OpenHuawei.org

LinuxCon 2012
August 30, 2012, San Diego, California

Sven-Thorsten Dietrich (sven@openhuawei.org)
Huawei Technologies

www.openhuawei.org

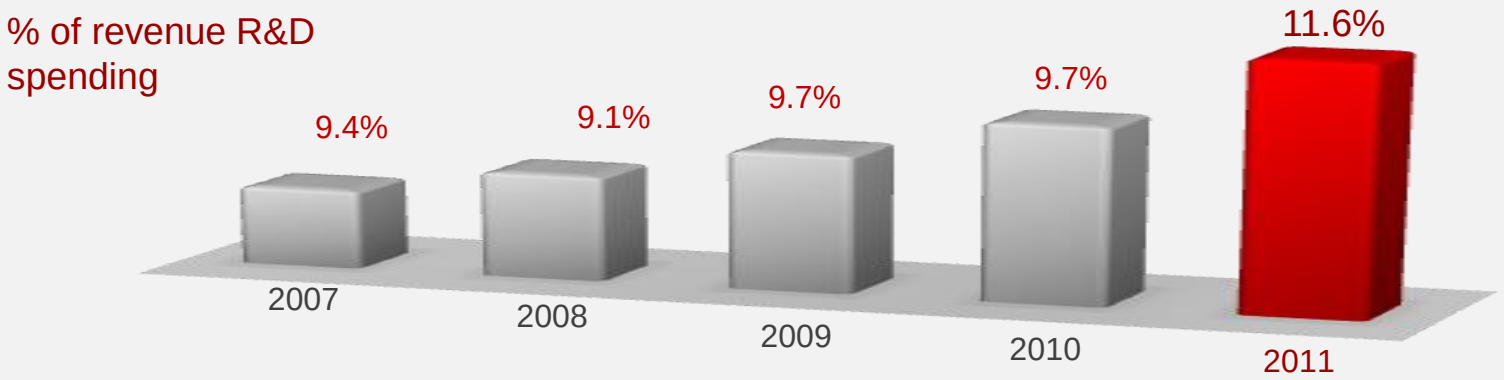
- Company Overview
- Linux / OSS Adoption: Challenges
- Linux Adoption at Huawei
- Linux Challenges at Huawei
- OpenHuawei.org: Fast-track to upstream
- OpenHuawei.org: Projects
- Conclusions

Huawei Company Overview

- Fast-growing Technology Company in Shenzhen, CN
- 62K(+) Engineering organization head count
- Multi-pronged Open Source strategy
- Using Canonical, Suse, RedHat, WindRiver distros
- Ambitious expansion from telecom to broader IT market
- Linux : strategic technology going forward
- Seeking to build sophisticated internal Linux expertise

Huawei R&D Innovation Investment

- USD 3.76 billion in 2011
- 62,000 R&D employees
- 23 R&D centers, 6 in Europe and 5 in NA
- 34 Joint Innovation Centers

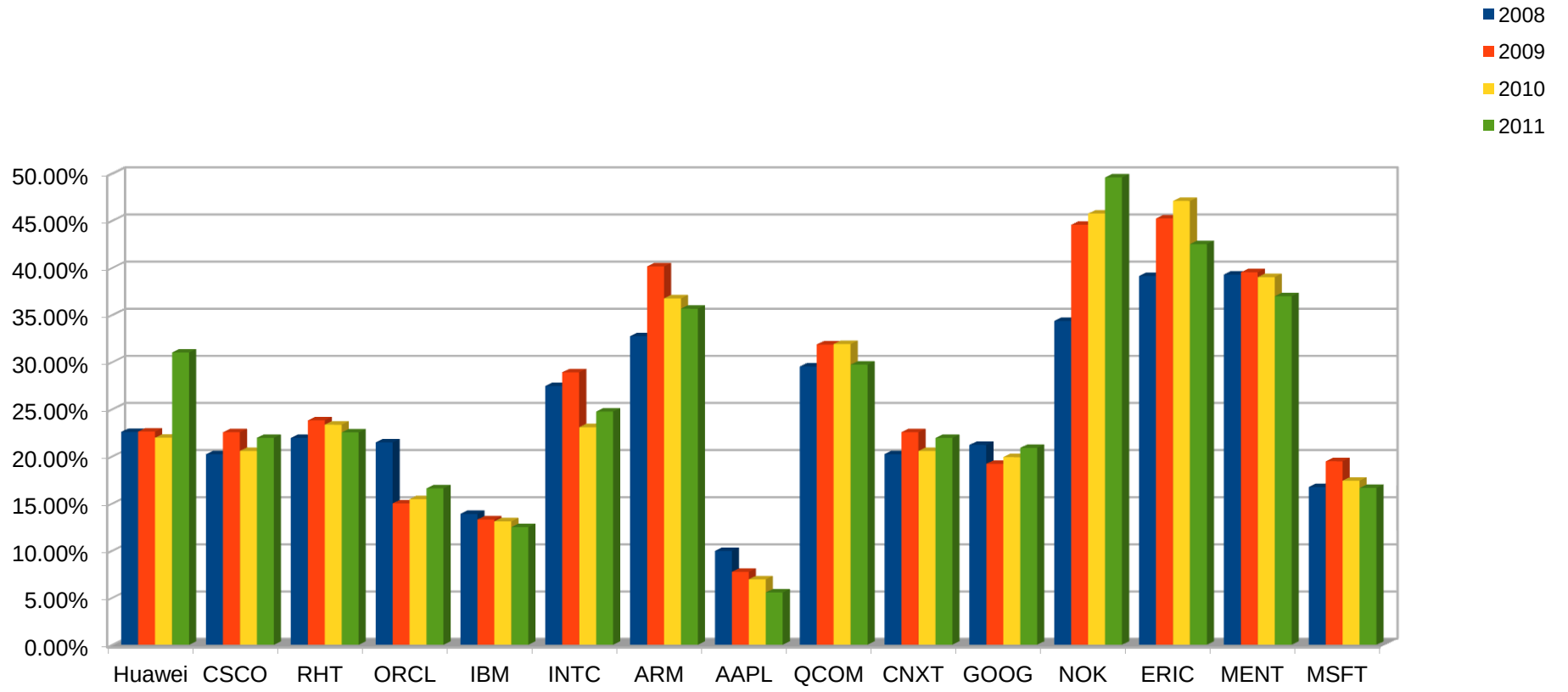


- Company Overview
- Linux / OSS Adoption: Challenges
- Linux Adoption at Huawei
- Linux Challenges at Huawei
- OpenHuawei.org: Fast-track to upstream
- OpenHuawei.org: Projects
- Conclusions

OSS Adoption: Bottom Line Driven

- **Market pressure / Customer demand**
- **Product complexity and short-term R&D effort**
  - Feature integration, maturity & availability
  - Arch enablement / support
  - Driver support & driver stability
  - Customizability & Tools
  - Emerging technologies
  - Quality in bugs/loc
- **Economic factors: CAPEX, TTM, TCO**
  - Up-front and short-term cost to market
  - Time to market
  - Royalty contributions

R&D cost / $ of Gross Profit (2011)



Source: NASDAQ, NYSE, Huawei Annual Report

OSS Adoption: Stages Evolve 2–5 years

- Evaluation / Experimentation
- Encapsulation, Abstraction, Unified Driver Code
  - Linux Sandwich
- Customization, Differentiation and Divergence
- Freeze and Back-port
- Re-Architect and Integrate
- Upstream Awareness, Alignment, Contribution
- Upstream Collaboration and Innovation

"Adopting OSS best-practices entails gradual re-education and re-organization of the classic software organization"

OSS Adoption: Pitfalls and Challenges

- Myths, FUD, Fear and Loathing
- Buy or Build → Buy and Build → Build
- Management and Strategy Execution
  - Intellectual property vs. competitive advantage
- Product Development
  - Adapting to OSS Culture, Values and Processes
  - R&D engineering, efficiency and cost

"Leveraging Open Source is more than using Linux in a product"

Linux Adoption: Sourcing - Buy / Build?

- Trusting developers working for competitors
- Supply chain conflict of interest
- Vendor technology road-map
- Feature integration and availability
- Maintenance: Farm out or in-house
- Support quality vs. response time
- Expertise: acquire or grow organically
- Cost projections

Linux Adoption: Myths & Misconceptions

- **Long-term costs of OSS / Linux**
  - Often poorly understood
- **GPL, IP and RISK impact on R&D processes?**
  - Compliance
  - IP segregation
  - Maintenance
  - Security
- **Kernel stability & upgrades**
  - Roll-forward vs. back-port
- **QA and code coverage**

OSS Adoption: Management & Strategy

- **Top Level Strategy and Execution**
  - Effective communication of OSS values
  - Contain internal politics and revenue tug-of-war
  - Reward and Recognition
- **Middle management and risk aversion**
  - Linux development, Linux road-map awareness
  - Decision-making and delegation
  - Encouraging collaboration and OSS best practices
  - Constrain NIH and re-invention
- **Engineering Management**
  - Adapting processes to OSS Community and Standards
  - Formal requirements, design, implementation phases

Linux Adoption: Intellectual Property

"To a Linux-based platform strategy, out-of-tree code is illogical "

1. **OSS code (upstream, in-tree) is higher quality / cost**
2. **Internal code (out-of-tree) is lower quality / cost**

- **Elementary Linux core value proposition**
- **Out-of-tree code TCO > in-tree Linux code**

- **Migration from proprietary software processes**
  - Understanding revenue-oriented differentiation
  - Proprietary "value-add" and IP in user-space licensing domain

Linux Adoption: Intellectual Property

A proprietary OS implementation is Intellectual Property. That proprietary IP's value may be very low or negative.

- **Differentiation**
  - Niche markets remain for OS-Platform differentiation
  - product cost vs. product features
  - platform cost across product line(s)
- **Separating OSS and Proprietary development**
  - HW arch, platform, driver support is not proprietary technology
  - Value-add, proprietary software must exist in user-space, non-GPL
- **OSS-aware innovation-oriented business model**

OSS Adoption: OSS Best Practices

- **Adapting to OSS Community and Processes**
  - Internal vs. external coding standard
- **Building the Open Source team**
  - Hire, acquire or grow organically
  - Creating 2-D organization-wide communication
- **Engineering interface to OSS**
- **Enabling upstream contribution and collaboration**
- **Parallel / out-of-tree development**
- **Upstream road-map (emergent technologies)**

Linux Adoption: Working with Linux

- **Linux built by engineer code contributions**
  - No decision-making by management
  - Code first. No requirements, design, implementation phases
  - Developers code based on input from contributors

- **Linux built via web-of-trust (established by quality of contribution)**
  - **Junior engineers build relationships with senior engineers**
  - **Face-to-face meetings (conferences) help build relationships**

"You cannot learn to swim without getting in the water"

OSS Adoption: Long Term Cost

- **Linux engineering (Labor) is not free**
- **Enterprise level quality assurance is not free**
- **Certification, compliance, indemnity is not free**
- **Security patches and updates are not free**
  - Efficiency of internal processes
  - Collaboration and code push-back
  - Appropriate code-re-use
- **Long-term cost dominated by labor**
  - Back-port vs. roll-forward
  - Upstream contribution vs. out-of-tree
  - Professional services
  - Unanticipated cost

Linux Adoption: OSS Product R&D

"Anyone can modify OSS code."

OSS expertise enables platform for a sustainable product cycle.

- **Understand Cost Factors: TTM, Product R&D, LTS**
  - Aggregate life-time COGS
  - Minimize internal code base / patch queue
  - Maintenance / Support
  - Add-on / out-of-tree must account for revenue
- **Align Biz Dev, PM, R&D planning with product cycles**
  - Planning and scheduling
  - Internal inter-product-team communication and feature development
  - co-utilize Kernel R&D and QA
  - Strategic innovation

Linux Adoption: OSS Product R&D

- **Upstream-aware Product development**
  - Develop new core/commodity platform features upstream
  - Early upstream collaboration, frequent development releases
  - Pull-back maturing code and branch to stabilize for product
  - Periodic stabilization merges between product and devel branches
- **Centralization of bug management R&D ←→QA**

  Finding lots of bugs is not bad engineering, its good QA
- **Migration from proprietary to OSS platform**
  - Re-work, abstract, adapt or re-write
  - Avoid tug-of-war with existing code base
  - Short term code-reuse vs. long-term maintenance effort

    "Middleware becomes Muddleware"

- Company Overview
- Linux / OSS Adoption: Challenges
- Linux Adoption at Huawei
- Linux Challenges at Huawei
- OpenHuawei.org: Fast-track to upstream
- OpenHuawei.org: Projects
- Conclusions

Huawei Linux / OSS Adoption Enablers

- **General industry-wide Linux adoption**
  - Market / Customer demand
  - Linux collaboration becoming industry SOP
- **Product complexity vs. R&D effort**
- **Top-level management buy-in and support**
  - Linux value proposition understood
  - Interest in improving engineering-level expertise with Linux/OSS
- **Language barrier**
  - Not applicable for source code
  - Communication via patches is Human language insensitive
- **Engineering staff enthusiasm**
  - Engineering teams are relatively young and eager to learn
- **Existing Linux-based product portfolio**
  - Double-edged sword
  - Adapting Linux to non-modified existing code base

Huawei Linux / OSS Adoption: Strategy

- Linux deployed in large number of Huawei products
- Plan to extend Linux usage in all domains
  - Cloud Platform ⟷ Network Gear ⟷ Consumer Devices (mobile, connected home, etc.)
  - Huawei investing a lot of engineering resources into Linux
- Linux / OSS (combined with commercial / proprietary software)
  - Quality, well known software, standards support
  - Commonality across products and R&D organizations (Architectural, tools, experience sharing, efficiency, R&D cost)
- Enable multi-core networking platforms
  - Different processor architectures
  - Scalability and integration, meet bandwidth demands
  - Participate in projects and advance the development of Linux

Huawei Linux / OSS Adoption: Objectives

- Leverage Linux & OSS in product cycle
- Interact with the Linux / OSS community
- Gain and retain Linux / OSS R&D & QA expertise
- Leadership and Recognition in Linux / OSS
- Innovation in OSS
- Enable differentiation

- Company Overview
- Linux / OSS Adoption: Challenges
- Linux Adoption at Huawei
-  Linux Challenges at Huawei
- OpenHuawei.org: Fast-track to upstream
- OpenHuawei.org: Projects
- Conclusions

Linux Adoption: Challenges

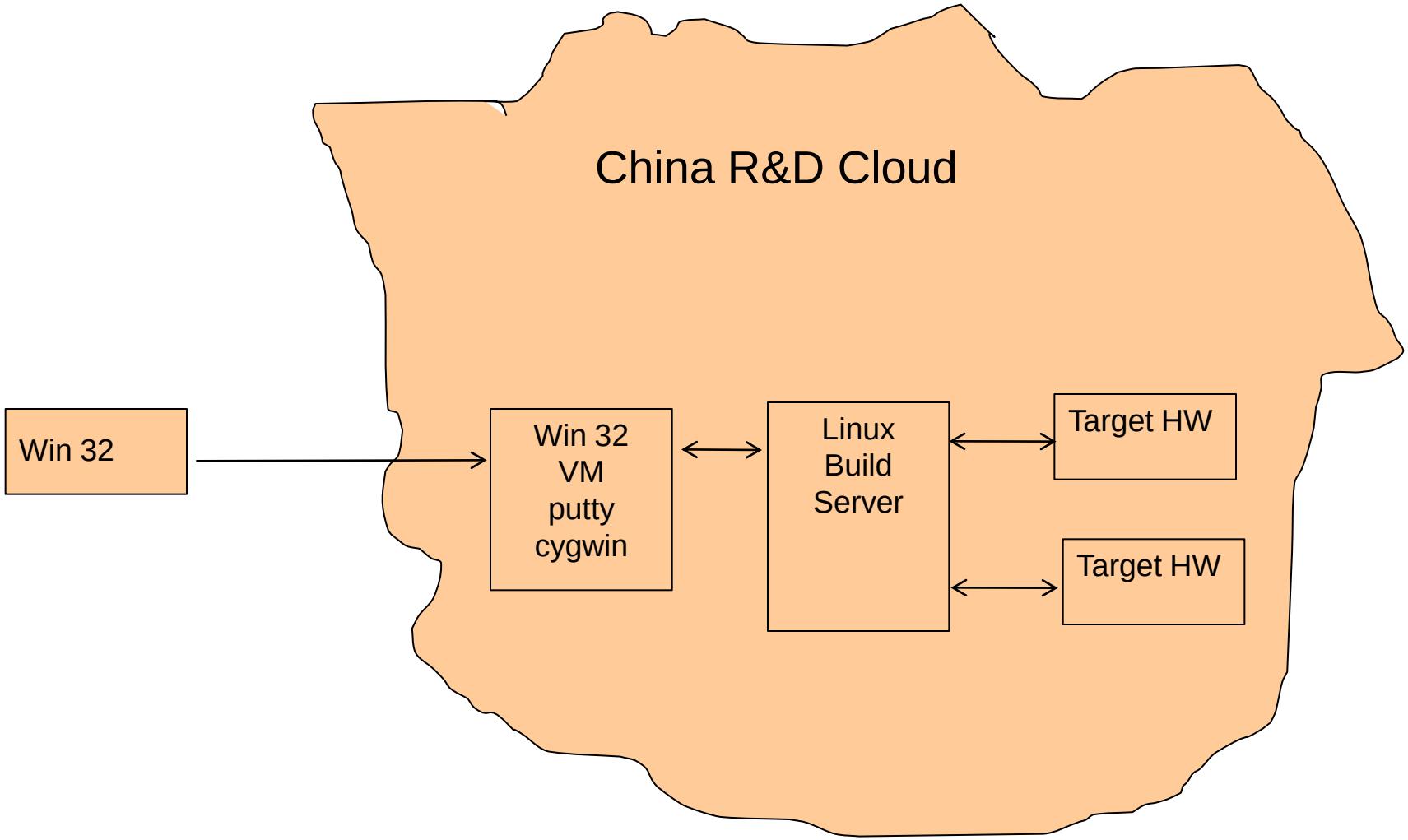
- **Linux development process**
  - Open-Source continuous improvements
  - QA and validation of fast moving code base
  - Upstream feature development vs. product requirements
  - Building and maintaining arch, platform, driver support
  - Identifying and adapting to emerging technologies
  - Project-oriented R&D
  - Identifying and adapting-to emerging technologies
- **Adapting to OSS processes**
- **Policy**
  - Publishing to code / mail patches
  - Differentiation between proprietary and OSS code
  - Transfer GPL/Kernel/OSS code from China R&D cloud

Huawei Linux / OSS R&D Challenges

- **Linux / Unix developer environment**
  - Windows 7 32bit workstations, Exchange MUA
  - Offshore R&D uses VM cloud / VDI to access R&D environment
  - Standard Linux/OSS tools
  - Local source code
  - Mailing patches and commenting inline per OSS conventions

- **Interaction with hardware**
  - Remote hardware and dev env
  - Cannot observe (fan, disk, led) hardware
  - No local jtag logic analyzer / oscilloscope / flash tools
  - No local NFS root file system for embedded Linux development

Legacy Product R&D Environment



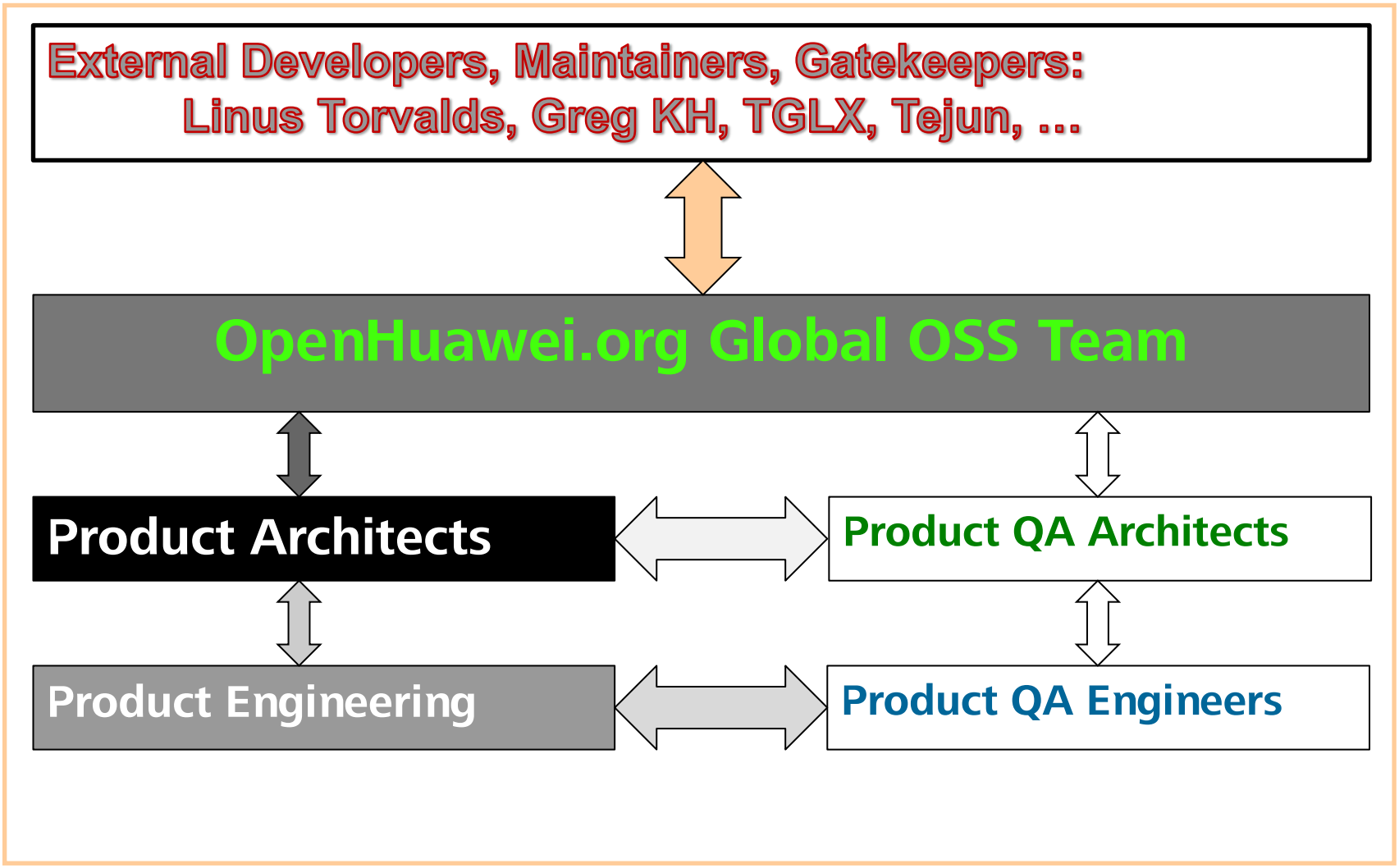
- Company Overview
- Linux / OSS Adoption: Challenges
- Linux Adoption at Huawei
- Linux Challenges at Huawei
-  OpenHuawei.org: Fast-track to upstream
- OpenHuawei.org: Projects
- Conclusions

OpenHuawei .ORG

- Information-bridge / community interface
- Allow bi-directional innovation and contribution
- Shield product planning and IP from direct scrutiny
- OpenHuawei evaluating and developing OS platform
- scalable, next-gen
- high-performance
  - enabling bare-metal performance
  - high-bandwidth networking and IO
  - low latency

- Enable user-space performance and differentiation

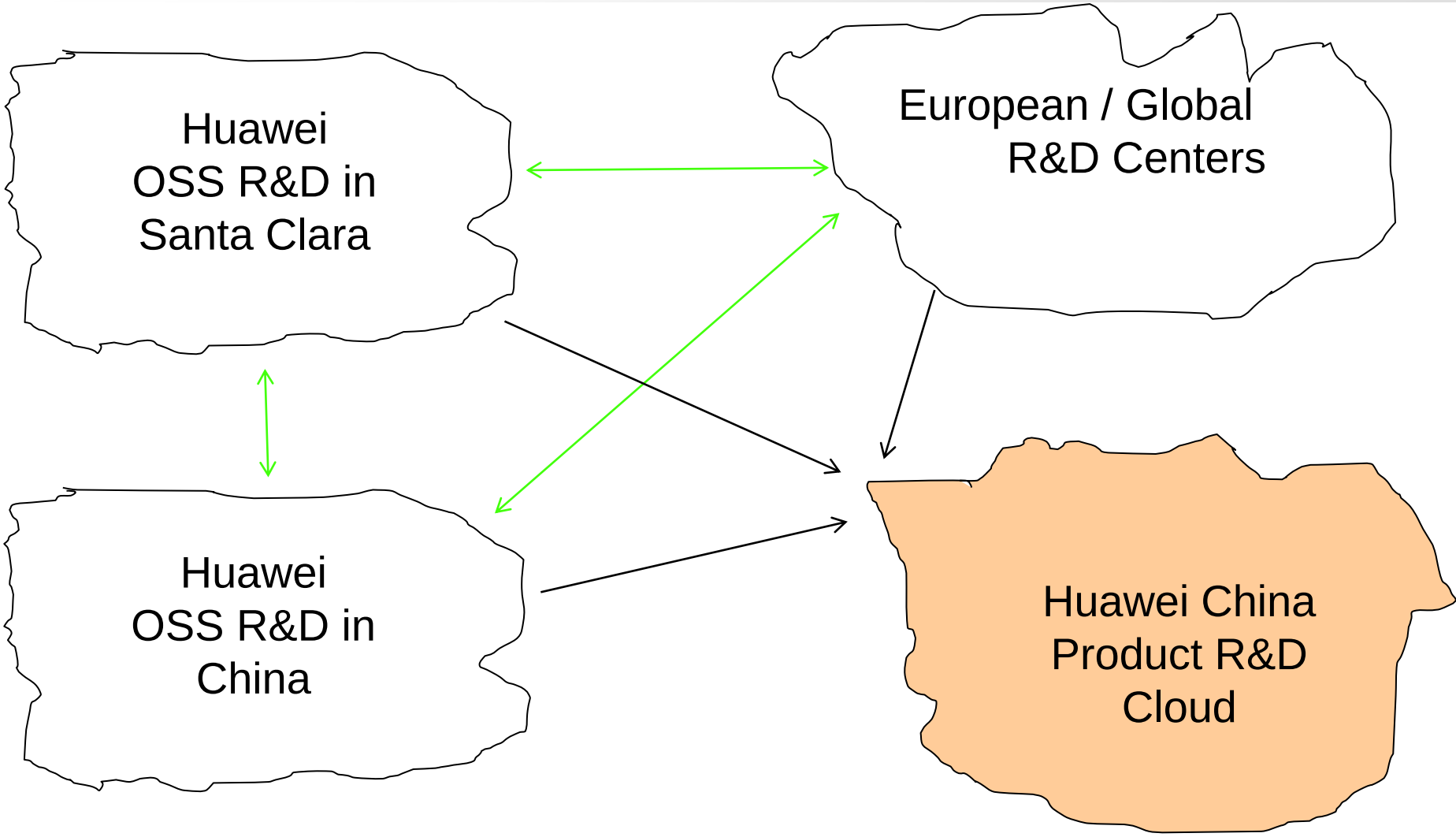
Community Interface: Kernel example



OpenHuawei R&D Lab Goals

- **Help recruit Linux engineers**
- **Enable productive Global OSS collaboration**
  - Ability to work with local hw
  - Enable development on native Linux workstations
- **Host/mirror upstream OSS source code**
  - u-boot, Linux, Virtualization, RT patches, etc.
- **Build service (OBS / Yocto)**
- **Host developer home directories as needed**
- **Access to OpenHuawei by Huawei teams worldwide**
- **Enable OSS best-practices by example**
  - Contributor to Apache Hadoop
  - Kernel contributions from Zefan Li (cgroup maintainer)

OpenHuawei OSS Code Exchange



OpenHuawei.org: R&D Lab Services

- **Services should be accessible by everyone at Huawei**
  - via http(s) and corporate proxy
  - compliant with monitoring
  - no direct connection to the internal Huawei networks
- **OpenVPN for global developer connectivity**
- **Linux/OSS distro environment (Suse)**
  - Git / gitweb
  - Subversion, mercurial, cvs, etc.
  - Bugzilla
  - Mail lists & list server
  - IRC & Wiki collaboration
  - Project management tools like Redmine
  - TFTP, Bootp, NFSroot, u-boot support
  - Linux Kernel build and debug tools
  - GCC (cross) tool chain

External OSS Development Environment

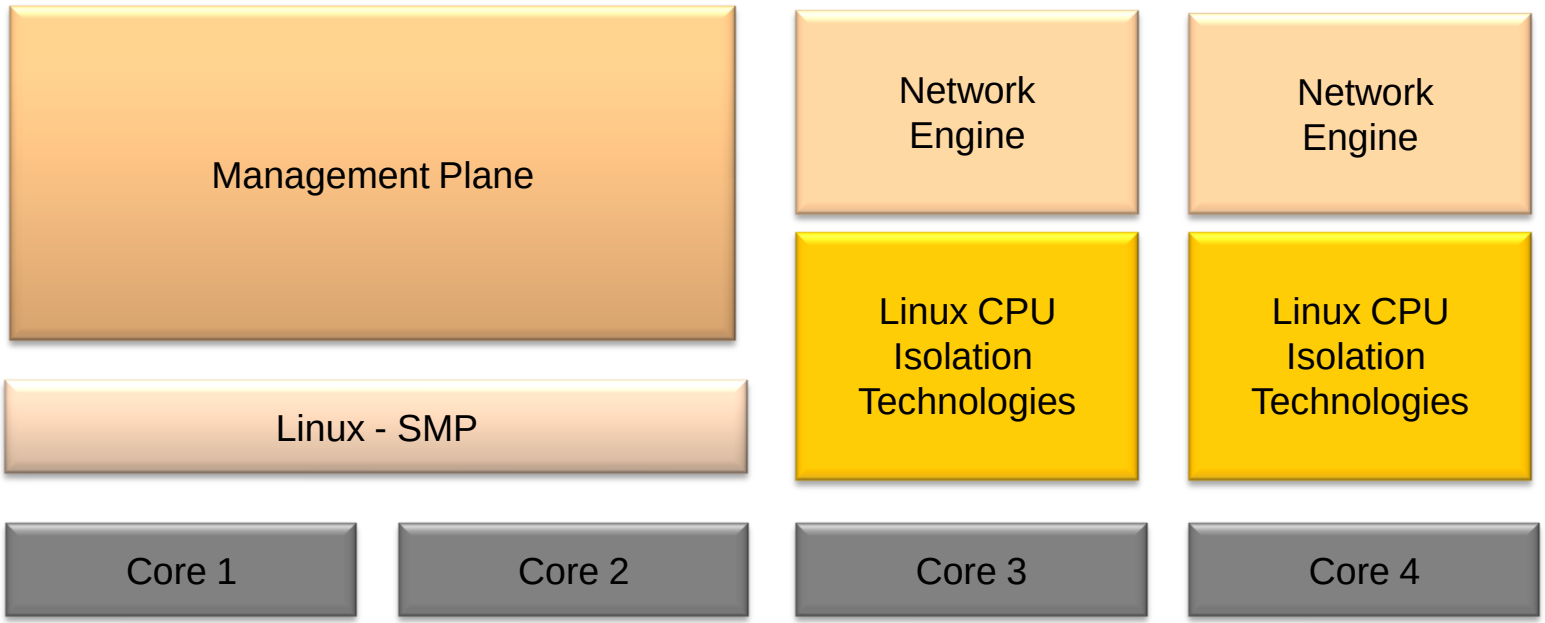
- Google Apps Domain
- OpenSuse 12.1 based (migration to 12.2 pending)
- Global Access via OpenVPN gateway @100Mb
- Internal DNS (dnsmasq)
- Shared (team) development server Dell
  - 4 x 10 HT cores
  - 128 G RAM
  - RAID
  - KVM
- Test / Development Platforms
  - HP blades 2 socket, 6x HT cores
  - 4 x Panda Board
- RackStation SAN via iSCSI
- 10 Gb backbone to SAN
- 1 Gb front end to ISP gateway

- Company Overview
- Linux / OSS Adoption: Challenges
- Linux Adoption at Huawei
- Linux Challenges at Huawei
- OpenHuawei.org: Fast-track to upstream
- OpenHuawei.org: Projects
- Conclusions

OpenHuawei Linux Projects

- **Adoption of multi-core designs**
  - Cost reduction through consolidation
  - Networking performance gains through accelerations techniques
- **Multi-core hardware architecture disruptive technology**
  - Linux plays crucial role
- **Multi-core adoption challenges in architecture R&D**
  - How to utilize existing software to multi-core environment
  - How to utilize all the advantages of multi-core in software
  - How can Linux help?
- **Common design is Control and Data Plane separation**
  - Fast path processing in the transport plane
  - SMP Linux does not achieve full power of multi-core networking
  - AMP Linux (Network stack parallel processing scalability)

Native Linux Data Plane / Control Plane Solution



- **Linux process isolation technologies**
- **User Space or Kernel space networking**

Time scale: 10Gbps net vs. 3Ghz CPU

- **Syscall ≈100 cycles**
- **Context switch ≈1000 cycles**
- **Memcpy 64B to 1KB ≈60 cycles best case/hot start, ≈700 cycles worst case**
- **IPI Memory latency 20-30 cycles**
- **Generating UDP header** ≈350 cycles (no UDP checksum)
- Simple UDP packet forwarding ≈100 cycles (no UDP checksum)
- **10G line up to 19 Mpps (million packets/second) ingres & egres**
- **3Ghz core: up to 5 Mpps (@ 500 cycles/128B UDP payload)**

By Vlad Buzov and Nikita Shulga

Linux packet processing pitfalls

- Traditional socket interface 500 kpps/core
- Kernel Network QoS scales poorly with increasing cores
- Vmslice
- Recvmsg() to receive several packets in one system call
- Memory-mapped RAW sockets: good asynchronous interface, but requires memcpy to move packet from RX to TX queue
- IRQ affinity: sometimes helps/sometimes hurts performance
- mitigate IRQ flood: MSI-X, NAPI, RX/TX queue watermark

By Vlad Buzov and Nikita Shulga

- Company Overview
- Linux Adoption: General History
- Linux Adoption at Huawei
- Linux Challenges at Huawei
- OpenHuawei.org: Fast-track to upstream
- OpenHuawei.org: Projects
- Conclusion

Thank you

sven@openhuawei.org
Sven.Dietrich@Huawei.com

www.huawei.com