Workgroup: Network Working Group
Internet-Draft:
draft-dreibholz-rserpool-applic-distcomp-32
Published: 26 September 2023
Intended Status: Informational
Expires: 29 March 2024
Authors: T. Dreibholz
SimulaMet

# Applicability of Reliable Server Pooling for Real-Time Distributed Computing

## Abstract

This document describes the applicability of the Reliable Server Pooling architecture to manage real-time distributed computing pools and access the resources of such pools.

## Status of This Memo



## Copyright Notice

Table of Contents



## 1. Introduction

Reliable Server Pooling defines protocols for providing highly available services. The services are located in a 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 transferred to the other server taking over.

The goal is to provide server-based redundancy. Transport and network level redundancy are handled by the transport and network layer protocols.

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

### 1.1. Scope

The scope of this document is to explain the way of using Reliable Server Pooling mechanisms to manage and access pools of Distributed Computing resources.

### 1.2. Terminology

The terms are commonly identified in related work and can be found in the Aggregate Server Access Protocol and Endpoint Handlespace Redundancy Protocol Common Parameters document [6].

## 2. Distributed Computing using RSerPool

### 2.1. Requirements

The application scenario for Distributed Computing is defined as follows:

* Clients generate large computation jobs. Jobs have to be processed by servers as soon as possible (real-time), i.e. unlike concepts like SETI@home [21], it is not possible to let clients fetch a job, process it later and may be some day upload the result.

* Jobs may be partitionable, i.e. they can be split up to smaller pieces which can be processed independently and the processing results can be concatenated to the processing result of the complete job. Jobs have to be processed by servers.

* Servers may be unreliable; i.e. user computers may be temporarily added to the pool of computing resources and may be revoked when they are used again by their owners. Furthermore, they may simply disappear because of broken network connections (modems, etc.) or power turned off.

* The processing power of servers in a pool of computing resources may be very heterogeneous, i.e. a few supercomputers and many low-end user PCs.

Maintaining a Distributed Computing pool for the scenario described above arises the following requirements to the pool management:

* It must be possible to manage large server pools, e.g. up to some hundreds or even thousands of servers.

* Due to heterogeneous processing resources within a pool, it must be possible to use appropriate server selection procedures to meaningfully utilize the available resources.

* It must be possible to dynamically add and remove servers.

* Servers may be unreliable, especially when the servers are represented by user PCs. Failover mechanisms are required to continue an interrupted computation session.

### 2.2. Architecture

All requirements for pool and session management of the Distributed Computing scenario defined in the previous section can be fulfilled by the Reliable Server Pooling architecture:

* An efficient implementation of the handlespace management structures allows pools to contain thousands of elements. Handlespace management structures have been proposed, implemented and analyzed in [15], [12].

* RSerPool allows to specify server selection rules by pool member selection policies [8]. A set of adaptive and non-adaptive policies is already defined. To fulfill the requirements of new applications, it is also possible to define new policies. Research has already been made on the subject of load distribution efficiency of pool policies in Distributed Computing scenarios: see [12], [14], [18], [19], [13] for details.

* Dynamic addition and removal of PEs is a feature of RSerPool [4].

* The control/data channel concept [3] of RSerPool realizes a session layer. That is, RSerPool already handles the main task of maintaining and monitoring connections between PUs and PEs; the only task of the application layer to provide full failover functionality is to realize an application-dependent failover procedure. By the usage of client-based state synchronization [14], [17] in the form of ASAP Cookies, a failover may be fully transparent to the PU while only a state restoration is necessary on the PE side. A demo application [22] using the RSerPool session layer in a Distributed Computing application is described in [16].

### 2.3. Limitations

Applying RSerPool for distributed computing applications, the duties of the RSerPool architecture are still limited to the management of pools and independent sessions only. It is in particular a non-goal to provide functionalities like data synchronization among sessions, user authentication, accounting or the support for more than one administrative domain. Such functionalities are considered to be application-specific and are therefore out of the scope of RSerPool.

## 3. Reference Implementation

The RSerPool reference implementation RSPLIB, including example Distributed Computing applications, can be found at [22]. It supports the functionalities defined by [3], [4], [5], [6] and [7] as well as the options [9], [11] and [10]. An introduction to this implementation is provided in [12].

## 4. Testbed Platform

A large-scale and realistic Internet testbed platform with support for the multi-homing feature of the underlying SCTP protocol is NorNet. A description of NorNet is provided in [20], some further information can be found on the project website [23].

## 5. Security Considerations

The protocols used in the Reliable Server Pooling architecture only try to increase the availability of the servers in the network. RSerPool protocols do 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.

The RSerPool architecture allows the use of different transport protocols for its application and control data exchange. These 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 (see [2]) applicability statement [1] for guidance on this issue.

## 6. IANA Considerations

This document introduces no additional considerations for IANA.

## 7. References

### 7.1. Normative References


**[1]** Coene, L., "Stream Control Transmission Protocol Applicability Statement", RFC 3257, DOI 10.17487/RFC3257, April 2002, <https://www.rfc-editor.org/info/rfc3257>.

**[2]** Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, DOI 10.17487/RFC4960, September 2007, <https://www.rfc-editor.org/info/rfc4960>.

**[3]** Lei, P., Ong, L., Tuexen, M., and T. Dreibholz, "An Overview of Reliable Server Pooling Protocols", RFC 5351, DOI 10.17487/RFC5351, September 2008, <https://www.rfc-editor.org/info/rfc5351>.

**[4]** Stewart, R., Xie, Q., Stillman, M., and M. Tuexen, "Aggregate Server Access Protocol (ASAP)", RFC 5352, DOI

10.17487/RFC5352, September 2008, <https://www.rfc-editor.org/info/rfc5352>.

**[5]** Xie, Q., Stewart, R., Stillman, M., Tuexen, M., and A. Silverton, "Endpoint Handlespace Redundancy Protocol (ENRP)", RFC 5353, DOI 10.17487/RFC5353, September 2008, <https://www.rfc-editor.org/info/rfc5353>.

**[6]** Stewart, R., Xie, Q., Stillman, M., and M. Tuexen, "Aggregate Server Access Protocol (ASAP) and Endpoint Handlespace Redundancy Protocol (ENRP) Parameters", RFC 5354, DOI 10.17487/RFC5354, September 2008, <https://www.rfc-editor.org/info/rfc5354>.

**[7]** Stillman, M., Ed., Gopal, R., Guttman, E., Sengodan, S., and M. Holdrege, "Threats Introduced by Reliable Server Pooling (RSerPool) and Requirements for Security in Response to Threats", RFC 5355, DOI 10.17487/RFC5355, September 2008, <https://www.rfc-editor.org/info/rfc5355>.

**[8]** Dreibholz, T. and M. Tuexen, "Reliable Server Pooling Policies", RFC 5356, DOI 10.17487/RFC5356, September 2008, <https://www.rfc-editor.org/info/rfc5356>.

**[9]** Dreibholz, T., "Handle Resolution Option for ASAP", Work in Progress, Internet-Draft, draft-dreibholz-rserpool-asap-hropt-29, 6 September 2021, <https://www.ietf.org/archive/id/draft-dreibholz-rserpool-asap-hropt-29.txt>.

**[10]** Dreibholz, T. and X. Zhou, "Definition of a Delay Measurement Infrastructure and Delay-Sensitive Least-Used Policy for Reliable Server Pooling", Work in Progress, Internet-Draft, draft-dreibholz-rserpool-delay-28, 6 September 2021, <https://www.ietf.org/archive/id/draft-dreibholz-rserpool-delay-28.txt>.

**[11]** Dreibholz, T. and X. Zhou, "Takeover Suggestion Flag for the ENRP Handle Update Message", Work in Progress, Internet-Draft, draft-dreibholz-rserpool-enrp-takeover-26, 6 September 2021, <https://www.ietf.org/archive/id/draft-dreibholz-rserpool-enrp-takeover-26.txt>.

### 7.2. Informative References

**[12]** Dreibholz, T., "Reliable Server Pooling – Evaluation, Optimization and Extension of a Novel IETF Architecture", 7 March 2007, <https://duepublico.uni-duisburg-essen.de/

servlets/DerivateServlet/Derivate-16326/
Dre2006_final.pdf>.

**[13]** Dreibholz, T., Zhou, X., and E. P. Rathgeb, "A Performance Evaluation of RSerPool Server Selection Policies in Varying Heterogeneous Capacity Scenarios", Proceedings of the 33rd IEEE EuroMirco Conference on Software Engineering and Advanced Applications Pages 157-164, ISBN 0-7695-2977-1, DOI 10.1109/EUROMICRO.2007.9, 29 August 2007, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/EuroMicro2007.pdf>.

**[14]** Dreibholz, T. and E. P. Rathgeb, "Overview and Evaluation of the Server Redundancy and Session Failover Mechanisms in the Reliable Server Pooling Framework", International Journal on Advances in Internet Technology (IJAIT) Number 1, Volume 2, Pages 1-14, ISSN 1942-2652, June 2009, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/IJAIT2009.pdf>.

**[15]** Dreibholz, T. and E. P. Rathgeb, "An Evaluation of the Pool Maintenance Overhead in Reliable Server Pooling Systems", SERSC International Journal on Hybrid Information Technology (IJHIT) Number 2, Volume 1, Pages 17-32, ISSN 1738-9968, April 2008, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/IJHIT2008.pdf>.

**[16]** Dreibholz, T. and E. P. Rathgeb, "An Application Demonstration of the Reliable Server Pooling Framework", Proceedings of the 24th IEEE INFOCOM, 16 March 2005, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/Infocom2005.pdf>.

**[17]** Dreibholz, T., "An Efficient Approach for State Sharing in Server Pools", Proceedings of the 27th IEEE Local Computer Networks Conference (LCN) Pages 348-349, ISBN 0-7695-1591-6, DOI 10.1109/LCN.2002.1181806, 8 November 2002, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/StateSharing-Paper-ShortVersion.pdf>.

**[18]** Dreibholz, T. and E. P. Rathgeb, "On the Performance of Reliable Server Pooling Systems", Proceedings of the IEEE Conference on Local Computer Networks (LCN) 30th Anniversary Pages 200-208, ISBN 0-7695-2421-4, DOI 10.1109/LCN.2005.98, 16 November 2005, <https://

www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/LCN2005.pdf>.

**[19]** Dreibholz, T. and E. P. Rathgeb, "The Performance of Reliable Server Pooling Systems in Different Server Capacity Scenarios", Proceedings of the IEEE TENCON ISBN 0-7803-9312-0, DOI 10.1109/TENCON.2005.300939, 22 November 2005, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/Tencon2005.pdf>.

**[20]** Dreibholz, T. and E. G. Gran, "Design and Implementation of the NorNet Core Research Testbed for Multi-Homed Systems", Proceedings of the 3nd International Workshop on Protocols and Applications with Multi-Homing Support (PAMS) Pages 1094-1100, ISBN 978-0-7695-4952-1, DOI 10.1109/WAINA.2013.71, 27 March 2013, <https://www.simula.no/file/threfereedinproceedingsreference2012-12-207643198512pdf/download>.

**[21]** SETI Project, "SETI@home: Search for Extraterrestrial Intelligence at home", 2022, <https://setiathome.ssl.berkeley.edu/>.

**[22]** Dreibholz, T., "Thomas Dreibholz's RSerPool Page", 2022, <https://www.nntb.no/~dreibh/rserpool/>.

**[23]** Dreibholz, T., "NorNet – A Real-World, Large-Scale Multi-Homing Testbed", 2022, <https://www.nntb.no/>.

## Author's Address

Thomas Dreibholz
Simula Metropolitan Centre for Digital Engineering
Pilestredet 52
0167 Oslo
Norway

Email: dreibh@simula.no
URI: https://www.simula.no/people/dreibh