Comparison of Protocols for Reliable Server Pooling
<draft-ietf-rserpool-comp-05.txt>

Status of this Memo

This document is an Internet-Draft and is subject to all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Distribution of this memo is unlimited.

Copyright (C) The Internet Society 2002. All Rights Reserved.

Abstract

This document compares protocols that may be applicable for the Reliable Server Pooling problem space. This document discusses the usage and applicability of these protocols for the Reliable Server Pooling architecture.

Abstract

1 Introduction
1.1 Overview
1.2 Terminology
2 Relation to Other Solutions
2.1 CORBA
2.2 DNS
2.2.1 Requirements
2.2.2 Technical Issues
2.2.3 Name/Address Resolution

Internet-Draft Comparison of Protocols for RSerPool June 30, 2002
1 Introduction

1.1 Overview

In creating a solution to provide reliable server pools [RSER-ARCH], there are a number of existing protocols, which appear to have similar properties as to what RSerPool is trying to accomplish. This document discusses the applicability of these protocols in meeting the requirements of Reliable Server Pooling [RFC3237].

This study does not intend to be complete, rather intends to highlight several protocols which working group members have suggested.

1.2 Terminology

This document uses the following terms:

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation Scope</td>
<td>The part of the network visible to pool users by a specific instance of the reliable server pooling protocols.</td>
</tr>
<tr>
<td>Pool</td>
<td>A collection of servers providing the same application functionality. Also called a Server Pool.</td>
</tr>
<tr>
<td>Pool Handle</td>
<td>A logical pointer to a pool. Each server pool will be identifiable in the operation scope of the system by a unique pool handle or &quot;name&quot;. Also called a Pool Name.</td>
</tr>
<tr>
<td>Pool Element</td>
<td>A server entity having registered to a pool.</td>
</tr>
<tr>
<td>Pool User</td>
<td>A server pool user.</td>
</tr>
<tr>
<td>Pool Element Handle</td>
<td>A logical pointer to a particular pool element in a pool, consisting of the name of the pool and a destination transport address of the pool element. Also called an Endpoint Handle.</td>
</tr>
<tr>
<td>Name Space</td>
<td>A cohesive structure of pool names and relations that may be queried by an internal or external agent.</td>
</tr>
<tr>
<td>Name Server</td>
<td>Entity which the responsible for managing and maintaining the name space within the RSerPool operation scope.</td>
</tr>
</tbody>
</table>

1.3 Abbreviations

DA: Directory Agent in SLP.
Relation to Other Solutions

This section is intended to discuss the applicability of some existing solutions with regards to Reliable Server Pooling requirements [RFC3237]. The protocols discussed have been suggested as possibly overlapping with the problems space of RSerPool.

2.1 CORBA

Often referred to as a Distributed Processing Environment (DPE), CORBA was mainly designed to provide location transparency for distributed applications. CORBA’s distribution model encourages an object-based view, i.e., each communication endpoint is normally an object.

CORBA has a number of variants, such as fault-tolerant CORBA, Real-time CORBA, etc. CORBA has been used in a number of situations, for example, Real-time CORBA has been used in fighter aircraft and weapon systems. Additionally, CORBA has been implemented in a wide range of devices, from attack submarines to Palm Pilots - the MICO open source ORB has been ported to the Palm Pilot, and the client-only application is 45 KB in size.

Currently, the applicability of CORBA for reliable server pooling is unclear, and interaction with other Internet protocols, such as AAA, IPsec and IPv6 may be problematic.

2.2 DNS

This section will answer the question why DNS is not appropriate as the sole solution for RSerPool. In addition, it highlights specific technical differences between RSerPool and DNS.

During the 49th IETF December 13, 2000 plenary meeting Randy Bush presented a talk entitled "The DNS Today: Are we overloading the Saddlebags on an Old Horse?" This talk underlined the issue that DNS
is currently overloaded with extraneous tasks and has the potential to break down entirely due to a growing number of feature enhancements.

One requirement to any solution proposed by RSerPool would be to avoid any additional requirements for DNS in order to support Reliable Server Pooling. Interworking between DNS and RSerPool will be considered so that additional burdens to DNS will not be added.

2.2.1 Requirements

Any solution for RSerPool should meet certain requirements [RFC3237]. These requirements are related to DNS.

"Servers should be able to register to (become PEs) and deregister from a server pool transparently without an interruption in service.

The RSerPool mechanisms must be able to support different server selection mechanisms. These are called server pool policies.

The RSerPool architecture must be able to detect server failure quickly and be able to perform failover without service interruption.

Server pools are identified by pool handles. These pool handles are only valid inside the operation scope. Interoperability between different namespaces has to be provided by other mechanisms."

2.2.2 Technical Issues

This section discusses the relationship between DNS and the requirements for RserPool.

2.2.2.1 Host Resolver Problems

A major issue that prevents the use of DNS as part of the RSerPool solution the issue is the architecture of host resolvers. These are stub resolvers - which means that they require their local DNS servers to do recursion for them.

In turn, this implies that setting TTL low or 0 will dramatically increase the load not only on the authoritative DNS servers - but also on these third party servers.

A secondary effect of this is that the authoritative DNS will not know the IP address of the DNS client - only the IP address of the local DNS. This affects the ability to do global load balancing correctly.
There is no way to get around these issues unless you require all hosts to be full resolvers. Putting full resolvers on newer hosts isn’t sufficient because the issues would still exist for all the legacy systems, which will form the bulk of the host population for years to come. The solution is not to use third party servers.

Additionally, if the client can contact the server directly, then the server knows the real IP address of the client. Since there is no third party involved, the caching TTL can be set as low as desired (even to zero). That will increase load on the server, but nowhere else.

Finally, DNS is based on a recursion. This recursion presents certain difficulties for RSerPool. Even if a host resolver is not a stub resolver, it has to go to another full resolver where 2 possibilities exists: either the mapping name-IP address is found or it has to do another recursive resolution of the name, staring from that intermediate resolver, until there is a cache hit in one of the intermediate resolvers or it is resolved by its root resolver (or home DNS server).

This process of recursion means that there is no end-to-end communication between the host and its server where the name-to-IP mapping resides. That also means that a lot of timers are running in intermediate systems. Any updating of the transient status of the pool element or of the pool may need to be propagated through the DNS.

2.2.2.2 Dynamic Registration

Registration / de-registration of servers is needed. It can be done with DNS by NOTIFY/IXFR. However, frequent updates and replication are incompatible. This is not a DNS problem per se, but it has an effect on DNS as it is deployed.

RSerPool MUST allow software server entities to register themselves with a name server dynamically. They can also de-register themselves for purposes of preventative maintenance or can be de-registered by a name server that believes the server entity is no longer operational. This is a dynamic approach, which is coordinated through servers in the pool and among RSerPool name servers.

2.2.2.3 Load Balancing

RFC 2782 itself points out some of the limitations of using DNS SRV for load balancing between servers.

Weight is only intended for static, not dynamic, server selection. Using SRV weight for dynamic server selection would require assigning unreasonably short TTLs to the SRV RRs, which would limit the usefulness of the DNS caching mechanism,
thus increasing overall network load and decreasing overall reliability.

Based on this, DNS can only really support stochastic load balancing, redirecting clients to servers randomly as various caches in various resolvers expire at random (although small) intervals. DNS offers excellent network scalability but poor control over load balance.

As mentioned previously, the issue of doing DNS-based dynamic load balancing on short time scales will have impacts on third parties, due to the presence of stub resolvers.

2.2.2.4 Heartbeating & Status Monitoring

DNS does not incorporate an application layer heartbeat. Heartbeating would dramatically boost traffic levels, and given the unavoidable third party dependencies of DNS, the resulting loading would be unacceptable. It is passive in the sense that it does not monitor or store information on the state of the host such as whether the host is up or down or what kind of load it is currently experiencing.

RSerPool SHOULD monitor the state of each server entity on various hosts on a continual basis and can collect several state variables including up/down state and current load. If a server is no longer operational, eventually it will be dropped from the list of available servers maintained by the name server, so that subsequent application name queries will not resolve to this server address.

2.2.3 Name/Address Resolution

The technical requirement for DNS name/address resolution is that given a name, find a host associated with this name and return its IP address(es). In other words, in DNS we have the following mapping:

Name       a host machine

Address(es)    IP address(es) to reach a (hardware) host machine

The technical requirement for RSerPool name/address resolution is that given a name (or pool handle), find a server pool associated with this name and return a list of transport addresses (i.e., IP addresses plus port numbers) for reaching a set of currently operational servers inside the pool. In other words, in RSerPool we have the following mapping:

Name       a handle to a server pool, which is often distributed across multiple host machines
2.3 Service Location Protocol (SLP)

2.3.1 Introduction

SLP is comprised of three components: User Agents (UA), Service Agents (SA) and Directory Agents (DA). User agents work on the user's behalf to contact a service. The UA retrieves service information from service agents or directory agents. A service agent works on behalf of one or more services to advertise services. A directory agent collects service advertisements.

The directory agent of SLP simply acts as a cache and is passive in this regard. The directory agent is optional and SLP can function without it. It is incumbent upon the servers to update the cache as necessary by re-registering. The directory server is not required in small networks as the user agents can contact service agents directly using multicast. Unicast queries to SAs are possible subsequent to the UA having discovered them. User agents are encouraged to locate a directory at regular intervals if they can't find one initially, otherwise they can detect DAs by listening passively for DA advertisements.

2.3.2 What to Use

Figure 1 shows how SLP might be realized to provide ENR services:

```
+-------------+                              +---------+
| APPLICATION |                              | SERVICE |
+-------------+                              +---------+
| ASP/RSERPOOL API | <--------------------> | ASP/RSERPOOL API|
+-------------+                              +---------+
| SLP UA | <----> | SLP DA | <----> | SLP SA | |
| SCTP |UDP|        | SCTP |UDP|        |UDP| SCTP |
+-------------+                              +---------+
```

Figure 1: RSERPOOL entities employing SLP for ENR services

Notes:
* Each box constitutes a host (running a PU, PE or ENR server ‘stack’), though one host could support more than one of these functions.

* As far as the Application is concerned, it is using a framework for exchanging messages with services reliably.

* As far as the Service is concerned, it is making itself available to a reliable server pool by interacting with the framework API.

* The ASAP/RSERPOOL API obtains endpoint name resolution data in a timely and robust manner and uses it to determine how to route PU requests to PEs.

* The ENR service function is performed using SLP. The PU employs a SLP UA to obtain information from a SLP DA.

* The ENR service function is performed using SLP. The PU employs a SLP UA to obtain information from a SLP DA.

* The PE employs a SLP SA to register information with a SLP DA. As the SLP SA is ‘mesh-enhanced,’ it only registers with one DA of this type (as long as it detects that this DA is alive & responsive & returns ‘OK’ results).

* The SLP DA is part of a mesh. It will forward PE state to other DAs in the mesh. For example, it will forward the registrations the SLP SA made on behalf of the PE on right of Figure 1.

* SCTP is used for communication between entities. Multicast UDP is used by SLP entities for active and passive discovery. While the RSERPOOL architecture cannot rely upon multicast mechanisms, it can profit from them when these are present in the network.

SLPv2 will be needed, but SLPv2 alone does not fulfill RSERPOOL update requirements for timeliness. This is achieved through mesh-enhancements to the Service Location Protocol (mSLP) [MSLP].

These enhancements make it possible for SAs to know of only a subset of all DAs. Mesh-enhanced SAs need only forward their registrations to only one mesh-enhanced DA. The mesh takes care of forwarding the message to the other DAs.

2.3.3 Summary

The most fundamental difference between SLP and RSerPool is that SLP is service-oriented while RSerPool is communication-oriented. More specifically, what SLP provides to its user is a mapping function from a name of a service to the location of the service provider, in the form of a URL string. The availability of the service provider is outside of the scope of SLP. How a service is accessable can be
described by the SLP attribute list associated with the service URL. SLP is essentially a discovery protocol, not a transport protocol. Therefore, the granularity of SLP operation is at application service level.

In contrast, RSerPool provides to its user is a mapping function from a communication destination name to a set of routable and reachable transport addresses that leads to a group of distributed software server entities registered under that name that collectively represent the named communication destination. With respect to SLP, this information could be represented in SLP attributes. RserPool, however, also has the responsibility of reliably delivering a user message to one of these server entities.

Currently, mSLP would need changes, for example it was designed to scale to ~10 DAs not ~100 DAs. Additionally, SLP is currently designed to run on top of UDP and TCP. If SCTP support is needed, some additional specification work would be needed.

SLP security makes no attempt to address the confidentiality of data transmitted between SLP agents. To properly address this concern, SLP agents would need to establish secure communication with each other. This would be achieved through the use of IPSec Encapsulating Security Payload.

Server discovery, however, is something which SLP does well, and if used for RserPool, this would be useful.

3 ASAP and ENRP

ASAP [ASAP] and ENRP [ENRP] are being developed the RserPool working group. Even though they are separate protocols, they are designed to work together.

3.1 ASAP

ASAP uses a name-based addressing model which isolates a logical communication endpoint from its IP address(es), thus effectively eliminating the binding between the communication endpoint and its physical IP address(es) which normally constitutes a single point of failure. In addition, ASAP defines each logical communication destination as a pool, providing full transparent support for server-pooling and load sharing. It also allows dynamic system scalability - members of a server pool can be added or removed at any time without interrupting the service.

ASAP is not designed to scale Internet wide. It uses a flat, peer-to-peer addressing model. Other protocols, such as DNS could be used to bridge the pools of servers. It uses a name-based addressing model to logically isolate the communication endpoint from its IP address(es). If multiple endpoints register under a the same name, a server pool is effectively created. ASAP is used to
select one Pool Element, based on a number of criteria, such as load sharing. ASAP monitors the reachability of the Pool Elements in order to provide fault tolerance.

3.2 ENRP

ENRP defines procedures and message formats of a distributed fault-tolerant registry service for storing, bookkeeping, retrieving, and distributing pool operation and membership information. It allows Pool Elements to be dynamically added, updated and removed from service. There are protocol mechanisms for detecting and removing unreachable Pool Elements.

4 Comparison Against Requirements

This section attempts to create a comparison table to compare the protocols which have been suggested as applicable to the RserPool architecture.

<table>
<thead>
<tr>
<th></th>
<th>CORBA</th>
<th>DNS</th>
<th>SLP</th>
<th>ASAP</th>
<th>ENRP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robustness</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Failover Support</td>
<td>Y</td>
<td>P</td>
<td>P</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Communication Model</td>
<td>N</td>
<td>P</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Processing Power</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Support of RserPool</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Unaware Clients</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registering and</td>
<td>N</td>
<td>P</td>
<td>P</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Deregistering</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Naming</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Name Resolution only to Active Elements</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Server Selection Policies</td>
<td>Y</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Timing Requirements and Scaling</td>
<td>P</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Scalability</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Security - General</td>
<td>Y</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Security - Name Space Services</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>
Internet-Draft     Comparison of Protocols for RSerPool     June 30, 2002

----------------------------+--------+-----+-----+------+------+
Y = Yes, meets requirement
P = Partially meets requirement
N = No, does not meet requirement
N/A = Not applicable

5 Security Concerns

This type of non-protocol document does not directly affect the security of the Internet.

6 Acknowledgements

The authors would like to thank Bernard Aboba, Erik Guttman, Matt Holdrege, Lyndon Ong, Christopher Ross, Micheal Tuexen and Werner Vogels for their invaluable comments and suggestions.

7 Normative References


[ENRP] Xie, Q, Stewart, R. R., "Endpoint Name Resolution Protocol (ENRP)", Work in progress.

[MSLP] Zhao, W., "mSLP - Mesh-enhanced Service Location Protocol" Work in progress.


8 Authors’ Addresses

John Loughney  
Nokia Research Center  
PO Box 407  
FIN-00045 Nokia Group  
Finland  
Email: john.loughney@nokia.com

Maureen Stillman  
Nokia  
127 W. State Street  
Ithaca, NY 14850  
USA  
Email: maureen.stillman@nokia.com

Qiaobing Xie  
Motorola, Inc.  
1501 W. Shure Drive, #2309  
Arlington Heights, IL 60004  
USA  
Email: qxie1@email.mot.com

Randall Stewart  
Cisco Systems, Inc.  
24 Burning Bush Trail  
Crystal Lake, IL 60012  
USA  
Email: rrs@cisco.com

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.
The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.