ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

High Availability Application platform

Mr. Syed Shareef Ali
M.Tech Final Year,
Department of Electronics and
Communication Engineering,
Global Institute of Engineering and Technology,
Hyderabad.

Mr. G. Ahmed Zeeshan
Associate Professor,
Department of Electronics and
Communication Engineering,
Global Institute of Engineering and Technology,
Hyderabad.

Abstract:

The high availability of service with the five nines probability is absolute requirements of the carrier grade network platform. The service availability emphasizes on both hardware and software fault tolerance for the mission critical applications such as air craft and nuclear reactors. Applications high availability is managed using the Availability Management Framework which is a middleware service. The application running on top of AMF is a logical organization of hardware and software resources to provide and protect services. Resources, namely components, are grouped into logical entities such as service units and service groups and are set together at configuration time to provide and protect services represented as component service instances and service instances.

The assignment of component service instances and service instances to components and service units, respectively, is performed at run time by the high availability middleware. Using high availability middleware service, the application is modeled to run as active and standby. All service requests initially will be handled by active instance. When an active instance becomes faulty, standby instance will become active instance and would continue performing the job requests. This modeling does the fault detection, error recoveries etc.. to make sure application service is continued with no downtime.

Introduction:

The Availability Management Framework (sometimes also called the AMF Framework or simply the Framework) is the software entity that enables service availability by coordinating other software entities within cluster. The Availability Management Framework provides a view of one logical cluster that consists of a number of cluster nodes. These nodes host various resources in a distributed computing environment. The Availability Management Framework provides a set of APIs to enable highly available applications. In addition to component registration and life cycle management, it includes functions for error reporting and health monitoring. The Availability Management Framework also assigns active or standby workloads to the components of an application as a function of component state and system configuration. The Availability Management Framework configuration allows prioritization of resources and provides for a variety of redundancy models. The Availability Management Framework also provides APIs for components to track the assignment of work or so-called component service instances among the set of components protecting the same component service instance.

To provide Service Availability, the system must be able to switchover from one component to an alternative component, without disrupting the service provided to the user or an application. If the component became faulty, the switchover is called. The switchover is also be required because the component must be removed from service for maintenance, or because parts of the system are overloaded and it is necessary to reassign resources and redistribute the load.Application service availability can be achieved by a preplanned controlled switchover. Broadly, a component quiesced by stopping its service to each user in an orderly manner and recording its state for that user in a checkpoint. The checkpoint is passed to another component which resumes service to the user from the checkpoint. For a fail over caused by a faulty component, the mechanisms try to achieve the same effect with disturbance to the service provided to the user, and in many cases they achieve it. However, for a fail over, no disturbance cannot be guaranteed because the mechanisms have no advance warning of the fault and cannot quiesce in an orderly manner.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In some cases, the component will have performed part of an action before it becomes faulty, and the part performed action is visible to the user. If it is essential that there is absolutely no disturbance to the user (e.g., if you are flying an airplane), more rigorous and more expensive mechanisms are available. In such circumstances, you will also need to expend considerably more effort to remove all (rather than only most) of the bugs from your software.

Description:

Today's market conditions have greatly reduced internal development resources; this building block consists of hardware and software stack to provide the service availability. The various layers of the solution stack as simplistically seen below. Building blocks must be integrated into a complete applications-ready solution that delivers at least a five nines level of availability. In addition, this integration must be completed in a fraction of the time traditionally experienced. This presents major challenges to all parties involved: Each building block must provide fault management capabilities (monitoring, fault detection and diagnosis, fault isolation, fault recovery, component replacement). Each building block must communicate configuration, availability status, faults, diagnostic test results, fault recovery actions, etc., to management middle ware. Integrated building blocks must be able to manage and recover from faults without impact to service being performed by the platform.

The system must provide the ability to monitor and control system hardware and communicate such to system and network management middle ware. The system must provide the ability to test and validate integrated building blocks to ensure that fault management actions and communications integrate and function properly and within the time constraints prescribed by the service or application. Assuming that each hardware and software building block provider will implement some level of fault management capabilities, the most significant challenge becomes interfacing many different hardware and software building blocks from many different providers with different implementation methodologies and architectures. The HPI is designed to solve this problem, and hence simplify the integration of hardware and software building blocks.

To make the application highly available on the middleware platform, the Application Management Framework has the static way of doing the AMF configuration to do the logical organization of the hardware & software resources. Static configuration will be in the form of XML. This XML contains all the logical representation of the application for ex:- the cluster size, on which all nodes of the cluster the application has to be spawned, the application executable name, the application executable location, the type of redundancy model used etc...Static approach doesn't support runtime creation/addition/modification of logical entities.

After bringing up the cluster using static XML, run time configuration support should be available to the users of highly available application middleware platform so that features like dynamic addition of the nodes to cluster or dynamic removal of nodes from the cluster, changing the application executable path etc... can be achieved using the dynamic or run time configuration of all the logical entities without disturbing the existing application cluster.

Run time approach has following advantages:

- Users of high available application middleware platform don't have to stop the cluster in order to add or remove nodes dynamically without affecting the deployed applications.

- The dynamic configuration approach enables cluster scalability.

- Enhances the existing functionality of the middleware platform. Hence would play a vital role in business market.

Block Diagram:

Applications Service
Middle ware
Operating Systems
Drivers
Chassis
I/O and Compute Boards
Processor And Chip Sets

Application Area:

- Telecommunications Industry
- Stock Exchanges
- Mission critical applications
- Reliable transaction systems
- Enterprise application management systems
- Carrier grade applications.

Conclusion:

The service availability is the most essential part of the high availability system applications. The framework developed right from the hardware and software stacks integrated, gives a robust fault tolerance system solution to the mission critical applications. The state management of the service or application gives the user a much needed fault recovery and service continuity during the system upgrades and maintenance.

References:

[1] www.opensaf.org/

[2] SAF_Interface_Specification_1.0.pdf

[3] http://en.wikipedia.org/

[4] SAI-AIS-B.01.01 specification