Host configuration

Enterprise applications

NetApp
April 25, 2024

This PDF was generated from https://docs.netapp.com/us-en/ontap-apps-dbs/mysql/mysql-containers.html on April 25, 2024. Always check docs.netapp.com for the latest.

Table of Contents

Host configuration 1
MySQL containerzation 1
MySQL and NFSv3 slot tables 1
I/O schedulers and MySQL 1
MySQL file descriptors 2

Host configuration

MySQL containerzation

Containerization of MySQL databases is becoming more prevalent.

Low-level container management is almost always performed through Docker. Container management platforms such as OpenShift and Kubernetes make management of large container environments even simpler. The benefits of containerization include lower costs, because there is no need to license a hypervisor. Also, containers allow multiple databases to run isolated from one another while sharing the same underlying kernel and operating system. Containers can be provisioned in microseconds.

NetApp offers Astra Trident to provide advanced management capabilities of storage. For example, Astra Trident allows a container created in Kubernetes to automatically provision its storage on the appropriate tier, apply export policies, set snapshot policies, and even clone one container to another. For additional information, see the Astra Trident documentation.

MySQL and NFSv3 slot tables

NFSv3 performance on Linux depends on a parameter called `tcp_max_slot_table_entries`.

TCP slot tables are the NFSv3 equivalent of host bus adapter (HBA) queue depth. These tables control the number of NFS operations that can be outstanding at any one time. The default value is usually 16, which is far too low for optimum performance. The opposite problem occurs on newer Linux kernels, which can automatically increase the TCP slot table limit to a level that saturates the NFS server with requests.

For optimum performance and to prevent performance problems, adjust the kernel parameters that control the TCP slot tables.

Run the `sysctl -a | grep tcp.*.slot_table` command, and observe the following parameters:

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

All Linux systems should include `sunrpc.tcp_slot_table_entries`, but only some include `sunrpc.tcp_max_slot_table_entries`. They should both be set to 128.

Caution
Failure to set these parameters may have significant effects on performance. In some cases, performance is limited because the linux OS is not issuing sufficient I/O. In other cases, I/O latencies increases as the linux OS attempts to issue more I/O than can be serviced.

I/O schedulers and MySQL

The Linux kernel allows low-level control over the way that I/O to block devices is

scheduled.

The defaults on various distributions of Linux vary considerably. MySQL recommends that you use `NOOP` or a `deadline` I/O scheduler with native asynchronous I/O (AIO) on Linux. In general, NetApp customers and internal testing show better results with NoOps.

MySQL's InnoDB storage engine uses the asynchronous I/O subsystem (native AIO) on Linux to perform read-ahead and write requests for data file pages. This behavior is controlled by the `innodb_use_native_aio` configuration option, which is enabled by default. With native AIO, the type of I/O scheduler has greater influence on I/O performance. Conduct benchmarks to determine which I/O scheduler provides the best results for your workload and environment.

See the relevant Linux and MySQL documentation for instructions on configuring the I/O scheduler.

MySQL file descriptors

To run, the MySQL server needs file descriptors, and the default values are not sufficient.

It uses them to open new connections, store tables in the cache, create temporary tables to resolve complicated queries, and access persistent ones. If mysqld is not able to open new files when needed, it can stop functioning correctly. A common symptom of this issue is error 24, "Too many open files." The number of file descriptors mysqld can open simultaneously is defined by the `open_files_limit` option set in the configuration file (`/etc/my.cnf`). But `open_files_limit` also depends on the limits of the operating system. This dependency makes setting the variable more complicated.

MySQL cannot set its `open_files_limit` option higher than what is specified under `ulimit 'open files'`. Therefore, you need to explicitly set these limits at the operating system level to allow MySQL to open files as needed. There are two ways to check the file limit in Linux:

- The `ulimit` command quickly gives you a detailed description of the parameters being allowed or locked. The changes made by running this command are not permanent and will erase after a system reboot.
- Changes to the `/etc/security/limit.conf` file are permanent and are not affected by a system reboot.

Make sure to change both the hard and soft limits for user mysql. The following excerpts are from the configuration:

```
mysql hard nofile 65535
mysql soft nofile 65353
```

In parallel, update the same configuration in `my.cnf` to fully use the open file limits.

Copyright information

Copyright © 2024 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.