Grid Engine Release Notes

Author: Univa Engineering

Version: 8.6.0

July 19, 2018
## Contents

1 License .......................... 1

2 Supported Operating Systems, Versions and Architectures 5

3 Supported and Tested Docker Versions 6
   3.1 Known Docker Issues That Affect Univa Grid Engine ................. 10

4 Fixes and Enhancements ............... 11
   4.1 Major Enhancements ........................................... 11
      4.1.1 New Core Binding Strategies and PE Behavior .................. 11
      4.1.2 Affinity Based Job Placement ................................ 11
      4.1.3 Managing Access to Devices with RSMAPs .................... 12
      4.1.4 Integration with Nvidia DCGM .................................. 12
      4.1.5 Performance Improvements ................................... 13
      4.1.6 Reworked Dispatch Job Information ............................ 14
      4.1.7 Data Compression at Communication Layer ...................... 14
      4.1.8 Using RSMAPs with Topology Masks and XOR Operator .......... 15
   4.2 Other ........................................ 16
      4.2.1 JSV Improvements ........................................... 16
      4.2.2 Reduce qhost Data Request Sizes at sge_qmaster ............ 17
      4.2.3 Upgrade Advance and Standing Reservations .................. 17
      4.2.4 per_pe_task_prolog and per_pe_task_epilog .................. 17
      4.2.5 The Master Task of a Parallel Docker Job in a Container .... 17
      4.2.6 Run the Container as root, Allow to Run Prolog etc. as a Different User . 17
      4.2.7 Automatically Map User ID and Group ID of a User Into the Container .... 18
      4.2.8 Create a container_pe_hostfile with all Container Hostnames ...... 18
      4.2.9 Docker Daemon Response Timeout ................................ 18
      4.2.10 Cgroups and Containers ...................................... 19
      4.2.11 Specify Arguments to Autostart Docker Jobs .................. 19
      4.2.12 New Client Command qralter ................................ 19
      4.2.13 Changes to the loadcheck Command ............................ 19
   4.3 Full List of Fixes and Enhancements .............................. 20

---

*Grid Engine Release Notes v 8.6.0*
1 License

TERM SOFTWARE LICENSE AGREEMENT

This agreement is between the individual or entity agreeing to this agreement and Univa Corporation, a Delaware corporation (Univa) with its registered office at 2300 N Barrington Road, Suite 400, Hoffman Estates, IL 60195.

1. SCOPE

This agreement governs the licensing of the Univa Software and Support provided to Customer.

- Univa Software means the Univa software described in the order, all updates and enhancements provided under Support, its software documentation, and license keys (Univa Software), which are licensed under this agreement. This Univa Software is only licensed and is not sold to Company.
- Third-Party Software/Open Source Software licensing terms are addressed on the bottom of this agreement.

2. LICENSE

Subject to the other terms of this agreement, Univa grants Customer, under an order, a non-exclusive, non-transferable, renewable term license up to the license capacity purchased to: (a) Operate the Univa Software in Customer’s business operations; and (b) Make a reasonable number of copies of the Univa Software for archival and backup purposes. Customer’s contractors and majority owned affiliates are allowed to use and access the Univa Software under the terms of this agreement. Customer is responsible for their compliance with the terms of this agreement. The initial contracted term of this license will be automatically renewed on its expiry for a one year period unless a written notification of termination has been received 60 days prior to term expiry.

3. RESTRICTIONS

Univa reserves all rights not expressly granted. Customer is prohibited from: (a) assigning, sublicensing, or renting the Univa Software or using it as any type of software service provider or outsourcing environment; or (b) causing or permitting the reverse engineering (except to the extent expressly permitted by applicable law despite this limitation), decompiling, disassembly, modification, translation, attempting to discover the source code of the Univa Software or to create derivative works from the Univa Software.

4. PROPRIETARY RIGHTS AND CONFIDENTIALITY

(a) Proprietary Rights. The Univa Software, workflow processes, designs, know-how and other technologies provided by Univa as part of the Univa Software are the proprietary property of Univa and its licensors, and all right, title and interest in and to such items, including all associated intellectual property rights, remain only with Univa. The Univa Software is protected by applicable copyright, trade secret, and other intellectual property laws. Customer may not remove any product identification, copyright, trademark or other notice from the Univa Software.
(b) Confidentiality. Recipient may not disclose Confidential Information of Discloser to any third party or use the Confidential Information in violation of this agreement.

(c) Confidential Information means all proprietary or confidential information that is disclosed to the recipient (Recipient) by the discloser (Discloser), and includes, among other things:

- any and all information relating to Univa Software or Support provided by a Discloser, its financial information, software code, flow charts, techniques, specifications, development and marketing plans, strategies, and forecasts;
- as to Univa the Univa Software and the terms of this agreement (including without limitation, pricing information).

(ii) Confidential Information excludes information that:

- was rightfully in Recipient’s possession without any obligation of confidentiality before receipt from the Discloser;
- is or becomes a matter of public knowledge through no fault of Recipient;
- is rightfully received by Recipient from a third party without violation of a duty of confidentiality;
- is independently developed by or for Recipient without use or access to the Confidential Information; or is licensed under an open source license.

Customer acknowledges that any misuse or threatened misuse of the Univa Software may cause immediately irreparable harm to Univa for which there is no adequate remedy at law. Univa may seek immediate injunctive relief in such event.

5. PAYMENT

Customer will pay all fees due under an order within 30 days of the invoice date, plus applicable sales, use and other similar taxes.

6. WARRANTY DISCLAIMER. UNIVA DISCLAIMS ALL EXPRESS AND IMPLIED WARRANTIES, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTY OF TITLE, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE UNIVA SOFTWARE MAY NOT BE ERROR FREE, AND USE MAY BE INTERRUPTED.

7. TERMINATION

Either party may terminate this agreement upon a material breach of the other party after a 30 days notice/cure period, if the breach is not cured during such time period. Upon termination of this agreement or expiration of an order, Customer must discontinue using the Univa Software, de-install it and destroy or return the Univa Software and all copies, within 5 days. Upon Univa’ request, Customer will provide written certification of such compliance.

8. SUPPORT INCLUDED

Univa’s technical support and maintenance services (Support) is included with the fees paid under an order. Univa may change its Support terms, but Support will not materially degrade during any paid term. More details on Support are located at www.univa.com/support
9. LIMITATION OF LIABILITY AND DISCLAIMER OF DAMAGES

There may be situations in which, as a result of material breach or other liability, Customer is entitled to make a claim for damages against Univa. In each situation (regardless of the form of the legal action (e.g. contract or tort claims)), Univa is not responsible beyond: (a) the amount of any direct damages up to the amount paid by Customer to Univa in the prior 12 months under this agreement; and (b) damages for bodily injury (including death), and physical damage to tangible property, to the extent caused by the gross negligence or willful misconduct of Univa employees while at Customer’s facility.

Other than for breach of the Confidentiality section by a party, the infringement indemnity, violation of Univa’s intellectual property rights by Customer, or for breach of Section 2 by Customer, in no circumstances is either party responsible for any (even if it knows of the possibility of such damage or loss): (a) loss of (including any loss of use), or damage to: data, information or hardware; (b) lost profits, business, or goodwill; or (c) other special, consequential, or indirect damages.

10. INTELLECTUAL PROPERTY INDEMNITY

If a third-party claims that Customer’s use of the Univa Software under the terms of this agreement infringes that party’s patent, copyright or other proprietary right, Univa will defend Customer against that claim at Univa’ expense and pay all costs, damages, and attorney’s fees, that a court finally awards or that are included in a settlement approved by Univa, provided that Customer: (a) promptly notifies Univa in writing of the claim; and (b) allows Univa to control, and cooperates with Univa in, the defense and any related settlement. If such a claim is made, Univa could continue to enable Customer to use the Univa Software or to modify it. If Univa determines that these alternatives are not reasonably available, Univa may terminate the license to the Univa Software and refund any unused fees. Univa’ obligations above do not apply if the infringement claim is based on the use of the Univa Software in combination with products not supplied or approved by Univa in writing or in the Univa Software, or Customer’s failure to use any updates within a reasonable time after such updates are made available.

This section contains Customer’s exclusive remedies and Univa’ sole liability for infringement claims.

11. GOVERNING LAW AND EXCLUSIVE FORUM

This agreement is governed by the laws of the State of Illinois, without regard to conflict of law principles. Any dispute arising out of or related to this agreement may only be brought in the state of Illinois. Customer consents to the personal jurisdiction of such courts and waives any claim that it is an inconvenient forum. The prevailing party in litigation is entitled to recover its attorneys’ fees and costs from the other party.

12. MISCELLANEOUS

(a) Inspection. Upon request by Univa, Customer must provide a usage report at least 60 days before the renewal anniversary. Univa, or its representative, may audit Customer’s usage of the Univa Software at any Customer facility. Customer will cooperate with such audit. Customer agrees to pay within 30 days of written notification any fees applicable to Customer’s use of the Univa Software in excess of the license.
(b) Entire Agreement. This agreement, and all orders, constitute the entire agreement between
the parties, and supersedes all prior or contemporaneous negotiations, representations or
agreements, whether oral or written, related to this subject matter.

(c) Modification Only in Writing. No modification or waiver of any term of this agreement is
effective unless signed by both parties.

(d) Non-Assignment. Neither party may assign or transfer this agreement to a third party,
except that the agreement and all orders may be assigned upon notice as part of a merger,
or sale of all or substantially all of the business or assets, of a party.

(e) Export Compliance. Customer must comply with all applicable export control laws of the
United States, foreign jurisdictions and other applicable laws and regulations.

(f) US Government Restricted Rights. The Univa Software is provided with RESTRICTED
RIGHTS. Use, duplication, or disclosure by the U.S. government or any agency thereof
is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical
Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and
(2) of the Commercial Computer Software Restricted Rights at 48 C.F.R. 52.227-19, as
applicable.

(g) Independent Contractors. The parties are independent contractors with respect to each
other.

(h) Enforceability. If any term of this agreement is invalid or unenforceable, the other terms
remain in effect.

(i) No PO Terms. Univa rejects additional or conflicting terms of a Customer's form-purchasing
document.

(j) No CISG. The United Nations Convention on Contracts for the International Sale of Goods
does not apply.

(k) Survival. All terms that by their nature survive termination or expiration of this agreement,
will survive.

Additional software specific licensing terms: Grid Engine and Univa Software incorporate certain
third-party software listed at the URL below. These licenses are accepted by use of the software
and may represent license grants with restrictions which Univa is bound to provide. We are
hereby notifying you of these licenses. * Third Party Software means certain third-party software
which is provided along with the Univa Software, and such software is licensed under the license
terms located at: http://www.univa.com/resources/licenses/ * Open Source Software means
certain opens source software which is provided along with the Univa Software, and such software
is licensed under the license terms located at: http://www.univa.com/resources/licenses/

Rev: Jun 2018
2 Supported Operating Systems, Versions and Architectures

Univa Grid Engine supports various platforms, hardware architectures and versions of operating systems. Find the full list in following table:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Version</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLES</td>
<td>10,11,12</td>
<td>x86, x86-64</td>
</tr>
<tr>
<td>RHEL</td>
<td>5 or higher, 6 or higher, 7</td>
<td>x86, x86-64</td>
</tr>
<tr>
<td>CentOS</td>
<td>5 or higher, 6 or higher, 7</td>
<td>x86, x86-64</td>
</tr>
<tr>
<td>Oracle Linux</td>
<td>5 or higher, 6 or higher, 7</td>
<td>x86, x86-64</td>
</tr>
<tr>
<td>Ubuntu</td>
<td>10.04LTS - 16.04LTS</td>
<td>x86, x86-64</td>
</tr>
<tr>
<td>Oracle Solaris</td>
<td>10, 11</td>
<td>x86_64, SPARC 64bit</td>
</tr>
<tr>
<td>HP-UX</td>
<td>11.0 or higher</td>
<td>64bit</td>
</tr>
<tr>
<td>IBM AIX</td>
<td>6.1 or later</td>
<td>64bit</td>
</tr>
<tr>
<td>Apple OS X</td>
<td>10.8 (Mountain Lion) or higher</td>
<td>x86, x86-64</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>XP Professional (SP3)</td>
<td>32 bit</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>Server 2003 / 2003 R2</td>
<td>32 bit</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>Vista Enterprise / Ultimate</td>
<td>32 and 64bit</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>Server 2008 / 2008 R2</td>
<td>32 and 64bit</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>7 Professional / Enterprise / Ultimate</td>
<td>32 and 64bit</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>Server 2012 / 2012 R2</td>
<td>32 and 64bit</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>8 / 8.1 Pro / Enterprise</td>
<td>32 and 64bit</td>
</tr>
<tr>
<td>Microsoft Windows</td>
<td>10 Pro / Enterprise</td>
<td>32 and 64bit</td>
</tr>
</tbody>
</table>

Table 1: Supported Operating Systems, Versions and Architectures

PLEASE NOTE: Hosts running the Microsoft Windows operating system cannot be used as master or shadow hosts.

PLEASE NOTE: Univa Grid Engine qmaster is fully supported on Linux and Solaris. We provide binaries in Univa Grid Engine for running the qmaster on other operating systems but they are not supported and delivered as a courtesy. If you require qmaster support on other architectures please contact us at support@univa.com.

PLEASE NOTE: if you require Univa Grid Engine support for older versions of the above operating systems please contact our sales or support team.
3 Supported and Tested Docker Versions

In principle, Univa Grid Engine supports these Docker versions on these Linux distributions:

<table>
<thead>
<tr>
<th>Linux Distribution</th>
<th>Docker versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat, CentOS, Debian and Ubuntu</td>
<td>1.12.0 to 1.13.0 and 17.03.0-ce to 17.12.0-ce</td>
</tr>
<tr>
<td>SLES and openSUSE</td>
<td>1.12.0 to 1.13.0 and 17.03.0-ce to 17.09.1-ce</td>
</tr>
</tbody>
</table>

But in the past some Docker versions did not work properly and were withdrawn later. There were different Docker builds provided under the same version number showing a slightly different behavior, so it is not possible to ensure Univa Grid Engine does work with all Docker versions between 1.12.0 and 17.12.0. The following table shows which Docker versions were tested on which Linux distribution. The table includes the “Git commit” ID of the Docker source code, the Go language version which was used to build Docker, the build date and - for completeness - the version of the Docker RemoteAPI. Other Docker versions were not tested or are too broken to test successfully.

- Red Hat Enterprise Linux 7.4 (Maipo), Kernel 3.10.0-693.11.1.el7.x86_64

<table>
<thead>
<tr>
<th>Docker Version</th>
<th>Git commit</th>
<th>Go Version</th>
<th>Build date</th>
<th>API version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.12.0</td>
<td>8eab29e</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.1</td>
<td>23cf638</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.2</td>
<td>bb80604</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.3</td>
<td>6b644ec</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.4</td>
<td>156402</td>
<td>go1.6.4</td>
<td>Mon Dec 12 23:41:49 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.5</td>
<td>7392c3b</td>
<td>go1.6.4</td>
<td>Fri Dec 16 02:23:59 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.6</td>
<td>78d1802</td>
<td>go1.6.4</td>
<td>Tue Jan 10 20:20:01 2017</td>
<td>1.24</td>
</tr>
<tr>
<td>1.13.0</td>
<td>49bf474</td>
<td>go1.7.3</td>
<td>Tue Jan 17 09:55:28 2017</td>
<td>1.25</td>
</tr>
<tr>
<td>17.03.0-ce</td>
<td>3a232c8</td>
<td>go1.7.5</td>
<td>Tue Feb 28 08:10:07 2017</td>
<td>1.26</td>
</tr>
<tr>
<td>17.03.1-ce</td>
<td>c6d412c</td>
<td>go1.7.5</td>
<td>Mon Mar 27 17:05:44 2017</td>
<td>1.27</td>
</tr>
<tr>
<td>17.03.2-ce</td>
<td>f5ec1e2</td>
<td>go1.7.5</td>
<td>Tue Jun 27 02:21:36 2017</td>
<td>1.27</td>
</tr>
<tr>
<td>17.06.2-ce</td>
<td>cec0b72</td>
<td>go1.8.3</td>
<td>Tue Sep 5 20:00:25 2017</td>
<td>1.30</td>
</tr>
<tr>
<td>17.07.0-ce</td>
<td>8784753</td>
<td>go1.8.3</td>
<td>Tue Aug 29 17:43:23 2017</td>
<td>1.31</td>
</tr>
<tr>
<td>17.09.0-ce</td>
<td>afdb6d4</td>
<td>go1.8.3</td>
<td>Tue Sep 26 22:42:49 2017</td>
<td>1.32</td>
</tr>
<tr>
<td>17.09.1-ce</td>
<td>19e2ef6</td>
<td>go1.8.3</td>
<td>Thu Dec 7 22:25:03 2017</td>
<td>1.32</td>
</tr>
</tbody>
</table>
## Docker Version and API Information

<table>
<thead>
<tr>
<th>Docker Version</th>
<th>Git commit</th>
<th>Go Version</th>
<th>Build date</th>
<th>API version</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.12.0-ce</td>
<td>c97c6d6</td>
<td>go1.9.2</td>
<td>Wed Dec 27 20:12:46 2017</td>
<td>1.35</td>
</tr>
<tr>
<td>18.03.0-ce</td>
<td>0520e24</td>
<td>go1.9.4</td>
<td>Wed Mar 21 23:13:03 2018</td>
<td>1.37</td>
</tr>
<tr>
<td>18.03.1-ce</td>
<td>9ee9f40</td>
<td>go1.9.5</td>
<td>Thu Apr 26 07:23:58 2018</td>
<td>1.37</td>
</tr>
</tbody>
</table>

- CentOS 7, Kernel 3.10.0-693.2.2.el7_x86_64

<table>
<thead>
<tr>
<th>Docker Version</th>
<th>Git commit</th>
<th>Go Version</th>
<th>Build date</th>
<th>API version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.12.0</td>
<td>8eab29e</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.1</td>
<td>23cf638</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.2</td>
<td>bb80604</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.3</td>
<td>6b644ec</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.4</td>
<td>156402</td>
<td>go1.6.4</td>
<td>Mon Dec 12 23:41:49 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.5</td>
<td>7392c3b</td>
<td>go1.6.4</td>
<td>Fri Dec 16 02:23:59 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.6</td>
<td>78d1802</td>
<td>go1.6.4</td>
<td>Tue Jan 10 20:20:01 2017</td>
<td>1.24</td>
</tr>
<tr>
<td>1.13.0</td>
<td>49bf474</td>
<td>go1.7.3</td>
<td>Tue Jan 17 09:55:28 2017</td>
<td>1.25</td>
</tr>
<tr>
<td>1.13.1</td>
<td>092c9a3</td>
<td>go1.7.5</td>
<td>Web Feb 8 06:38:28 2017</td>
<td>1.26</td>
</tr>
<tr>
<td>17.03.0-ce</td>
<td>3a232c8</td>
<td>go1.7.5</td>
<td>Tue Feb 28 08:10:07 2017</td>
<td>1.26</td>
</tr>
<tr>
<td>17.03.1-ce</td>
<td>c6d412e</td>
<td>go1.7.5</td>
<td>Mon Mar 27 17:05:44 2017</td>
<td>1.27</td>
</tr>
<tr>
<td>17.03.2-ce</td>
<td>5f8c8e2</td>
<td>go1.7.5</td>
<td>Tue Jun 27 02:21:36 2017</td>
<td>1.27</td>
</tr>
<tr>
<td>17.06.2-ce</td>
<td>cec0b72</td>
<td>go1.8.3</td>
<td>Tue Sep 5 20:00:25 2017</td>
<td>1.30</td>
</tr>
<tr>
<td>17.07.0-ce</td>
<td>8784753</td>
<td>go1.8.3</td>
<td>Tue Aug 29 17:43:23 2017</td>
<td>1.31</td>
</tr>
<tr>
<td>17.09.0-ce</td>
<td>afdb6d4</td>
<td>go1.8.3</td>
<td>Tue Sep 26 22:42:49 2017</td>
<td>1.32</td>
</tr>
<tr>
<td>17.09.1-ce</td>
<td>19e2c6f6</td>
<td>go1.8.3</td>
<td>Thu Dec 7 22:25:03 2017</td>
<td>1.32</td>
</tr>
<tr>
<td>17.12.0-ce</td>
<td>c97c6d6</td>
<td>go1.9.2</td>
<td>Wed Dec 27 20:12:46 2017</td>
<td>1.35</td>
</tr>
<tr>
<td>18.03.0-ce</td>
<td>0520e24</td>
<td>go1.9.4</td>
<td>Wed Mar 21 23:13:03 2018</td>
<td>1.37</td>
</tr>
<tr>
<td>18.03.1-ce</td>
<td>9ee9f40</td>
<td>go1.9.5</td>
<td>Thu Apr 26 07:23:58 2018</td>
<td>1.37</td>
</tr>
</tbody>
</table>

Note: The version 1.13.1 is the one provided by Docker. The version 1.13.1-63 provided by CentOS is broken and NOT supported by Univa Grid Engine!

- Ubuntu 16.04.3 LTS, Kernel 4.4.0-103-generic x86_64
<table>
<thead>
<tr>
<th>Docker Version</th>
<th>Git commit</th>
<th>Go Version</th>
<th>Build date</th>
<th>API version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.12.0</td>
<td>8eab29e</td>
<td>go1.6.3</td>
<td>Thu Jul 28 22:11:10 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.1</td>
<td>23cf638</td>
<td>go1.6.3</td>
<td>Thu Aug 18 05:33:38 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.2</td>
<td>bb80604</td>
<td>go1.6.3</td>
<td>Tue Oct 11 18:29:41 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.3</td>
<td>6b644ec</td>
<td>go1.6.3</td>
<td>Wed Oct 26 22:01:48 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.4</td>
<td>1564f02</td>
<td>go1.6.4</td>
<td>Tue Dec 13 00:08:34 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.5</td>
<td>7392c3b</td>
<td>go1.6.4</td>
<td>Fri Dec 16 02:42:17 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.6</td>
<td>78d1802</td>
<td>go1.6.4</td>
<td>Tue Jan 10 20:38:45 2017</td>
<td>1.24</td>
</tr>
<tr>
<td>1.13.0</td>
<td>49bf474</td>
<td>go1.7.3</td>
<td>Tue Jan 17 09:58:26 2017</td>
<td>1.25</td>
</tr>
<tr>
<td>17.03.0-ce</td>
<td>3a232c8</td>
<td>go1.7.5</td>
<td>Tue Feb 28 08:01:32 2017</td>
<td>1.26</td>
</tr>
<tr>
<td>17.03.1-ce</td>
<td>c6d412e</td>
<td>go1.7.5</td>
<td>Mon Mar 27 17:14:09 2017</td>
<td>1.27</td>
</tr>
<tr>
<td>17.03.2-ce</td>
<td>f5ec1e2</td>
<td>go1.7.5</td>
<td>Tue Jun 27 03:35:14 2017</td>
<td>1.27</td>
</tr>
<tr>
<td>17.06.2-ce</td>
<td>cec0b72</td>
<td>go1.8.3</td>
<td>Tue Sep 5 19:59:11 2017</td>
<td>1.30</td>
</tr>
<tr>
<td>17.09.0-ce</td>
<td>af8b6d4</td>
<td>go1.8.3</td>
<td>Tue Sep 26 22:40:56 2017</td>
<td>1.32</td>
</tr>
<tr>
<td>17.09.1-ce</td>
<td>19e2cf6</td>
<td>go1.8.3</td>
<td>Thu Dec 7 22:23:00 2017</td>
<td>1.32</td>
</tr>
<tr>
<td>17.12.0-ce</td>
<td>c97c6d6</td>
<td>go1.9.2</td>
<td>Wed Dec 27 20:09:53 2017</td>
<td>1.35</td>
</tr>
<tr>
<td>18.03.0-ce</td>
<td>0520e24</td>
<td>go1.9.4</td>
<td>Wed Mar 21 23:08:31 2018</td>
<td>1.37</td>
</tr>
<tr>
<td>18.03.1-ce</td>
<td>9ec9f40</td>
<td>go1.9.5</td>
<td>Thu Apr 26 07:15:30 2018</td>
<td>1.37</td>
</tr>
</tbody>
</table>

- Ubuntu 16.10, Kernel 4.8.0-59-generic x86_64

<table>
<thead>
<tr>
<th>Docker Version</th>
<th>Git commit</th>
<th>Go Version</th>
<th>Build date</th>
<th>API version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.13.0</td>
<td>49bf474</td>
<td>go1.7.3</td>
<td>Tue Jan 17 10:05:19 2017</td>
<td>1.25</td>
</tr>
<tr>
<td>17.03.0-ce</td>
<td>3a232c8</td>
<td>go1.7.5</td>
<td>Tue Feb 28 08:05:01 2017</td>
<td>1.26</td>
</tr>
<tr>
<td>17.03.1-ce</td>
<td>c6d412e</td>
<td>go1.7.5</td>
<td>Mon Mar 27 17:17:43 2017</td>
<td>1.27</td>
</tr>
<tr>
<td>17.03.2-ce</td>
<td>f5ec1e2</td>
<td>go1.7.5</td>
<td>Tue Jun 27 03:59:22 2017</td>
<td>1.27</td>
</tr>
</tbody>
</table>

- Ubuntu 17.04, Kernel 4.10.0-42-generic x86_64

<table>
<thead>
<tr>
<th>Docker Version</th>
<th>Git commit</th>
<th>Go Version</th>
<th>Build date</th>
<th>API version</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.09.0-ce</td>
<td>af8b6d4</td>
<td>go1.8.3</td>
<td>Tue Sep 26 22:41:24 2017</td>
<td>1.32</td>
</tr>
<tr>
<td>17.12.0-ce</td>
<td>c97c6d6</td>
<td>go1.9.2</td>
<td>Wed Dec 27 20:09:19 2017</td>
<td>1.35</td>
</tr>
</tbody>
</table>

* Grid Engine Release Notes v 8.6.0 *
```markdown
### Contents

- **Ubuntu 17.10, Kernel 4.13.0-19-generic x86_64**

<table>
<thead>
<tr>
<th>Docker Version</th>
<th>Git commit</th>
<th>Go Version</th>
<th>Build date</th>
<th>API version</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.06.2-ce</td>
<td>cec0b72</td>
<td>go1.8.3</td>
<td>Tue Sep 5 19:57:44 2017</td>
<td>1.30</td>
</tr>
<tr>
<td>17.09.0-ce</td>
<td>afd16d4</td>
<td>go1.8.3</td>
<td>Tue Sep 26 22:41:24 2017</td>
<td>1.32</td>
</tr>
<tr>
<td>17.09.1-ce</td>
<td>19e2c6f6</td>
<td>go1.8.3</td>
<td>Thu Dec 7 22:23:07 2017</td>
<td>1.32</td>
</tr>
<tr>
<td>17.12.0-ce</td>
<td>e97c6d6</td>
<td>go1.9.2</td>
<td>Wed Dec 27 20:09:47 2017</td>
<td>1.35</td>
</tr>
</tbody>
</table>

- **Ubuntu 18.04, Kernel 4.15.0-24-generic x86_64**

<table>
<thead>
<tr>
<th>Docker Version</th>
<th>Git commit</th>
<th>Go Version</th>
<th>Build date</th>
<th>API version</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.12.0-ce</td>
<td>e97c6d6</td>
<td>go1.9.2</td>
<td>Wed Dec 27 20:09:47 2017</td>
<td>1.35</td>
</tr>
<tr>
<td>18.03.0-ce</td>
<td>0520e24</td>
<td>go1.9.4</td>
<td>Wed Mar 21 23:08:36 2018</td>
<td>1.37</td>
</tr>
<tr>
<td>18.03.1-ce</td>
<td>9ec9f40</td>
<td>go1.9.5</td>
<td>Thu Apr 26 07:15:45 2018</td>
<td>1.37</td>
</tr>
</tbody>
</table>

- **openSUSE Leap 42.3, Kernel 4.4.92-31-default x86_64**

<table>
<thead>
<tr>
<th>Docker Version</th>
<th>Git commit</th>
<th>Go Version</th>
<th>Build date</th>
<th>API version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.12.0</td>
<td>8eab29e</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.1</td>
<td>23cf638</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.2</td>
<td>bb80604</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.3</td>
<td>6b6443e</td>
<td>go1.6.3</td>
<td></td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.4</td>
<td>156f802</td>
<td>go1.6.4</td>
<td>Mon Dec 12 23:41:28 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.5</td>
<td>7392c3b</td>
<td>go1.6.4</td>
<td>Fri Dec 16 02:24:38 2016</td>
<td>1.24</td>
</tr>
<tr>
<td>1.12.6</td>
<td>78l1802</td>
<td>go1.6.4</td>
<td>Tue Jan 10 20:20:13 2017</td>
<td>1.24</td>
</tr>
<tr>
<td>1.13.0</td>
<td>49bf474</td>
<td>go1.7.3</td>
<td>Tue Jan 17 10:09:08 2017</td>
<td>1.25</td>
</tr>
<tr>
<td>17.03.0-ce</td>
<td>60cbe22</td>
<td>go1.7.5</td>
<td>Thu Feb 23 10:55:03 2017</td>
<td>1.26</td>
</tr>
<tr>
<td>17.03.1-ce</td>
<td>e6d412c</td>
<td>go1.7.5</td>
<td>Fri Mar 24 00:53:12 2017</td>
<td>1.27</td>
</tr>
<tr>
<td>17.09.1-ce</td>
<td>f4f2d25</td>
<td>go1.8.7</td>
<td>Tue Jun 12 12:05:08 2018</td>
<td>1.32</td>
</tr>
</tbody>
</table>
```
3.1 Known Docker Issues That Affect Univa Grid Engine

- With Docker 17.09.0-ce and Docker 17.12.0-ce, specifying the `--oom-kill-disable` switch has no effect, when using the `docker` command line client nor the `-xd "--oom-kill-disable"` switch of the Univa Grid Engine submit clients.
4 Fixes and Enhancements

4.1 Major Enhancements

4.1.1 New Core Binding Strategies and PE Behavior

For PE-jobs it is hard or even impossible to know in advance how many tasks are going to be scheduled and on how many hosts. Therefore, with Univa Grid Engine version 8.6.0, the core-binding request behavior has changed to accommodate this fact. The binding-request is now changed to a “per PE-task” request where previously it was a “per host” request. This means that the requested amount of cores for a PE-job are assigned “per PE-task”.

For example, if a job with

```
qsub -pe mype 5-7 -binding linear:2 -b y sleep 60
```

is submitted, it means that each PE-task will get 2 cores, no matter on which host or on how many hosts the tasks are scheduled.

There are now different binding-strategies, most of them exist in two versions: “host aware” and “host unaware” strategies. For example, there are two versions of linear binding strategies: `linear` and `linear_per_task`. Host unaware strategies have the suffix “_per_task”.

With “host aware” strategies, all tasks that end up on a host have to adhere to the given strategy together. For “host unaware” strategies, each task has to adhere to the strategy on its own. This is less strict and usually more tasks can fit on a host.

All available core binding strategies are:

- `linear:<amount>[:<socket>,<core>]`
- `linear_per_task:<amount>`
- `striding:<amount>:<n>[:<socket>,<core>]`
- `striding_per_task:<amount>:<n>`
- `explicit:[<socket>,<core>;...]<socket>,<core>`
- `explicit_per_task:[<socket>,<core>;...]<socket>,<core>`
- `balance_sockets:<amount>`
- `pack_sockets:<amount>`
- `one_socket_balance:<amount>`
- `one_socket_per_task:<amount>`

See man page `submit(5)` for more details and examples.

4.1.2 Affinity Based Job Placement

Univa Grid Engine 8.6.0 adds an affinity job placement policy that allows assignment to each host or queue an affinity value for each consumed resource of jobs that are running on the host or queue.
Affinity can be positive or negative. Positive affinity will attract other pending jobs, negative affinity will reject other pending jobs. Attraction/rejection will work on host and/or queue level if this is enabled by setting the weighting parameters \texttt{weight\_host\_affinity} and/or \texttt{weight\_queue\_affinity}.

Sorting based on this affinity value will cause

- affinity (so that jobs are packed on clusters of hosts or queues),
- anti-affinity (so that jobs are distributed on hosts in the cluster or queues residing on hosts)
- or best fit (if a mixture of positive and negative affinity values is defined for different resources)

Find more information concerning job placement according to affinity values in the section 3.7.1 (Host/Queue Sorting) of the Admin Guide.

### 4.1.3 Managing Access to Devices with RSMAPs

Univa Grid Engine 8.6.0 allows to manage access to host devices via RSMAPs. Each id of a RSMAP complex can be configured to represent a device on the host by setting the new parameter "device". Each device can be represented by more than one RSMAP id. In the example below a RSMAP complex \texttt{gpu} is initialized with two ids and each id is mapped to a Nvidia GPU:

\[
\text{complex\_values} \quad \text{gpu}=2(\text{gpu0[device=/dev/nvidia0]} \ \text{gpu1[device=/dev/nvidia1]})
\]

The assigned devices are shown in the qstat output of a job:

\[
\text{granted devices} \quad \text{host: /dev/nvidia0}
\]

In a default environment the configuration and assignment of devices has no effect on the scheduling, but if cgroups are available the cgroups parameter "devices" can be set to a list of devices that should be managed by Univa Grid Engine. Read/write access to all devices in the list will be blocked via cgroups and jobs will only be able to access devices that were assigned to them via RSMAPs. With the following configuration Univa Grid Engine will manage access to all Nvidia GPUs (i.e. all devices from /dev/nvidia0 to /dev/nvidia255):

\[
\text{cgroups\_params cgroup\_path=/sys/fs/cgroups devices=/dev/nvidia[0-255]}
\]

### 4.1.4 Integration with Nvidia DCGM

Univa Grid Engine 8.6.0 is integrated with NVIDIA’s Data Center GPU Manager (DCGM) that provides detailed information about GPU resources.

Support for DCGM can be enabled on host level by setting the execd parameter \texttt{UGE\_DCGM\_PORT} to the port DCGM uses to communicate on the specific host. If DCGM is running Univa Grid Engine will automatically retrieve load values for the installed and supported GPU devices from DCGM. For each available device the load values are reported in the format \texttt{<hostname>\_cuda.<cuda\_id>\_<attribute>=<value>} and are visible via qconf -se:
If RSMAP complexes are used to manage GPU devices, each RSMAP id can be mapped to a CUDA device with the new parameter `cuda_id`. The reported load values can then be used by UGE during the scheduling of GPU devices:

```
complex_values gpu=2(gpu0[cuda_id=0] gpu1[cuda_id=1])
```

If DCGM is enabled, Univa Grid Engine 8.6.0 allows requesting the special load value `affinity`. If a job requests a GPU and `affinity`, it will automatically be bound to the cores that have a good affinity to the assigned GPU. This ensures that the data between the CPU and GPU is transferred in the fastest way possible. Currently `affinity` is treated as a hard request, if it is requested and Univa Grid Engine cannot bind the CPU cores needed for a GPU device, the job will not be scheduled. If less cores are needed the request can be combined with the `-binding` switch.

The following requests a GPU device and binds the cores that have a good affinity to the assigned GPU:

```
% qsub -l gpu=1[affinity=true] -b y sleep 1000
```

### 4.1.5 Performance Improvements

Performance and Scalability of Univa Grid Engine has been improved in the following areas:

- general improvements affecting all components by optimizations of the Univa Grid Engine data store and updated memory allocator (jemalloc library)
- scheduler optimizations for various job profiles, esp. for parallel jobs with and without resource reservation
- higher throughput of the communication library by introducing an additional thread pool for handling incoming/outgoing data
• compression of communication data can reduce network load and avoid network saturation on slow network connections
• the impact of running many qsub -sync, drmaa or drmaa2 clients on sge_qmaster has been reduced

4.1.6 Reworked Dispatch Job Information

The mechanisms that provide Dispatch Job information have been reworked and many issues have been fixed.

`qalter -w p` is deprecated. Instead administrators should enable scheduler job information by setting `schedd_job_info` to `if_requested` in the scheduler configuration so that users can request dispatch job information for individual jobs via `-rdi` submit switch and make the Dispatch job information visible with `qstat -j jid`

Administrators have additional possibilities to define limits that reduce memory requirements for dispatch job information.

4.1.7 Data Compression at Communication Layer

The Univa Grid Engine communication library was enhanced to support data compression before transferring data over the network.

Large Univa Grid Engine clusters may produce high network data traffic. The network load depends mainly on cluster size and average job run times. In order to reduce the amount of data sent over the network Univa Grid Engine 8.6.0 supports data compression.

The additional effort to compress the data before sending is handled by introducing a commlib work thread pool which is also used for uncompressed data. On multi core architectures uncompressed data transfer should show a performance improvement.

Setting up the new compression mode and configure thread pool settings in Univa Grid Engine requires new configuration parameters that are specified in the following Univa Grid Engine configuration areas:

<table>
<thead>
<tr>
<th>man page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bootstrap(5)</td>
<td>New parameter <code>communication_params</code> in bootstrap file</td>
</tr>
<tr>
<td>sge__diagnostics(5)</td>
<td>Verify and Adjust compression and thread pool setup</td>
</tr>
<tr>
<td>sge__conf(5)</td>
<td>New <code>qmaster_params</code> and <code>execd_params</code> parameter <code>CL_WP_THREADS</code> for setting up work threads (overwrite bootstrap settings)</td>
</tr>
<tr>
<td>qping(1)</td>
<td>New compression specific output and options</td>
</tr>
<tr>
<td></td>
<td>(see also qping enhancements section below)</td>
</tr>
</tbody>
</table>

Summary of changes and enhancements:
• Compression cannot be configured at Installation time. It must be enabled manually after installation or update of Univa Grid Engine.
• Univa Grid Engine 8.6.0 uses zlib for data compression - additional compression methods are planned to be supported in follow-up Univa Grid Engine releases.
• Univa Grid Engine 8.6.0 zlib compression is supported for following architectures: darwin-x64, lx-amd64, lx-arm64, lx-arm7, lx-x86, sol-amd64, sol-sparc64, sol-x86
• The qping binary can be used to verify compression settings.
• It is possible to enable compression only for single clients like (qconf, qstat) by setting up an environment variable (see bootstrap(5) man page).
• Compression setting changes in bootstrap configuration require a restart of Univa Grid Engine daemons.
• Commlib work pool setting changes at sge_qmaster and sge_execd can be done during runtime of the affected daemon.
• Univa Grid Engine 8.6.0 commlib modifications show a higher performance compared to older Univa Grid Engine versions if compression is not enabled; however this depends on availability of cpus on sge_qmaster host.
• Turning on compression will cause cpu overhead but dramatically reduces network traffic. Depending on the network speed individual compression settings may need to be adjusted (see sge_diagnostics(5) man page).

Also qping is enhanced for data compression and commlib work pool settings

• qping -dump shows new data compression specific columns (compressed and uncompressed message length, compression ratio and method)
• qping -info shows active commlib work threads and min/max settings for the requested daemon
• The sge_conf(5) qmaster_params PROF_COMMLIB_TIME now shows also the nr of active work threads used by commlib in the sge_qmaster messages log file.
• qping has new command line switches for basic output filtering (-from, -to and -format)

All these additions are described in the qping(1) man page.

4.1.8 Using RSMAPs with Topology Masks and XOR Operator

RSMAP complex attributes with topology masks and the XOR operator (\^\^) can be combined to achieve flexible PE task placement and best application performance.

Assume the following definition of the complex attributes gpu and hgpu:

```
$ qconf -sc|egrep "#name|gpu"
#name shortcut type relop requestable consumable default urgency aapre
#-----------------------------------------------------------------------------
gpu  gpu   RSMAP <= YES  YES  0  0  NO
hgpu hgpu   RSMAP <= YES HOST 0  0  NO
```

Sample setting on a node on host level for complex_values:
2 sockets, 14 cores per socket, 28 sockets in total

"grid engine release notes v 8.6.0"
Assume the _allocation_rule_ of the _mpi_ PE is 28.

The first example reserves 4 machines with 28 slots on each machine for one hour immediately. Host level resources are only reserved in the second example (4 _hgpu_ s per host, 16 in total):

```bash
% qrsub -pe mpi 112 -d 1:0:0
% qrsub -pe mpi 112 -l hgpu=4 -d 1:0:0
```

Submit a job with 4 PE tasks on one host and request 4 _hgpus_:

```bash
% qsub [-ar <id>] -pe mpi 4 -par 4 -l hgpu=4
```

While only 4 queue slots on the host will be used, the topology mask of the _hgpu_ resource will mark all 28 cores to be in use. If the cluster is configured that all jobs are submitted with a binding request no other jobs will be dispatched to this machine.

Schedule a job on 4 machines and get any two _gpus_ on each host.

```bash
% qsub [-ar <id>] -pe mpi 4 -par 1 -l hgpu=2
```

A parallel job which should get the same _gpu_ for each PE task on each host would be submitted as follows:

```bash
% qsub [-ar <id>] -pe mpi 4 -par 1 -l 'hgpu=1(\"\")'
```

This is a shortcut for

```bash
% qsub [-ar <id>] -pe mpi 4 -par 1 -l 'hgpu=1(1**2**3**4)'
```

If the job should get the two PE tasks with one _gpu_ for each task on the same socket on each host, the XOR operator can be used:

```bash
% qsub [-ar <id>] -pe mpi 8 -par 2 -l 'hgpu=2([0-1]**[2-3])'
```

### 4.2 Other

#### 4.2.1 JSV Improvements

The Univa Grid Engine packages now contain a Python implementation for JSVs. Also new example scripts for JSV and core-binding were added.

Scripts and examples can be found at “$SGE_ROOT/util/resources/jsv”.
4.2.2 Reduce qhost Data Request Sizes at sge_qmaster

By setting the environment variable `SGE_GDI_REQUEST_REDUCE_LEVEL` it is possible to reduce the amount of data transferred from `sge_qmaster` to qhost clients. A detailed description can be found in the `qhost(1)` man page (ENVIRONMENTAL VARIABLES).

4.2.3 Upgrade Advance and Standing Reservations

When upgrading from Univa Grid Engine versions 8.5.4 and higher to Univa Grid Engine 8.6.0 it is now possible to upgrade advance and standing reservations as well by calling

```bash
# ./inst_sge -upd-ars
```

after the qmaster and exed upgrade are completed.

Prerequisite is that the host on which the upgrade is done (usually the master host) is a submit host and can resolve the user names of the AR owners and in AR acl_list/xacl_list.

See also the chapter about upgrading in the Installation Guide.

4.2.4 per_pe_task_prolog and per_pe_task_epilog

It is now possible to define a special prolog and epilog script that is started for each slave task of a tightly integrated parallel job that is started by `qrsh -inherit`. The prolog is started before the slave task command or script, the epilog is started after it. See man page `sge_pe(5)` for details.

4.2.5 The Master Task of a Parallel Docker Job in a Container

The master task of a parallel Docker job can now be started in a container. Previously, the master task was started on the host itself but not in a container, while all slave tasks were started in containers. This new behavior requires a properly setup network, container network and DNS. See the Administrators Guide for details.

4.2.6 Run the Container as root, Allow to Run Prolog etc. as a Different User

With the `execd_params START_CONTAINER_AS_ROOT` it is now possible to start all Docker containers as root and allow the `prolog`, `pe_start`, `per_pe_task_prolog`, `per_pe_task_epilog`, `pe_stop` and epilog scripts to be started as a different user than the job owner. This change does not apply to “autostart Docker jobs”, i.e. jobs that specify `-b y NONE` as job script in order to use the entrypoint that is defined in the Docker image instead of using the `sge_container_shepherd` as the container entrypoint.
4.2.7 Automatically Map User ID and Group ID of a User Into the Container

If the `START_CONTAINER_AS_ROOT` parameter is set to “true”, it is now necessary that the Univa Grid Engine admin user, the job user and all pre and post script users are defined inside the container. Because this is usually not the case, by setting the `AUTOMAP_CONTAINER_USERS` parameter to “TEMPORARY”, Univa Grid Engine transfers the user ID and group ID of any of these users from the host to the container. But only Univa Grid Engine itself can use this user ID information there, it is not available for the job or any of the scripts started by Univa Grid Engine!

If `AUTOMAP_CONTAINER_USERS` is set to `PERSISTENT`, Univa Grid Engine writes an entry to the “/etc/passwd” file inside the Docker container for all these users. This allows lookup of the user information in a script, but it does not allow to switch to this user!

Caution! If `AUTOMAP_CONTAINER_USERS=PERSISTENT` is specified, if an user maps the “/etc/passwd” and “/etc/group” file into the container, the host files are modified!

4.2.8 Create a container_pe_hostfile with all Container Hostnames

If a parallel Docker job is started where the container hostnames are selected from RSMAPs, the execution daemon of the master task writes a `container_pe_hostfile` with all the container hostnames in the `pe_hostfile` format if the `execd_params CONTAINER_PE_HOSTFILE_COMPLEX` is set to the name of the RSMAP complex that defines the hostnames.

E.g.: If there is a RSMAP “cont_hosts” declared and on each execution host it defines values like:

```
cont_host=4(host1_cont1 host1_cont2 host1_cont3 host1_cont4)
```

and a job is submitted using

```
# qsub -pe mype 4 -l docker,docker_images="*image:latest*",cont_host=1 job_script.sh
```

and the scheduler decides to schedule the master task to host1, two slave tasks to host2 and one slave task to host3, the “container_pe_hostfile” might contain:

```
host1_cont3 1 <NULL> <NULL>
host2_cont1 1 <NULL> <NULL>
host2_cont4 1 <NULL> <NULL>
host3_cont2 1 <NULL> <NULL>
```

This allows to read this information in a `per_pe_task_prolog` and set the hostnames of the containers inside of the containers accordingly.

4.2.9 Docker Daemon Response Timeout

A new `execd_params DOCKER_RESPONSE_TIMEOUT` is introduced. This defines the time Univa Grid Engine waits for a response from the Docker daemon to a request Univa Grid Engine sent to the Docker daemon. This does not mean the full response must be received within the timeout;

Grid Engine Release Notes v 8.6.0
the timeout counter is reset after each character Univa Grid Engine receives from the Docker daemon in response to a specific request.

If this parameter is not specified, the default value of 60 s is used. The minimum timeout is 10 s, the maximum timeout is 86400 s. If `DOCKER_RESPONSE_TIMEOUT` is not within this range, the default value is used.

### 4.2.10 Cgroups and Containers

The cgroups generated by Univa Grid Engine are now used in Docker containers. There is no special configuration value, when Univa Grid Engine is configured to use cgroups, these cgroups are also used within Docker containers.

There is a known issue with cgroups and Docker that is not yet understood:

On some Linux distributions some Docker versions expects the cgroups path to contain a “slice” postfix, which makes Univa Grid Engine unable to start Docker jobs in cgroups - the jobs will fail then. This was observed on CentOS 7 with Docker 1.12.6 build c4618fb, while it was not observed with Docker 1.12.6 build 78d1802 on the same host. It was also not observed on different distributions than CentOS 7.

There are indications that it could be related to the Docker SELinux packages that are provided for CentOS 7, if they do not match the Docker version exactly, this error seems to be more likely.

### 4.2.11 Specify Arguments to Autostart Docker Jobs

The autostart Docker jobs, i.e. Docker jobs that were submitted as binary jobs with the keyword `NONE` as job script, now accept arguments. These arguments are provided to the binary or script defined in the `ENTRYPOINT` of the Docker image and are appended to arguments that are defined in the `ENTRYPOINT`. This works only with suitable Docker images. Whether a Docker image is suitable can be tested by manually starting

```bash
$ docker run -it image:latest arg1 arg2 arg3
```

on the execution host. If the script or binary defined in the `ENTRYPOINT` gets these arguments, it should also work with an Univa Grid Engine job.

### 4.2.12 New Client Command qralter

A new `qralter` client command has been added to Univa Grid Engine allowing to modify existing advance reservations, see User Guide -> Reservations.

### 4.2.13 Changes to the loadcheck Command

`loadcheck` will now optionally output information about GPUs available on a host like the GPU name, available memory and cpu affinity.

`loadcheck` outputs a name and a value column, with Univa Grid Engine 8.6.0 the width of the name column has been increased.
4.3 Full List of Fixes and Enhancements

4.3.1 8.6.0 (and Also Fixed for a 8.5.*, 8.4.*, 8.3.* Patch Release)

- GE-3721 qstat -j "*" -u "user1" is not working
- GE-5290 qstat (-xml) does not accept filter switches when -j "*" is specified
- GE-5831 UGE REST Server Crashing when POSTing new complex
- GE-5941 renewing certificates makes sgepasswd file unreadable
- GE-6575 stree-edit utility broken
- GE-6697 qstat -j "*" does not support "$user" placeholder set in sge_qstat request file
- GE-6734 wrong scheduler info messages shown for jobs
- GE-6764 very long dispatching time due to RQS rule result in scheduler timeout
- UWSA-77 Basic authentication fails under solaris frequently
- UWSA-81 Add support for listening only on localhost
- UWSA-169 REST Service: 'none' auth method doesn't work
- UWSA-186 requested jobEnvironment is not shown in jobs
- UWSA-188 jobEnvironment ugeresdk contains error in json converter
- UWSA-189 ugerest is showing same scheduler conf twice
- UWSA-190 upgrade of restlet-jse-2.3.6 to restlet-jse-2.3.10
- UWSA-199 UGERest and other jar files do not show their build version
- UWSA-200 Add version info route to ugerest

4.3.2 8.6.0 (and Also Fixed for a 8.5.* and 8.4.* Patch Release)

- GE-4389 enhance qsub man page with JSV modification examples for core binding
- GE-5835 long scheduling times with wildcard PEs and resource reservation
- GE-6018 Setting limits based on percentage.
- GE-6103 jobs are bound to cores even if no binding is requested
- GE-6356 drmaa2_j_get_info does not provide full job information -> not all fields are filled
- GE-6402 Scheduler might not respect RQS limits during the time RQS rules are changed
- GE-6432 qdel -u "*" is only allowed to managers, not to operators
- GE-6462 on native Windows (win-x86), environment variable values containing an equal sign are truncated
- GE-6478 Very long load value of loadsensor causes segfault of execd
- GE-6510 Core binding: striding-strategy counts needed cores wrong and rejects viable hosts
- GE-6511 gdi_request_limits does not behave as documented
- GE-6534 sge_execd crash with core dump with GPGPU jobs
- GE-6568 Allow for forced job deletion through UGE REST
- GE-6595 Docker interactive job can't be deleted by qdel
- GE-6671 sudo requests for same user as ugerest service user are rejected
- GE-6682 exec host cannot startup if no admin or submit host
GE-6709 huge virtual memory requirements with test DRMAA1 application
GE-6728 sched_conf.5 man page is inaccurate regarding the
PREFER_SOFT_REQUESTS scheduler param
GE-6771 qsub -sync stops immediately on native Windows (win-x86)
GE-6787 qmaster and execd logging "invalid value (33026) for ar->op"
GE-6855 sge_qmaster abort() with critical logging "got NULL element
for RUE_used_now"
GE-7237 UGE qmaster daemon may crash in 'lo_thread' thread
GE-7240 UGE qmaster does not handle LO delete cluster request
UWSA-193 additional job usage values for execd_params
   ENABLE_MEM_DETAILS=1 are missing

4.3.3 8.6.0 (and Also Fixed for a 8.5.* Patch Release)

GE-285 allow parallel job allocation scheme be specified at submission time
GE-3146 backfilling a resource reservation before a calendar or an AR is broken
GE-4305 Better documentation for basic share tree use case
GE-4726 adding a way that multiple jobs can be bound to the same socket
   using all cores
GE-5436 qmon should not reset value "-1" for sharetree usage_weight_list list
GE-5569 native Windows (win-x86) sge_execd exits if it cannot access the
   act_qmaster file
GE-5636 qmaster error logging "rc_add_job_utilization 0 slot amount" indicates
   wrong job reservation
GE-5650 with allocation rule $fill_up, slave resource requests are not obeyed
   when master resource and queue requests are provided
GE-5806 parallel jobs might not startup due to wrong RQS calculations
GE-5830 Scheduler fails to handle a job submitted with option "-mbind cores"
GE-5848 allow to use UGE cgroups in Docker jobs
GE-6020 allow native Windows (win-x86) functions to retry to logon users
   several times
GE-6105 RSMAP attributes should support per slot resource allocation for PE jobs
GE-6183 sge_shepherd sets limits too high for master task
GE-6236 accounting file broken on Solaris if accounting line has 1023 characters
GE-6237 jobs are being restarted even if the '-r no' was specified
   during submission
GE-6270 allow to run prolog etc. as root inside a Docker container
GE-6271 sge_container_shepherd fails if the prolog is started as root
GE-6407 add documentation for config-api
GE-6413 shepherd does not handle all error responses to a pull Docker image request
GE-6434 incomplete binding requests shown in qstat for long lists of binding requests
GE-6454 improve documentation of Docker integration
GE-6480 wrong/missing error messages and wrong exit status when initializing
   invalid RSMAP ranges
GE-6483 document placeholders in Docker requests in the UserGuide and man pages
GE-6489 using of external load sensors might deadlock execution daemon
GE-6490 load sensor specific errors are not logged into execd messages file
GE-6493 RSMAP map entry selection request does not work
Contents

- GE-6497 support halftime -1 setting in scheduler config to disable past usage for sharetree
- GE-6506 qstat does not show granted PE
- GE-6508 Accounting shows wrong fail and exit states for 'qdel -f jobID'
- GE-6521 Qmaster Crashes with LO Enabled and Job Dependencies
- GE-6524 AR shouldn't be scheduled to unknown queue instances at time of submission
- GE-6527 document qsub -xd --help
- GE-6528 allow to specify arguments to autostart Docker jobs
- GE-6536 No error message when a job cannot get scheduled due to RSMAP-ranges
- GE-6537 Submitting RSMAP-range jobs via -adds is broken
- GE-6549 qstat/qhost outgoing packet size is much bigger than the data which is finally displayed
- GE-6553 RQS limits incorrectly applied when PE job submitted with "-l h=<host>"
- GE-6566 qsub rejects valid RSMAP resource requests
- GE-6569 Allow for Deletion of non jobsession jobs in DRMAA2
- GE-6573 reduce resulting qhost client requested data sizes transferred via network
- GE-6583 scheduler is wrongly skipping hosts or queue instances for parallel jobs that are using master task specific requests
- GE-6597 execd uses 100% CPU if it can't delete a finished Docker container
- GE-6615 enhance error logging if load sensor cannot be started
- GE-6617 provide workaround in native Windows (win-x86) sgeexecd.bat script for start /b bug on Windows 10, version 10.0.15063
- GE-6618 native Windows (win-x86) execution daemon crashes if qloadsensor does not work
- GE-6620 error messages from qloadsensor.exe (win-x86) are not forwarded to the execd messages file
- GE-6623 add the UGE admin user to the Performance Monitor Users group on native Windows 10 (win-x86)
- GE-6629 qhost NSOC and NCOR incorrect on lx-arm64
- GE-6637 using hostgroups in rqs limit definition can trigger short qmaster hang at startup or rqs modify request
- GE-6640 qloadsensor.exe report "no error" if an error occurs while initializing the PDH service
- GE-6644 any epilog SIGSEGVs and sets queue in error state with execed_params INHERIT_ENV=false
- GE-6646 When host aliases are configured qsub --sync y reports "commlib info: successfully updated host aliases (add: 0, del: 0)"
- GE-6647 while a non allocated reservation is "active" in a standing reservation jobs submitted into the SR get rejected
- GE-6670 The global configuration parameter "gdi_request_limits" not working for aliased hostnames
- GE-6672 Improve accept() handling in commlib
- GE-6678 RSMAP-topology-masks jobs and -binding jobs result in wrong scheduling decisions
- GE-6695 qrstat does not output cal_depth and cal_jmp information for standing reservations
- GE-6696 save_sge_config.sh needs to dump advance/standing reservations for upgrade to newer versions
- GE-6706 unexpected logging and possibly incorrect accounting if multiple
array tasks of a job are running on a host
GE-6713 shadow on Solaris cannot start sge_qmaster
GE-6737 queue calendars closing queues not considered when max_reservation > 0
GE-6739 Parallel job requesting pe range not scheduled even if resources available
GE-6741 jobs submitted into AR with RSMAP resources are not scheduled
GE-6744 qrstat does not output queue request (-q) and immediate request (-now y)
GE-6746 support a "pack socket" core binding strategy
GE-6754 Adding new session (qconf -asi) not working on admin only host
GE-6755 save_sge_config.sh does not dump all advance/standing reservations
GE-6774 Jobs do not get the correct binding when a specific RSMAP id with topology mask is requested
GE-6775 RSMAP ids are granted even if the requested cores are already in use
GE-6777 huge erroneous reader thread logging at calendar state transition
GE-6782 qstat -njd is not working as documented in the man page
GE-6785 Supplementary groups in manager and operator lists are ignored during access validation for job deletions.
GE-6786 false logging for 'qmaster_params': 'gdi_timeout', 'gdi_retries' and 'gdi_ping'
GE-6788 qmaster logging about receiving older load report
GE-6796 Calendar modification/state transition might cause repeated timed calendar events for up to one second.
GE-6799 unexpected logging messages in sge_qmaster messages file
GE-6800 implement a per task prolog and epilog for tightly integrated parallel jobs
GE-6812 started docker job at execd may result in sge_shepherd process eating up all memory
GE-6814 removing non "lo_*" complex entry triggers error logging of lothread
GE-6818 error logging: getgrgid(...) failed: Numerical result out of range
GE-6826 Spaces within a comma separated list of switch arguments are not allowed
GE-6829 disallow mapping /etc/passwd and /etc/group into the container if AUTOMAP_CONTAINER_USERS=PERSISTENT is specified
GE-6835 update host configurations for changed hostnames not working if a changed hostname is matching as substring of another host
GE-6836 improve execd reconnect behavior after qmaster shutdown and restart
GE-6841 allow to configure the Docker daemon communication timeout
GE-6843 Integration with latest Docker CE version(17.09)
GE-6852 interactive Docker jobs take very long to finish if the qrsh client is killed and an epilog is configured
GE-6853 Sometimes not able to delete pending job if docker device-mapping is used
GE-6862 Release notes contain 'current version' also for features that were introduced with FCS
GE-6869 arch script does not support Mac OS X High Sierra
GE-6874 normal docker job required no_root_quash if execd spooling directory shared via nfs
GE-6884 qconf help menu lists ss switch twice
GE-6901 "messages" file can have the name ".<digit>" instead of "messages" in case of file creation issues
GE-6907 Allow to specify allocation rule (-par switch of qsub) via JC's
GE-6913 expose scheduler thread ID in the qping output
GE-6916 sge_ca script does not show output for skip install question if CA
directories are already existing
GE-6924 PE jobs that request a combination of consumable HOST/JOB and YES
RSMAP complexes might not get the correct amount of ids assigned
GE-6931 the main shepherd of an interactive Docker job uses 100% CPU and
doesn't end if output file is deleted
GE-6932 child shepherd waits infinite for mutex in shepherd_trace()
GE-6934 jobs that are submitted into AR and request RSMAP ids block ids
that were not assigned
GE-6939 very long ticket calculation times for parallel array tasks
GE-6945 cannot create AR for queues with jc_list "ANY_JC"
GE-6946 Delete an array task will break the -tc <number> condition
GE-6948 parse error message from Docker response and provide it as job
error reason
GE-6959 sge_execd calls sched_setaffinity function repeatedly
GE-6972 Problem with gridengine/kernel
GE-6973 Problem with gridengine/kernel
GE-6974 Problem with gridengine/kernel
GE-6983 if job is killed by execd enforced limit the job failed state in
accounting file should reflect this
GE-6997 the native Windows (win-x86) execd should log that it cannot find
or start the qloadsensor.exe
GE-7002 test Docker integration with Docker versions up to 17.12
GE-7008 Jobs requesting RSMAP complexes that are defined without Ids are
not scheduled
GE-7009 no job dispatching with max_reservation > 0 and
consumable+PE+binding combination request
GE-7027 In AFS/KRB Grid Engine installations coshepherd/set_token_cmd is
not cleaned up
GE-7039 AR slot oversubscription
GE-7042 Resource reservation due to start time prevents dispatching the
next job of the same category
GE-7056 Failed/deleted docker interactive job’s shepherd remains and
increase trace file forever
GE-7060 Possible crash of qmaster clients
GE-7062 reservation for PE jobs with RSMAP requests is too far in the future
GE-7063 submitting an AR to a PE can violate the PE’s fixed allocation rule
GE-7067 jobs with reservation and RSMAP complex requests cause error
logging in qmaster messages file
GE-7068 PE jobs that request binding and RSMAP complexes defined as
consumable HOST might not be scheduled
GE-7069 running jobs requesting host level consumable prevent ARs to
be scheduled
GE-7078 ARs not backfilled with backfilling=h_rt
GE-7085 Logging of PDC create execd message owned by root user
GE-7108 CRITICAL "error: lGetElemStrFirst(ST_name): run time type error"
when submitting advance reservation with project
GE-7111 CSP effective secret key algorithm is not used correctly
GE-7112 qmaster error logging "rc_add_job_utilization 0 slot amount
(job <job_id> obj global type RUNNING)"
GE-7144  UGE 8.5.5 grid master continuously segfaulting
GE-7184  provide host level configurable job umask setting
GE-7187  epilog scripts are not always executed when running docker job
GE-7263  unset closed file pointers after call to sge_peclose()
UWSA-194 JobClass name of submitted job is missing
UWSA-195 fix the OpenSSL security warning

4.3.4  8.6.0

GE-168  enhance qacct [-A account_string] to qacct [-A [account_string]]
GE-639  job umasks should be configurable per job
GE-1969 qconf -me reports success on incorrect file
GE-2186  sched job info shows old and outdated message although
job is meanwhile running.
GE-2242  Inadequate job_info messages for resource quotas with
parallel jobs
GE-2464  sched_job_info can cause immense memory consumption
GE-2739  No way to remove a -notify flag, once it was set
GE-2748  Jobs with no suitable queues at all should be easier detect
GE-3279  Description of 'job_is_first_task' in 'man sge_pe' should be rephrased
GE-3614  add a way to distribute a parallel job on different
sockets without knowing the exact architecture on
submission time
GE-3621  add a placeholder for the -binding parameter which
aligns the amount of slots with the amount of requested cores
GE-3754  write documentation for sge_share_mon
GE-3787  string load values are reported only up to 1024 characters
GE-3803  qacct failes with accounting files bigger than 4GB on
native Windows (win-x86)
GE-3864  qsub -w e/-w v do not consider attributes which are load
values only
GE-3965  qalter -w p of a job in user hold prints 'verification:
job is already running'
GE-4028  DOS CR-LF in submitted scripts causes shell search errors
GE-4059  qalter -clearp silently exits
GE-4275  qalter -w e|w|v|p <job_id> identifies jobs as running when
they are not eligible for scheduling
GE-4358  misleading diagnose message for qalter -w p that refers to
queue 'global' instead of host
GE-4429  Improve qalter -w option documentation
GE-4430  Qalter -w p on a held job says 'job is already running'
GE-4433  qalter returns nothing when run without a job
GE-4498  man lacks detailed information about the new pss, rss,
smem, vmem values
GE-4678  job class parameter V does not work like descibed in
sge_job_class man page
GE-4733  finding rqs excluded queues is printing a useless
error message
GE-4734 reduce impact of qalter -w p on sge_qmaster operation
GE-4785 qmod -rq can trigger a qmaster crash when the queue is in o state
GE-4793 qalter -tc messages are not displayed, each time of execution
GE-4794 Information about the setting of tc switch can not be retrieved
GE-4817 Add automatic corebinding magnitude selection when used with PE's
GE-4820 Document the '-w' Options to qsub as a way to validate job submissions
GE-4825 provide Python binding for JSVs
GE-4839 rescheduling might produce unexpected error messages
GE-5069 new binding strategy "linear_socket" instead of requesting "linear" plus -l sched_binding_per_socket=1
GE-5084 qmaster_params "OLD_RESCHEDULE_BEHAVIOR" should not be declared deprecated
GE-5100 forwarding of job error messages from native Windows (win-x86) exec hosts to qmaster
GE-5248 remove deprecated gdi_multi_read_req setting
GE-5281 Add support for devices subsystem in cgroup integration
GE-5304 bug in mirror interface causes segfault in drmaa2 application
GE-5395 adding a way for listing the state of all global resources
GE-5403 Job should run on free core first
GE-5471 Need tool to provide overview of used and requested cluster resources
GE-5535 enable schedd_job_info for specific jobs only
GE-5537 standing reservation, which enhances the AR
GE-5542 request for a rerun limit for jobs
GE-5571 qalter -w e|v|p does not take RQS limit rules into account that contain job class filters
GE-5573 scheduler triggers job delivery before complete execd cleanup of rescheduled job
GE-5575 pe_hostfile documentation is not exact enough
GE-5686 prolog/epilog race conditions when jobs are rescheduled
GE-5691 qrsh -cwd is broken
GE-5706 gdi_request_limits parameter is missing in global config
GE-5725 upgrade procedure fails if LO_ROOT is set
GE-5733 job in hold state gets tickets and is reported as running by "qalter -w p"
GE-5773 Ability to change the cgroup name 'UGE' to another name
GE-5799 Deliver Univa Grid Engine software as RPMs
GE-5857 handle delete requests of event clients preferred
GE-5872 gdi_request_limits should support to limit event client registrations
GE-5888 resources defined in the global host are not available for -masterl requests
GE-5910 make DRMAA2 compatible with the AR object changes
GE-5911 make WebService API compatible with the AR object changes
GE-5919 correct sge_diagnostics man page header and footer
GE-5924 Standing Reservation XML output
GE-5926 Standing Reservation must go in Error state when no more allocations are possible
GE-5927 support to upgrade standing reservations
GE-5928 support Python configuration API for standing reservations
GE-5939 reimplement manual rescheduling done by qmod -r
GE-5947 drmaa2 complete reservation session features
GE-5956 re-connect request for an event client even on qmaster shutdown
GE-5963 qalter -w p doesn't report correctly when -masterl switch is used
GE-5987 examine fopen() system call problem
GE-6014 Enhance sge_diagnostics man page with info about scheduler profiling
GE-6038 schedd_job_info true/false is not case insensitive
GE-6071 Support for Affinity/Anti-Affinity in UGE
GE-6086 Fix Java code (JGDI, UGEREST etc.) to support standing reservations
GE-6125 for demo binaries the version string printed by -help command line option and in messages files should contain "demo"
GE-6129 update berkeleydb to version 6.2
GE-6141 DRAMA2 still does not use GDI sessions which will have an impact on performance
GE-6143 qalter -w p requests are executed by worker threads
GE-6147 category string should be created and normalized in qmaster thread when jobs are added or modified
GE-6159 add qralter to modify end time of Advance Reservations
GE-6249 remove qtcsh from distribution
GE-6255 execd job reports get lost due to a race condition in sge_qmaster
GE-6256 Parallelize sorting in CULL to improve scheduler performance
GE-6260 Compression/uncompression of data passed/received from commlib layer
GE-6261 event_master thread performance might be improved by processing events for different event clients in parallel
GE-6275 Determine number of unused cores (threads) in qmaster and execd
GE-6277 Improve performance of basic CULL functionality
GE-6287 Move scheduler category to master
GE-6328 out-of-the-box functionality to show utilization of global variables
GE-6334 hostgroup changes via -*attr do not update queue instance settings
GE-6339 QERROR message should include task ID for array jobs
GE-6364 check if username needs to be part of the scheduler category string
GE-6412 deny job submission with empty non boolean request
GE-6415 core binding jobs fail to be scheduled on free cores.
GE-6418 Implement host_sort_formula in schedd configuration object
GE-6424 Adapt config API to reflect object changes
GE-6439 possible performance degradation in scheduler when debiting dispatched jobs
GE-6440 create and maintain affinity cache in worker / scheduler thread
GE-6441 output affinity information with qstat -F and qhost -F
Contents

GE-6450 change naming scheme of fallback messages files
GE-6455 Look at all open, fdopen, close, etc. calls and map them to a
global function
GE-6458 qconf -srqsl and other "show list" options report an error if
no data is configured
GE-6468 Introduce and document new parameter that influences host
resolving timeout handling
GE-6476 Enhancing sge-diagnostic man page
GE-6482 scheduler profiling does not cover full scheduler code
GE-6484 Wrong/missing error-logging for loadsensors
GE-6491 affinity shall only be reduced for preempted jobs for complex
variables which are preemptable
GE-6492 change host and queue sorting in scheduler thread to reflect
affinity
GE-6494 Wrong binding-parameters are logged in qmaster-messages file
instead of sent to client
GE-6499 Add entry for "used_slots" to sge_pe man page & admin guide
GE-6500 fix compensation_factor description in sched_conf(5)
GE-6526 Excessive Memory Usage with large RSMAP Ranges
GE-6530 Core binding: Binding request for PE-jobs should be "per task"
instead of "per host"
GE-6531 Core binding: execd applies granted core-binding for PE-jobs
in a wrongish way
GE-6532 Core binding: Create new binding strategies with better names
GE-6533 Core binding: Number of granted cores should be part of accounting
GE-6538 Add RSMAP-ranges as <name>:<amount> to a host does not work
GE-6540 remove queue_sort_method from scheduler config (replaced
by weight_queue_seqno)
GE-6544 Core-binding: qsub checking of binding-strategy is not strict
enough (linearasidf:4 is accepted)
GE-6545 adapt Docker version check functions to new Docker version scheme
GE-6561 Core binding: Remove scheduler parameter "sched_binding_per_socket"
GE-6567 ranges in combination with RSMAP id requests do not work
GE-6578 Duplicate calendar entries associated with host_aliases
GE-6600 Core-binding: cgroups with PE-jobs not correct
GE-6611 build unsupported platforms with classic spooling support only
GE-6622 Order of fields in the accounting man page does not match
qacct output
GE-6638 accounting(5) man page field description is broken for "ioops"
GE-6642 operators cannot delete a (standing) reservation
GE-6646 Add support for NVIDIA DataCenter GPU Manager
GE-6647 Automatically use affinity for GPU and CPU where possible
GE-6649 Use environment variable CUDA_VISIBLE_DEVICES to hide disabled GPUs
GE-6655 Document how to tag CPU cores as 'already in use' so that they
will not be considered for core and memory binding
GE-6666 qhost man page is lacking description for resources that are
free-but-still-bound
GE-6677 DRMAA jobs submit always with -w e which has a negative performance
impact on qmaster
GE-6691 support halftime -1 setting in scheduler config to disable past usage for sharetree
GE-6692 cleanup of CULL for halftime -1 that requires minor release change
GE-6693 allow to set halftime to -1 in qmon/config API and UGE Rest
GE-6707 qalter -w e l w does not return a validation message.
GE-6708 qping and other uge binaries dump core if local hostname is not resolveable
GE-6720 calender off times not considered by resource reservation
GE-6736 Remove Solaris dtrace support and related things from UGE and LO
GE-6738 drmaa2_open_jsession does not recreate htable of job, jtemplate and jinfo
GE-6752 update jemalloc to current version (5.1.0)
GE-6753 update hwloc lib to current version (1.11.7)
GE-6758 switch to a newer OS (e.g. CentOS 6) for the lx-* build
GE-6761 qrstat -ar <ar_id> prints the PE request as granted_parallel_environment
GE-6766 xml schemas need to be verified, updated and made UGE version specific
GE-6772 update man page for queue initial_state to include qmaster migration scenario
GE-6781 qrstat -j reports job as running but shows also schedd job information why the job cannot be started
GE-6783 shepherd trace file grows too large for interactive Docker jobs
GE-6831 Alternative solution to limit the number of event clients in use by abusive users
GE-6834 unexpected logging on rescheduling of tight pe jobs
GE-6837 config-api test in master branch fails due to missing field per_pe_task_prolog
GE-6839 sge_execd is not properly handling short outages of sge_qmaster resulting in delayed reporting of load values
GE-6840 Generate all nroff man pages from markdown source files
GE-6854 pe_n, pe_min_X, pe_max_X missing in jsv(1) man page
GE-6857 requesting RSMAP complexes as soft request with -soft and/or l_soft does not work
GE-6859 qacct -A does not show column name with account string
GE-6860 Implement a way to disable the chaining of RSMAP ids
GE-6887 reduce the number of copy operations done in event master thread when distributing events to clients
GE-6891 qalter -clearp does not provide a proper response message
GE-6902 enable immediate load report sending
GE-6903 improve communication connection shutdown and gdi timeout handling
GE-6917 chaining of multiple RSMAP complexes with XOR parameter does not work
GE-6918 it is possible to specify invalid PE allocation rules
GE-6919 config-api missing field
GE-6920 update openssl library to most current version
GE-6923 Add job category field to 'qstat -j' output
GE-6947 show user name trying exceed max_u_jobs on qmaster message log
GE-6955 cgroups definition of subdir_name only working if beginning with "/" on some docker versions
GE-6982 accounting should contain the pe task id for tasks of tightly
integrated pe jobs
GE-6991 extend monitoring to track time waiting in mutex_lock
GE-6993 environment of Docker pe task shows wrong RSMAP ids if complex
is consumable YES
GE-7001 wrong behaviour if RSMAP topology mask does not fit on actual
host topology
GE-7017 support XOR RSMAP request per PE task only
GE-7020 Enhance current commlib message protocol to support compression
GE-7026 build process for markdown man pages is broken
GE-7031 display requested RSMAP selection in qstat -j/qstat -r
GE-7038 enhance qping to show compressed data size if applicable
GE-7041 qalter might cause error message and reject the modification
if combined with -w
GE-7045 backup might fail with "mv: cannot move ... File exists"
GE-7054 Integrate zlib into build process
GE-7057 Rescheduling due to preemption does not respect force-switch
GE-7058 make compression configurable
GE-7064 SGE_HGR_ variables are set to wrong values when consumables
are requested via -master1 and -l
GE-7065 debiting of PE jobs is wrong if they contain a "-masterl" request
GE-7099 correct the message displayed during upgrade procedure
GE-7122 write -xd parameters into job config for non docker job
GE-7134 update berkeley db libs to current version (6.2.x)
GE-7135 update postgresql libs to current version 10.x
GE-7146 mbind nlocal not functioning properly (even in 8.6.0)
GE-7159 Increase performance of worker threads for job delivery
GE-7177 integrate performance tools support for UGE/L0 build process and TS
GE-7179 optimize RQS code that consumes much cpu time in parsing for lWhere
GE-7180 optimize scheduler runtime by optimizing CULL functions and
other low level UGE functions
GE-7185 AR verification of jclass parameter is either missing or does
not handle special keywords
GE-7190 update nvidia cudatoolkit used for cuda_load_sensor to current version
GE-7197 interactive Docker jobs do not end if the docker container is
killed and no input is provided to the qrsh client
GE-7202 category string in accounting file is wrong for RSMAP requests
GE-7205 greedy RR: job is backfilled although it should not
GE-7206 greedy RR: qstat -rr shows negative job ID
GE-7225 remove GUI installer from distribution packages
GE-7234 cgroups support does not work with Docker 1.13.1-63 provided by CentOS
GE-7245 Can't update from 8.4.5 to 8.5.5 with CSP configured
GE-7249 The qacct -A switch is showing wrong values
GE-7250 restarting sge_qmaster with bdb spooling invalidates sharetree node
ids in qconf -sstree output
GE-7252 No config-update for execution daemons if reader threads are disabled
GE-7275 wrong handling of thread local storage can trigger crash on
qmaster shutdown
GE-7281 qalter -V exit code is incorrect when environment variables
like LD_LIBRARY_PATH are set
GE-7282  performance degradation in scheduling of parallel jobs
UWSA-86  authentication issues with RESTapi
UWSA-149 get jobs call hangs for ugerest
UWSA-192 REST should handle multiple simultaneous requests of addJob
UWSA-201 Resource usage not reported for pe jobs via ugeREST
5 Upgrade Notes

5.1 Upgrade Requirements

This is a summary of the Upgrade Matrix that describes how you can carry out the transition from Sun or Oracle Grid Engine 6.2uX, Univa Grid Engine 8.X.X to Univa Grid Engine 8.6 when you are currently using classic, BDB local spooling or PostgreSQL spooling.

<table>
<thead>
<tr>
<th>Version</th>
<th>Upgrade Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Univa Grid Engine 8.X.X</td>
<td>Backup/Restore</td>
</tr>
<tr>
<td>Oracle Grid Engine 6.2u6-6.2u8</td>
<td>Backup/Restore</td>
</tr>
<tr>
<td>Sun Grid Engine 6.2u5</td>
<td>Backup/Restore</td>
</tr>
<tr>
<td>Sun Grid Engine 6.2u1-6.2u4</td>
<td>Upgrade to SGE 6.2u5 and then Backup/Restore</td>
</tr>
<tr>
<td>Sun Grid Engine 6.2 FCS</td>
<td>Upgrade to SGE 6.2u5 and then Backup/Restore</td>
</tr>
</tbody>
</table>

Table 12: Upgrading from SGE, OGE, UGE 8.X.X to Univa Grid Engine 8.6.X
6 Compatibility Notes

6.1 Changes in Windows Execution Host sgepasswd File

The encryption algorithm for the “$SGE_ROOT/$SGE_CELL/common/sgepasswd” file passwords has been changed from RC4 to AES-256-CBC:

If you upgrade to the current version of Univa Grid Engine you need to convert your existing sgepasswd file during the upgrade procedure.

Become root and execute the following command on the master machine:

```
# sgepasswd -c
```

This will create a backup of your original ‘sgepasswd’ file as ‘sgepasswd.old_algorithm_backup’ and create the new compatible sgepasswd file. Otherwise encryption related error messages may show up.

If you create a new sgepasswd file from scratch no additional steps compared to previous versions are required.

6.2 Scheduler Log File

In previous versions of Univa Grid Engine if both the master and some slave tasks of a parallel job were scheduled to the same queue instance, one line was written to the scheduler log file containing the number of slots that was granted to this job on the queue instance of the master task. Such a line looked like this:

```
249:1:STARTING:1522328357:2772638938:Q:B@host1:slots:3.00000
```

From Univa Grid Engine 8.6.0 on, two lines are written, one for the master task, one for the slave tasks:

```
249:1:STARTING:1522328357:2772638938:Q:B@host1:slots:1.00000
249:1:STARTING:1522328357:2772638938:Q:B@host1:slots:2.00000
```

This is because of changed handling of the master task which was necessary to fix GE-5888, see the “Fixes and Enhancements” section for details.

6.3 Removed Scheduler-Parameter queue_sort_method

With the newly implemented affinity-feature, the scheduler-parameter queue_sort_method has been replaced with the new parameters weight_host_sort, weight_queue_host_sort, weight_queue_seqno. With these it is not only possible to configure a queue-sorting solely based on seq_no, or on the host_load_formula, as it was with queue_sort_method. But it is now also possible to configure a mixed sorting strategy, where the amount of the weight-parameters decide the ratio with which they influence the sorting.

For configuring what used to be queue_sort_method = load, one now has to set weight_queue_seqno to a significantly smaller number than weight_queue_host_sort. For queue_sort_method = seq_no, simply set weight_queue_seqno a significantly higher value than weight_queue_host_sort.
6.4 Changes for qconf Exit States

The qconf will no longer report an error when showing the content of empty configuration objects. If an configuration object contains no elements the exit status of qconf will be 0 and there will be no error message. The change affects following qconf command line options:

- `qconf -scall` show a list of all calendar names
- `qconf -sckptl` show all ckpt interface definitions
- `qconf -sconfi` show a list of all local configurations
- `qconf -sel` show a list of all exec servers
- `qconf -sh` show a list of all administrative hosts
- `qconf -shgrpl` show host group list
- `qconf -sjcl` show job class list
- `qconf -sm` show a list of all managers
- `qconf -so` show a list of all operators
- `qconf -spl` show all parallel environments
- `qconf -sprjl` show a list of all projects
- `qconf -sql` show a list of all queues
- `qconf -srqsl` show resource quota set list
- `qconf -ss` show a list of all submit hosts
- `qconf -sul` show a list of all userset lists
- `qconf -suserl` show a list of all users

6.5 Changes for Scheduler Profiling

The scheduler profiling option is revised and cleaned up. The most important changes are:

- Enhanced sge_diagnostics man page with info about scheduler profiling
- The scheduler profiling table shows new profiling levels: “ticket calculation”, “scheduler thread”, “ssos init”, “config update”, “wait for order completion”, “mirror events” and “set event client params”.
- The scheduler profiling level “pending ticket calculation” was renamed to “priority calculation”.
- The scheduler profiling level “copy lists” was renamed to “data preparation”.
- The scheduler profiling level “scheduler” was renamed to “scheduling”.
- The scheduler profiling level “wait” was renamed to “waiting for events”.

6.6 Changed Limit Calculations

The resulting limit calculation is revised and fixed. The most important changes are:

- The configured consumable type (NO, YES, JOB, HOST) will have no influence on any resulting limit for tight integrated parallel jobs.
- Previous version calculated the resulting limits by far to high (depending on pe and consumable settings)
- This also affects the cgroups h_vmem observation
Univa Grid Engine versions prior to 8.5.1 showed, by far, a too high limit value setting for limits like e.g. `h_vmem`. The limit adjustments are now adapted. A detailed overview of the limit observation and how it works is described in the `sge_diagnostics(1)` man page (JOB LIMITS).

ATTENTION: If you’re updating to 8.6.0 from a previous version it is recommended to verify the used limit requests of your jobs. It might be necessary to change the request value. If the limit is set too low or adjusted to fit the old limit adjustment jobs which were running fine might fail after installing this version.

### 6.7 New Default for Job Verification of DRMAA Submitted Jobs

In the past a job verification (`-w e`) was triggered for all jobs submitted via DRMAA. This had a negative performance impact on `qmaster` for all DRMAA submitted jobs which might have had also a negative impact on the cluster throughput. The verification default for DRMAA jobs has now been disabled (`-w n`) for Univa Grid Engine 8.6.0. Users that nevertheless want to get jobs verified can still enable this through the DRMAA native specification.

### 6.8 Default for Integer Complexes

Older Univa Grid Engine installations allowed to request an integer complex resource during job submission without the need to specify the amount. Implicitly 1 was used when the amount of requested resources was omitted. This is not possible anymore beginning with 8.6.0.
6.9 Deprecated Functionality

`qalter -w p` is deprecated. Instead scheduler job information has to be used.

6.10 Removed Functionality

Following components/features were removed with version 8.6.0 of Univa Grid Engine:

- Graphical Installer
- qtesh
7 Known Issues and Limitations