dumps.  
- It should be possible for kernel dumps to be stored:  
  - To disk.  
  - Across a network.  
  - To memory.  
- Regardless of specific dump target, dumps must be preserved across the next system boot.  
- Information pertaining to all CPUs and all kernel threads must be pre-
served.

- The kernel dump provides the ability to capture enough of the kernel state to permit post-crash analysis of the conditions that lead to a server problem, which may be hardware, software, resources, or a configuration issue.

Requirement: **4.2.2 Kernel Summary Dump**
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support for producing summary dumps containing only specified data structures. These list of data structures shall be able to be specified by the system administrator.

Requirement: **4.3 Kernel Message Structuring**
Version Assignment: Configurable
- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t yet consider this to be a core function.

Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support that allows the structuring of kernel messages using an event log format to provide more information to better identify the problem and its severity, and to allow client applications registered for the fault event to take policy-based corrective action. The format can be tested against the event log specification (evlog.sf.net).

Requirement: **4.4 Dynamic Debug/Probe Insertion**
Version Assignment: Configurable
- This requirement specifies a tool that will normally be used in specialized circumstances, and also reflects a methodology not currently standardized in Linux.

Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support for the ability to dynamically insert software instrumentation into a running system in the kernel or applications.

- The instrumentation must be insertable to any part of the kernel.
- The instrumentation should allow control to be passed to a user-provided module.
- The instrumentation should not require interactive direction, i.e., no user sitting at the kernel debugger.
- The user-provided modules should have access to data the kernel would normally be expected to have access to, e.g., hardware registers, kernel
memory, and such.

Requirement: **4.5 Platform Signal Handler**  
Version Assignment: Core  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux shall provide, where possible, an infrastructure to allow interrupts generated by “hardware errors” to be logged using the event logging mechanism. A default handler shall be provided.

Requirement: **4.6 Remote Access to Event Log**  
Version Assignment: Core  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux shall provide support for a remote access capability that allows a centralized system to access the Linux OS event log information of a remote system.

**Category: Tools**

Requirement: **5.1 User-Level (gdb) Debug Support for Threads**  
Version Assignment: Core  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux shall provide support to fully enable debugging of multi-threaded programs via gdb. This support should allow any actions available for debugging a single-threaded (non-threaded) process be extended to be available for every thread in a multi-threaded process.  
OSDL CGL shall provide specific additional debugging capabilities that are unique to multi-threaded applications:  
- Automatic notification of a new thread.  
- List of threads and the ability to switch among them.  
- Apply specific gdb commands to a list of threads.  
- Hide/unhide a given thread from the thread list

Requirement: **5.2 Kernel Dump Analysis**  
Version Assignment: Core  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux shall provide support for tools to
enable enhanced analysis of kernel dumps. These enhancements should include, but should not be seen as limited to, the following capabilities, accessing kernel structures, virtual to physical address translation, and module access.

Kernel Debugging
OSDL CGL shall support the ability to utilize kernel debuggers.

Requirement: **5.3.1 Kernel Debugger**
Version Assignment: Configurable

- This requirement specifies a tool that is useful in specialized circumstances. In addition, there is no commonly accepted methodology in providing a kernel debugger in Linux so OSDL CGL expects that vendors and installations will choose implementations most suitable to their environment.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for a kernel debugger that provides the ability to break into a running kernel at boot time or during system operation, set breakpoints, examine memory contents in several formats, change memory contents or machine registers, and other debug functionality. The debugger shall be SMP-aware. It can support remote kernel debug sessions on headless systems using a tty connection.

- Support can be configured out if desired.

**Category: Performance**

Soft Real-Time Support
OSDL CGL shall provide the capability of configuring the scheduler to provide soft real-time support so that the real-time scheduling latency of a given task will not exceed a target offered by the vendor.

Requirement: **6.1.1 Soft Real-Time Performance**
Version Assignment: Core
Application Type: G, S

Description: OSDL CGL specifies that carrier grade Linux shall provide the capability of configuring the scheduler to provide soft real time support so that the real time scheduling latency of a given task will not exceed a target offered by the vendor.

- OSDL CGL needs to support scheduling latencies of 10 ms.
• The Validation suite can likewise specify a methodology to allow a vendor to specify a latency value and provide verification that the specified value is satisfied.

Requirement: 6.1.2 Kernel Preemption
Version Assignment: Configurable
Application Type: G, S
Description: OSDL CGL specifies that carrier grade Linux shall provide support for a preemptible kernel.
• Support can be configured out if desired.

Requirement: 6.2 RAID 0 Support
Version Assignment: Configurable
• RAID support is up to the installation to choose.
Application Type: M
Description: OSDL CGL specifies that carrier grade Linux shall provide RAID 0 (Striping) support that stripes data across multiple disks without any redundant information to enhance performance in either a request-rate-intensive or transfer-rate-intensive environment.
• Support can be configured out if desired.

Requirement: 6.3 Application Loading
Version Assignment: Configurable
• The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t yet consider this to be a core function.
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support for fully loading and pinning an application’s pages before beginning execution so that none of its pages are demand paged during execution.
• This should be configurable for each application and also as a global setting.
• Security needs to be addressed as mis- or over-use of this capability can have severe impacts on the system.

Scaling Analysis
OSDL CGL needs to support systems that scale in a number of ways not directly related to the number of CPUs. Rather, OSDL CGL systems will be deployed where they need to efficiently support large numbers of objects of various types, e.g., timers, threads, processes, and network
connections.

Gateway and Signaling applications tend to deal in large numbers of messages. This tends to translate to large numbers of threads, timers or processes, depending on the application structure.

Analysis reports shall be produced that analyzes the system behavior as the number of concurrent elements increases to some multiple of 1000 consistent with the system resources available such as the amount of memory. This analysis report shall include the performance of following system behaviors. Degradation of the elapsed time of a process execution as the total number of elements increases:

- File I/O access time degradation as the total number of elements being measured increases.
- Daemon process response time degradation as the total number of elements being measured increases. It shall including the performance of the network daemon.
- Real time scheduling performance degradation as the total numbers of concurrent elements being measured increases.
- Numeric computation time degradation as the total number of elements being measured increases.

Requirement: 6.4.1 Concurrent Timers Scaling Behavior and Report

Version assignment: NA
Application Type: G, S
Description: OSDL CGL specifies that carrier grade Linux shall provide a characterization of the system behavior when supporting applications that require scaling of total count of system timers into the 1000s. This report will include a description of the environment used to determine the characteristics.
OSDL CGL Requirements, Priority 2

Category: Standards

IPv6, IPSECv6, MIPv6

These portions of IPv6 are listed as priority 2 due to length of time to get all RFCs supported. By supporting the initial priority 1 items, OSDL CGL will allow initial roll-out of IPv6 networks.

Requirement: 1.7.1 IPv6 RFCs

Version Assignment: Core

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the following core IPv6 IETF RFCs in the second release:

- RFC 2553: Basic Socket Interface Extensions for IPv6
- RFC 2529: Transmission of IPv6 over IPv4 Domains without explicit tunnels
- RFC 3056: Connection of IPv6 domains over IPv4 clouds
- RFC 2080: RIPng for IPv6
- RFC 2465: Management Information Base for IP Version 6 - Textual Conventions and General Group
- RFC 2466: Management Information Base for IPv6 - ICMPv6 group
- RFC 2452: IP Version 6 Management Information Base for the Transmission Control Protocol
- RFC 2454: IP Version 6 Management Information Base for the User datagram Protocol
- RFC 2491: IPv6 over Non-Broadcast Multiple Access (NBMA) networks
- RFC 2492: IPv6 over ATM Networks
- RFC 2710: Multicast Listener Discovery (MLD) for IPv6
- RFC 2711: IPv6 router alert option
- RFC 2675: IPv6 jumbograms
- RFC 3019: IPv6 MIB for MLD
- Advanced Sockets API for IPv6
- Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)
OSDL CGL Requirements Definition Version 1.1
OSDL CGL Requirements, Priority 2

Requirement: 1.7.2 IPSECv6 RFCs
Version Assignment: Configurable

- See earlier description in previous IPSECv6 section.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the following core IPSECv6 IETF RFCs in the second release:

- RFC 2401: Security Architecture for the Internet Protocol (Supporting tunnelling mode also)
- RFC 2409: The Internet Key Exchange (IKE)
- RFC 2451: The ESP CBC-Mode Cipher Algorithms
- RFC 2764: A Framework for IP Based Virtual Private Networks
- RFC 3173: IP Payload Compression Protocol (IPComp)

Requirement: 1.7.3 MIPv6 RFCs
Version Assignment: Configurable

- See earlier description in previous MIPv6 section.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the following core MIPv6 IETF RFCs in the second release:

- Mobility support in IPv6
- Fast hand overs for MIPv6
- Vertical handoff for MIPv6 between media
- User level:
  - Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

Requirement: 1.8 SCTP RFCs
Version Assignment: Core

- Accepted into the base for future Linux kernels (2.5/2.6).

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the following core Stream Control Transmission Protocol (SCTP) IETF RFCs will be supported:

- RFC 2960
- RFC 1112
No Console Operation
OSDL CGL must provide the capability to boot and administer a system that has no mouse, keyboard nor VGA monitor physically attached.

Requirement: 2.6.2 Network Console Operation

Version Assignment: Core

- There is no serious project to provide this currently on Linux so the availability of a feature will not exist in the 1.1 timeframe.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for a connection to a system console via a network port on the system where a remote console can be connected across the network. All output that would appear on a local console must appear on the remote console.

Requirement: 2.7 Automatic Alternate Boot Selection

Version Assignment: Core

- There is no serious project to provide this currently on Linux so the availability of a feature will not exist in the 1.1 timeframe

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for detecting boot failures and automatically selecting an alternate boot image or device. The alternate device may be either local or remote.

Requirement: 2.8 Hyperthreading of CPUs

Version Assignment: Configurable

- This is an installation-specific choice to use this feature based on their environment.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for hyperthreading, the ability of a CPU to appear to be multiple CPUs to the operating system and applications where the hardware supports such capability. Performance benefits of such support are not stated and OSDL CGL cannot provide guarantees of hyperthreading as depending on the platform they may be very sensitive to the behavior and profiles of running applications.

- Support can be configured out if desired.
Category: Availability

Watchdog Timer
OSDL CGL shall utilize a watchdog timer if one is configured. Existing implementations should be used to work with various offered hardware cards and drivers to enable this support.

Requirement: 3.2.2 Watchdog Timer Pre-Timeout Interrupt
Version Assignment: Core

- There is no serious project to provide this currently on Linux so the availability of a feature will not exist in the 1.1 timeframe

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for a watchdog timer pre-timeout interrupt. Where the hardware supports such a capability an interrupt handler routine will be called before the real timeout occurs.

Software Live Upgrade
OSDL CGL shall provide a software live-update feature. It shall include functions that allow kernel and application software to be upgraded while minimizing downtime of the system. Specific downtime targets will need to conform to product-specific requirements for a system based on the application for which it's being used.

- Usual industry practice refers to 30 and 60 second downtime periods for downtime, regardless of cause. If downtime is required for software upgrade, it would need to fit within this target.
- OSDL CGL does not specify a single value here to support a wide range of platforms and applications for which CGL will be targeted.

Version Assignment: Configurable

- There are a multiplicity of methods to perform installation and upgrade of large numbers of remote systems. None of them currently provide the level of support desired for OSDL CGL. As these features are provided, they will continue to be provided through distributions and through specific function products. OSDL CGL will not specify a single such installer so long as whatever is provided meets all of the functional pieces in this requirement.

Requirement: 3.8.1 Software Live Upgrade Minimal Reboot
Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide mechanisms that allow software to be upgraded while minimizing rebooting of the system.
No reboot shall be required for upgrade of kernel components such as device drivers or kernel modules.

- The ability to replace the kernel itself without rebooting should be defined eventually.
- Application (user-space-level software) upgrades shall require no reboot on any platform.

**Requirement:** 3.8.2 Software Live Upgrade Version Check

**Application Type:** G, S, M

**Description:** OSDL CGL specifies that carrier grade Linux shall provide software upgrade capabilities that must include provisions for version compatibility and dependency checking. The upgrade process shall allow the coexistence of new and old executables, shared libraries, configuration files, and data.

**Requirement:** 3.8.3 Software Live Upgrade Log

**Application Type:** G, S, M

**Description:** OSDL CGL specifies that carrier grade Linux shall provide software upgrade capabilities that must include logging of dates, times, changes, and identity of the user.

**Requirement:** 3.8.4 Software Live Upgrade Rollback

**Application Type:** G, S, M

**Description:** OSDL CGL specifies that carrier grade Linux shall provide software upgrade mechanisms that allow rollback to a previous version of the software without having to reinstall the previous version.

**Category:** Serviceability

**Requirement:** 4.7 Fast Boot-Up Time Enabling

**Version Assignment:** Core

- There is no serious project to provide this currently on Linux so the availability of a feature will not exist in the 1.1 timeframe

**Application Type:** G, S, M

**Description:** OSDL CGL specifies that carrier grade Linux shall provide mechanisms to allow rapid booting of the system. On non-redundant systems, the total elapsed time of the reboot process once the firmware has finished and given control to the kernel
on a stand alone machine (not boot over the network) shall not exceed 30 seconds.

**Online Diagnostics**

OSDL CGL shall provide an online diagnostics capability that supports a set of system health checks can be run to determine the system health including OS health and hardware health. The diagnostic check capability must be provided so that it can be run remotely or locally.

Version Assignment: Configurable

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don't yet consider this to be a core function

Requirement: **4.9.1 Online Diagnostics Specification**

Application Type: G, S, M

Description: OSDL CGL shall introduce a CGL Common Diagnostics Specification. This specification shall define set common interfaces/APIs and a framework so that a set of devices can register to the framework dynamically and perform the diagnostic.

Requirement: **4.9.2 Online Diagnostic Framework**

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide an online diagnostics framework that complies with the CGL Common Diagnostics Specification. As per the specification, this framework will allow an new application to register to it dynamically and to perform diagnostic function to any available devices. It will also allow a device to be added to the framework at run time and provide the diagnostic service.

Requirement: **4.9.3 Diagnostics Support to CIM Provider and CIMOM**

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for CIM and CIMOM to access the support provided by the online diagnostics framework.

Requirement: **4.10 Force Unmount**

Version Assignment: Core

- There is no serious project to provide this currently on Linux so the availability of a feature will not exist in the 1.1 timeframe

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for forced
unmounting of a filesystem.

Requirement: **4.13 Fast Reboot**
Version Assignment: Configurable

- This support needs to be applicable in a platform-agnostic way and work with or without a local storage to be considered core.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the ability to reboot a system within 1 minute. This is specific to systems with limited numbers of I/O devices.

**Category: Tools**

**Kernel Debugging**
OSDL CGL shall support the ability to utilize kernel debuggers.

Requirement: **5.3.2 Kernel Debugger Authentication**
Version Assignment: Configurable

- This requirement specifies a tool that is useful in specialized circumstances. In addition, there is no commonly accepted methodology in providing a kernel debugger in Linux so OSDL CGL expects that vendors and installations will choose implementations most suitable to their environment.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support for a kernel debugger that requires basic authentication to ensure that only authorized users with access to a system can use the debugging features. This level of security is analogous to that assumed by an authorized user logging into a system.

**Fault Injection**
OSDL CGL shall provide a Fault Injection capability that can be used as the test harness for injecting faults to device drivers and their resource areas for testing purpose.

Version Assignment: Configurable

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don't yet consider this to be a core function
Requirement: **5.4.1 Fault Injection Specification**
Application Type:
Description: OSDL CGL shall introduce a CGL Fault Injection Specification. This specification shall define a set of common interfaces/APIs and a framework so that a set of devices can interface with the framework to perform the injection of fault.

Requirement: **5.4.2 Fault Injection Framework**
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide a fault injection framework that complies with the CGL Fault Injection Specification. As per this specification, this framework will allow a new application to register to it and inject fault into an available device. It shall also allow a device to be added to the framework to provide the injection point for that device.

Requirement: **5.4.3 Fault Injection Points**
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide a Fault Injection Framework that provides, but is not limited to, the following injection points:
- Linux Memory Mapped I/O.
- Interrupts and DMA.
- Linux PCI API.

**Kernel**
OSDL CGL shall support profiling of the running kernel to identify bottlenecks and other interesting information.

Version Assignment: Configurable

- There are a number of methods extant for providing these features, but none are accepted as mainstream Linux methods. OSDL CGL is not targeting this as a 1.1 timeframe effort. In any case, once these exist they will be used in specialized circumstances and will not be considered Core items.

Requirement: **5.5.1 Kernel Flat/Graph Execution Profiling**
Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support for profiling of the running kernel using a prof or gprof style recording of trace information during system execution.
Requirement: **5.5.2 Kernel Sampling for Profiling**  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux shall provide support for profiling of the running kernel by providing profiling based on interrupt sampling, for example:.  
- Take an interrupt.  
- Record execution point.  
- Repeat on rapid (micro- or milli-second multiples).  
- Analyze to build up profile of system execution history.

Requirement: **5.6 System Tools to Analyze Execution Profiles**  
Version Assignment: Configurable  
- This meshes with the ability to take traces and measures of system execution, and likewise there is no accepted as mainstream Linux method. OSDL CGL is not targeting this as a 1.1 timeframe effort. In any case, once these exist they will be used in specialized circumstances and will not be considered Core items.

Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux shall provide support for interfaces to allow utilities to collect data from trace kernel and application level activity.

Requirement: **5.8 Multithreaded Core Dump Support for Threaded Applications**  
Version Assignment: Core  
Application Type: G, S, M  
Description: OSDL CGL specifies that carrier grade Linux shall provide support for correctly storing core dumps of multi-threaded user-space applications. These core dumps can then be analyzed using gdb enhanced for also supporting multi-threaded user-space applications..

**Category: Performance**

**Scaling Analysis**  
OSDL CGL needs to support systems that scale in a number of ways not directly related to the number of CPUs. Rather, OSDL CGL systems will be deployed where they need to efficiently support large numbers of objects of various types, e.g., timers, threads, processes, and network connections.

Gateway and Signaling applications tend to deal in large numbers of messages. This tends to translate to large numbers of threads, timers or processes, depending on the application structure.

Analysis reports shall be produced that analyzes the system behavior as the number of concurrent elements increases to some multiple of 1000 consistent with the system resources available such
as the amount of memory. This analysis report shall include the performance of following system behaviors. Degradation of the elapsed time of a process execution as the total number of elements increases:

- File I/O access time degradation as the total number of elements being measured increases.
- Daemon process response time degradation as the total number of elements being measured increases. It shall including the performance of the network daemon.
- Real time scheduling performance degradation as the total numbers of concurrent elements being measured increases.
- Numeric computation time degradation as the total number of elements being measured increases.

Version Assignment: NA

Requirement: 6.4.2 Concurrent Thread Scaling Behavior and Report
Application Type: G, S
Description: OSDL CGL specifies that carrier grade Linux shall provide a characterization of the system behavior when supporting applications as the number of concurrent threads increases to counts into multiples of 1000. This report will include a description of the environment used to determine the characteristics.

Requirement: 6.4.3 Concurrent Process Scaling Behavior
Application Type: G, S
Description: OSDL CGL specifies that carrier grade Linux shall provide a characterization of the system behavior when supporting applications as the number of concurrent processes increases to concurrent process counts into multiples of 1000. This report will include a description of the environment used to determine the characteristics.

Enhancements for Scaling
Enhancements needed in the system identified by the benchmarks to ensure that the system scales to support appropriate numbers of various elements.

Version Assignment: Core

Requirement: 6.5.1 Enhancements for High Concurrent Timer Scaling
Application Type: G, S
Description: OSDL CGL specifies that carrier grade Linux shall provide support needed in the system to address deficiencies identified by the benchmarks to ensure that large numbers of concurrent timers can be supported.
Requirement: **6.5.2 Enhancements for High Concurrent Thread Scaling**

Application Type: G, S

Description: OSDL CGL specifies that carrier grade Linux shall provide support needed in the system to address any deficiencies identified in the reference work for multiple thousands of concurrent threads.

Requirement: **6.5.3 Concurrent Process Scaling**

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support needed in the system to address any deficiencies identified in the reference work for multiple thousands of concurrent processes.

Requirement: **6.5.4 SMP and Lock Contention Scaling**

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support that reduces uses of the Big Kernel Lock and improves efficiency of locking activities where reasonable in the kernel.

Requirement: **6.6 Kernel I/O Performance Analysis**

Version Assignment: NA

Application Type: G, S, M

Description: Currently there is an unidentified hardware limitation that restricts the throughput of block devices (e.g.; disk I/O) to 100 megabits per second (100 Mb/s) regardless of the number or types of I/O controllers. OSDL CGL shall identify the source of this restriction and remove it. Performance should be scalable based on the number of controllers/disks/adapters.

Requirement: **6.8 Page Flushing**

Version Assignment: Configurable

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists, and this requirement is for specialized installations, we don’t consider this to be a core function.

Application Type: G, S, M

Description: OSDL CGL specifies that carrier grade Linux shall provide support that allows either applications or the operator to specify control of page-flushing operations to control system impact.

- This capability should be configurable on a per-process or per-application basis and also as a global setting.
- The Proof of Concept and Architecture subgroups need to consider what
forms an API to support this should take.
- Security needs to be addressed as mis- or over-use of this capability can have severe impacts on the system.

Requirement: **6.9 Process Affinity**
Version Assignment: Core
- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t yet consider this to be a core function.

Application Type: G, S, M
Description: OSDL CGL specifies that carrier grade Linux shall provide support for forcing processes to execute on a specified CPU to avoid interfering with interrupt handling.
- This should be configurable for each application.
- The new O(1) scheduler addresses this, using the processor affinity mask, but provides no utilities to set or control it.
- Security needs to be addressed as mis- or over-use of this capability can have severe impacts on the system.
- Spec needed here to provide utilities to exploit the function.
- API to be defined in spec.

Requirement: **6.10 Selectable Scheduler Policy Framework**
Version Assignment: Configurable
- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t yet consider this to be a core function.

Application Type: G, S
Description: OSDL CGL specifies that carrier grade Linux shall provide a framework to allow multiple scheduler policies to be made available in the OSDL CGL kernel and to allow any one of these policies to be selected and compiled into the kernel.
OSDL CGL Requirements, Priority 3

**Category:** Standards

Requirement: [1.9 Service Availability Forum Proposed Standards Compliance](#)

Version Assignment: Configurable

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t yet consider this to be a core function.

Application Type: NA

Description: OSDL CGL shall fully comply with the Service Availability Forum proposed standards where there is specific overlap.

- Architecture group and full technical board needs to determine if we're to work with non-open industry standards such as this. The Service Availability Forum has not specified licensing and compliance conditions and so it is not known if an open source based effort can be applied using them.

- These areas will most likely be limited to a few platform services such as system-provided checkpoint if such is specified by OSDL CGL.

- The Service Availability Forum proposed standards will be available as two separate proposed standards:
  - A Hardware Platform Interface to monitor and control carrier system platform chassis and hardware resources. Available in the third quarter of 2002.
  - An Applications Interface to monitor and control highly available carrier applications. A specific availability date is not yet announced, but a tentative date of first quarter of 2003 is announced.

- OSDL CGL will therefore align itself as reasonably as possible given existing information available from the Service Availability Forum.

- A key assumption is that licensing of the Service Availability Forum specifications is compatible with OSDL CGL reference and with implementation capabilities to support OSDL CGL.
**Category: Availability**

**Requirement: 3.9 Application Fail-Over Enabling**

Version Assignment: Core

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t yet consider this to be a core function

Application Type: NA

Description: OSDL CGL specifies that carrier grade Linux shall provide the capability that enables an application to resume execution even after software failure, without clustering. It shall be compliant with available open source and/or industry standards applicable to this function.

- API to be in spec.

**Requirement: 3.10 Data Checkpointing**

Version Assignment: Configurable

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t consider this to be a core function

Application Type: NA

Description: OSDL CGL specifies that carrier grade Linux shall provide a data checkpoint mechanism to applications to store vital data locally and to sync the data with a backup instance. It shall be compliant with available open source and/or industry standards applicable to this function.

**Requirement: 3.11 Multi-Node Volume Management**

Version Assignment: Configurable

- Specific use of such management capabilities is up to the installation

Application Type: NA

Description: OSDL CGL specifies that carrier grade Linux shall provide support that allows volumes to be managed in a clustered environment. Volumes in such an environment are usually on physical disks accessible to multiple nodes.

- OSDL CGL shall provide support for the remote node(s) to be informed of the new volume definitions (see Disk and Volume Management).

Clustering middleware or administrative actions may be required to issue instructions on secondary (and tertiary, etc.) nodes to inform them of the volume redefinitions.
Requirement: **3.12 Cluster Filesystem**
Version Assignment: Configurable

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t consider this to be a core function

Application Type: NA
Description: OSDL CGL specifies that carrier grade Linux shall provide support that allows installation of a cluster filesystem.

Category: Serviceability

Requirement: **4.11 Linux Panic Handler Enhancement**
Version Assignment: Core

- The goal of OSDL CGL is to utilize a standard API such that applications can be guaranteed to work across distributions. As no such standard yet exists we don’t consider this to be a core function

Application Type: NA
Description: OSDL CGL specifies that carrier grade Linux shall support enriched capabilities on system panic. Currently the default system panic behavior is to print a short message to the console and halt the system. OSDL CGL shall provide a set of configurable functions including log panic event to system event log as well as the options to reboot, power down, or power cycle when panic event occurs.

Requirement: **4.12 Kernel Dump of Live System**
Version Assignment: Core

- This is a follow on to earlier kernel dump requirements. As no work on this feature is currently available, it will not be considered a core part of 1.1.

Application Type: NA
Description: OSDL CGL specifies that carrier grade Linux shall provide support that allows a kernel dump image to be taken of a live system while minimizing disturbing the system execution.
**Category:** Tools

Requirement: 5.7 Debugging Support for Fork
Version Assignment: Core

- There is not work extant on this feature and it remains a long-term goal.

Application Type: NA

Description: OSDL CGL specifies that carrier grade Linux shall provide support for the ability to follow the fork() system call and to debug into a child process.

---

**Category:** Performance

Requirement: 6.11 Managing Transient Data
Version Assignment: Configurable

- A number of approaches are currently being looked at on Linux to satisfy this, but they are immature and it is unclear the future direction so we maintain this as a long-term goal.

Application Type: NA

Description: OSDL CGL specifies that carrier grade Linux shall provide support for a self-resizing filesystem for transient data that can be limited to a maximum size.