



## **Avere OS 5.3.4.2 Release Notes**

2020-03-05

## Table of Contents

<b><i>New in Avere OS 5.3.4.2</i></b> .....	<b>1</b>
<b>Resolved issues</b> .....	<b>1</b>
Filesystem .....	1
<b><i>New in Avere OS 5.3.4.1</i></b> .....	<b>1</b>
<b><i>New in Avere OS 5.3.3.1</i></b> .....	<b>4</b>
<b><i>New in Avere OS 5.3.2.3</i></b> .....	<b>6</b>
<b><i>New in Avere OS 5.3.2.2</i></b> .....	<b>6</b>
<b><i>New in Avere OS 5.3.2.1</i></b> .....	<b>6</b>
<b><i>New in Avere OS 5.3.1.2</i></b> .....	<b>7</b>
<b><i>Contact Microsoft Customer Service and Support</i></b> .....	<b>11</b>

## Copyright Information

Copyright © 2020 Microsoft Corporation. All rights reserved.  
Specifications are subject to change without notice.

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.

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

# New in Avere OS 5.3.4.2

## Resolved issues

### Filesystem

- |         |   |
|---------|---|
| 6278225 | Fixed a bug so that the system correctly releases resources when accessing large files. Before this change, the system could consume excessive amounts of memory when handling large files. |
| 6290767 | Removed an incorrect assert statement that could cause a filesystem restart.  |

# New in Avere OS 5.3.4.1

This software release includes required updates for products offered from the Azure Portal, including Microsoft [Azure HPC Cache](#) and [Avere vFXT for Azure](#). It also includes fixes that affect all cache products, including [Azure FXT Edge Filer](#) and legacy FXT products.

## Resolved issues

### Cloud object store

- |         |  |
|---------|--|
| 5789368 | Fixed an issue that could cause process restarts when computing the space in a cloud storage container. The problem was caused by Azure request throttling due to multiple list calls. |
|---------|--|

### Filesystem

- |         |  |
|---------|--|
| 5094453 | Added code to save metadata content as a precaution when metadata failures are detected.                                     |
| 5186590 | Improved cache population speed by fixing a slowdown in the cache filesystem consistency check during read-ahead.            |
| 5339078 | Improved efficiency while establishing SSL connections.  |
| 5407127 | Improved metafile performance in the token manager during failover.  |
| 5447895 | Removed some locking code that caused contention that slowed certain front-end operations.                                   |
| 5488596 | Fixed a defect that could cause bucket keys with hash collisions to be skipped when populating the key-attribute hash table. |
| 5519684 | Created a background process to find and free stale file locks that could block access to the file.                          |
| 5567025 | Eliminated a race condition in code for checking a cloud filesystem.   |

- 5627850 Fixed a bug that could cause a filesystem restart if an HTTPS connection setup took longer than expected.
- 5636721 Eliminated misleading snapshot alerts that could be shown when flushing the cache in Azure HPC Cache.
- 5645667 Fixed a space management problem that could cause trouble during failover events.
- 5651782 Corrected a space management issue associated with rare failures.
- 5592404 Updated locking code to prevent a process restart while invalidating a core filer.
- 5707278 Fixed a lock order problem that could occur when invalidating a file handle.
- 5842386 Corrected code to prevent a loop while booting.
- 5866048 Reduced the amount of memory allocated to cloud locks to avoid a possible out-of-memory error.
- 5890695 Fixed a bug that caused a filesystem restart while securely connecting to a cloud backend.
- 5911314 Corrected a race condition that affected cloud connection management.
- 5912761 Added instrumentation for tracking task execution.
- 5932889 Corrected a race condition in the system initialization sequence.
- 5955269 Fixed a problem that could cause operations to fail to complete when a space limit was reached.
- 5961765 Changed logging to avoid multiple log entries for a slow-running task.
- 5986301 Corrected a flaw in state check and internal accounting for metafile operations.
- 5986543 Fixed traces to eliminate a possible null point exception.
- 5990301 Resolved a race condition that was caused by reinitialization of OpenSSL algorithms and allocation functions when using proxy configurations.
- 5997314 Corrected a possible race condition error related to a metadata flush operation.
- 6041890 Fixed a locking hierarchy violation that could result in process restart because of an uncommon race condition.
- 6054860 Corrected logic in the replay path for metadata operation.

## General

- 5533584 Cluster-generated certificates now include extended key usage for TLS web server authentication. This change makes the certificates compliant with the browser TLS extension.
- 5679687 Fixed an issue where clusters upgraded to v5.3 would no longer allow RemoteCommand service.
- 5695188 Fixed an XML-RPC bug that prevented users from creating a read-only Cache Policy.

- 5771119 Fixed a bug in the Azure FXT Edge Filer's NIC driver that caused LACP link aggregate interfaces to fail when cluster services were started.
- 5820727 Disabled pre-production debugging code that caused excessive log entries.
- 5926431 Fixed a general XML-RPC bug that caused all integers to be encoded as long integers instead of <int>. In XML-RPC this encoding applies to signed 32-bit integers, which are the vast majority of uses in Avere XML-RPC.
- 5991352 Updated code to prevent a deadlock in the process responsible for refreshing cloud service credentials.
- 5998813 Fixed a bug that caused GCE vFXT nodes to fail to connect to GCP API endpoints through an HTTP proxy.
- 6037362 The default XML-RPC encoding for 32-bit signed integers has been changed back to <int>. Larger integers are encoded as <double>. This change restores historical behavior and removes a change in Avere OS 5.2 that encoded all XML-RPC integers as <i8> by default.

The standard XML-RPC encoding for 32-bit signed integers is <int>. Larger integers must be encoded differently and use a non-standard encoding: <double> (our historical default), <i8>, or <string>.

An XMLRPC client can set its preferred encoding by calling `system.setInt64Representation` or by setting one of these HTTP cookie values:

- 'int64Representation=i8'
- 'int64Representation=double'
- 'int64Representation=string'

## SMB/CIFS

- 5132513 Fixed a defect that sometimes caused the incorrect error status to be returned when a path involving NFS symbolic links was accessed using SMB.
- The system now correctly returns the error `NT_STATUS_ACCESS_DENIED` if an NFS symbolic link cannot be evaluated for SMB access of a file or directory on a path with the link.
- (Before this fix, the errors `NT_STATUS_OBJECT_NAME_NOT_FOUND` or `NT_STATUS_OBJECT_PATH_NOT_FOUND` were returned in some cases.)
- 5534066 Fixed a bug for the SMB/CIFS multi-component lookup case of “..” detecting a loop when one or more of the name path components is a symbolic link.

# New in Avere OS 5.3.3.1

This software release supports the general availability (GA) release for the Microsoft [Azure HPC Cache](#) service. It also includes bug fixes that affect other caching products, including Avere vFXT for Azure and Azure FXT Edge Filer.

## Cache policy change

A default setting has been changed in the built-in cache policy Full Caching. The maximum writeback delay in this policy now defaults to one hour. Previously, the maximum writeback delay was set to 10 minutes by default. The value can be customized by using the **Core Filer > Manage Cache Policies** settings page for an FXT or Avere vFTX for Azure cluster.

This change only affects clusters created with Avere OS 5.3.3.1 or later software. Clusters upgraded to 5.3.3.1 keep the original 10 minute default writeback delay.

## Resolved issues

### Filesystem

5105615	Fixed a problem that could interfere with cluster node removal. Before this change, a long-running internal operation could cause the removal to fail to complete.
5156906	Fixed a race condition in statistics-gathering code that could cause a file system process restart.
5227453	Updated code to fix a problem revoking tokens when a node is removed from a cluster.
5279880	Fixed a defect that caused a filesystem restart when removing a cluster node.
5283106	Updated hash table code to proactively flush operations when a large number of metadata operations accumulate in the hash table. Hash tables are used to maintain directory operations that will be flushed to the back-end core filer, but if a hash table chain becomes too long, it can increase lock contention.
5289152	Updated code to ensure that directory operation cache entries are maintained using least-recently-used (LRU) algorithms.
5304368	Corrected a race condition that impacted the token manager during startup.
5321702	Updated object version handling to fix a bug.
5369616	Fixed an issue related to cloud response handling.
5409769	Updated code to ensure that a memory reservation is released after data is flushed to back-end storage. In a certain situation this bug caused excessive memory use.
5410203	Fixed a logic problem that could cause issues at the end of a local directory manager rebalance operation.
5439093	Improved a diagnostic tool used to diagnose potential problems with cached directory operations.

5459515	Updated the default metadata write back time to 10 minutes.
5460843	Corrected an issue in available space detection during a data flush.
5481433	Corrected a race condition in token processing.
5534498	Fixed a race condition in name cache.
5581410	Changed the Full Caching cache policy's default writeback time (described above).
26959	Corrected handling of metadata from the back end to avoid panics under stress.
27445	Corrected a problem handling an error related to the cloud module's version database.

## NFS

5160193 This release adds a per-junction language setting. Possible values are utf-8 and undefined.

This language setting is enforced by NFS operations that create or remove directory entries (dirents).

- Operations that create new directory entries: CREATE, MKDIR, SYMLINK, MKNOD, LINK, RENAME
- Operations that remove existing directory entries: REMOVE, RMDIR, RENAME

These operations return a status of INVALID when the language setting is UTF-8 and the dirent is not valid UTF-8.

Before this change, the system behaved as if all junctions had a language setting of undefined.

Upgrading to a release containing this change does not affect existing junctions.

Additional information about this setting:

- The XMLRPC method `vserver.addJunction()` gives a default language setting of utf-8.
- The XMLRPC method `vserver.modifyJunction()` preserves the language setting unless specified.
- For cloud core filers, the best practice is to specify the UTF-8 language setting.
- For NAS core filers, the language should match the core filer language. Only set the language to undefined if the core filer's cache policy does not have Local Directories enabled.

## Platform

- 5265161 Power saving settings were updated for clusters running in Azure to give better performance.
- 5462020 Fixed a bug in a custom setting that controls LACP mode.
- 5475751 Updated drivers for FXT 6000-series hardware to correct MAC address filtering when running in LACP mode.

## XML-RPC

- 5352452 Updated the XML-RPC help for `node.allowToJoin()`.

# New in Avere OS 5.3.2.3

The 5.3.2.3 release includes improvements for the Microsoft [Azure HPC Cache](#) service, which is in public preview. No additional fixes for GA products are included in this release.

# New in Avere OS 5.3.2.2

The 5.3.2.2 release includes improvements that affect the Microsoft [Azure HPC Cache](#) service, which is in public preview. No customer-visible updates for GA products are included.

# New in Avere OS 5.3.2.1

This release includes bug fixes and updates to support the new Microsoft [Azure HPC Cache](#) service, which is in public preview.

## Resolved issues

### Filesystem

- 27448 Removed an automatic debugging data collection routine from the nightly collection list to prevent a possible system restart on nodes with very large caches.
- 26768 Fixed a missing lock to prevent a possible database corruption and restart.
- 26359 Prevent blocked operations and possible system restarts by allowing more operations to abort when the associated core filer is being removed or suspended.



## NFS

27057 Added the ability to restrict which NFS exports should be probed by the cluster. This change protects the cluster from attempting to investigate overwhelming numbers of core filer exports, and also gives administrators the ability to protect restricted exports which should not be mounted by the Avere cluster.

The new setting is controlled by the new XML-RPC methods `corefiler.getExportFilters()` and `corefiler.setExportFilters()`. Refer to their help files for details.

## General

27399 Fixed a software error that could cause the system to loop while reassigning IP addresses to cluster nodes.

27373 Updated descriptions in the Avere Control Panel to clarify that the “use 1Gb management network” setting makes those ports unavailable for client-facing data.

# New in Avere OS 5.3.1.2

This release introduces support for the new Azure FXT Edge Filer hardware platform. Read more at <https://azure.microsoft.com/services/fxt-edge-filer/>.

## Resolved issues

### Cloud object store

26155 Avere OS now prevents users from creating two core filers that point to the same cloud storage container. Each active core filer must control a different set of data. The system now checks unique IDs to detect whether the new core filer is a cloned version of an existing core filer on the cluster and gives a failure message if it is.

26267 Updated locking code for multithread-safe data structures. Before this change, it was possible that running a debugging command which accessed certain internal data structures could cause a crash because the service utility did not obtain the proper locks.

26390 Fixed a problem in cloud core filer shutdown that could cause a filesystem restart because a timer was not correctly cleaned up.

26647 Fixed a problem that could cause a core loop after a cross-directory rename. The issue was caused by a failed sanity check when the directory contents on the cloud storage did not match the expected contents.

26941 Fixed a bug so that files can flush to the core filer even in the rare situation when a file handle hash collides with the hash of a reserved file handle.

27264 Fixed an issue that could cause TCP connection requests to the cloud back end to get stuck during SSL negotiation, requiring a filesystem restart.

## Filesystem

- 23176 Eliminated a false report of a long-running operation.
- 24341 Improved core filer invalidation code to fix a race condition that could cause a panic if the cluster restarted during invalidation.
- 25315 Removed a problem with an internal data structure that could sometimes cause a file system restart when truncating a very large file.
- 25853 Enhanced data migration to use all nodes in the cluster.
- 25929 Removed a conflict between the system that decides to when to write data to the back end and the system that reserves space for future writes.
- 25930 Updated code for failover to avoid unnecessarily marking cache content as stale. This change ensures that non-idempotent file operations are executed exactly once, even when cluster nodes crash.
- 26059 Fixed code related to the transaction log to prevent a data inconsistency problem that could occur if cluster nodes are not shut down cleanly. This defect was found in vFXT clusters after nodes were shut down from the Azure portal instead of using the cluster control panel; it is safer to always shut down the cluster before suspending or powering off individual nodes.
- 26061 Installed proper locking to prevent future file system crashes while examining internal log entries. Diagnostic debugging tools were also added to help diagnose any future crashes in related systems.
- 26068 Fixed an internal state issue that could cause an out of memory error when flushing data and using cloud file system buffers directly.
- 26321 Corrected a resource limit configuration that could cause an unexpected filesystem service restart in a vFXT node running on a VM with more than 144G of memory.
- 26374 Corrected a code error that caused a lock to be held too long, potentially causing a deadlock and system restart.
- 26455 Fixed a rare race condition that could leave internal logs in an unexpected state after a restart.
- 26532 Changed FlashMirror™ code to ensure that all files marked out-of-sync on both the primary node and the mirror node are copied if there is a failover event.
- 26576 Corrected code that resets pointers to metadata files during failover.
- 26608 Changed a default so that the system gives alerts and disables directories in a specific invalid state instead of crashing.
- 26622 Fixed an issue in always-forward mode so that a readdirplus operation now provides all attributes and file handles.

- 26679 Fixed an issue that could cause errors when purging metadata files if filenames in the log included non-UTF-8 characters.
- 26682 Fixed a timing-related memory corruption that could be triggered when the file count was extremely high. Before this change, this problem could cause a restart and core file when using HA features.
- 26706 Fixed an issue to prevent using freed pointer values.
- 26768 Fixed a missing lock to prevent a possible database corruption and restart.
- 26786 Fixed an issue where a lock used while scanning a large data structure could cause causing which can cause other threads in armada to watchdog.
- 26796 Sped up the file scan time to improve performance during file validation after restarting the cluster.
- 26802 Fixed a file system restart that was caused by the improper state of an out-of-sync item during a migration.
- 26817 Improved token manager operations to boost per-operation performance by reducing disk writes.
- 26829 Fixed an issue that could causing a cache manager crash when reporting long running operations in log messages or to the Avere Control Panel.
- 26859 Improved buffer management controls to avoid deadlocks while flushing data to cloud storage.
- 26879 Improved flush log free space estimation to prevent errors or dropped requests when space ran out.
- 26880 Fixed an issue where an aborted rename could be logged incorrectly. Before this fix, the incorrect state could permanently block operations on some files or directories.
- 26914 Changed an algorithm in the name cache to prevent the number of allocated entries from exceeding the maximum.
- 26923 Fixed a crash related to an invalid file handle during a rename operation in a system with always-forward mode enabled.
- 26926 Fixed a problem where processing a truncation and running out of log space could cause a crash.
- 26967 Fixed an issue where improper locking could cause a restart after a migration.
- 27074 Fixed a problem that could cause the cluster to deny access to a cloud core filer when SMB access was enabled.
- 27087 Fixed an issue to prevent a crash in read-only mode when the target of an object pointer would be removed, causing a null pointer.
- 27099 Improved locking to prevent a filesystem core if identical rename operations happened simultaneously.
- 27113 Added code to automatically clear a dangling reference. In some rare circumstances the reference problem could prevent file access indefinitely.

- 27134 Fixed an out-of-memory problem that could cause the cache flush component to crash.
- 27184 Fixed an issue that interfered with running the ping command directly on cluster nodes. The ping command no longer runs within a secure sandbox.
- 27187 Fixed a race condition when reading the core filer directories to populate the cache.
- 27196 Fixed a problem that could cause a deadlock when opening metadata files.
- 27215 Updated code to ensure that inactive flush requests are cleared from the request log.
- 27234 Fixed a small leak in the code to detect long-running operations.
- 27244 Fixed a memory issue that could occur during heavy reading from a core filer.
- 27278 Fixed a problem that could cause a node to fail during migration to or from a cloud core filer.
- 27279 Added locking and proper initialization code to fix a bug that could cause a crash when deleting cache flush records.

## General

- 26056 Added validation to prevent an empty entry for the KMIP server hostname.
- 26723 Fixed a problem where bootloader messages and prompts were not being shown on the serial-over-LAN (SOL) console for some 2000-, 3000-and 4000-series FXT models.
- 26756 Increased the frequency of checks for upcoming cloud credential expirations.
- 26887 Increased a ulimit setting to prevent certain internal processes from unnecessarily restarting.
- 26952 The Data Management tab no longer appears in the control panel dashboard on Azure vFXT systems.
- 27014 Cluster proxy server connection tests now tolerate failures with short timeouts. This change allows the proxy configuration to be honored even if the proxy service is not always reliable.
- 27031 Fixed a software defect that could cause the management layer to fail and restart when the number of assigned client IP addresses in the cluster was reduced. This failure affected each cluster node in turn as management control switched to healthy nodes.  
  
If you need to reduce the number of virtual interfaces (client- or cluster-facing) before updating, contact Microsoft Customer Service and Support for advice.
- 27055 A divide-by-zero error in a network driver could cause the system to unexpectedly restart. The issue has been corrected.
- 27060 Enhanced logging to improve the supportability of token management, especially during cache invalidation.
- 27120 Improved memory usage, fixing an issue where the system requested a larger buffer for packets than required, and consumed extra memory unnecessarily.

- 27146 Fixed a problem where a web page error could prevent cluster creation.
- 27174 Updated the Core Filer Details settings page to clarify the “filer class” control and update the list of options. The control is now labeled SMB filer class, and the help text has been revised.
- 27217 Corrected a problem that caused the private preview software for the Azure FXT Edge Filer (v5.2.1.1) to return an error when trying to configure a syslog server.

## NFS

- 26285 Fixed a defect in the network lock manager so that an internal server lock recovers correctly. Before this change, lock operations could receive “lockgrace” errors after the end of the grace period.
- 26979 Addressed a defect where specific errors (like ROFS) during junction add or modify operations returned generic "system is busy" errors instead of appropriate descriptions.
- 27135 Updated code to avoid a file system service restart (with associated core file) if an NFS client attempted to access a junction that did not have an NFS export policy.

## SMB/CIFS

- 26634 Added a CIFS admin field in the Core Filer Details configuration page. If provided, this user’s credentials are supplied when reading SMB ACLs from the core filer.
- 26815 Fixed an issue where junctions in a migration from a cloud source to a NAS target with the “force default ACL” option set could cause a filesystem service restart.
- 27095 Changed code to correctly handle non-terminal errors that could cause a core file from SMB processes.

# Contact Microsoft Customer Service and Support

Microsoft Customer Service and Support can be reached by website, phone, or email.

**By web:** Use the links under Support Information on <https://www.microsoft.com/avere/contact-us>

**By phone:** 1-888-88-AVERE, Option 2 (Toll-Free)  
1-412-894-2570, Option 2

**By email:** [averesupport@microsoft.com](mailto:averesupport@microsoft.com)