Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 14, 2020

T. Dreibholz
SimulaMet
M. Tuexen
Muenster Univ. of Appl. Sciences
M. Shore
No Mountain Software
N. Zong
Huawei Technologies

September 11, 2019

The Applicability of Reliable Server Pooling (RSerPool) for Virtual Network Function Resource Pooling (VNFPOOL)

draft-dreibholz-vnfpool-rserpool-applic-09

Abstract

This draft describes the application of Reliable Server Pooling (RSerPool) for Virtual Network Function Resource Pooling (VNFPOOL).

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

1.1. Abbreviations

- PE: Pool Element

- PR: Pool Registrar

- PU: Pool User

- RSerPool: Reliable Server Pooling

- SCTP: Stream Control Transmission Protocol

- VNFPOOL: Virtual Network Function Resource Pooling

2. Virtual Network Function Resource Pooling

Virtualised Network Function (VNF) (e.g. vFW, vLB) -- as introduced in more detail in [I-D.zong-vnfpool-problem-statement] -- provides the same function as the equivalent network function (e.g. FW, LB), but is deployed as software instances running on general purpose servers via virtualisation platform. The main features of VNF include the following aspects:

1. A service consists of a sequence of topologically distributed VNF instances where the data connections are preferably directly established between the instances.

2. There are potentially more factors that cause VNF instance transition or even failure; VNF pool refers to a group of VNF instances providing same network function.

Virtualisation technology allows network function virtualisation operators to build a reliable VNF by pooling the underlying resources, such as CPU, storage, networking, etc. to form a cluster of VNF instances. VNF pool refers to a cluster or group of VNF instances providing same network function. Each VNF pool has a Pool Manager (PM) to manage the VNF instance such as instance selection, monitoring, etc. There will be a redundancy mechanism for a reliable PM to achieve reliable VNF. More details on VNF pool can be found in [I-D.zong-vnfpool-problem-statement].

3. Reliable Server Pooling

3.1. Introduction

```
                      +---------------+
                      |   Pool User   |
                      +---------------+
                              ^
                              | ASAP
                              V
                      +---------------+   ENRP   +---------------+
                      |   Registrar   |<-------->|   Registrar   |
                      +---------------+          +---------------+
                              ^
                              | ASAP
                              V
      +------------------------------------------------------------+
      | +--------------+ +--------------+         +--------------+ |
      | | Pool Element | | Pool Element | ... ... | Pool Element | |
      | +--------------+ +--------------+         +--------------+ |
      | Server Pool                                                |
      +------------------------------------------------------------+
```

Figure 1

An overview of the RSerPool framework -- which is defined as RFC in [RFC5351] -- is provided in Figure 1. There are three types of components:

- o Pool Element (PE) denotes a server in a pool. PEs in the same pool provide the same service.

- o Pool User (PU) denotes a client using the service of a pool.

- o Pool Registrar (PR) is the management component for the pools.

The set of all pools within an operation scope (for example: an organisation, a company or a department) is denoted as handlespace. Clearly, a single PR would be a single point of failure. Therefore, PRs also have to be redundant. Within the handlespace, each pool is identified by a unique pool handle (PH).

3.2. Registrar Operations

The PRs of an operation scope synchronise their view of the handlespace by using the Endpoint haNdlespace Redundancy Protocol (ENRP, defined as RFCs in [RFC5353], [RFC5354]). In contrast to for instance the Domain Name System (DNS), an operation scope is restricted to a single administrative domain. That is, all of its components are under the control of the same authority (for example: a company). This property leads to small management overhead, which also allows for RSerPool usage on devices having only limited memory and CPU resources (for example: telecommunications equipment). Nevertheless, PEs may be distributed globally to continue their service even in case of localised disasters (like for example an earthquake). Each PR in the operation scope is identified by a PR ID, which is a randomly chosen 32-bit number.

3.3. Pool Element Operations

Within their operation scope, the PEs may choose an arbitrary PR to register into a pool by using the Aggregate Server Access Protocol (ASAP, defined as RFCs in [RFC5352], [RFC5354]). The registration is performed by using an ASAP_REGISTRATION message. Within its pool, a PE is characterised by its PE ID, which is a randomly chosen 32-bit number. Upon registration at a PR, the chosen PR becomes the Home-PR (PR-H) of the newly registered PE. A PR-H is responsible for monitoring the availability of its PEs by ASAP_ENDPOINT_KEEP_ALIVE messages (to be acknowledged by a PE via an ASAP_ENDPOINT_KEEP_ALIVE_ACK message within a configured timeout). The PR-H propagates the information about its PEs to the other PRs of the operation scope via ENRP_UPDATE messages.

PEs re-register regularly in an interval denoted as registration lifetime and for information updates. Similar to the registration, a re-registration is performed by using another ASAP_REGISTRATION message. PEs may intentionally deregister from the pool by using an ASAP_DEREGISTRATION message. Also like for the registration, the PR-H makes the deregistration known to the other PRs within the operation scope by using an ENRP_UPDATE message.

3.4. Takeover Procedure

As soon as a PE detects the failure of its PR-H (that is: its request is not answered within a given timeout), it simply tries another PR of the operation scope for its registration and deregistration requests. However, as a double safeguard, the remaining PRs also negotiate a takeover of the PEs managed by a dead PR. This ensures that each PE again gets a working PR-H as soon as possible. The PRs of an operation scope monitor the availability of each other PR by using ENRP_PRESENCE messages, which are transmitted regularly. If there is no ENRP_PRESENCE within a given timeout, the peer is assumed to be dead and a so-called takeover procedure (see also [AINA2009] for more details) is initiated for the PEs managed by the dead PR: from all PRs having started this takeover procedure, the PR with the highest PR ID takes over the ownership of these PEs. The PEs are informed about being taken over by their new PR-H via an ASAP_ENDPOINT_KEEP_ALIVE with Home-flag set. The PEs are requested to adopt the sender of this Home-flagged message as their new PR-H.

3.5. Pool User Operations

3.5.1. Handle Resolution and Response

In order to access the service of a pool given by its PH, a PU requests a PE selection from an arbitrary PR of the operation scope, again by using ASAP. This selection procedure is denoted as handle resolution. Upon reception of a so-called ASAP_HANDLE_RESOLUTION message the PR selects the requested list of PE identities and returns them in an ASAP_HANDLE_RESOLUTION_RESPONSE message.

3.5.2. Pool Member Selection Policies

The pool-specific selection rule is denoted as pool member selection policy or shortly as pool policy. Two classes of load distribution policies are supported: non-adaptive and adaptive strategies (a detailed overview is provided by [Dre2006], [LCN2005], [IJIIDS2010], [IJHIT2008]). While adaptive strategies base their selections on the current PE state (which requires up-to-date information), non-adaptive algorithms do not need such data. A basic set of adaptive and non-adaptive pool policies is defined as RFC in [RFC5356].

Defined in [RFC5356] are the non-adaptive policies Round Robin (RR), Random (RAND) and Priority (PRIO) as well as the adaptive policies Least Used (LU) and Least Used with Degradation (LUD). While RR/RAND select PEs in turn/randomly, PRIO selects one of the PEs having the highest priority. PRIO can for example be used to realise a master/backup PE setup. Only if there are no master PEs left, a backup PE is selected. Round-robin selection is applied among PEs having the same priority. LU selects the least-used PE, according to up-to-date application-specific load information. Round robin selection is applied among multiple least-loaded PEs. LUD, which is evaluated by [ICDS2008-LUD], furthermore introduces a load decrement constant that is added to the actual load each time a PE is selected. It is used to compensate inaccurate load states due to delayed updates. An update resets the load to the actual load value.

3.5.3. Handle Resolution and Response

PE may fail, for example due to hardware or network failures. Since there is a certain latency between the actual failure of a PE and the removal of its entry from the handlespace -- depending on the interval and timeout for the ASAP_ENDPOINT_KEEP_ALIVE monitoring -- the PUs may report unreachable PEs to a PR by using an ASAP_ENDPOINT_UNREACHABLE message. A PR locally counts these reports for each PE and when reaching the threshold MAX-BAD-PE-REPORT (default is 3, as defined in the RFC [RFC5352]), the PR may decide to remove the PE from the handlespace. The counter of a PE is reset upon its re-registration. More details on this threshold and guidelines for its configuration can be found in [IJAIT2009].

3.6. Automatic Configuration

RSerPool components need to know the PRs of their operation scope. While it is of course possible to configure a list of PRs into each component, RSerPool also provides an auto-configuration feature: PRs may send so-called announces, that is, ASAP_ANNOUNCE and ENRP_PRESENCE messages which are regularly sent over UDP via IP multicast. Unlike broadcasts, multicast messages can also be transported over routers (at least, this is easily possible within LANs). The announces of the PRs can be heard by the other components, which can maintain a list of currently available PRs. That is, RSerPool components are usually just turned on and everything works automatically.

3.7. State Synchronisation

RSerPool has been explicitly designed to be application-independent. Therefore, RSerPool has not intended to define special state synchronisation mechanisms for RSerPool-based applications. Such state synchronisation mechanisms are considered as tasks of the applications themselves. However, RSerPool defines two mechanisms to at least support the implementation of more sophisticated strategies: Cookies and Businesss Cards. Details on these mechanisms can also be found in Subsection 3.9.5 of [Dre2006].

3.7.1. Cookies

ASAP provides the mechanism of Client-Based State Sharing as introduced in [LCN2002]. Whenever useful, the PE may package its state in form of a state cookie and send it -- by an ASAP_COOKIE message -- to the PU. The PU stores the latest state cookie received from the PE. Upon PE failure, this stored cookie is sent in an ASAP_COOKIE_ECHO to the newly chosen PE. This PE may then restore the state. A shared secret known by all PEs of a pool may be used to protect the state from being manipulated or read by the PU.

While Client-Based State Sharing is very simple, it may be inefficient when the state changes too frequently, is too large (the size limit of an ASAP_COOKIE/ASAP_COOKIE_ECHO is 64 KiB) or if it must be prevented that a PU sends a state cookie to multiple PEs in order to duplicate its sessions.

3.7.2. Businesss Cards

Depending on the application, there may be constraints restricting the set of PEs usable for failover. The ASAP_BUSINESS_CARD message is used to inform peer components about such constraints.

The first case to use a Business Card is if only a restricted set of PEs in the pool may be used for failover. For example, in a large pool, each PE can share its complete set of session states with a few other PEs only. This keeps the system scalable. That is, a PE in a pool of n servers does not have to synchronise all session states with the other n-1 PEs. In this case, a PE has to tell its PU the set of PE identities being candidates for a failover using an ASAP_BUSINESS_CARD message. A PE may update the list of possible failover candidates at any time by sending another Business Card. The PU has to store the latest list of failover candidates. Of course, if a failover becomes necessary, the PU has to select from this list using the appropriate pool policy -- instead of performing the regular PE selection by handle resolution at a PR. Therefore, some literature also denotes the Business Card by the more expressive term "last will".

In symmetric scenarios, where a PU is also a PE of another pool, the PU has to tell this fact to its PE. This is realised by sending an ASAP_BUSINESS_CARD message to the PE, providing the PH of its pool. Optionally, also specific PE identities for failover may be provided. The format remains the same as explained in the previous paragraph. If the PE detects a failure of its PU, the PE may -- now in the role of a PU -- use the provided PH for a handle resolution to find a new PE or use the provided PE identities to select one. After that, it can perform a failover to that PE.

3.8. Protocol Stack

The protocol stack of a PR provides ENRP and ASAP services to PRs and PEs/PUs respectively. But between PU and PE, ASAP provides a Session Layer protocol in the OSI model. From the perspective of the Application Layer, the PU side establishes a session with a pool. ASAP takes care of selecting a PE of the pool, initiating and maintaining the underlying transport connection and triggering a failover procedure when the PE becomes unavailable.

The Transport Layer protocol is by default SCTP (as defined in [RFC4960]) -- except for the UDP-based automatic configuration announces (see Section 3.6) -- over possibly multi-homed IPv4 and/or IPv6. SCTP has been chosen due to its support of multi-homing and its reliability features (see also [Dre2012]).

3.9. Extensions

A couple of extensions to RSerPool are existing: Handle Resolution Option defined in [I-D.dreibholz-rserpool-asap-hropt] improves the PE selection by letting the PU tell the PR its required number of PEs to be selected. ENRP Takeover Suggestion introduced in [I-D.dreibholz-rserpool-enrp-takeover] ensures load balancing among PRs. [I-D.dreibholz-rserpool-delay] defines a delay-sensitive pool policy. [RFC5525] defines an SNMP MIB for RSerPool.

3.10. Reference Implementation and Deployment

RSPLIB is the Open Source reference implementation of RSerPool. It is currently -- as of February 2016 -- available for Linux, FreeBSD, MacOS and Solaris. It is actively maintained. Particularly, it is also included in Ubuntu Linux as well as in the FreeBSD ports collection. RSPLIB can be downloaded from [RSerPoolPage]. Further details on the implementation are available in [Dre2006], [Globecom2010-Demo].

RSerPool with RSPLIB is deployed in a couple of Open Source projects, including the SimProcTC Simulation Processing Tool-Chain for distributing simulation runs in a compute pool (see [OMNeTWorkshop2008] as well as the simulation run distribution project explained in [Dre2012] for a practical example) as well as for service infrastructure management in the NorNet Core research testbed (see [ComNets2013-Core], [PAMS2013-NorNet]).

4. Usage of Reliable Server Pooling

**** TO BE DISCUSSED! ****

The following features of RSerPool can be used for VNFPOOL:

- Pool management.
- PE seclection with pool policies.
- Session management with help of ASAP_BUSINESS_CARD.

The following features have to be added to RSerPool itself:

- Support of TCP including MPTCP as additional/alternative transport protocols.
- Possibly add some special pool policies?

- See also [I-D.dreibholz-rserpool-nextgen-ideas] for ideas on a next generation of RSerPool.

The following features have to be provided outside of RSerPool:

- State synchronisation for VNFPOOL.

- Pool Manager functionality as an RSerPool-based service.

5. Security Considerations

Security considerations for RSerPool can be found in [RFC5355]. Furthermore, [IJIIDS2010] examines the robustness of RSerPool systems against attacks.

6. IANA Considerations

This document introduces no additional considerations for IANA.

7. Testbed Platform

A large-scale and realistic Internet testbed platform with support for Reliable Server Pooling and the underlying SCTP protocol is NorNet. A description of and introduction to NorNet is provided in [PAMS2013-NorNet], [Sydney2019], [Haikou2019-NorNet-Tutorial]. Further information can be found on the project website [NorNet-Website] at https://www.nntb.no.

8. Acknowledgments

The authors would like to thank Xing Zhou for the friendly support.

9. References

9.1. Normative References

[I-D.dreibholz-rserpool-asap-hropt]
Dreibholz, T., "Handle Resolution Option for ASAP", draft-dreibholz-rserpool-asap-hropt-24 (work in progress), March 2019.

[I-D.dreibholz-rserpool-delay]
Dreibholz, T. and X. Zhou, "Definition of a Delay Measurement Infrastructure and Delay-Sensitive Least-Used Policy for Reliable Server Pooling", draft-dreibholz-rserpool-delay-23 (work in progress), March 2019.

[I-D.dreibholz-rserpool-enrp-takeover]
Dreibholz, T. and X. Zhou, "Takeover Suggestion Flag for the ENRP Handle Update Message", draft-dreibholz-rserpool-enrp-takeover-21 (work in progress), March 2019.

[I-D.dreibholz-rserpool-nextgen-ideas]
Dreibholz, T., "Ideas for a Next Generation of the Reliable Server Pooling Framework", draft-dreibholz-rserpool-nextgen-ideas-11 (work in progress), March 2019.

[I-D.zong-vnfpool-problem-statement]
Zong, N., Dunbar, L., Shore, M., Lopez, D., and G. Karagiannis, "Virtualized Network Function (VNF) Pool Problem Statement", draft-zong-vnfpool-problem-statement-06 (work in progress), July 2014.

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

[RFC5351] 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>.

[RFC5352] 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>.

[RFC5353] 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>.

[RFC5354] 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>.

[RFC5355] 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>.

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

[RFC5525] Dreibholz, T. and J. Mulik, "Reliable Server Pooling MIB Module Definition", RFC 5525, DOI 10.17487/RFC5525, April 2009, <https://www.rfc-editor.org/info/rfc5525>.

9.2. Informative References

[AINA2009]
Zhou, X., Dreibholz, T., Fa, F., Du, W., and E. Rathgeb, "Evaluation and Optimization of the Registrar Redundancy Handling in Reliable Server Pooling Systems", Proceedings of the IEEE 23rd International Conference on Advanced Information Networking and Applications (AINA) Pages 256-262, ISBN 978-0-7695-3638-5, DOI 10.1109/AINA.2009.25, May 2009, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/AINA2009.pdf>.

[ComNets2013-Core]
Gran, E., Dreibholz, T., and A. Kvalbein, "NorNet Core - A Multi-Homed Research Testbed", Computer Networks, Special Issue on Future Internet Testbeds Volume 61, Pages 75-87, ISSN 1389-1286, DOI 10.1016/j.bjp.2013.12.035, March 2014, <https://www.simula.no/file/simulasimula2236pdf/download>.

[Dre2006] Dreibholz, T., "Reliable Server Pooling - Evaluation, Optimization and Extension of a Novel IETF Architecture", March 2007, <https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-16326/Dre2006_final.pdf>.

[Dre2012] Dreibholz, T., "Evaluation and Optimisation of Multi-Path Transport using the Stream Control Transmission Protocol", Habilitation Treatise, March 2012, <https://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-29737/Dre2012_final.pdf>.

[Globecom2010-Demo]
Dreibholz, T. and M. Becke, "The RSPLIB Project - From Research to Application", Demo Presentation at the IEEE Global Communications Conference (GLOBECOM), December 2010, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/Globecom2010-Demo.pdf>.

[Haikou2019-NorNet-Tutorial]
Dreibholz, T., "NorNet at Hainan University: Getting Started with NorNet Core", Tutorial at Hainan University, College of Information Science and Technology (CIST), April 2019, <https://www.simula.no/file/china2019-nornet-tutorialpdf/download>.

[ICDS2008-LUD]
Zhou, X., Dreibholz, T., and E. Rathgeb, "A New Server Selection Strategy for Reliable Server Pooling in Widely Distributed Environments", Proceedings of the 2nd IEEE International Conference on Digital Society (ICDS) Pages 171-177, ISBN 978-0-7695-3087-1, DOI 10.1109/ICDS.2008.12, February 2008, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/ICDS2008-LUD.pdf>.

[IJAIT2009]
Dreibholz, T. and E. 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>.

[IJHIT2008]
Dreibholz, T. and E. 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>.

[IJIIDS2010]
Dreibholz, T., Zhou, X., Becke, M., Pulinthanath, J., Rathgeb, E., and W. Du, "On the Security of Reliable Server Pooling Systems", International Journal on Intelligent Information and Database Systems (IJIIDS) Number 6, Volume 4, Pages 552-578, ISSN 1751-5858, DOI 10.1504/IJIIDS.2010.036894, December 2010, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/IJIIDS2010.pdf>.

[LCN2002] 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, November 2002, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/StateSharing-Paper-ShortVersion.pdf>.

[LCN2005] Dreibholz, T. and E. 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, November 2005, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/LCN2005.pdf>.

[NorNet-Website]
Dreibholz, T., "NorNet -- A Real-World, Large-Scale Multi-Homing Testbed", Online: https://www.nntb.no/, 2017, <https://www.nntb.no/>.

[OMNeTWorkshop2008]
Dreibholz, T. and E. Rathgeb, "A Powerful Tool-Chain for Setup, Distributed Processing, Analysis and Debugging of OMNeT++ Simulations", Proceedings of the 1st ACM/ICST International Workshop on OMNeT++ ISBN 978-963-9799-20-2, DOI 10.4108/ICST.SIMUTOOLS2008.2990, March 2008, <https://www.wiwi.uni-due.de/fileadmin/fileupload/I-TDR/ReliableServer/Publications/OMNeTWorkshop2008.pdf>.

[PAMS2013-NorNet]
Dreibholz, T. and E. 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, March 2013, <https://www.simula.no/file/threfereedinproceedingsreference2012-12-207643198512pdf/download>.

[RSerPoolPage]
Dreibholz, T., "Thomas Dreibholz’s RSerPool Page", Online: https://www.uni-due.de/˜be0001/rserpool/, 2019, <https://www.uni-due.de/˜be0001/rserpool/>.

[Sydney2019]
Dreibholz, T., "NorNet at the University of Sydney: From Simulations to Real-World Internet Measurements for Multi-Path Transport Research", Invited Talk at University of Sydney, January 2019, <https://www.simula.no/file/sydney2019-presentationpdf/download>.

Authors' Addresses

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

Phone: +47-6782-8200
Fax: +47-6782-8201
Email: dreibh@simula.no
URI: https://www.simula.no/people/dreibh

Michael Tuexen
Muenster University of Applied Sciences
Stegerwaldstrasse 39
48565 Steinfurt, Nordrhein-Westfalen
Germany

Email: tuexen@fh-muenster.de
URI: https://www.fh-muenster.de/fb2/personen/professoren/tuexen/

Melinda Shore
No Mountain Software
PO Box 16271
Two Rivers, Alaska 99716
U.S.A.

Phone: +1-907-322-9522
Email: melinda.shore@nomountain.net
URI: https://www.linkedin.com/pub/melinda-shore/9/667/236

Ning Zong
Huawei Technologies
101 Software Avenue
Nanjing, Jiangsu 210012
China

Email: zongning@huawei.com
URI: https://cn.linkedin.com/pub/ning-zong/15/737/490