CGL Requirements

Laila Daniel

February 12, 2003

Abstract

The Carrier Grade Linux Working Group has been chartered by the Open Source Development Lab to enhance the Linux operating system to achieve an open source platform that is highly available, secure, scalable and easily maintained. This is in view of an operating system which can be tailor-made to the needs of the emerging communication scenario which comprises the Internet, Telecom and Datacom.

1 Introduction

The Open Source Development Lab Carrier Grade Linux (ODSL CGL) definition outlines an enhanced form of Linux that is tailored for the needs of the emerging communications scenario which is a convergence of Datacom, Telecom, and Internet domains. Today's communication networks are characterised by the following features.

- 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.

This paper focuses on the CGL requirements definition. Section 2 deals with an overview of cgl requirements. Section 3 gives a description of each of these requirements.

2 Overview of the CGL Requirements

The CGL Working Group has identified three main categories of application areas into which the majority of applications implemented in the first stage on CGL platforms to fall. These application areas are gateways, signaling servers and management servers. The CGL Working Group focuses initially on Linux platform requirements to support applications in these areas.

2.1 Application Areas

- Gateways

  Gateways provide bridging services between different technologies or administrative domains. Gateway applications are characterized by supporting a large number of connections in real-time over a large number of interfaces, withthe requirement of not losing any frames or packets. Gateways are implemented on dedicated platforms with replicated (rather than clustered) systems used for redundancy.

- Signaling Servers

  Signalling server applications which include SS7 products, handle control services for calls, such as routing, session control and status. Signaling server applications are characterized by sub-millisecond real-time requirements and large numbers of simultaneous connections (10,000 or more). 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 can handle traditional service and billing operations, as well as network management. Management 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.

2.2 CGL Requirements

The requirements for Carrier Grade Linux give emphasis to functionality, hardening and conformance to standards.
Requirements for the OSDL CGL definition have been grouped into the following categories:

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

The requirements are also given three priorities

- Priority 1 Required for 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.

3 CGL Requirements

This section provides a detailed description of CGL requirements. The requirements for different catefories at different priority levels are described.

3.1 Standards

The standards and interfaces defined will fall into two categories:

- Mandatory- Required for CGL compliance.
- Optional - Specified by the standard but not required for compliance.

In the standards category, the CGL working group's first priority is given to mandatory components which includes Linux Standards Base (LSB) compliance, some IPv6 (including IPSECv6 and MIPv6) compliance, SNMP support, and POSIX interface compliance in the areas of timers, signals, message queues, semaphores, event logging and threads.

Optional components include Policy Framework The IETF Policy framework Common Open Policy Service (COPS) together with QoS and Internet Protocol Security Policy Information Base (IPSec PIB) will be offered as an alternative to SNMP.

The application type includes gateway, signalling server and management applications.

3.1.1 Linux Standard Base compliance

ODSL 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.

3.1.2 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 specifies that carrier grade Linux shall provide support for timer interface,signals, message queue and semaphore to fully comply with the POSIX 2001 standard. 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).

3.1.3 IPv6, IPSECv6, MIPv6

Since IPV6 is the IP protocol for the next generation networks, the ODSL CGL should support IPV6. The CGL should support IPSECv6. mobile IPv6 is very important for next generation mobile networks for delivering mobile multi-media solutions to the mobile embedded devices so it is required that CGL should support MIPv6.

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

OSDL CGL Requirements Definition Version 1.1 gives a list of IETF RFCs that deals with IPv6, IPSECv6 and MIPv6 that CGL shall provide support.

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.

ODSL CGL 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.

3.1.4 Priority 2 & 3

Priority 2 adds support for the stream control transmission protocol (SCTP) (which is a mandatory component) and further IPv6, IPSECv6 and MIPv6 RFCs compliance. SCTP is a next generation multi-link end-to-end transport layer that provides reliable transmission of messages over potentially unreliable connectionless packet services such as IP.

In future releases (priority 3), the standards category will also include requirements for service availability middleware. These specs are expected to follow the efforts of the Service Availability Forum (www.saforum.org) to definestandard interfaces, but may branch off to other requirements depending on the openness and availability of the Service Availability Forum's specification.

3.2 Platform Requirements

The platform includes the underlying hardware, firmware, and minimum required set of devices (and their interfaces) to support a Carrier Grade Linux operating system.

CGL will include several hardware interfaces such as hardened device drivers, interfaces to hardware configuration and management, and co-processor interfaces. The device drivers will be hardened to withstand at least some aspects of malfunctioning hardware. The ability to use redundant hardware components will be further enhanced. The hardware configuration and management interfaces will provide unified interfaces to manage and to configure certain classes of hardware units. The co-processor interfaces will provide standardized ways of communicating with external processors like digital signal processors, network processors and cryptographic chips.

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

The main goal of defining the platform is to include support for hot-swap, remote boot, diskless operation, and console-less operation. The carrier-grade Linux definition of hot swap also includes the concept of hot insert (adding cards to the system not originally in place at boot time), hot remove (not replacing cards that are removed), and maintaining device identities across hot swaps and system boots.

The application types include gateway, signalling server and management applications.

3.2.1 Priority 2 & 3

Once these standards are set, the working group will turn its attention to the second priority, which adds network console and automatic alternate boot selection. Network console refers to the ability to have the system console for a CGL system appear on a remote system. Alternate boot selection refers to the ability to have the system self select an alternate kernel to boot if it detects that there are too many reboots within a specified period of time.

There are currently no priority 3 platform requirements for carrier-grade Linux under development by CGLWG.

3.3 Availability

High availability is a must in the case of telecommunication networks. In telecommunication systems, the service availability requirement for the entire system is typically expressed as five or six nines (99.999%). So the CGLWG is working on a set of specifications for building high availability into the Linux OS.

The CGL availability enhancements will establish a more robust and highly available Linux operating system. The availability of the CGL operating system layer will be measured in terms of Defects-Per-Million (DPM). DPM is a statistically valid record of how many actions per million were not successfully completed for any reason. It should be noted that DPM measures the robustness of CGL, not the availability of the system. How a certain DPM measure contributes to system availability depends on the actual system configuration (including both hardware and software).

The application type includes gateway, signalling server and management applications.

The first priority here is driver hardening, watchdog timer support, applicationheartbeat monitor support, Ethernet bonding and aggregation, RAID 1 support, journaling file system support, and disk and volume management support.

3.3.1 Hardened Driver Support

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.

3.3.2 Watchdog Timer Support

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. The 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.

3.3.3 Application Heartbeat Support

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.

3.3.4 Ethernet bonding and aggregation

CGL specifies that carrier grade Linux shall support multiple ethernet cards to be bonded for bandwidth aggregation. Support can be configured out if desired. 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.

3.3.5 RAID Support

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.

3.3.6 Resilient File system

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 such as JFS, XFS, ext3and ReiserFS.

3.3.7 Disk and volume management

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.

3.3.8 Priority 2 & 3

The second priority is to add watchdog timer pre-interrupt (for platforms that support this feature) and software live upgrade features to the Carrier Grade Linux specification.

The third priority here will be the addition of application fail-over support, data checkpointing services, multi-node volume management, and support for cluster file systems.

3.4 Serviceability

To provide this serviceability, CGL will provide APIs to define and monitor various resources and services. The monitoring facilities and APIs will be extended to cover the entire software stack. CGL will provide a mechanism to generate and log structured events. These events will have a well-defined structure that can be easily interpreted by external software including middleware. Hooks will be provided to middleware vendors to collect these events and provide advanced fault management capabilities. CGL will provide guidelines to driver developers to provide online diagnostic capabilities. On the serviceability front, the CGLWG is currently trying to define requirements for resource monitoring, kernel crash dump and analysis features, structured kernel messages, dynamic kernel probing, hardware error logging, and remote access to the event log.

3.4.1 Resource Monitoring

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.

The 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.CGL will also provide resource monitor subsystems such as Disk Space Usage, Kernel Resources (CPU utilization, Memory Usage), Network Usage, and /proc status.

3.4.2 Kernel Dumps

OSDL CGL specifies that carrier grade Linux shall provide support for producing and storing kernel dumps.It should be possible for the kernel dumps to be stored to a disk, across a network or to a memory. 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).

3.4.3 Dynamic debug

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. Dynamic kernel probing is a technology that allows designers to examine data structures and establish break points without the requirement that debugging code be compiled into the kernel itself. Kernel probing allows developers to observe system behavior easily and quickly at the debugger level in a running system, without having to compile the modules under observation with debug capabilities beforehand, or indeed even having to decide in advance which modules may need to be probed. Dynamic kernel probing also allows system maintainers to examine a deployed system without needing to bring it down.

3.4.4 Hardware error logging

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.

3.4.5 Remote Access to Event Log

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.

3.4.6 Priority 2 & 3

Priority 2 will add fast boot requirements, consistent system device enumeration, online diagnostics, and forced unmount of file systems. Consistent device enumeration refers to a capability by which applications can itemize all devices in the system and register these devices so they receive hot-swap events (insertions, removals). All devices in the system shall maintain consistent identification across removals, insertions, and reboots. Note that most of this capability is required in the priority 1 platform requirements for hot swap.
The additional priority 2 requirement here is the API by which applications can access the consistent device identities. Through the priority 2 development work, the carrier-grade Linux spec will allow applications to register with an online diagnostic framework and perform diagnostics on registered devices. Devices will be able to add themselves to the diagnostic framework at any time.Forced unmount functions will also be provided. Forced unmount refers to the capability to force an unmount of a mounted file system, even if there are open files or process working directories on it. Pending requests on the file system will be terminated and return an error indication and the file system will be left in a clean state (if possible).

Priority 3 in the serviceability category adds panic handler enhancements to allow more flexible responses to a panic, and kernel dump generation for a live system. System panics currently halt Linux-based systems. The carrier-grade Linux spec will add the capability to have the panic add an event to a (persistent) error log and then provide the option to reboot, power off, or power cycle the system when a panic occurs Kernel dump generation of a live system refers to taking a snapshot memory and resources dump of a live system without unduly perturbing the execution of the system.

3.5 Tools

Priority 1 in the tools category will focus on the development of debugger support for threaded programs, kernel debugger support, and a kernel crash dump analysis tool.

3.5.1 Debug support for threads

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 such as automatic notification of a new thread, listing of threads and the ability to switch among them and applying specific gdb commands to a list of threads and Hiding/unhiding a given thread from the thread list

3.5.2 Kernel Debugger

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.

3.5.3 Kernel dump analysis

The tools in this category 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.

3.5.4 Priority 2 & 3

Priority 2 in the tools category will add support for fault injection testing, kernel tracing and kernel profiling. Fault injection testing is a method by which software faults can be injected into the kernel to test the ability of the kernel to recover from errors, without the need to create elaborate tests to create the same fault conditions through normal kernel operations, allowing for easier and more complete testing of seldom used paths in the kernel.

Kernel tracing provides a set of trace points in the kernel for most significant kernel operations, and an interface for enabling and filtering them, as well as a graphical analysis tool. Tracing is a very powerful diagnostic technique for analyzing system activity, especially for debugging problems caused by interactions between programs and the kernel.

Profiling refers to the ability to build a picture of how the kernel spends its time, either by using the tracing facility (above), or by sampling kernel execution via interrupts and recording the kernel activities that are interrupted.

Priority 3 will add debugging support for following a fork system call into the child process.

3.6 Performance

In the performance category, priority 1 performance features required include soft real-time (less than 10-ms worst case latencies), pre-emptible kernel, RAID 0 (striping), application pre-loading, and a scaling analysis and report to identify scaling bottlenecks.

3.6.1 Soft Real-time performance

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.It should support latencies of the order of 10 ms.

3.6.2 Kernel Preemption

CGL specifies that carrier grade Linux shall provide support for a pre-emptible kernel.

3.6.3 RAID support

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.

3.6.4 Application loading

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.

An important aspect of performance in Carrier Grade Linux is ensuring that the reliability, availability and serviceability enhancements do not unduly reduce performance. This will require benchmark comparisons with traditional Linux platforms.

3.6.5 Priority 2 & 3

Priority 2 in this category will add remediation for problems found in the scaling analysis performed during the Priority 1 phase, virtual memory system improvements (page pinning and flushing APIs), SMP CPU process affinity and support for multiple scheduler policies.

Priority 3, on the other hand, will bring support for a self-resizing file system for transient data.

3.7 Security

Security mechanisms must be able to authenticate users and applications, authorize user access to resources, maintain usage logs for off-line billing and manage prepaid services, and audit all legal and illegal accesses performed.

The scope of the CGL Working Group is to ensure that the kernel will operate in a secure manner and that security hooks are provided for applications and middleware components. Since a secure system becomes insecure if used incorrectly, the CGL Working Group will also provide guidelines for the correct use and installation of CGL.

The current version (V1.1) of the carrier-grade Linux specification does not have any requirements for security, but this will be addressed in a future version of the document. Security is recognized by the CGLWG as a key component of high availability.

3.8 Scalability

OSDL CGL needs to support systems that scale in a number of ways not directly related to the number of CPUs.Carrier Grade Linux must be scalable in a variety of dimensions including:

- Number of processes,threads, and timers
- Number of network connections
- Network bandwidth
- Number of I/O devices
- Number of other I/O related entities (such as disks, partitions, files)
- I/O bandwidth
- Memory size and Disk size
- Number of processors

When hardware (such as the number of I/O devices or processors) is scaled up, the performance enhancement should be close to linear. When logical resource usage (such as number of processes or network bandwidth usage) is scaled up, increases in processor utilization should be linear or better. Scalability also means removing artificial limitations on logical resources.

Improvements to the Linux scheduler will be included to enhance scalability in the number of processes, as well as provide support for realtime processes. The scheduler improvements, combined with improvements to the preemptibility of the kernel, will guarantee a maximum latency between timeslices given to a particular process and provide high-resolution timers to applications.

Symmetric multiprocessor (SMP) scaling will support up to at least 8- processor machines. One important aspect of SMP scaling is support for finer-granularity mutual exclusion objects (mutexes) for inter-processor synchronization.

4 Conclusion

The Carrier Grade Linux Working Group has been chartered by the Open Source Development Lab to enhance the Linux operating system to achieve an open source platform that is highly available, secure, scalable and easily maintained. The carrier-grade enhancements to the Linux operating system will be contributed to the standard Linux kernel. A CGL platform will contain the standard Linux kernel, kernel patches, kernel modules and modified libraries. The CGL Working Group has identified eight areas in which enhancements may be required, including standards, platform, high availability enhancements, security enhancements, serviceability enhancements, performance and scalability enhancements, and tools.

5 References

- OSDL CGL Requirements Definition Version 1.1
  http://www.osdl.org/projects/cgl/osdl_cgl_requirements_definition_1_1.pdf
- Carrier Grade Linux Technical scope white paper
  http://www.osdl.org/projects/cgl/cgl_technical_scope_whitepaper.pdf
- OSDL - http://www.osdl.org/
- Linux Standard Base - http://www.linuxbase.org/
- Open Source Software - http://www.opensource.org/
- Internet Engineering Task Force - http://www.ietf.org/
- " Optimize Linux to Meet Carrier Grade Requirements"
  John Mehaffey, CommsDesign Magazine, December 2002
- "Carrier Grade Linux: What You Need to Know"
  John Mehaffey, CommsDesign Magazine, August 2002