Carrier Grade Linux: A to Z!
The full story, or almost ... 

Ibrahim Haddad, OSDL

Ottawa Linux Symposium
July 2005
Agenda

- Linux – Disruptive Technology
- Telecommunications Market
- Carrier Grade Linux
- Deployment
- Conclusion
Why companies adopt Linux?
Why Adopt Linux and Open Source?

• GPL – Full access to source code
• Platform independence
  – Lower hardware costs and extended system life-cycle
• Lower Maintenance and Support
  – Minimal system administration support required after installation
• Pricing
  – Lower costs vs. legacy / proprietary OS
  – Entire supply chain looking to lower TCO
• Multiple providers
• Higher system performance, reliability and security
• Source code quality, Innovation rate, peer reviews and testing resources
• Established eco-system: Hardware and Software
Linux is a Disruptive Technology. Really.

Linux can’t do [fill in blank] is an invitation to Open Source developers
Disruptive Technologies

• What is disruptive technology?
  – Technology with significant deficiencies that is targeted to a niche segment but providing significant cost benefits
  – Usually targeted to early adopters that are the ‘risk takers’ with a much slower adoption into the mainstream

• Disruptive technologies in telecommunication
  – Voice Over IP
  – Linux Operating System

• Adoption of Disruptive technology always starts with non mission-critical applications
Disruptive Technologies

- Reinvent Business Models
- Decentralize Vertical Markets
- Clarify Market Vision
  - Linux/Open Source help identify real value-added
  - Those who do not provide clear value are automatically out of the market

The Innovator’s Dilemma
Clayton M. Christensen
Yesterday and Today

Yesterday …
Communications and data service networks built on proprietary platforms to meet specific requirements for
• Availability
• Reliability
• Performance
• Service response time

Today …
Communications service providers must drive down cost while maintaining carrier-class platforms with:
– High availability
– Scalability
– Security
– Reliability
– Predictable performance
– Maintenance & upgrade
Why the attention to Linux from Telecom?

Service Provider

Proprietary/Closed Technology

FOR ONLY A MILLION DOLLARS, YOU CAN UPGRADE TO OUR NEWEST SOFTWARE VERSION.

OR YOU CAN SLOWLY DECOMPOSE IN THE MIASMA OF OUR PLANNED OBSOLESCENCE.

WE CAN'T AFFORD TO UPGRADE NOW. SAY GOODBYE TO THE DIGITS THREE AND NINE.
Telephony Business in (R)evolution

- Voice telephony is a premium service.
- Few players
- Profits

- Voice telephony is a commodity
- More players in the field
- Shrinking Business – fierce competition and pricing models
- Much less profits
- Losses in some areas
- More threats (VoIP, Broadband telephony)
Telecom Platform (R)evolution

**PAST**

- Proprietary/Legacy systems
- No clear separation of “building blocks”
- Proprietary technologies & interfaces
- Expensive to develop, maintain & scale
- Single Provider

**FUTURE**

- Open standard-based systems
- Interchangeable “building blocks”
- Standardized interfaces
- COTS SW and HW
- Multiple Providers
Why this shift, now?

• Service providers and carriers are in a position today where they must move away from specialized proprietary architectures, and towards COTS approaches and building practices driven by two key motivations:

  – **Faster time to market**: They need to be able to deliver new services based on common standardized platforms. They are in a constant race to deliver faster to the market.
    • Building with proprietary and specialized technologies that are offered by a very limited number of providers is one obstacle from this perspective.

  – **Reduce costs**: They need to reduce the design and operation costs by using COTS hardware and software components
    • COTS components are offered by multiple providers
    • COTS components are compliant or registered towards standards of industry agreed specifications.
Telecom Platform (R)evolution

Application Enabling Platforms

Network Element

- Proprietary Applications and 3rd Party Applications
- Proprietary HA Middleware and Protocols
- CARRIER GRADE LINUX (OSDL)
- Standard-Based HW Management
- Standards-Based HW Platform (AdvancedTCA/CompaPCI)

2005

Network Element

- Proprietary Applications and 3rd Party Applications
- Proprietary Protocols
- Standard-Based Middleware (SA Forum)
- CARRIER GRADE LINUX (OSDL)
- Standard-Based HW Management
- Standards-Based HW Platform (AdvancedTCA/CompaPCI)

2006

Network Element

- Applications
- Standardized Protocols
- Standard-Based Middleware (SA Forum)
- CARRIER GRADE LINUX (OSDL)
- Standard-Based HW Management
- Standards-Based HW Platform (AdvancedTCA/CompaPCI)

2007

Rapid Adoption Promotes Industry Innovation – Resources are focused on new applications
The Results …

One subrack
Multiple Network Elements

Network Element
Interchangeable standard based building blocks with standardized interfaces

- Proprietary and 3rd Party Applications
- Application Interface
- HA Middleware
- Hardware Interface
- Carrier Grade Linux
- Standard HA Hardware
Challenges

• **Reduce Costs**
  – Using COTS building blocks (SW and HW)

• **Seamless integration** of carrier-grade components
  – Integrated solution must be validated for carrier-grade availability

• **Maintain carrier-class characteristics**
  – Delivering **increasing levels of availability** and dependability
  – **Growth of packet traffic** putting pressure on communication networks
  – Platforms in all-IP environment must maintain their carrier-class characteristics

• **Decrease time to market**

• **Fast delivery of new services** by shorten new service dev time
  – Unifying platforms

• **Increase profits!**
Don’t Shoot the Messenger!

• Not the doom of NEPS and Carriers
  – Not “turning the industry on its head” either

• Instead a “sea change”
  – Transition to COTS architectures and practices
  – **Embrace of Linux and Open Source**
  – Re-alignment at multiple levels
    • **Before 1999/2000:** incompatible platforms, protocols, etc., high barriers to entry, circuit switched, ….
    • **Today:** Telecom resurgence with COTS, Linux & OSS
  – New players, new businesses

• **Key to success is understanding difference between core value add and marginal business**
Carrier Grade Linux
## CGL WG

An industry forum to support & accelerate the development of Linux functionality for telecommunication applications.
CGL Goal

• **Goal:** Making Linux Better for the Telecom Industry

• **Vision:** Next generation and multimedia communication services are delivered using Linux-based open standard Carrier Grade Platforms.

A Linux kernel with Carrier Grade characteristics is an essential building block component for telecom platforms and architectures.
CGL History

OSDL Founded, Labs established in
- Beaverton, OR, USA
- Yokohama, Japan

01/2002: CGL Working Group Established

Linus Torvalds & Andrew Morton join OSDL

02/2005: CGL 1.1 is released

10/2002: CGL 1.1 distributions are available

05/2005: First meeting with Carriers

02/2005: CGL 3.0 Technology Release

10/2003: CGL 2.0 is released

CGL 2.0 distributions are available

06/2005: CGL 3.1 is released
Carrier Grade Linux – Scope

- Software Development Tools
  - Linux OS with Carrier Grade Enhancements
  - High Availability Hardware Platforms
  - Middleware Components
    - Java
    - CORBA
    - Databases
  - High Availability Components
  - HA Platform Interfaces
  - HA Application Interfaces
- Applications
  - Call (Packet) Processing
  - Service Control
  - SIP
  - SS7

Scope of the Carrier Grade Linux Working Group
CGL Context – IT Networks

Carrier Grade Linux

Access
- Edge Devices
  - Proxy/caching
  - VPN
  - RAS
  - Firewall
  - Wireless edge
  - All-in-one
  - VoIP gateway
  - GPRS gateway

Corporate Network
- Infrastructure Server
  - Directory
  - Security
  - Load balancing
  - File/Print
  - Web
  - Mail
  - NAS
  - Soft switch
  - Telco features

Business Application Server
- E R P
- S C M
- C R M
- M R O
- S F A
- H R
- I V R

Enterprise Data Store
- Data/Content Server
  - Databases
  - Multimedia
  - Documents
  - New Objects
  - HPC

Client Devices
CGL Context – Next Generation Networks

Service Layer
- Application
- Service Capability Application Servers

Control Layer
- MSC SGSN
- GGSN
- HSS
- HLR
- GMSC/Transit
- CSCF
- MGCF

Connectivity Layer
- Media Gateway
- Backbone Switches/Router
- Media Gateway
- PSTN/ISDN
- Internet Intranets
- User data
- Control

CGL-based Server Node
Carrier Grade Linux – Lifecycle Process
CGL 3.1 Requirements Areas

- Standards-compliance / APIs
- Hardware Support
- Availability
- Clustering
- Security
- Performance
- Serviceability
Current Activities

• Focusing on development
  – Concentrate on promoting quality implementations of the 3.1 spec in 2005-2006
  – Resources on the CGL tech board are being realigned accordingly
  – New SIGs established

• Extend gap analysis and use case descriptions

• Refine CGL specification

• Document carrier inhibitors in MRD

• Work on specific focus areas (next slide)
Beyond CGL 3.1 -- Specs

First half of 2005

Current Specification Structure
Separate Document per Focus

New Focus Areas

- Smaller, more focused releases.
- Break out deep technical items.
- Separate new work from updates.

Updates to previous Focus Areas

Secure
Availability
Hardware
Serviceability
Performance
Clustering
Standards/APIs

July 2005 onward

Real-time
Device Drivers
Testing
Mobile IPv6
... Focus Z
Special Interest Group – Focusing on Dev Efforts

• Existing SIGs
  – Storage SIG
    • Participants: OSDL, IBM, Novell, EMC, RH, Unisys, Bull, SUN, DreamWorks, HP, Cicso, Intel
  – Hotplug:
    • Participants: OSDL, HP, Virtual Iron, Fujitsu WindRiver, Intel, Bull, Unisys, EMC, SGI
  – Security
    • Participants: OSDL, Novell, IBM, Sun, Ericsson, MontaVista, HP, SE Linux, NCSC
  – Clusters
    • Participants: OSDL, RH, IBM, HP, Novell, MontaVista, SUN, Intel, Oracle, ORNL

• New SIGs:
  – Robust mutexes
  – Live patching S
  – System management
End User Forums

- Carriers Forum
- Linux User Advisory Councils (LUACs)
- Independent software vendors (ISV forum)
- Independent hardware vendors (Open Source Driver Forum)
CGL and Src Code
Sequence of events

1. CGL specs calls out specific function needs in the Linux kernel – address usage models
2. Gap analysis is conducted to identify the current open source implementation gap against a given requirement
3. Member companies works with community and open source projects to fill the gap
4. As result, some of these features are mainlined
Mapping between requirements and code

- There are many projects (with src code) that member companies are working on that are mapped to CGL's needs
  - Some projects carry the CGL flag
  - Others don’t
- Sometime, it works better that way!
- At the end, as long as there is src code available in open source that fulfill CGL's need, we have achieved a milestone
CGL Member Participation in Development

OpenHPI
IBM, Intel, MontaVista

Openais
MV, OSDL, Intel, NTT

SAF Conformance
Intel, Community

DigSig
Ericsson

uSDE
Intel, MontaVista

AEM
Ericsson

TIPC
Ericsson, Intel, WindRiver

Mutexes
Intel, OSDL

OpenIPMI
Intel, MontaVista, Cyclades

Live Update
NTT

Andrew Morton

PRODUCTION KERNEL
MAINTAINER

Linus Torvalds

SUBSYSTEM
MAINTAINERS

SOURCE
CODE

SOURCE
CODE

CONTRIBUTORS

SOURCE
CODE

DEVELOPMENT KERNEL
MAINTAINER
Integration with Linux Kernel

- Kernel integration takes time
- Some enhancements already integrated with 2.6 kernel
  - Others will follow
- All enhancements available from SourceForge or project web sites
Forking?

- We shouldn't fear the fork
  - It is a normal part of the Linux development process
- There are short term forks that occur all of the time with the intention of being a proof of the technology before it is integrated into the mainstream.
- What is important?
  - The **purpose of the fork**
    - With CGL, most of the times forks occur with the intention of proving the concept to the wider audience
    - Yes, sometimes that fork can last for over a year
  - But the **goal is to work it back**
    - It is a temporary thing not an idea to run off in a different direction
A permanent patch?

• Different than a fork
  – A way to get technology to track the mainstream that might be in opposition to the needs of the larger user population

• Linux tools support this environment and makes it easy for somebody to add in their little bit of required differences (based upon business needs) without incurring the expense of doing the whole stack

• Is this such a bad thing? Maybe not?
  – It allows for the special case were something that greatly benefits one industry segment but would hurt others can still track the normal development process and benefit from the larger amount of work being done.
Let's look at CGL 2.0 Requirement Document

- Total: 71 requirements
  - 18 in stock kernel
  - 6 in stock kernel + glibc
  - 7 in glibc
  - 8 in stock kernel + RPM (utilities)
  - 26 as RPMs
  - 5 as kernel patches
  - 1 unimplemented
What about the CGL 3.1 Requirements

• The focus now is on development efforts!
Carrier Grade Linux Deployments
(Linux domination in telecom networks)

NEC Case Study

BT Case Study
NEC New Platform Architecture

- Service and Application Layer
- Middleware compliant with Service Availability Forum Open Interfaces
- Carrier Grade Linux (OSDL)
- AdvancedTCA™

Advanced New Platform for Mobile Operators Employing OSDL Carrier Grade Linux and PICMG Forum Advanced-TCA(TM)
Application to Mobile Infrastructure
Current Status

• 100+ nodes in several carrier networks
• 10M+ subscribers in total
• Running for 6 months in actual commercial services
Future Planned Deployments

New Platform Architecture With Carrier Grade Linux
BT Global – Pilot Project in Belgium

- Provide voice communication services
Firebird System Architecture

Firebird IN Evolution

21C Application Defined Network

In-country Applications

Global Applications

Application Build GUI

Service Logic Execution Environment (with open interfaces) – SAF, JAIN

Carrier Grade Linux -- OSDL

i86 Commercial Off-the-shelf Platform or i86 ATCA

Northbound Open Interfaces
SIP, SML, PARLAY -- IETF

Soft Switch

MGW

MGW

Open Interfaces
IP, TDM, ATM - ITU, IETF
Results

- CGL is suitable
- It has been a good experience
- Problems were fixed quickly
- Lower cost solution than proprietary solutions

- *Investigating roll out of the platform on all EU network*
Conclusions
CGL Initiative: Jan 2002 – Present

• Increasing number of OSDL member companies involved with CGL

• 3 major releases of the requirement definition

• 8 Linux distributors are shipping CGL distributions, 5 of them have registered against CGL 2.0

• Over 20 providers for CGL based platforms

Service Providers/Carriers are deploying CGL based platforms on their networks providing voice and data communications services to their subscribers
Thank You for the Linux Kernel community

Helping us extend and further the adoption of Linux in Telecom!