Reliable Server Pooling Working group
Internet-Draft
Expires: September 1, 2003

L. Coene
Siemens
March 3, 2003

Reliable Server pool applicability Statement <draft-coene-rserpool-applic-01.txt>

Status of this Memo

Copyright Notice

Abstract

This document describes the applicability of the reliable server pool architecture and protocols to applications which want to have High avialebility services. This is accomplished by using redundant servers and failover between servers of the same pool in case of server failure. Processing load in a pool may de distributed/shared between the members of the pool according to a certain policy. Also some guidance is given on the choice of underlying transport protocol (and corresponding transport protocol mapping) for transporting application data and Rserpool specific control data.

1. INTRODUCTION

Reliable server pooling provides protocols for providing higly available services. The services are located in pool of redundant servers and if a server fails, another server will take over. The only requirement put on these servers belonging to the pool is that if state is maintained by the server, this state must be transfered to the other server taking over. The mechanism for transfering this state information is NOT part of the Reliable server pooling architecture and/or protocols and must be provided by other protocols.

The goal is to provide server based redundancy. Transport and network level redundancy are handle by the transport and network layer protcols.

The application may choose to distribute its traffic over the servers of the pool conforming to a certain policy.

The application wishing to make use of Rserpool protocols may use different transport layers(such as UDP, TCP and SCTP). However some transport layers may have restrictions build in in the way they might be operating in the Rserpool architecture and its protocols.

1.1 Scope

The scope of this document is to explore the different ways that Reliable server pool protocols can be used in order to provide a higly available service towards applications with different requirements.

1.2 Terminology

The terms are commonly identified in related work and can be found in the Aggregate Server Access Protocol and Endpoint Name Resolution Protocol Common Parameters documentRFC COMM [5].

2. Reliable serverpool

2.1 Architecture

A overview of the reliable server pool architecture is given in the Rserpool architecture document RFC ARCH [2].

The Rserpool architecture is made up of clients(Pool Users - PU) and servers(Pool Elements - PE). Both PU and PE's can be grouped into a pool in which a PE provides a service(File transfer, storage, bank transaction) to a PU. The PU's may try to find out via the endpoint resolution protocol(ENRP) which PE's are active. The PU can set up a communication channel with a particular PE(chosen out of the server pool) by using the Aggregate Server Access Protocol (ASAP) or by using directly any of the transport protcols(UDP/TCP/SCTP/RTP). ASAP may be running on top of UDP, TCP or SCTP.

The minimum mode of using Rserpool is to use only the ENRP for Endpoint name resolution. The PU may setup the client - server communication WITHOUT ASAP, but using present transport protocols(such as UDP, TCP..)

The normal use of Rserpool is to use ENRP for Enpoint name resolution and ASAP for client - server communication. ASAP may be using as underlying transport protocol UDP, TCP or SCTP.

2.2 ASAP/ENRP applicability

2.2.1 Minimal rserpool service

The minimum service provided by Rserpool is the use of ENRP for Endpoint name resolution. The ENRP procol may be running over TCP or SCTP.

- Endpoint name resolution
- no automatic failover from one PE to another, has to be done by the application itself
- bussinesscard or cookie mechanism not possible
- May be used by allready existing applications which do not want to change the interface between PU and PE.
- Only PU-NS and PE-NS communication will use Rserpool protocols

2.2.2 Full Rserpool service

The fullservice provided by Rserpool is the use of ENRP for Endpoint name resolution and the Use of ASAP for PU - PE communication . ENRP may be running over TCP or SCTP while ASAP may be running over TCP, SCTP, UDP or RTP.

- Endpoint name resolution
- automatic failover from one PE to another is transparent for the application itself
- bussinesscard exhange for determining if a PU is a pool or not. It allows the PE to treat the PU's as pool and use Rserpool protocols for it
- cookie mechanism can be used for state transfer between PE's
- May be used by allready existing applications which do not want to change the interface between PU and PE.
- All entities wil use Rspool protocols for communication withs their respective peers

3. Application and Control data Transport

3.1 Rserpool use between 2 pools

Bussinesscards will allow to detect if their peer is part of a pool itself. Both the PU and the PE can be part of their own pools. If the PU or PE would fails, then the businesscard will have informed the respective peer to contact a alternative fellow PE/PU belonging to the pool.

3.2 state sharing via the cookie

Every time a response is send back, a cookie could be send along the response. The cookie is "encrypted" and is stored by the PU, no modification at all it done to the cookie . If a PE fails then the cookie is send to a alternate PE, the PE check if the cookie is valid. The contents of the cookie is only provided and validated by the PE. It can be used for state sharing between the PE.

4. Transport protocols used by ENRP & ASAP

4.1 ASAP on top of UDP

UDP is a unreliable message transport delivery protocol, so if a message gets lost due to a changeover of server(or client), then the message will not be retransmitted after changeover has occured. New messages will be sent to alternate server/client within the serverpool.

This service may be of some importance to services where realtime constraints apply.(Example video servers: a few lost message ain't that important as long as the big bulk of messages get through). No conegstion control is done and as such no real measure of the congestion status on the server(or client) is taken into account, thus making loadsharing harder. Only the ENRP server responsible for that particular server pool will have a up to date view of the load distribution in the pool.

4.2 ASAP on top of TCP

TCP provides full reliable delivery with congestion control of the message to its peer node. It provides for a single homed, single stream delivery of a byte stream from or to the server. Change over will retrieve the unsent messages and send them on another TCP connection to a different server of the server pool.

4.3 ASAP on top of SCTP

PR-SCTP is the only know protocol which allows the choice of full, partial or no reliable delivery with congestion control of the message to its peer node. If the no-reliable delivery option is selected of SCTP, then ASAP will function as described in ASAP over UDP and including congestion control.

if multihoming, streams, unsequenced and/or assured delivery are required for the application, then SCTP should be used for ASAP. See SCTP aplicability statement RFC 3257 [9].

5. Issues for Reliable Server pooling

5.1 State transfer accoss the server pool

Rserpool protocols(ENRP and ASAP) do NOT provide any service for transfering state information of a application from one Processing Element(PE) to another.

6. Security considerations

The protocols used in the Reliable server pool architecture only tries to increase the availability of the servers in the network. Rserpool protocols does not contain any protocol mechanisms which are directly related to user message authentication, integrity and confidentiality functions. For such features, it depends on the IPSEC protocols or on Transport Layer Security(TLS) protocols for its own security and on the architecture and/or security features of its user protocols.

A overview of possible treats to Reliable Server pooll protcols is detailed in RFC TREAT [8].

Rserpool architecture allows the use of different Transport protocols for its application and control data exchange. Those transport protocols may have mechanisms for reducing the risk of blind denial-of-service attacks and/or masquerade attacks. If such measures are required by the applications, then it is advised to check the SCTP applicability statement[RFC3057] for guidance on this issue.

7. Acknowledgments

The authors wish to thank X, Y and M. Stillman and many others for their invaluable comments.

References

[1] Tuexen, M., Stewart, R., Shore, M., Xie, Q., Ong, L., Loughney, J. and M. Stillman, "Requirements for Reliable Server Pooling", RFC 3237, January 2002.

[2] Tuexen, M., Stewart, R., Shore, M., Xie, Q., Ong, L., Loughney, J. and M. Stillman, "Architecture for Reliable Server Pooling", Draft in progress , October 2002.

[3] Stewart, R., Xie, Q., Stillman, M. and M. Tuexen, "Aggregate Server Access Protocol (ASAP)", Draft in progress , October 2002.

[4] Xie, Q., Stewart, R. and M. Stillman, "Endpoint Name Resolution Protocol (ENRP)", Draft in progress , October 2002.

[5] Stewart, R., Xie, Q., Stillman, M. and M. Tuexen, "Aggregate Server Access Protocol and Endpoint Name Resolution Protocol Common Parameters", Draft in progress , October 2002.

[6] Conrad, P. and P. Lei, ""Services Provided By Reliable Server Pooling", Draft in progress , January 2003.

[7] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, ""Stream Control Transmission Protocol"", RFC 2960, October 2000.

[8] Stillman, M., Gopal, R., Sengodan, S., Guttman, E. and M. Holdrege, ""Threats Introduced by Rserpool and Requirements for Security in response to Threats"", RFC zzzz, Nov 2002.

[9] Coene, L., ""Stream Control Transmission Protocol Applicability statement"", RFC 3257, April 2002.

Author's Address

Lode Coene
Siemens
Atealaan 32
Herentals 2200
Belgium

Phone: +32-14-252081
EMail: lode.coene@siemens.com

Intellectual Property Statement

Full Copyright Statement

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.