OpenSAF –
A Standardized HA Solution

LinuxCON – Edinburgh, UK – 2013-10-21

Anders Widell
Ericsson AB
Outline

● What are OpenSAF and SA Forum?
● What is Service Availability?
● Simple Use Case: Web server
● The OpenSAF architecture
● Information model & model-based management
● New features in OpenSAF 4.3 and 4.4
● Roadmap and future
OpenSAF and SA Forum

• SA Forum is a standardisation body formed in 2001 by a group of companies in the communication and computer industry

• The SAF specifications describe APIs for 14 services in a framework for Service Availability

• OpenSAF is the only open source (LGPLv2.1) implementation of the SAF specifications
Service Availability

- The probability that a system provides its service at a (randomly chosen) point in time
- Telecom requirement is often 99.999% (five nines) availability – which allows a total downtime of around five minutes per year
- Planned downtime (hardware and software updates, system maintenance) is included in the total downtime
SAF Services

- AMF - Availability Management Framework
- CLM - Cluster Membership Service
- IMM - Information Model Management Service
- LOG - Log Service
- NTF - Notification Service

Optional services include: CKPT (checkpoint), EVT (event), MSG (message queue), SMF (software management), ...
The SAF APIs

- C language APIs for applications running in the framework
- Bindings for Java and Python exist
- Backwards compatibility ensured when new API versions are introduced
- Applications don't have to use C APIs – for simple use cases, a shell script (resource agent) can be sufficient
OpenSAF and Availability Spectrum

- 99%
- 99.9%
- 99.99%
- 99.999%
- 99.9999%

Some of the Application Domain segments

- OpenSAF
- Other MW (Proprietary, Commercial, Open-source), Typically domain driven

Integration support

Characteristics
Simple Use Case: Web server

- Requirements
  - Static content => No shared disk needed
  - Migrating IP address
  - Lifecycle management
  - Health Monitoring
  - No altering of the code
  - Cold standby
Simple Use Case: Web server

- **Suggested Solution**
  - Wrapper component for httpd interacts with AMF
  - Use httpd init script for lifecycle and health checks
  - PID supervision
  - IP part is non-saf aware component
OpenSAF Clusters

- Min: 1 node
- Max: >100 nodes
- Nodes can have different processor architectures (x86, ARM, PPC, ...)
- During upgrade, nodes can run different versions of OpenSAF (and applications)
OpenSAF Architecture

System Controller
- Server
- Director
- Node Director
- Application

Payload Node
- Node Director
- Application

SAF API (though agent library)
OpenSAF internal protocol
Information Model

Northbound Interface

Information Model Management Service (IMM)

Object tree

Object Manager

Object Implementer

Objects support:
- Configuration data
- Runtime data
- Administrative operations (FEVS safe communication)
Three different ways to configure the persistent backend:

- No PBE – Initial configuration is reloaded at cluster restart
- 1 PBE – Shared disk is used
- 2 PBEs* – Each SC has a local copy of the PBE

* 2 PBEs is a new feature in OpenSAF 4.4
Information Model Notifications

Northbound Interface

Send notifications

Information Model Change Notifier (IMCN)*

- Create/delete objects
- Modify config attributes
- Modify cached runtime attributes

IMM

Object with notifiable attributes

Object Manager

Object Implementer

* IMCN is a new feature in OpenSAF 4.3
SAF API Trace* Using LTTng

- **LTTng consumerd**
- **liblttng-ust**
- **Application process**
- **libSaAmf**

**Benefits of LTTng:**
- Almost zero overhead when tracing is disabled
- Lockless algorithms and per-CPU data structures

*SAF API Trace is a new feature in OpenSAF 4.4*
Container* Components

Java VM
  Java component 1
  Java component 2

Linux Container
  Linux component 1
  Linux component 2

VirtualBox VM
  Linux component 3
  Linux component 4

Start / Stop components
SA-aware
Health check
Start / Stop components

* Container / contained components is on the wish list for future releases
OpenSAF 4.4

- IMM two persistent backends
- IMM reference integrity checks
- SAF API trace using LTTng
- OpenSAF unit files for systemd
- LOG support systems without shared disk
- CKPT callback when active replica is updated

OpenSAF 4.5 and beyond

- AMF container / contained components
- AMF component readiness state
- CLM Enhanced cluster management
- NAM Implement SAF Naming Service
- Utilise Linux control groups
- Man pages (OpenSAF tools & SAF APIs)
- Complete Java SAF API bindings
- “Pythonic” Python SAF API bindings
- C++ SAF API bindings