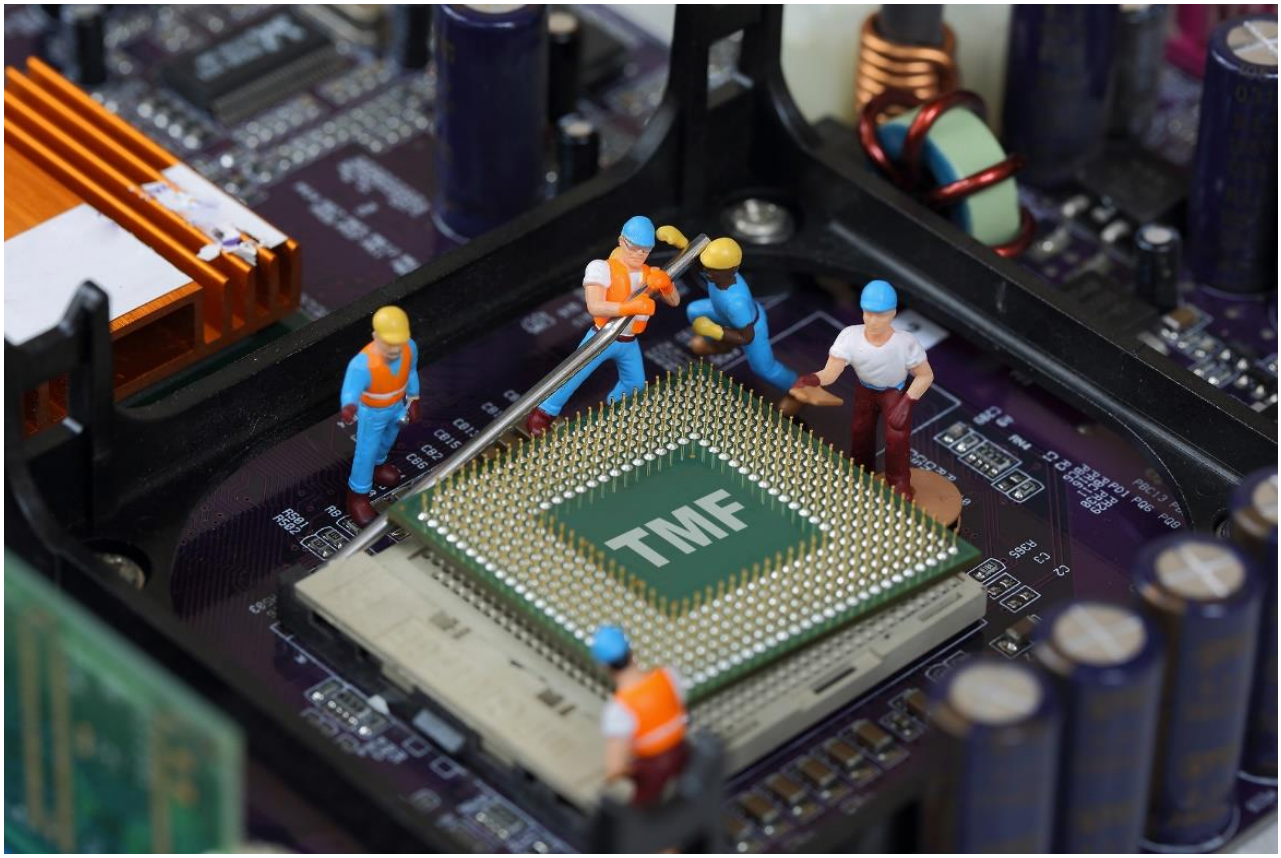
GRAVIC®
Shadowbase

Only the Truth:
Debunking TMF NonStop Data Protection Myths

A Gravic, Inc. Article



Authors' Note: *We feel it is critically important to address this topic because we continue to find that considerable misinformation still exists regarding the use of TMF and data auditing in general on HPE NonStop systems. Perceptions remain that the use of TMF is superfluous, costly, or has bad performance issues, as well as a lack of understanding or disregard for the data integrity protection benefits offered by TMF. Debunking these myths and leveraging this transformative technology is critical to empowering mission- and business-critical applications and services, and for competitively positioning NonStop applications well into the future.*

HPE NonStop Shadowbase data replication software captures change data (data created or updated by users and applications) from a transaction log, for example the TMF log on NonStop systems. As data is changed, TMF writes the changes to the transaction log, where they are read by HPE Shadowbase software for replication to a target system or application.

The Transaction Log

On an HPE NonStop system, this transaction log is maintained by the HPE NonStop Transaction Management Facility (TMF) subsystem, and is known as the TMF Audit Trail (TMF AT). Data managed in this way by TMF are known as audited files, or audited data (conversely, non-audited-files and non-audited data)[1]. As changes are made to audited data (Enscribe files, SQL/MP and SQL/MX tables), these changes are permanently made to the TMF AT when the application's transaction commits, or is rolled back (undone) as the application's transaction aborts. The changes are then read and replicated by HPE Shadowbase data replication. Therefore, for the replication engine to operate, the changes to data must be TMF audited.[2]

Myths from the Early Days

However, in the early days of Tandem Computers, this TMF auditing function was not particularly efficient, and could negatively impact application performance. These issues were quickly resolved, but the reputational damage to TMF was already done.

A myth evolved that using TMF audited files and tables should be avoided at all costs if you wanted your application to perform well. This myth took root, and is unfortunately still prevalent today, even though times have significantly changed, and *the myth is now manifestly untrue*.

As a consequence of this myth and other falsehoods, we sometimes meet resistance from customers when they learn that for HPE Shadowbase replication to function, they must use audited files and tables. *The two main myths cited in opposition are: TMF will slow down my application and using TMF will require an application rewrite.* Neither of these myths – nor any other additional reasons – are valid, and do not provide *any* reason for the customer to not gain TMF protection and implement an HPE Shadowbase data replication solution.

Debunking the Two Main Myths

MYTH: *TMF will slow down my application.*
TRUTH: Far from creating performance issues, use of TMF auditing will almost always result in *dramatically* improved application performance. A primary reason for this myth is because it evolved during the period before significant enhancements were made to TMF and the disk process (DP2) software.[3] These enhancements include:

- Most importantly, the database disk processes can eliminate physical I/O operations by caching if updates are audited[4] with no possibility of data loss. When a system fails, updates in the audit trail are reapplied instead of being lost.

[1]Note: In this paper, non-audited files and data includes non-audited SQL tables and is interchangeable with the term "unaudited files and data."
[2]There are a few exceptions to this rule, but they are not germane to this discussion.
[3]DP2 is an HPE NonStop software component that provides the interface between TMF, the database, and file subsystems.
[4]To achieve optimal performance, ensure that the DP2 disk cache size is set to a sufficiently large value. As a rough guide, the DP2 disk cache should be at least 10-20% of the total amount of main memory available. DP2 disk cache size can be checked for each data volume by using the SCF INFO <volume>, CACHE and SCF STATS <volume> commands.

- TMF overhead is minimal, since it is implemented as part of the NonStop OS kernel and disk process. HPE NonStop as a platform was optimized for transaction processing; therefore, transaction processing requires a low-level of resources.
- Audit records sent from the database disk process to the audit trail disk process are blocked together, using a technique called *boxcarring*. A few shorter messages can efficiently represent a large number of transactions.
- Audit trail writes are also boxcarred. Audit for many transactions from many disks is collected and written to the end of the audit trail with a single I/O.
- TMF itself can be massively parallelized using many parallel audit trails (i.e., master and auxiliary audit trails).

These enhancements have busted the myth about poor performance when using audited data. Since some may still be skeptical, a performance analysis was undertaken,[5] comparing the transaction rates and response times for a sample application using audited and non-audited files. Much of the following performance discussion was taken from that study.

In this analysis in Figure 1, the absolute best transaction rate that could be achieved using non-audited files was 960 transactions per second (TPS), whereas using audited files reached 2,450 TPS. Note that this best non-audited transaction rate was achieved when using database buffering, which comes at the expense of possible data loss in the event of a failure – something which is not possible when using audited files.[6] For the same risk of data loss between file buffering vs TMF protection (using an unaudited, unbuffered database), only 150 TPS was achieved, far below the rate that the audited files achieved.

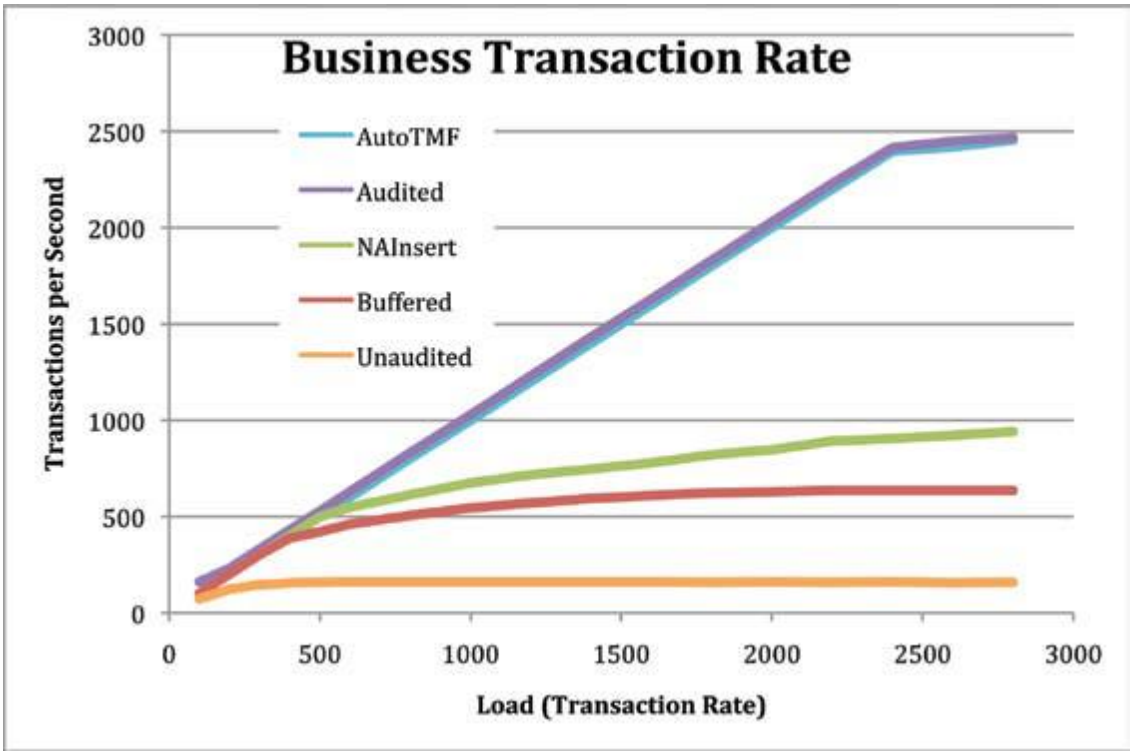


Figure 1 – Audited vs. Non-Audited Transaction Rate

The same result is true for transaction response times in Figure 2. For the same load on the system, when using unaudited files, the transaction response time was as much as 10x slower (> 500 ms) than when using audited files (< 50 ms).

[5] *Best Practices: Using TMF to Implement Business Continuity/Disaster Recovery*, Richard Carr, Carr Scott Software Inc., *The Connection*, Sept-Oct 2013 | Volume 34, No. 5.

[6] Note that "file buffering" (setting the BUFFERED file attribute) is different than TMF data caching. BUFFERED data can be lost if a failure occurs. However, TMF-cached data is always guaranteed to be recoverable.

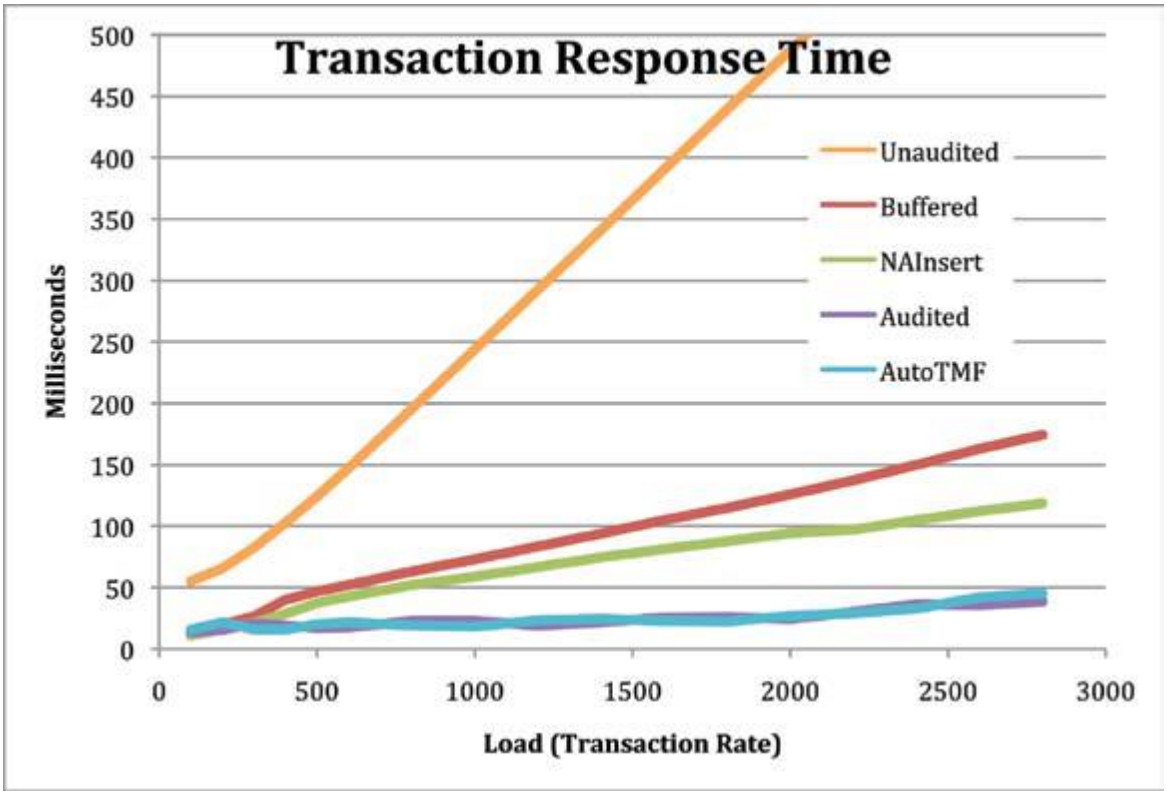


Figure 2 – Transaction Response Time

As confirmed by the performance analysis, the net result of these TMF/DP2 enhancements is to turn the myth on its head. If you want your application to perform well, and leverage the full capacity of your NonStop system, then you should use TMF audited files and tables.

MYTH: *Using TMF will require an application rewrite.*
TRUTH: The second objection most often raised about using audited data is that *the use of TMF will require an application rewrite*. For applications that do not currently use audited data, HPE NonStop AutoTMF software automatically provides TMF transactions without requiring any change to the application code or logic. AutoTMF works as effectively and efficiently as explicit use of TMF in terms of performance, online backup and recovery, and capturing database changes. AutoTMF is used in hundreds of customer applications. Some adopted AutoTMF to support TMF-based data replication such as RDF or HPE Shadowbase software, and many use AutoTMF to improve application performance.

Debunking More Myths

MYTH: *Since AutoTMF is an extra step in the transaction workflow, it degrades performance.*
TRUTH: Both non-audited data collection and AutoTMF use intercept technology to implement their key functions; however, far more work is performed in the non-audited intercept than in the AutoTMF intercept.

- If a non-audited data replication solution is being used, then *every* data change (insert, update, or delete) requires an extra I/O to the data replication processes and/or log files. This step adds *considerable* overhead to the non-audited file I/O pathlength and latency, and significantly degrades system and application performance (more than any additional overhead introduced by AutoTMF).
- Using AutoTMF as an intercept method does not add more overhead than using other intercept methods to log non-audited I/O data changes. (However, there is no such thing as a free lunch.) The key is to understand what happens after the intercept is invoked. AutoTMF simply makes sure that a TMF transaction is active when one needs to be. It is the file system (and TMF) that process the data collection by using caching; as previously explained, this sequence is very efficient and fault-tolerant.

MYTH: *Using AutoTMF and TMF can lose data that needs to be replicated.*
TRUTH: Non-audited replication can lose data that needs to be replicated; however, this loss *cannot* happen with AutoTMF and TMF, and for several reasons.

- When an I/O intercept mechanism is being used to log data changes for non-audited data, it is very easy to miss individual files, rows, or even entire tables when adding the intercept library to new applications or performing application updates. When this omission happens, the source application silently runs, making data changes at the source that are not collected and replicated to the backup database. The backup database is inconsistent with the primary database, and worse, the customer has no idea that it has occurred. Such database inconsistency cannot occur when using AutoTMF, because all data changes are automatically audited, and thus are collected and replicated. Forgetting to add the AutoTMF intercept does not allow the source application I/O's to occur against the source's audited file, and the problem is easily seen and corrected very quickly. No data loss occurs at the target.
- The extra steps to save the non-audited I/O into the replication engine's log files are also fraught with additional failure modes, possibly leading to data loss when failures occur (e.g., certain application failures, I/O cancellations, replication process failures, CPU failures, even system failures and restarts). These data loss scenarios cannot happen when using AutoTMF since TMF is completely fault-tolerant and has been specifically designed, implemented, heavily tested, and is literally used at thousands of sites. If you are still not convinced, please run some system failure tests while using both forms of collection (non-audited intercepting vs AutoTMF and TMF).
- On a related note, when any part of the non-audited data replication intercept or solution fails, it may negatively impact your source application's processing, even preventing it from running. This issue is not the case when TMF is used. The replication engine is a completely separate component from the application and TMF subsystem, and failure of any part of the replication engine does not directly impact the application processing.

MYTH: *Our data is temporal – it is old news after one or two minutes. Our business continuity plan is to simply fail-over to a standby node, and not to try to recover the failure. Therefore, we see no value in TMF recovery, since we do not want to have to conduct TMF ONLINE DUMPS or AUDITTRAIL DUMPS or perform any other additional work.*
TRUTH: The use of TMF auditing and AutoTMF does not mandate the use of TMF recovery, or the use of TMF dumps. An existing business continuity plan that does not rely on these TMF features can be maintained as is.

MYTH: *AutoTMF and TMF requires more disk space and retention for the audit trails.*
TRUTH: Yes, probably, but it is similar to the needs of the non-audited solution. Regardless of the collection method used, sufficient (and persistent) disk space must be allocated to contain the change data that needs to be replicated to the target system, and the files that hold the change data need to remain available until that data is replicated successfully. The audit trail requirements to satisfy this need should not be substantially different from the disk needs that the non-audited solution requires.

MYTH: *AutoTMF license fees are an added expense.*
TRUTH: Yes, AutoTMF is an add-on product; however, the license cost of AutoTMF and, for example, HPE Shadowbase data replication, is competitive with the license costs of other competing replication products. However, the key point is that AutoTMF should more than offset this *cost* by eliminating the potential risk involved with inconsistent data, data loss, and broken file links, which do not occur in a TMF-protected database and are far more significant and can occur in a non-audited environment.

MYTH: *AutoTMF and TMF does not preserve database consistency.*
TRUTH: Another TMF benefit that must not be overlooked is that of *data integrity* (consistency). Without TMF and its inherent ACID[7] transaction semantics, applications may make data changes that are not fully completed, especially if a data change requires multiple database operations, and/or is split across several applications or sub-routines. Without the use of transactions, recovery from failures in such circumstances is complex and error-prone, and is very likely to leave the database in an inconsistent state (which can be very costly). By using TMF and audited data, data integrity and error recovery is completely handled by TMF; either all of the database operations complete or none of them do, always leaving the database in a consistent state.

[7]Atomicity, Consistency, Isolation, Durability (ACID) are the foundational properties of transactional database operations. Paul J. Holenstein, Dr. Bruce Holenstein, Dr. Bill Highleyman, *Breaking the Availability Barrier III: Active/Active Systems in Practice*, Chapter 16 (AuthorHouse, 2007).

Adding AutoTMF also avoids the possibility of a base file becoming inconsistent with its alternate key file or index. This possibility can (and does) occur for non-audited files and tables (for example, resulting in the dreaded file system error 59 situation).

MYTH: *Knowledgeable professional services personnel are not available in my region to assist us with implementing and deploying a TMF-based solution.*
TRUTH: HPE and select partners have implemented and deployed hundreds of AutoTMF and TMF-based replication solutions for a wide variety of customers, and have provided global and regional training to enable the customer and reseller staff to maintain the solutions.

MYTH: *My non-audited application and database are 'old' and 'legacy?' No way!*
TRUTH: Newer and younger management teams tend to view a non-audited application and database that cannot maintain (or guarantee) data integrity and consistency as "antiquated" or "legacy" – out of touch with current best-practices for relational database management systems (RDBMS). Such a view may lead to replacement of the application, DBMS, or even staff, with something "newer and better." Using TMF eliminates these concerns and gives you these features now.

MYTH: *AutoTMF and TMF does not future-proof applications.*
TRUTH: Future technical improvements in NonStop data protection will require TMF auditing to be in place in order to leverage these advanced capabilities. For example, HPE Shadowbase software has implemented a new and unique zero data loss (ZDL)[8] capability to further protect customer data. This technology ensures that customers' data changes are safe-stored on a target system before the source data changes are allowed to commit. Any subsequent failure of the source system will not lose any critical data, regardless of the type of failure that occurs at the source system, datacenter, or communications network. This additional capability is simply not available without the use of TMF auditing.

Summary

Far from being an impediment, having a TMF audited database not only offers significant operational, reliability, and data integrity advantages, it also improves overall system performance and can lead to improved capacity utilization. The migration from a non-audited to an audited application is very simply made using the facilities provided by AutoTMF, and introduces no significant overhead. The myths and other objections raised against the use of TMF auditing and AutoTMF simply do not stand up to scrutiny.

- The use of TMF and AutoTMF does not impact performance; in fact, in most cases it dramatically improves it.
- The use of AutoTMF does not require an application rewrite, nor typically any application modifications at all.
- The use of TMF does not require use of TMF recovery or the taking/management of TMF dumps.
- The use of AutoTMF is not expensive in comparison to the potential costs of inconsistent or missing data.
- The use of TMF ensures database consistency and brings an unaudited database up-to-date to the best practices available for DBMS data management.
    - Therefore, it may help to save one's application (or job) and database from being replaced by 'something' more modern.
- The use of TMF helps to future-proof applications, enabling the exploitation of new capabilities (such as zero data loss).

Thus, the path to improving your overall non-audited environment runs through a TMF implementation, and the existence of applications using non-audited data is not an impediment to the deployment of an HPE Shadowbase audit-based data replication solution.

We feel strongly about helping companies improve their NonStop applications and databases with auditing, so along with our HPE colleagues, we will gladly assist with running a trial/proof-of-concept with your application and your data, in your IT environment. Seeing is believing!

More details about the benefits of using TMF, the performance analysis, and the use of AutoTMF can be found in the article, *Best Practices: Using TMF to Implement Business Continuity/Disaster Recovery, written* by

[8]For more information, visit Shadowbase Zero Data Loss (ZDL).

Richard Carr of Carr Scott Software Inc., and a TMF survey article, *Boosting Performance with Every Transaction*, by Charles Johnson (former *Distinguished Technologist* and *TMF Elder* at Tandem/Compaq/HPE). We are grateful to Richard for allowing us to reference his article.[9]

[9] For more information, please visit: CarrScott.com

International Partner Information

Global

Hewlett Packard Enterprise
6280 America Center Drive
San Jose, CA 95002
USA
Tel: +1.800.607.3567
www.hpe.com

Japan

High Availability Systems Co. Ltd
MS Shibaura Bldg.
4-13-23 Shibaura
Minato-ku, Tokyo 108-0023
Japan
Tel: +81 3 5730 8870
Fax: +81 3 5730 8629
www.ha-sys.co.jp

Gravic, Inc. Contact Information

17 General Warren Blvd.
Malvern, PA 19355-1245
USA
Tel: +1.610.647.6250
Fax: +1.610.647.7958
www.shadowbasesoftware.com
Email Sales: shadowbase@gravic.com
Email Support: sbsupport@gravic.com

Hewlett Packard Enterprise Business Partner Information

Hewlett Packard Enterprise directly sells and supports Shadowbase Solutions under the name ***HPE Shadowbase***. For more information, please contact your local HPE account team or visit our website.