

FairCom White Paper

Linux Performance and Safety Advisory

Audience

Developers

Subject

**Learn how a known Linux bug affects data integrity
and the precautions you should take**

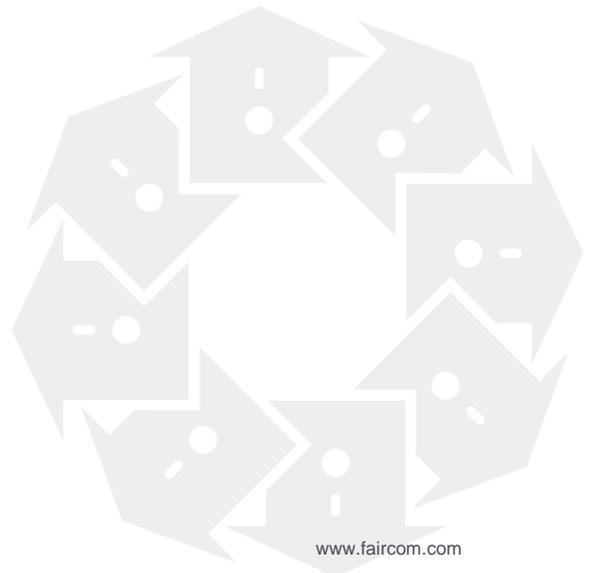
© Copyright 2021, FairCom Corporation. All rights reserved. For full information, see the FairCom Copyright Notice (page vii).



FairCom[®]

Contents

1.	Linux Implementation Affects Data Safety & Performance	1
1.1	Workaround for Synchronous Write Mode on Linux	1
2.	Caching Considerations	3
2.1	Write Barriers.....	3
2.2	Commit Delay and Disk Write Cache	4
3.	Settings for Best Performance	6
4.	Document History	6
5.	Index	9



1. Linux Implementation Affects Data Safety & Performance

Tech Bulletin: FairCom DB V10.3 - Performance Change on Linux Systems

The FairCom DB V10.3.1.21834 (Build 140307) release changed behavior from prior Linux releases:

- Releases prior to this build were vulnerable to a data integrity issue due to a known issue in the Linux kernel.
- Releases beginning with this build (and later) address this issue to provide data integrity with some loss in performance.

FairCom DB releases prior to the build listed above are vulnerable to this issue and should be updated to the latest release.

WARNING: By default, most current Linux systems might not be configured in a way that guarantees recoverability of data in the event of a loss of power to the system. This paper discusses the effects of changes to address this issue and a possible impact on performance.

This document highlights a potential transaction data loss on recovery operations for FairCom DB Servers running on Linux systems due to the implementation of the Linux file system. A workaround allows recovery in case of a power failure. The configuration keywords discussed below can have an impact on performance, but may be necessary to ensure data integrity depending on your environment.

1.1 Workaround for Synchronous Write Mode on Linux

A limitation in all implementations of the Linux file system has caused occurrences of FairCom DB errors **64**, **66**, **96**, and **571**.

In a system power-off test on a Linux system, we noticed cases in which automatic recovery failed with the errors listed above. This was because some data had not been written to disk in the transaction logs, data files, and index files.

These errors were seen when using the following FairCom DB Server configuration options:

COMPATIBILITY LOG_WRITETHRU	Affects transaction log files
COMPATIBILITY TDATA_WRITETHRU	Affects TRNLOG data files
COMPATIBILITY TINDEX_WRITETHRU	Affects TRNLOG index files



COMPATIBILITY SYNC_LOG

Affects transaction logs and TRNLOG index files (this is an old option that we don't expect to be used anymore)

On Linux systems, when these configuration options are used, an `fcntl()` call is supposed to enable the `O_SYNC` file mode. However, Linux ignores this mode, which causes disk writes to be stored in the file system cache instead of being written to disk before the write call returns. **This gives very fast performance, but it does not provide the data integrity that was intended.**

The following Linux kernel bug report confirms this limitation in the file system portion of the Linux kernel: **Linux Kernel Bug Tracker - 5994** (https://bugzilla.kernel.org/show_bug.cgi?id=5994). The report includes a proposed patch, but we found no indication that the patch has been applied to the Linux kernel for any Linux distributions as of February 2014.

Beginning with FairCom DB V10.3.1.21834 (Build 140307), a change in the FairCom DB logic resolves the problem by enabling `O_SYNC` when opening or creating a file instead of calling `fcntl()` after opening the file. This change also disables the use of `COMPATIBILITY SYNC_LOG` on Linux because that keyword would turn off the `O_SYNC` bit.

Note: Releases prior to FairCom DB V10.3.1.21834 (Build 140307) are vulnerable to this issue and should be updated to the latest FairCom DB release.

The Effects of the Change

When rebooting with a power source (e.g., an uninterruptable power supply, UPS) connected to the system with the disk write cache enabled, it was possible to recover the data without any errors. However, when rebooting by disconnecting the power source with the disk cache enabled, we saw the errors mentioned above during automatic recovery.

Based on these results, it appears safe to keep the disk write cache enabled if there is an uninterruptable power supply (UPS) connected to the system. The UPS must be properly configured to safely shut down the computer and FairCom DB Server before the UPS battery fails.

2. Caching Considerations

The use of write barriers and commit delay has an impact on data safety and performance.

Write barriers enforce the orderly flushing of caches to disk by ensuring that all requests issued at levels above the barrier must be satisfied before continuing. Write barriers affect performance because they require the write to be complete before the write call continues. Asynchronous (cached) vs. synchronous (flushed) writes are examined in the section titled Write Barriers (page 3).

Tests were conducted to determine the performance impact of commit delay and disk write cache. The results of these tests and configuration suggestions are presented in the section titled Commit Delay and Disk Write Cache (page 4).

2.1 Write Barriers

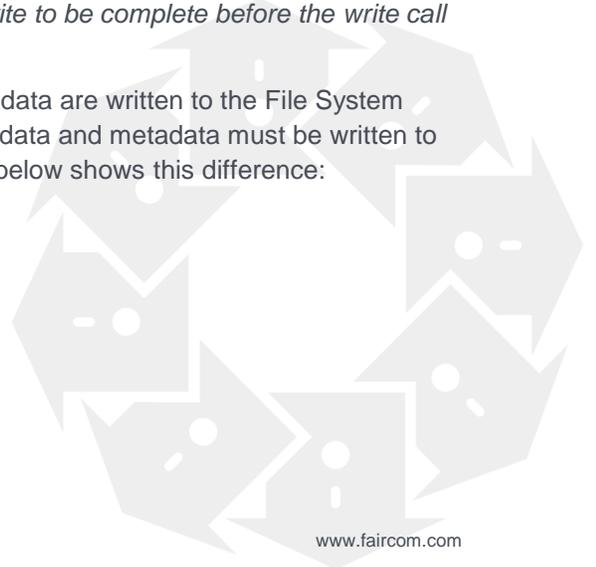
Linux file systems offer several levels of caching, including the user process, file system, transport layer, storage controller, and the hard disk drive (HDD), to name just a few. Think of these layers of caching as a stack with the application at the top and the physical media to which the data is written (the hard disk platter) at the bottom.

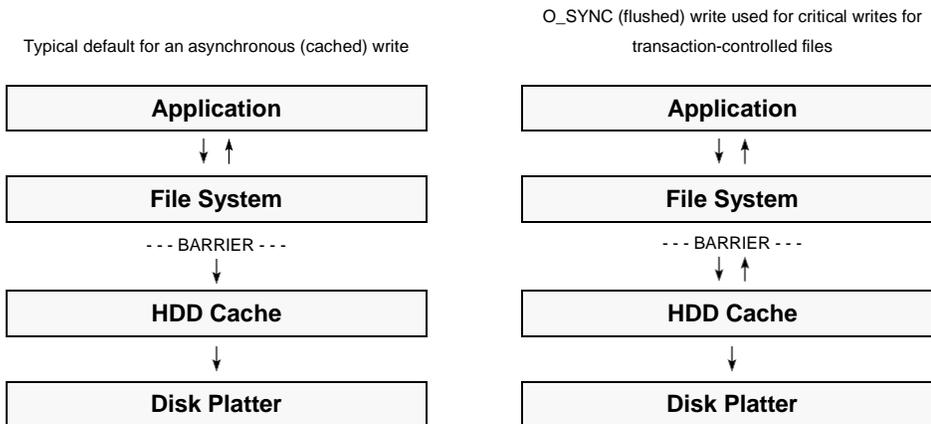
When a disk has write cache enabled and the system loses power, the contents of the disk write cache are lost. This means that file data and file system metadata that the application or the file system believed to have been written to disk might not have made it to disk. It is also possible for the data and metadata to have been written to disk in a partial or out-of-order manner, leading to inconsistencies which can cause problems for the file system.

The "write barrier" (or simply "barrier") enforces orderly flushing of these caches to disk. Essentially, the barrier ensures that all requests issued at levels above the barrier must be satisfied before continuing. Write barriers guard against data loss and inconsistency of file system metadata

The barrier affects performance because it requires the write to be complete before the write call continues.

In the default mode, the barrier ensures the data and metadata are written to the File System cache before the write call returns. In O_SYNC mode, the data and metadata must be written to the HDD cache before the write call returns. The diagram below shows this difference:





Asynchronous (cached) vs. synchronous (flushed) writes. Notice that for O_SYNC writes, the call does not return until the data has been written to the HDD cache.

The bug in the Linux kernel mentioned earlier was preventing O_SYNC mode from taking effect, which meant that data was not written to disk before the call returned. This resulted in unexpectedly high performance at the expense of data recoverability in the case of a power loss.

It should be safe to disable write barriers (using `barrier=0` in `/etc/fstab` or using the `-o nobarrier` option for `mount`) in either of the following situations:

- if disk write cache is disabled; or
- if disk write cache is enabled and the disk has a *properly configured* battery backup or uninterruptable power supply (UPS).

2.2 Commit Delay and Disk Write Cache

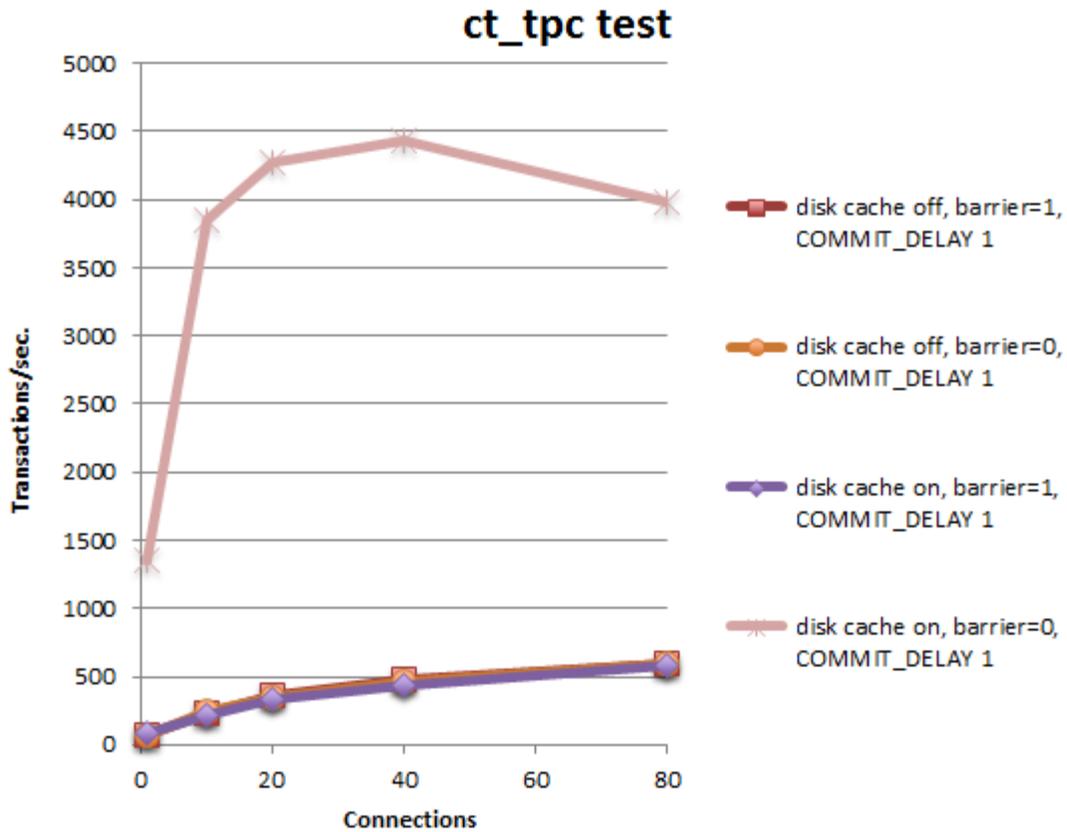
The effects of commit delay and disk write cache on FairCom DB Server performance were tested. All tests used the following options:

```
COMPATIBILITY LOG_WRITETHRU
COMPATIBILITY INDEX_WRITETHRU
```

The test results are as follows (tps = transactions per second; larger numbers are better):

ct_tpc test				With fcntl() bug fix				
ext4 file system, data=ordered				Avg. tps for 20 sec. with # clients:				
	Disk Write Cache	Barrier	COMMIT_DELAY	1	10	20	40	80
	off	1	1	76	236	358	483	600
	off	0	1	76	248	349	463	592
	on	1	1	79	220	332	434	586
	on	0*	1	1358	3848	4270	4435	3980

* Battery backup or uninterruptable power supply (UPS) required for data integrity.



The results shown above highlight the significant cases from a battery of tests run at different settings.

The following conclusions were drawn based on these tests:

- For all combinations of write cache and barrier settings, using a small commit delay rather than no commit delay improves the performance.
- When disk write cache is enabled (the default for many systems), large commit delays can slow down performance compared to no or small commit delay.

Some options, such as putting an uninterruptable power supply (UPS) on a system, improve recoverability of data without impacting performance.

If you rely on a UPS, be sure to configure it to cleanly bring the system down before the battery is exhausted! Note that the FairCom DB Server will generally come down cleanly if it receives a shutdown signal from the operating system. However, we strongly recommend testing this operation on your system!

The best performance can be achieved by shutting off the barrier, which requires a battery backup or UPS to ensure data integrity.

3. Settings for Best Performance

With the patch installed, the highest transaction rates are seen with the following configuration:

- Disk write cache enabled
- File system mounted with `barrier=0`
- `COMMIT_DELAY 1`

Best Practices

The best configuration for your system depends on many factors. There is no substitute for performing tests to determine the best settings for your environment.

- Use the FairCom DB Load Test Utility to get a general overview of performance on your system. This test program is supplied with the FairCom DB Professional Developer's Kit.
- Use your own application and sample data to experiment with different settings.

Checklist

In evaluating your configuration, consider the questions in this checklist:

1. Which file system are you using and what caching options does it provide?
2. Which layers of cache do you have on your system (file system, disk controller, etc.)?
3. Do you have an uninterruptable power supply available?
 - Have you properly configured the UPS to bring the system down cleanly before the battery is exhausted?
 - Have you tested to be sure the FairCom DB Server will be brought down in a safe manner by your UPS-invoked shutdown?
4. Should disk write cache be enabled?
5. What is the best setting for the barrier option?
6. What is the best setting for commit delay?

4. Document History

10/01/2015	Corrected the build date of the change from FairCom DB V10.3.1.20772 (Build 140216) to V10.3.1.21834 (Build 140307).
7/1/2014	Updated information on the first and second pages about affected builds and the fact that FairCom DB releases prior to this change are vulnerable to an issue with the Linux kernel.
6/20/2014	Document first published.

Last published Thursday, April 15, 2021.

Copyright Notice

Copyright © 1992-2021 FairCom USA Corporation. All rights reserved.

No part of this publication may be stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of FairCom USA Corporation. Printed in the United States of America.

Information in this document is subject to change without notice.

Trademarks

FairCom DB, FairCom EDGE, c-treeRTG, c-treeACE, c-treeAMS, c-treeEDGE, c-tree Plus, c-tree, r-tree, FairCom, and FairCom's circular disc logo are trademarks of FairCom USA, registered in the United States and other countries.

The following are third-party trademarks: Btrieve is a registered trademark of Actian Corporation. Amazon Web Services, the "Powered by AWS" logo, and AWS are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries. AMD and AMD Opteron are trademarks of Advanced Micro Devices, Inc. Macintosh, Mac, Mac OS, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. Embarcadero, the Embarcadero Technologies logos and all other Embarcadero Technologies product or service names are trademarks, service marks, and/or registered trademarks of Embarcadero Technologies, Inc. and are protected by the laws of the United States and other countries. HP and HP-UX are registered trademarks of the Hewlett-Packard Company. AIX, IBM, POWER6, POWER7, POWER8, POWER9, POWER10 and pSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. Intel, Intel Core, Itanium, Pentium and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. ACUCOBOL-GT, Micro Focus, RM/COBOL, and Visual COBOL are trademarks or registered trademarks of Micro Focus (IP) Limited or its subsidiaries in the United Kingdom, United States and other countries. Microsoft, the .NET logo, the Windows logo, Access, Excel, SQL Server, Visual Basic, Visual C++, Visual C#, Visual Studio, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Oracle and Java are registered trademarks of Oracle and/or its affiliates. QNX and Neutrino are registered trademarks of QNX Software Systems Ltd. in certain jurisdictions. CentOS, Red Hat, and the Shadow Man logo are registered trademarks of Red Hat, Inc. in the United States and other countries, used with permission. SAP® Business Objects, SAP® Crystal Reports and SAP® BusinessObjects™ Web Intelligence® as well as their respective logos are trademarks or registered trademarks of SAP. SUSE" and the SUSE logo are trademarks of SUSE LLC or its subsidiaries or affiliates. UNIX and UNIXWARE are registered trademarks of The Open Group in the United States and other countries. Linux is a trademark of Linus Torvalds in the United States, other countries, or both. Python and PyCon are trademarks or registered trademarks of the Python Software Foundation. isCOBOL and Veryant are trademarks or registered trademarks of Veryant in the United States and other countries. OpenServer is a trademark or registered trademark of Xinuos, Inc. in the U.S.A. and other countries. Unicode and the Unicode Logo are registered trademarks of Unicode, Inc. in the United States and other countries.

All other trademarks, trade names, company names, product names, and registered trademarks are the property of their respective holders.

Portions Copyright © 1991-2016 Unicode, Inc. All rights reserved.

Portions Copyright © 1998-2016 The OpenSSL Project. All rights reserved. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

Portions Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved. This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Portions © 1987-2020 Dharma Systems, Inc. All rights reserved.

This software or web site utilizes or contains material that is © 1994-2007 DUNDAS DATA VISUALIZATION, INC. and its licensors, all rights reserved.

Portions Copyright © 1995-2013 Jean-loup Gailly and Mark Adler.

Portions Copyright © 2009-2012 Eric Haszlkiewicz.

Portions Copyright © 2004, 2005 Metaparadigm Pte Ltd.

Portions Copyright © 2008-2020, Hazelcast, Inc. All Rights Reserved.

Portions Copyright © 2013, 2014 EclipseSource.

Portions Copyright © 1999-2003 The OpenLDAP Foundation.

Open Source Components

Like most software development companies, FairCom uses third-party components to provide some functionality within our technology. Often those third-party components are selected because they are a standard in the industry, they offer specific functionality that is easier to license than to develop and maintain in the long run, or they provide a proven and inexpensive solution to a particular business need. Examples of third-party software FairCom uses are the OpenSSL toolkit that provides Transport Layer Security (TLS) for secure communications and the ICU Unicode libraries to provide wide character support (think international characters and emojis).

Some of these third-party components are the subject to commercial licenses and others are subject to open source licenses. For open source solutions that we incorporate into our technology, we include the package name and associated license in a notice.txt file found in the same directory as the server.

The notice.txt file should always stay in the same directory as the server. This is particularly important in instances where your company has redistribution rights, such as an ISV who duplicates server binaries and (re)distributes those to an eventual end-user at a third-party company. Ensuring that the notice.txt file "travels with" the server binary is important to maintain third-party and FairCom license compliance.

4/15/2021

5. Index

A
affected releases 1
Asynchronous (cached) vs. synchronous
 (flushed) writes 3

B
barrier 3
build 1

C
cached writes 3
Caching Considerations 3
Checklist 6
Commit Delay and Disk Write Cache 4
COMPATIBILITY LOG_WRITETHRU 1, 4
COMPATIBILITY SYNC_LOG 1
COMPATIBILITY TDATA_WRITETHRU 1
COMPATIBILITY TINDEX_WRITETHRU 1, 4
Copyright Notice vii

D
Document History 6

E
errors 64, 66, 96, 571 1

F
FairCom DB affected releases 1
flushed write 3
flushed writes 3

L
Linux Bug 5994 1
Linux Implementation Affects Data Safety &
 Performance 1
Load Test Utility 6
LOG_WRITETHRU 1

O
O_SYNC 3

R
release 1

S
Settings for Best Performance 6
SYNC_LOG 1
synchronous (flushed) writes 3
Synchronous Write Mode 1

T
TDATA_WRITETHRU 1
TINDEX_WRITETHRU 1

W
Workaround for Synchronous Write Mode on
 Linux 1
Write Barriers 3

