

open



USE



IMPROVE



EVANGELIZE

OpenSolaris Introduction

Vladimir Kotal & Sasha Nedvedicky

Solaris RPE (Security technologies)

Sun Microsystems, Inc.

開
放
的
열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
πικρό
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை



OpenSolaris

- Released on June 14th, 2005
 - > Opened Solaris source code under CDDL
 - > Now in its 2nd year
- OpenSolaris project
 - > From opensolaris.org:
 - > **“The OpenSolaris project is an open source community and a place for collaboration and conversation around OpenSolaris technology.”**
- OpenSolaris Governing Board (OGB)
 - > Manages and directs OpenSolaris community
 - > Community feedback is important and always considered



TM

open source



OpenSolaris content

- Communities

- > Covering A-Z (almost)
- > Academic and research, Dtrace, Desktop, Device drivers, Laptop, Networking, Performance, Security, Testing, Xen, ZFS, Zones, ...

- Projects

- > Bluetooth stack, CIFS, Caiman (new installer), Clearview (Net administration), Crossbow (NIC virtualization), Crypto framework, Image packaging system (new packaging), Indiana (OpenSolaris distribution framework), Live media, Network automagic, Open Sound system, OpenGrok (fast source browser), Packet event framework, Tesla (enhanced power management), Suspend/resume, ZFS encryption, ...

How Does Innovation Work in OpenSolaris?

Technology
Ports

Distributions



opensolaris™

Nevada Source Gate



Distributions

- Collection of source bases (Consolidations)
 - > ON (OS-Networking), JDS (Gnome), Install, ...
- Existing distributions
 - > SXCE (Solaris Express Community Edition)
 - Build snapshot every 2 weeks
 - <http://www.opensolaris.org/os/downloads/>
 - > SXDE (Developer edition)
 - Every 3 months
 - <http://developers.sun.com/sxde/download.jsp>
 - > Nexenta, Belenix, ...
 - > Indiana



Indiana

- Project started by Ian Murdock
- A binary distribution of OpenSolaris developed by OpenSolaris community
 - > People external and internal to Sun
 - > Can be used as a basis for derived distributions
 - > Developer preview was released on October 31st
- Live CD, installation from CD
 - > It will be possible to install other packages from repositories at the internet
- The first version should be available in March 2008



The source !

- Where is the source ?
 - > Mercurial
 - Distributed SCM
 - ON gate
 - > OpenGrok
 - src.opensolaris.org
- CDDL (aka cud'l)
 - > Common Development and Distribution License
 - > GPL/CDDL/BSD differences:
 - http://blogs.sun.com/chandan/entry/copyright_s_licenses_and_cddl_illustrated



Quality

- Code reviews
 - > webrevs
 - > cr.opensolaris.org
- Architectural reviews
 - > ARC community
 - > PSARC
- Testing
 - > Test suites
 - > Unit testing
 - Before filing a RTI (Request to Integrate)



Contributing to OpenSolaris

- Be part of a user group
 - > CZOSUG (Czech OpenSolaris User Group)
 - cz.opensolaris.org
 - Meetings once every 2 months or so
- Sponsor request program
 - > http://blogs.sun.com/jimgris/entry/request_sponsor
 - > <http://opensolaris.org/os/communities/participation/>
- Post Mercurial transition
 - > <http://www.opensolaris.org/os/project/scm-migration/>
 - > You'll do the putback (commit) by yourself !



Technology Ports

- DTrace (Dynamic Tracing)
 - > Mac OSX 10.5 Leopard
 - > FreeBSD (<http://people.freebsd.org/~jb/dtrace/>)
 - > QNX (http://blogs.sun.com/bmc/entry/dtrace_on_qnx)
- ZFS (Zettabyte File System)
 - > Mac OSX 10.5 Leopard
 - > ZFS is being integrated in FreeBSD 7-CURRENT and will be available as “experimental feature” in FreeBSD 7.0-RELEASE
 - > Linux – ZFS on FUSE



Dtrace intro

- General debugging paradigm
 - > In order to **see** software, we make it slow (debugging statements)
 - > We want to debug production systems (no debug version of program needed)
 - Change the program text that is being executed
- Want to dynamically instrument running system
 - > Both userland and kernel
 - > With no (or almost) performance impact
 - Activate/deactivate the probes **on the fly**, make them emit **data**
 - > **Safely** instrument (no disruption of the system)



Dtrace basics

- `dtrace -l`
 - > List every point in the system that we know how to instrument (lots of probes ! Solaris has many (?) probes now)
- Providers
 - > Each provider knows how to instrument different portion of the system
 - syscall, pid, fbt, sdt, lockstat, mib, io, sched, profile, vminfo, sysinfo, ... (others are coming !)
- syscall provider
 - > Knows how to instrument system calls
- fbt (Function boundary tracing) provider
 - > Instrument **every** function entry and return in the kernel



Dtrace examples

- `dtrace -n syscall:::entry`
 - > Instruments **every** system call entry in the system (as it happens) with default action (indication that the probe fired)
 - truss/strace do it only on per-process basis ! (and they are stopping the process)
- D language
 - > Combination of C and awk (somewhat)
 - > Observe who makes the syscalls (including us !)
 - dtrace -n 'syscall:::entry { trace(execname); }'
 - > Aggregations
 - dtrace -n 'syscall:::entry { @[execname]=count(); }'



Diving into the system with dtrace

- Answer to one question provokes the next Q
 - > We have counts of systems calls by various processes. What is process X doing with so many system calls ?!
- Predicates help us to filter out data
 - > `dtrace -n 'syscall:::entry/execname == "xterm"/ { @[probefunc]=count(); }'`
 - Now we see number of system calls issued by **all** xterm processes
 - Aggregate also by pid (`[probefunc,pid]`)
- What are the programs doing issuing read() ?
 - > `dtrace -n 'syscall::read*:entry/execname == "xterm"/ { @[pid,ustack()] = count(); }'`

Getting back to processes

- Want to see when Xpending() is called

```
> dtrace -n 'pid1281::Xpending:entry
  { printf("Xpending called at %Y\n",
    walltimestamp); }' -q
```

- When script overgrows one-liner it's time for a D script

```
#!/usr/sbin/dtrace -Fs
pid1281::Xpending:entry
{ self->follow = 1; }    # thread-specific variable
pid1281:::entry
pid1281:::return
/self->follow/
{}
pid1281::Xpending:return
{ self->follow = 0; exit(0); }
```



Understanding interactions

- flowident (-F) helps us to see the whole code flow
 - > Add fbt probes
 - > Not only for debugging but also for understanding the code !

```

0  -> XPending
0   -> sys_rtt_common
0   <- sys_rtt_common
0   -> _XEventsQueued
0     -> sys_rtt_common ← This is kernel !
0     <- sys_rtt_common
0     -> _Xflush ← This is userland !
...
    
```



Dtrace possibilities

- SDT provider
 - > Custom probes in your app !
- Profiling your app !
 - > Using ticks (Solaris has excellent cyclic timer subsystem)
- Troubleshooting
 - > Failures to open a file in the system:

```
-dtrace -n 'syscall::open*:entry '{self->file=copyinstr(arg0); } syscall::open*:return/arg1== -1/ { printf("%s[pid:%d] failed to open %s ", execname, pid, self->file); }'
```



Dtrace resources

- Brendan Gregg's
 - > Dtrace one-liners
 - http://www.brendangregg.com/DTrace/dtrace_oneliners.txt
 - > Dtrace toolkit
 - <http://www.opensolaris.org/os/community/dtrace/dtracetoolkit/>
- www.solarisinternals.com
- `/usr/demo/dtrace`
- Bryan Cantrill's presentation at Google
 - > <http://video.google.com/videoplay?docid=-8002801113289007228>



Why to Use OpenSolaris

- An alternative to other open source operating systems, like Linux, FreeBSD
- A unique combination
 - > Commercial OS, which is now open sourced
 - > Benefit from proven code
 - > Backward compatibility and stability
- Direct access to cutting edge technologies
 - > DTrace, ZFS, Zones, SMF
- Option to participate
 - > OpenSolaris projects and communities
 - > Create your own project



OpenSolaris and Solaris Resources

- OpenSolaris discussions
 - > <http://www.opensolaris.org/os/discussions/>
- Talks and presentations
 - > http://www.opensolaris.org/os/community/os_user_groups/os-presentations
- Sun Developer Network (SDN)
 - > <http://developers.sun.com>
- BigAdmin
 - > <http://www.sun.com/bigadmin/home>

open



USE



IMPROVE



EVANGELIZE

**Questions ? Come to talk to us
while having a beer !**

Vladimir Kotal & Sasha Nedvedicky

<http://blogs.sun.com/vlad/>

<http://blogs.sun.com/SashaN/>

“open” artwork and icons by chandan:

<http://blogs.sun.com/chandan>

開
放
的
열린
مفتوح
libre
मुक्त
ಮುಕ್ತ
livre
libero
ముక్త
开放的
açık
open
nyílt
:::~::~
πικρ
オープン
livre
ανοικτό
offen
otevřený
öppen
открытый
வெளிப்படை