



Developing Agents for Open HA Cluster

Prasad Dharmavaram
Sun Microsystems.

Agenda

Introduction

Making applications HA on Open HA Cluster

Application characteristics

SC Resource management model

Agent development methods

GDS coding template

Introduction

- Agent is the glue between an application and Open HA Cluster framework.
- Open HA Cluster has an extensive portfolio of agents
- APIs and tools available for custom Agents

Making Applications HA on Open HA Cluster

- 1) Study the application characteristics
- 2) Understand the Open HA Cluster Resource Management framework.
- 3) Familiarize with various tools available for agent development on Open HA Cluster
- 4) Pick a tool and develop an agent
- 5) Install and test on a cluster using SCATE/DATS + any additional application specific manual tests
- 6) Deploy.

Making Applications HA on Open HA Cluster

-
-
-
- **Step 1** : Study the application characteristics

Application Characteristics – Failover Services

- Crash tolerance
 - Be able to restart correctly after an unclean shutdown
 - Sometime requires cleanup of lock/pid files
- Independence from physical node hostname
 - THAT changes with a failover
 - Should be able to co-exist with multi-homed hosts
- Multi-hosted data
 - Application should not hard code data/config path



Application Characteristics – Scalable Services

- All characteristics that apply to failover services plus the following
 - The application must have the ability to run multiple instances, all operating on the same application data in the cluster file system.
 - The application must provide data consistency for simultaneous access from multiple nodes.
 - The application must implement sufficient locking with a globally visible mechanism, such as the cluster file system.

Application Characteristics

- Understand the application
 - Installation – config, data, binaries
 - Environment variables
 - How to start the application
 - How to stop the application
 - How to validate the application
 - How to monitor the application

•

Making Applications HA on Open HA Cluster

-
-
-
- **Step 2** : Understand the Open HA Cluster Resource Management framework.

Resource Management Model

- Key concepts
 - > Resource Type (RT): Is a representation of an HA entity
 - Example: SUNW.apache, SUNW.nfs
 - > Resource: Is a specific instance of a RT
 - Example: apache resource, nfs resource
 - > Resource Group(RG): Is a collection of resources
 - Example: A RG containing
 - A failover filesystem
 - A failover IP Address (aka LogicalHostname resource)
 - A failover apache http server

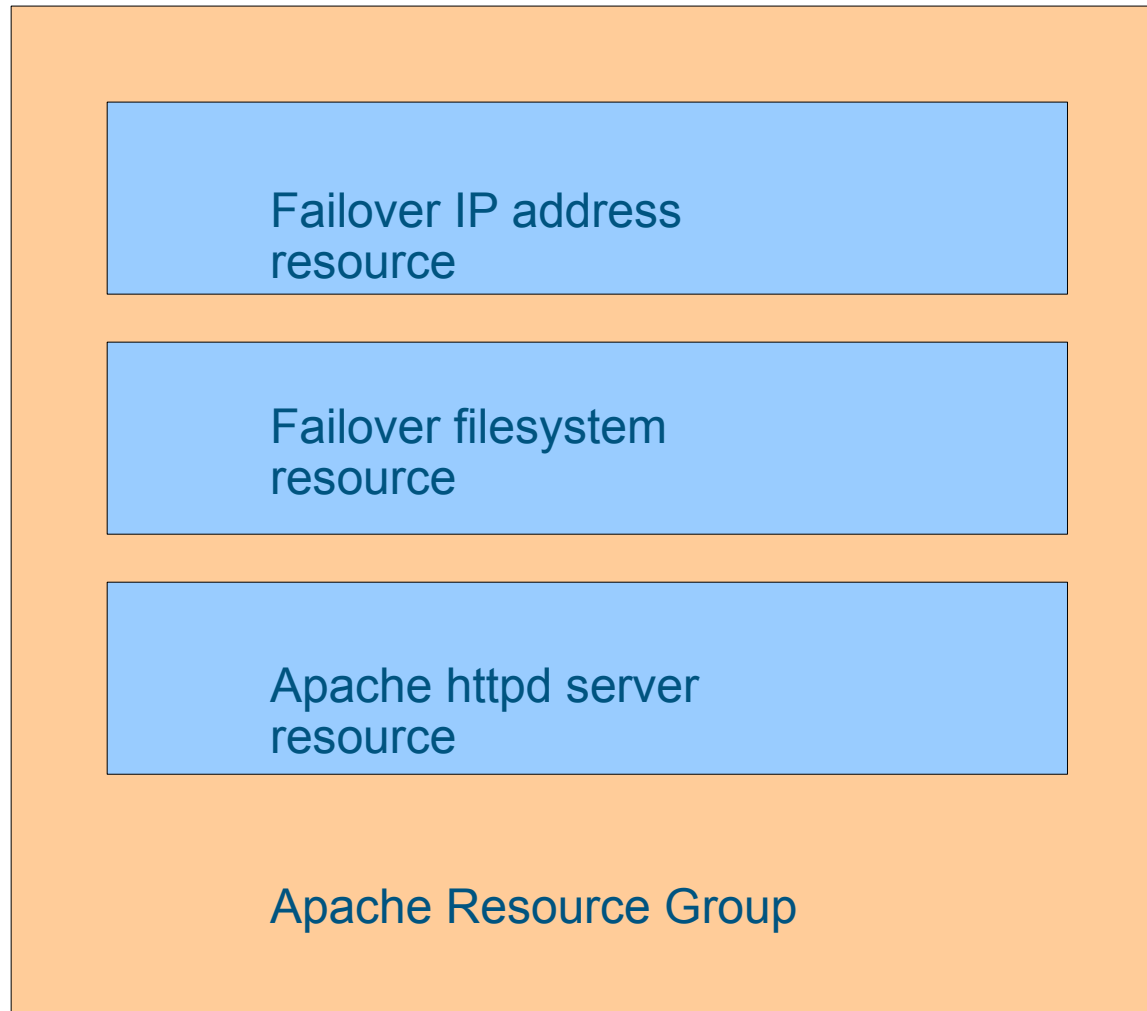
•

•

Resource Management Model

- Key Concepts contd..
 - Resources can have dependencies between them
 - *Facilitates proper startup/shutdown*
 - *Dependencies can have various flavors such as strong/weak/restart*
 - *Works across different cluster nodes*
 - Resource Groups can have affinities between them
 - *Positive or Negative affinity between resource groups*
 - *Various flavors – weak positive, strong positive, strong positive with failover delegation, weak negative, strong negative.*
 - Implemented by the Resource Group manager (RGM)

Example of a Failover RG



Making Applications HA on Open HA Cluster

-
-
-
- **Step 3** : Familiarize with various tools available for agent development on Open HA Cluster

Tools available for Agent development

- Develop agent from scratch using DSDL/SCHA SPI
- SCDS builder using C
- SCDS builder using KSH
- Generic Data Service (GDS)
 - Use SCDS builder
 - Use GDS directly and create a resource
 - Use GDS coding template

Available API for Agent Development

- SCHA API
 - *scha_resource_get(1HA)*
 - *scha_control(1HA)*
 - *scha_resource_serstatus(1HA)*
 - *Available in C and CLI form*
- PMF API
 - Process control
- Run commands under a timeout
 - `hatimerun()`

Available API for Agent Development


- Data Service Development Library (DSDL)
 - *Brings together SCHA, PMF and hatimerun*
 - *Provides integrated fault monitoring*
 - *Available in C only*
- *More details : Sun Cluster Data Services Developer's Guide*
- <http://opensolaris.org/os/community/ha-clusters/ohac/Documentation/Agents/writingagents/>

Agent Builder

- GUI based code generation tool
- You specify start/stop/validate and probe scripts
 - Only start is mandatory
 - Uses PMF to monitor applications if no probe is supplied
 - Reliable application shutdown
 - Can generate Agent in ksh, C and GDS
 - Creates a Solaris package for the Agent

Agent Builder for C/KSH/GDS

File Edit



Vendor Name: DEVELOP

Application Name: apache

RT Version: 6

Working Directory: /home/ashu/develop [Browse ...](#)

Scalable Failover Network Aware

Type of the generated source for the Resource Type C ksh GDS

Step 1 of 2: [Create](#) [<<Previous](#) [Next>>](#) [Exit](#)

Output Log

```
Creating a failover resource type.  
Creating the rtconfig file ...done.  
Cloning and modifying Makefile ...done.  
Cloning and modifying README.apache ...done.  
Cloning and modifying prototype ...done.
```

Agent Builder

File Edit



Start Command (or file):	<input type="text" value="/usr/apache2/bin/apachectl start"/>	<input type="button" value="Browse ..."/>	Timeout (in Secs):	<input type="text" value="300"/>
Stop Command (optional):	<input type="text"/>	<input type="button" value="Browse ..."/>	Timeout (in Secs):	<input type="text" value="300"/>
Validate Command (optional):	<input type="text" value="/usr/apache2/bin/apachectl configtest"/>	<input type="button" value="Browse ..."/>	Timeout (in Secs):	<input type="text" value="300"/>
Probe Command (optional):	<input type="text" value="/usr/bin/wget -o /dev/null -O /dev/null http://\$hostnames/index.ht"/>	<input type="button" value="Browse ..."/>	Timeout (in Secs):	<input type="text" value="30"/>

Step 2 of 2:

Output Log

```
Creating a failover resource type.  
Creating the rtconfig file ...done.  
Cloning and modifying Makefile ...done.  
Cloning and modifying README.apache ...done.  
Cloning and modifying prototype ...done.
```

Agent Builder

- SCDS Builder based (C & KSH)
- Good for simple applications
- Hard coded start/stop/probe/validate values
- Not very easy to provided extension properties
- Template not available for coding non PMF start/stop
- Template code not available for making Apps Failover Zone aware
- Should understand and in some cases know SCDS API
- Template code not available for creating SMF services
- Difficult to test the start/stop/validate/probe without registering the resource

Generic Data Service

- Layered on top of DSDL
- Create Resource directly using the SUNW.gds RT
- or
- Use Agent Builder to create a package with utility scripts
- e.g:
 - *clrt register SUNW.gds*
 - *clrs create -g tomcat-rg -t SUNW.gds -p Start_command=/global/develop/agent-dev/scripts/start-tc -p Stop_command=/global/develop/agent-dev/scripts/stop-tc -p Probe_command=/global/develop/agent-dev/scripts/probe-tc -p Port_list=8080/tcp -p Network_resources_used=fang-1 tomcat-rs*

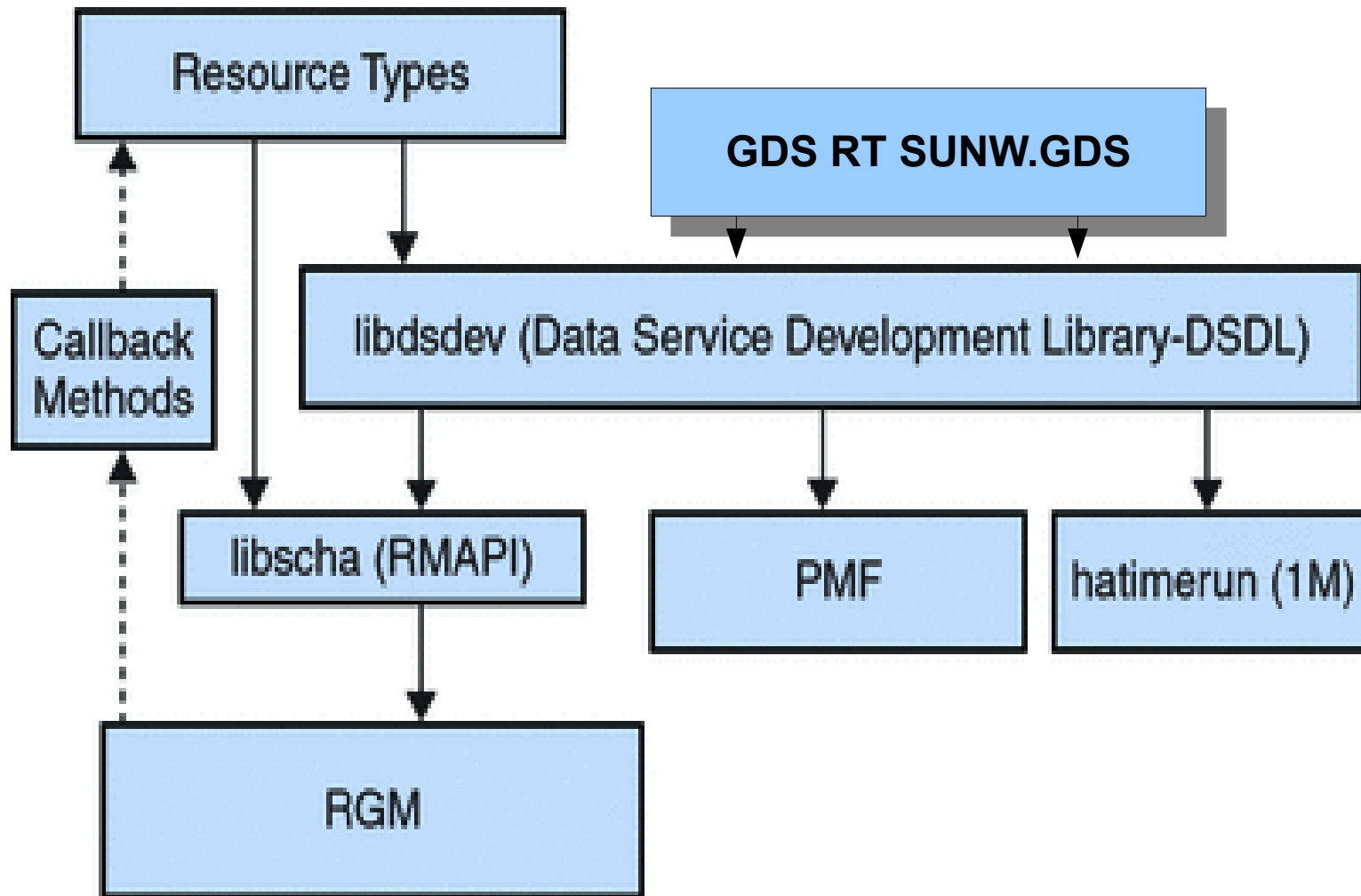
Generic Data Service

- No coding against cluster framework
- Less to code, so less bugs
- Inherits future GDS improvements automatically
- No need to know SCDS or SCHA API
- Scripts can be tested before registering
- Deal with application and not the framework
- Easy to debug on customer systems

•

•

Components interaction diagram



GDS Coding Template

- Use SUNW.gds with the GDS coding template to overcome some limitations with simple GDS
- Download the coding template from
- <http://opensolaris.org/os/community/ha-clusters/ohac/GDS-template/>
- All the GDS based agents in Open HA Cluster use it!

GDS Coding Template

- Easy to use and to package
- Easy to Debug
- Easy to register
- Disabling of PMF integrated
- Modular extendable function library
- Ability to use the scripts for multiple instances
- Ability to add extension parameters
- Ability to use SMF/Failover Zone aware

Making Applications HA on Open HA Cluster

-
-
-
- **Step 4** : Pick a Tool and develop the Agent

Making Applications HA on Open HA Cluster

-
-
-
- **Step 5** : Install and test on a cluster using SCATE/DATS + any additional application specific manual tests

Making Applications HA on Open HA Cluster

-
-
-
- **Step 6 : Deploy!**



Thank You

Prasad Dharmavaram

31st May 2009, San Francisco, California, USA