OpenSolaris Introduction

Jerry Jelinek
Solaris Engineering
Sun Microsystems

Agenda

- What is OpenSolaris
- Participating in the OpenSolaris community
- Suns internal development process
- OpenSolaris source code
- Developer guidelines and tips
- Getting started

What is OpenSolaris

- We are making Solaris *open source*
  - License allows royalty-free use, modification, derived works
- Not just source code – also *open development*
  - Non-Sun developers can contribute code, documentation, etc.
  - Collaboration between Sun and non-Sun engineers
- Want to build a community around Solaris
  - Developers *and* users

Why Open Source?

- Customers are demanding it
- Good for security and overall quality
- Good for the future of the technology
  - Stand on each other's shoulders
- Open development can take Solaris to places closed development could not

Interactions with Community

- Sun and non-Sun developers work together on open source projects
  - Also Sun-only, community-only projects
- Projects integrated into open source code base after full design and code reviews
  - Transparent, open review process including reviewers from community
- Commercial Solaris product assembled from open source code
- Other distributions as well (`http://schillix.org/`)

How this will change Solaris

- Commercial Solaris product still available with testing, support, ISV certification, etc.
- Same level of quality, compatibility, security as before (if not better)
- Community provided features, drivers, bug fixes, etc. improve both open source and commercial products

opensolaris.org

- Central resource for information
- Browse & download source
- Announcements
- Communities
  - Nevada, SVM, UFS
  - Dtrace, Fault Management, Zones, PPC, ...
- Discussions
  - Bugs, code, RFEs, tools, user groups, Xen, ...

Community Advisory Board

- Board to guide decisions regarding OpenSolaris
- Represents both Sun and external community
- Initial members:
  - Casper Dik (Senior Staff Engineer, Sun)
  - Roy Fielding (Chief Scientist, Day Software)
  - Al Hopper (Engineering Consultant, Logical Approach)
  - Simon Phipps (Chief Technology Evangelist, Sun)
  - Rich Teer (Freelance Consultant)
- Community: cab

Licensing

- Most source will be released under the CDDL
  - Community Development and Distribution License
  - OSI-approved, open source license
  - Source for modifications must be published, but proprietary extensions possible
  - Patent protections for developers and users
  - See http://opensolaris.org/os/about/faq/licensing_faq/
- Some code under existing open source license (e.g. GPL) will remain under the same license

Source Content

- Currently released
  - Kernel and core commands and libraries – except for code with legal encumbrances
  - Includes Solaris 10 innovations: DTrace, Predictive Self-Healing, Zones, etc.
  - GNOME, X11, etc. already available
- Long term
  - Release source for all components of Solaris except those with legal encumbrances
  - Start rewriting encumbered code

Participating in the OpenSolaris Community

- Connect with Sun-internal engineers and others who share a common interest
  - Share your knowledge
  - Provide feedback to developers (internal & external)
  - Ask questions
- Contribute to development
  - Minor bug fixes - major projects

Solaris Development Process

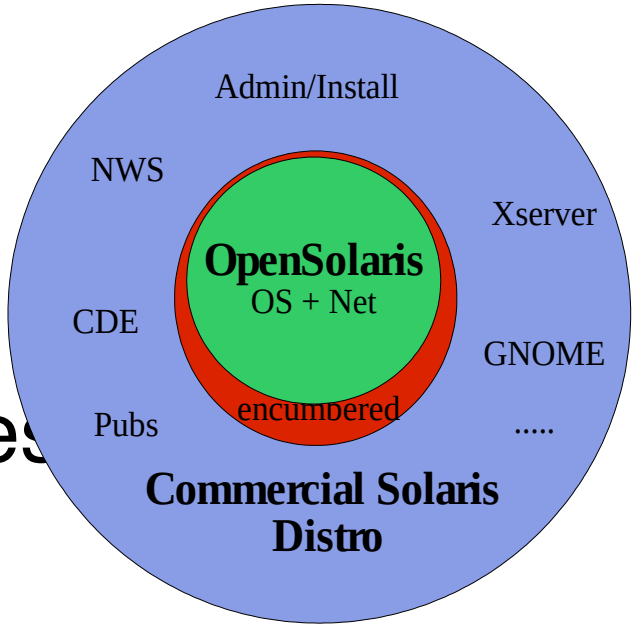
- Our internal development model works well
  - Scales from small fixes to large projects
  - Supports distributed developers
  - Supports distributed reviewers
  - Emphasizes technical decision making

Solaris Development Today

- Hundreds (perhaps thousands) of developers
  - Distributed around the world
- The whole product is called the WOS (Wad Of Stuff)
- WOS composed of *consolidations*
  - Collections of similar technology
  - ON, man pages, CDE, GNOME, Java, etc. are each their own consolidation
  - Each has a technical leadership body called a consolidation team, or c-team
  - Development practices can vary from consolidation to consolidation, but mostly the same
- New WOS build every two weeks

ON Consolidation

- ON == OS/Networking
- Largest consolidation
- Contains kernel, core libraries commands, daemons, networking, etc.
  - > Pretty much everything needed to boot
- Initial OpenSolaris release is ON
  - > Other consolidations will follow



ON Source Code Management

- One repository, called the *gate*
- Cared for by *gatekeepers*
  - Engineers who volunteer for the task
  - Have ultimate, complete discretion over what changes stay, and what changes go
- Source/revision control: Teamware
  - Hierarchically structured
  - Supports disconnected workspaces
  - Filesystem-based
  - bringover, modify, resolve, putback

ON Process

- Centered around technical decision making
- Based on principle of “shrink to fit”
  - Unnecessary parts of process eliminated
- Goal: Release-level quality all the time
  - Avoid the “quality death spiral”

Technical Considerations

- Compatibility
- Correctness
- Debuggability
- Good interface design
- Performance
- Completeness

Process: large-scale project

- Business review
- Formal architectural review (primarily of interfaces)
- Design review
- Code review
- Test plan
- C-team review
- Putback

Applying Shrink-to-fit

- Not all projects require all these steps
- bug fix/small RFE
  - ~~Business review~~
  - ~~Formal architectural review (primarily of interfaces)~~
  - ~~Design review~~
  - Code review
  - ~~Test plan~~
  - ~~C-team review~~ Request To Integrate
  - Putback

Evolving for OpenSolaris

- Seems like our processes could easily accommodate open development
  - Ultimately want to include community members in all of the review bodies
  - But we could be wrong; time and experience will tell
- Discussion
  - OpenSolaris: arc

OpenSolaris Source Tour

- `http://opensolaris.org/os/downloads/`
- usr/src
  - uts (kernel)
    - i86pc
    - common/dtrace
  - lib (libraries)
    - libc
    - libdtrace
  - cmd (commands)
    - ls
    - dtrace

OpenSolaris Source Tour

- SVM
  - usr/src/uts/common/io/lvm
  - usr/src/lib/lvm
  - usr/src/cmd/lvm
- UFS
  - usr/src/uts/common/fs/ufs
  - usr/src/cmd/fs.d
- Discussion
  - OpenSolaris: code

Developer Playbook

- As an OpenSolaris developer, how to reach important engineering goals?
  - Tools, techniques, tips
    - Correctness
    - Compatibility
    - Debuggability
    - Performance

Correctness

- The most important constraint, but also the most difficult to satisfy
- Not feasible to prove the software correct
- The common answer is testing
- It is better to crash than to corrupt data
- In general, stop as soon as you notice a problem
  - `cmn_err(9F)`, `assert(3C)`
  - Also easier to debug

Compatibility

- Another difficult one
- Rigorous testing finds many problems
  - Standards tests, regression tests, desktop usage
- Automated API checking
- Senior engineers are a big help here
  - History often repeats itself
  - Advocates often offer advice to RTI submitters
- We have well-defined interface boundaries
  - Drivers should stick to the DDI
  - Applications shouldn't consume undocumented interfaces, either

Debuggability

- Unfortunately, no-one is perfect
- Want to minimize time it takes to debug a problem
- Want to maximize information available after a single failure
  - Customers don't like hearing that their application/machine/datacenter needs to fail again before the problem can be fixed
- Write readable code
  - Design and code review feedback extremely valuable
- Core files
  - Never turn them off from within an application

Performance

- Poor Solaris performance is a bug, not an RFE
- As important as performance is, correctness always takes priority
- Performance fixes without benchmarks aren't
  - Measurement is essential
- Dtrace – use it!
- Search for MT bottlenecks
  - In kernel, use `lockstat(1M)`
  - In userland, use `plockstat(1M)`

OpenSolaris Mechanics

- Initially bug fix-driven
  - Lots of hand-holding
  - Community developers will have engineering “sponsors” at Sun
  - Discussion - OpenSolaris: request-sponsor
- Eventually automated access to repository
- Biggest hurdle: technology
  - Source code management
  - Access to test suites
  - Access to hardware

Getting Started

- Build the source
  - See: `http://www.opensolaris.org/os/downloads/`
- Fix a simple bug
  - Get your feet wet
  - `http://opensolaris.org/bugdatabase/`
  - Search for: `oss-bite-size`
    - 70 bugs
- Discussions
  - tools: discuss
  - tools: gcc

Questions ?

OpenSolaris Introduction

Jerry Jelinek
gerald.jelinek@sun.com