

Scalable Dynamic Clustering

Dynamic Clustering and Management
White Paper

Mohamed.abdelaziz@Sun.COM

Shreedhar.Ganapathy@Sun.COM

July 2007

Abstract

In today's information technology world, fault tolerance has become an expected system characteristic, as demands on such systems, not only requires of the availability of data, but also the efficiency of such systems. By clustering a set of servers, with minimal, or no configuration, through a dynamic discovery protocol, a compute cluster can be formed to increase compute power, availability, security, and geographical distribution.

Table of Contents

Introduction..... 1

Server Identification.....2

Cluster Identification.....2

Server Clustering.....2

Monitoring.....2

Connection types.....3

Unicast channels.....3

Multicast channels.....3

Deployment.....3

Requirements.....4

Summary.....4

Links.....4

Master Node :

- A dynamically elected node, which takes on the role of the leader of the cluster.
- Secondary are automatically selected based on a sort order of the cluster view asserted by the master node.

Inner cluster addressing is done using a node name, rather than an IP address, thus simplifying connection establishment, while allowing network mobility, without requiring any configuration changes.

Executive Summary

In today's information technology world, fault tolerance has become an expected system characteristic, as demands on such systems, not only requires the availability of data, but also the efficiency of such systems.

By clustering a set of servers, with minimal, or no configuration, through a dynamic discovery protocol, a compute cluster can be formed to increase compute power, availability, security, and geographical distribution.

As a fault tolerance strategy, the Shoal clustering framework devises a self organization protocol, allowing nodes to autonomously elect a master for the cluster (based on discovery views), as well as candidates in the case of master failure. This allows for dynamic deployment of clusters as well as expanding/shrinking an existing cluster.

Introduction

Shoal, the cluster management framework, provides the foundation for network configuration and dynamic and autonomous cluster formation. The goals of the framework are:

- Provide dynamic network configuration that support seamless discovery and cluster formation
- Uniquely identify and virtualize node, cluster, application addressing
- Provide autonomous recovery from network and application failures
- Provide a foundation topology for fault tolerant applications
- Provide a simple, easy to use API for applications to consume clustering services

The cluster management framework supports applications by providing:

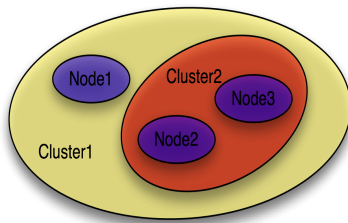
- Access to current cluster topology
- Flexible schema that allows definition of and access to node configuration
- Services which determine health state of cluster members and organization
- APIs that allow access to cluster members and configuration specifics

Server Advertisement :

- XML Document
- Defines instance name encoded ID
- SW/HW Configuration meta-data

Cluster Identification :

- Defines cluster name encoded ID
- Traffic scoped to cluster
- An instance may be a member of multiple clusters



Illustrates multi-cluster participation

Server Identification

A server is described by an XML document containing data such as a name, SHA1 hash name encoded ID, physical transports, software and hardware configuration. In addition to the default elements of server advertisement, applications can attach additional meta-data allowing for extensibility, i.e. system load maybe exchanged in such advertisement allowing for dynamic load balancing. These advertisements are used as the identifier element in all protocol messages, thus provide the most up to date representation of a server, at any one given point (e.g. mobility is no longer an issue, as the proper mapping is always available).

Cluster Identification

Clusters are identified by a given name (uniqueness of names is outside the scope of the framework, a naming service can be used to ensure uniqueness in a given network), where it is encoded into a SHA1 hash encoded 128 bit identifier. This identifier is then used to scope traffic on the network and avoid cross cluster communication.

Server Clustering

Instead of relying on a central server to bootstrap a cluster, each server utilizes a discovery mechanism, by which, servers form a representation of a network which is later used in determining the organization of the network. The protocol utilizes the server identifiers in a sorted order to elect master server for the cluster. The same sort order is also utilized for independent election/promotion of a secondary master in case of a master failure, as well as when coalescing of a disjoint cluster. It is key to note that this greatly simplifies conflict resolution, fail-over, and most importantly it is performed autonomously without message exchange, with the exception of cluster change event assertions.

Monitoring

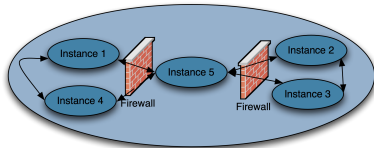
The Shoal cluster management framework defines a protocol by which the state of the cluster membership can be maintained. In order to reduce network traffic and resources, monitoring relies on a heartbeat mechanism where each server is expected to broadcast an alive message, at predefined period, to the cluster. Cluster members can then independently detect troubled members, then proceed according to their standing in the cluster, where a master node is expected to verify (through directed messages) the health state of a suspected server, followed by an assertion of the cluster view state, whereas ordinary members follow the same process in the case of suspect master server.

Communication channels:

- Mixed channels fully supported
- Self healing. i.e., on a multi-homed node, a network interface may suffer a failure, the virtual channel is automatically repaired

Cross Internet deployment:

- ws for geographical distribution of a cluster
- Close to zero-config deployment
- Requires no application specific configuration
- Supports network mobility



Connection types

There are two of types logical message based communication channels exposed to applications, unicast and multicast, these channels can simply be a direct connection between two servers, or rely on IP multicast in the multicast channel case, or they can be a composite of connections.

Unicast channels

- Direct connections between two servers
- Datagram transport
- Composite connections (in the case firewalled networks, where message relays are required)

Multicast channels

- Can be based on IP multicast
- Virtual multicast (A combination of direct connections and IP multicast)

Unlike other group communication providers, the Shoal cluster management framework only requires a server or channel name to establish a communication channel. It is also key to note that the framework does not impose any limitation of physical transport types (unicast/multicast). Since channels are established by a logical name, it provides for great flexibility of the location of the channel endpoints, a server may change physical addresses (with minor interruption manifested through slow message delivery, until the route is repaired), or a channel endpoint can reside on multiple servers, providing for inherently fault tolerant channels.

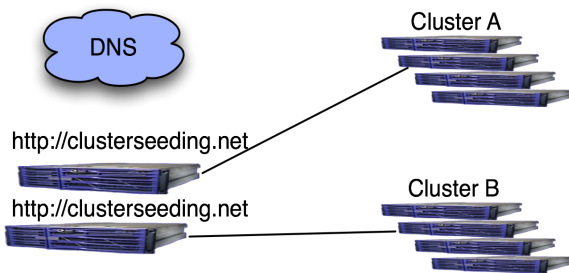
Deployment

The clustering management framework relies on a dynamic cluster formation protocol which relies on all available system transports defined (TCP/IP, UDP, IGMP), as well as virtual multicast channels which rely on composite interface connections. These virtual channels are constructed dynamically and don't require any special setup¹ requirements, as the underlying framework will construct such channel using the most optimum path.

¹ with the exception of a cross sub-net or Internet deployment, where a seeding node(s) must be configured

Requirements

- A deployment where IP multicast is supported, the framework does not require any setup.
- A cross sub-net deployment where IP multicast is not supported, a single bootstrapping node (seed) or more, for load distribution and availability) within the cluster must be configured to take on the role propagating messages to the cluster. This configuration is typically done prior to startup, or maybe discovered at runtime. See bootstrapping for more details
- A cross Internet deployment carries the same requirement as cross sub-net, it also may require a message relaying node in the case where cluster members are not directly reachable to one another.



Bootstrapping

Bootstrapping can be achieved using static addresses, which is fine for a lot of deployments, however it becomes a cumbersome task if such addresses happen to change. This can be avoided through a deployment of a web service. This service provides the following functionalities :

- Act as a seed, when no other seed is available
- Publication service for nodes advertising their ability to act as cluster bootstrapping seeds.
- Seed discovery

To keep configuration to a minimum, such a service is mapped through DNS, and possibly hosted by multiple addresses for high availability. This level of indirection minimizes configuration to a DNS record(s), and avoids having to perform the configuration on cluster instances.

It is key to note that this bootstrapping configuration is only required at cluster deployment, there's no configuration requirement by the application running within the clustering framework.

Summary

The clustering management framework is an open source project under shoal.dev.java.net, currently utilized in GlassFish V2 to provide scalable clustering capabilities, a distributed state cache, in memory session high availability. In addition the framework is being utilized by JonAS (Java Open Application Server) for cluster formation and message exchange.

Links

<http://shoal.dev.java.net>

http://weblogs.java.net/blog/sdo/archive/2007/07/sjsas_91_glassf.html



Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN (9786) Web sun.com