SMF – Solaris Service Management Facility

Sebastian Marius Kirsch
skirsch@luusa.org

24th August 2006
What is SMF?

- Service Management Facility
- feature of Solaris 10 and successors (OpenSolaris, Solaris 11)
- ‘A mechanism to define, deliver and manage long-running services for Solaris’
- ‘/etc/init.d on steroids’
What is a service?

‘A service is a long-lived software object with a well-defined state, error boundary, definition of start and stop, and relationships to other services. A service is often critical to operation of system or fulfillment of business objectives.’

What does SMF do?

- replaces the conventional method of starting services via scripts in `/etc/rc?.d`
- uses a single daemon (`svc.startd`) to start, monitor, restart, and stop services
- configuration and state are kept in a database (SQLite, in `/etc/svc`)
- database is populated from XML files (manifests, profiles)
- handles startup, dependencies, fault detection and management, ...
Service dependency hell
Components: Architecture schematic

init(1M)
svc.startd(1M)
observability/
management
agent
commands

inet-service
deged
service

inetd(1M)
delegated
restarter
deged
service

svc.configd(1M)
repository API

svc.startd(1M)

init(1M)

KERNEL
contracts

A typical first encounter with SMF

- /etc/rc?.d and /etc/inetd.conf are suspiciously empty.
- Where are all my init scripts?
- I kill daemon X, and it just reappears!
- How do I stop this f*****g daemon?
SMF administration

- svcs: print the state of all services, including reasons for failure and dependant services
- svcadm: enable/disable services, clear fault states
- svccfg: import new service descriptions, change properties
Example: svcs

$ svcs -x svc:/network/smtp:sendmail
svc:/network/smtp:sendmail (sendmail SMTP mail transfer agent)
  State: online since Wed Nov 30 18:29:34 2005
  See: sendmail(1M)
  See: /var/svc/log/network-smtp:sendmail.log
Impact: None.

$ svcs -D svc:/network/smtp:sendmail
STATE   STIME   FMRI
online  Nov_30 svc:/milestone/multi-user:default

$ svcs -d svc:/network/smtp:sendmail
STATE   STIME   FMRI
online  Nov_30 svc:/system/identity:domain
online  Nov_30 svc:/system/filesystem/local:default
online  Nov_30 svc:/milestone/name-services:default
online  Nov_30 svc:/network/service:default
online  Nov_30 svc:/system/filesystem/autofs:default
online  Nov_30 svc:/system/system-log:default
Adding a new service

- services are described by a manifest (XML file)
- components of a manifest:
  - service name
  - dependencies
  - methods for starting, stopping, and refreshing the configuration of a service
  - specific instances
  - environment for methods (user/group, environment variables)
  - when to restart a service (after processes exit, on core dumps, on signals received, ...)
- is imported via svccfg
A manifest example

```xml
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM
"/usr/share/lib/xml/dtd/service_bundle.dtd.1">

<service_bundle type='manifest'
  name='OracleListener9.2.0'>
  <service
    name='application/oracle/listener'
    type='service'
    version='1'>
    <single_instance />
  </service>
</service_bundle>
```
A manifest example (cont’d)

15    <dependency
16       name='fs-local'
17       grouping='require_all'
18       restart_on='none'
19       type='service'>
20       <service_fmri
21          value='svc:/system/filesystem/local'/>
22    </dependency>
23
24    <!-- ... -->
25
26    <dependent
27       name='oracle-multiuser-server'
28       grouping='require_all'
29       restart_on='none'>
30       <service_fmri
31          value='svc:/milestone/multi-user-server'/>
32    </dependent>
A manifest example (cont’d)

34     <instance name='v9-2-0' enabled='false'>
35
36     <dependency
37         name='config'
38         grouping='require_all'
39         restart_on='restart'
40         type='path'>
41     <service_fmri
42         value='file://localhost/local/oracle
43         /9.2.0/network/admin/listener.ora'/>
44     </dependency>
A manifest example (cont’d)

<exec_method type='method' name='start' 
  exec='/local/oracle/9.2.0/bin/lsnrctl start' 
  timeout_seconds='60'>
<method_context>
  <method_credential 
    user='oracle' group='dba' />
  <method_environment>
    <envvar 
      name='ORACLE_HOME' value='/local/oracle/9.2.0'/>
  </method_environment>
</method_context>
</exec_method>

<exec_method type='method' name='refresh' 
  exec='/local/oracle/9.2.0/bin/lsnrctl reload' 
  timeout_seconds='60'>
<!-- ... -->
</exec_method>
A typical second encounter with SMF

- Ick, XML!
- Ick, a database! (‘That’s just like a Registry for Unix!’)
- Isn’t all this stuff needlessly complicated?
- Can’t I just turn it off?
What’s so great about SMF?

▶ services are first-class objects
▶ faults can be detected
  without SMF: No way of monitoring a service
    ▶ if it detaches or spawns child processes
    ▶ for signals received or core dumps, or hardware errors
▶ dependencies can be restarted
▶ everything is restartable, even svc.startd
▶ automatic logging of all output
▶ parallel startup
▶ kill processes belonging to a service, and only them
  ▶ no fiddling around with pgrep or pid files
▶ no fiddling with su to run service as a different user
▶ transaction-safe configuration repository
Under the hood: Contracts

- How does svc.startd detect exiting processes or signals received?
- conventional Unix API does not support this kind of service monitoring
- contracts are a kernel feature of Solaris 10
- contracts have
  - an owner (may be orphaned)
  - member processes
  - event sets (informative, critical, fatal)
- member processes and their children stay in the same contract, unless they explicitly create a new contract
- contracts can be monitored for being empty, for signals, for core dumps, and hardware errors
- contracts can be regained (after owning process exits)
Looking at a contract

bash-3.00 ctstat -i 6549 -v

<table>
<thead>
<tr>
<th>CTID</th>
<th>ZONEID</th>
<th>TYPE</th>
<th>STATE</th>
<th>HOLDER</th>
<th>EVENTS</th>
<th>QTIME</th>
<th>NTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>6549</td>
<td>150</td>
<td>process</td>
<td>owned</td>
<td>13964</td>
<td>0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

  - cookie: 0x20
  - informative event set: none
  - critical event set: hwerr empty
  - fatal event set: none
  - parameter set: inherit regent
  - member processes: 20828 20830 20832 20834 20836 20838 20840 20842
  - inherited contracts: none

bash-3.00 ps -f -p 13964,20828,20830,20832,20834,20836,20838,20840,20842

<table>
<thead>
<tr>
<th>UID</th>
<th>PID</th>
<th>PPID</th>
<th>C</th>
<th>STIME</th>
<th>TTY</th>
<th>TIME</th>
<th>CMD</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>13964</td>
<td>13949</td>
<td>0</td>
<td>Sep 22</td>
<td>?</td>
<td>1:00</td>
<td>/lib/svc/bin/svc.startd</td>
</tr>
<tr>
<td>oracle</td>
<td>20828</td>
<td>13949</td>
<td>0</td>
<td>Sep 29</td>
<td>?</td>
<td>1:31</td>
<td>ora_pmon_ZTEST</td>
</tr>
<tr>
<td>oracle</td>
<td>20830</td>
<td>13949</td>
<td>0</td>
<td>Sep 29</td>
<td>?</td>
<td>0:40</td>
<td>ora_dbw0_ZTEST</td>
</tr>
<tr>
<td>oracle</td>
<td>20832</td>
<td>13949</td>
<td>0</td>
<td>Sep 29</td>
<td>?</td>
<td>1:39</td>
<td>ora_lgwr_ZTEST</td>
</tr>
<tr>
<td>oracle</td>
<td>20834</td>
<td>13949</td>
<td>0</td>
<td>Sep 29</td>
<td>?</td>
<td>4:21</td>
<td>ora_ckpt_ZTEST</td>
</tr>
<tr>
<td>oracle</td>
<td>20836</td>
<td>13949</td>
<td>0</td>
<td>Sep 29</td>
<td>?</td>
<td>0:26</td>
<td>ora_smon_ZTEST</td>
</tr>
<tr>
<td>oracle</td>
<td>20838</td>
<td>13949</td>
<td>0</td>
<td>Sep 29</td>
<td>?</td>
<td>0:00</td>
<td>ora_reco_ZTEST</td>
</tr>
<tr>
<td>oracle</td>
<td>20840</td>
<td>13949</td>
<td>0</td>
<td>Sep 29</td>
<td>?</td>
<td>1:41</td>
<td>ora_cjq0_ZTEST</td>
</tr>
<tr>
<td>oracle</td>
<td>20842</td>
<td>13949</td>
<td>0</td>
<td>Sep 29</td>
<td>?</td>
<td>2:36</td>
<td>ora_qmn0_ZTEST</td>
</tr>
</tbody>
</table>
What else?

- store properties in the repository
  For example:
  - store name of Apache config file in repository
  - have several instances with different config files
- SMF is integrated with RBAC (role-based access control)
  - you can delegate rights to restart services or change their configuration to ordinary users
- supports delegated restarters
  - example: inetd
- is integrated with zones
  - pgrep/pkill in the global zone? Bad idea...
What is SMF not?

- SMF is not network aware (no dependencies/restarts across different machines)
- SMF just does process monitoring, not monitoring of functionality
- SMF is not provisioning.
Where can I find out more?

- SMF introduction at BigAdmin:
  [http://www.sun.com/bigadmin/content/selfheal/smf-quickstart.html](http://www.sun.com/bigadmin/content/selfheal/smf-quickstart.html)

- Liana Praza’s blog:

- SMF design criteria:

- SMF and RBAC:

- Link collection at del.icio.us:
  [http://del.icio.us/tag/solaris+smf](http://del.icio.us/tag/solaris+smf)
Questions? Feedback?
Thanks for listening!

Slides are available at

http://www.sebastian-kirsch.org/moebius/docs/smf.pdf
SMF – Solaris Service Management Facility

Sebastian Marius Kirsch
skirsch@luusa.org

24th August 2006