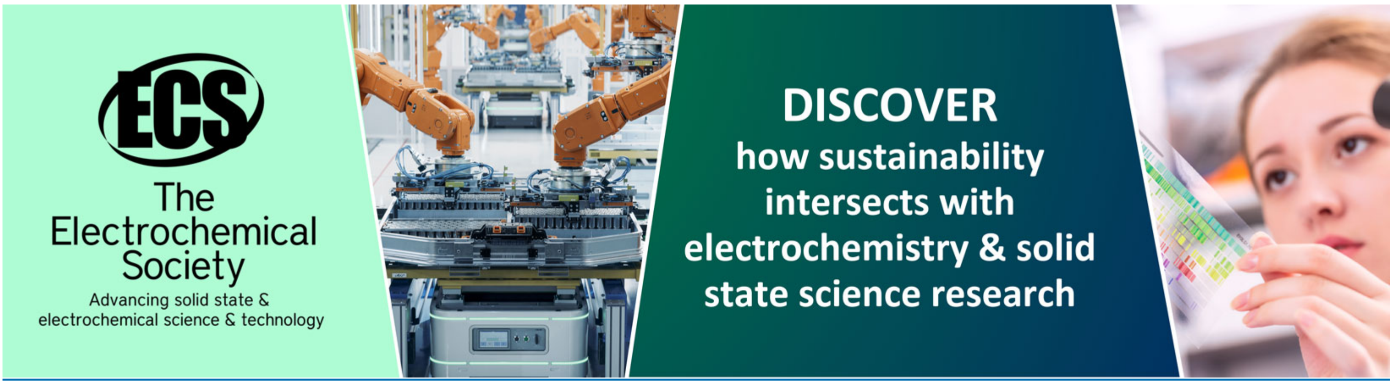
Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Tape SCSI monitoring and encryption at CERN

To cite this article: Stefanos Laskaridis *et al* 2017 *J. Phys.: Conf. Ser.* **898** 062005

View the article online for updates and enhancements.


ECS
The Electrochemical Society
Advancing solid state & electrochemical science & technology
DISCOVER
how sustainability intersects with electrochemistry & solid state science research

Tape SCSI monitoring and encryption at CERN

Stefanos Laskaridis, V Bahyl, E Cano, J Leduc, S Murray, G Cancio and D Kruse
CERN – European Organization for Nuclear Research, Geneva, Switzerland

E-mail: {Steve.Laskaridis,Vladimir.Bahyl}@cern.ch

Abstract. CERN currently manages the largest data archive in the HEP domain; over 180PB of custodial data is archived across 7 enterprise tape libraries containing more than 25,000 tapes and using over 100 tape drives. Archival storage at this scale requires a leading edge monitoring infrastructure that acquires live and lifelong metrics from the hardware in order to assess and proactively identify potential drive and media level issues. In addition, protecting the privacy of sensitive archival data is becoming increasingly important and with it the need for a scalable, compute-efficient and cost-effective solution for data encryption.
In this paper, we first describe the implementation of acquiring tape medium and drive related metrics reported by the SCSI interface and its integration with our monitoring system. We then address the incorporation of tape drive real-time encryption with dedicated drive hardware into the CASTOR [1] hierarchical mass storage system.

1. Introduction

High Energy Physics is a scientific area which involves the production of large datasets. These data are characterized by their sheer volume, high burst write rates and infrequent reads; traits which impose a need for efficient cold storage solutions.

CASTOR, the CERN Advanced STORage manager [1], is a hierarchical storage manager developed at CERN for storing LHC physics data. It consists of a disk caching level and a tape backend for permanent data storage. Currently amounting for more than 25,000 tapes, 100 tape drives spread across 7 libraries saved are approximately 500 million files. Security of these data poses a main concern raising two issues: protection against unanticipated malfunctions and prying eyes.

At this scale, hardware failures are a frequent phenomenon. Thus, we are proposing a way to proactively identify actual and potential drive and/or media level issues by acquiring system-level information made available via drive SCSI log pages.

On the other hand, privacy of the saved data is another topic of widespread concern. Therefore, we opted to incorporate real-time encryption on sensitive data saved on tape utilizing the dedicated hardware in the tape drives.

2. SCSI Monitoring

Monitoring has always been an important aspect of CASTOR. Saving large amounts of data is a delicate process that needs to be systematically observed for post-mortem diagnostics or in order to prevent data loss due to internal or external factors. The aim of SCSI monitoring is to dig into the internal reporting tools of the infrastructure to acquire relevant metrics from tape drives and channel them into CASTOR's logs. From there, these values would be retrieved by the monitoring infrastructure tools, evaluated and alerts would be raised on anomalies. This, along with the already in place SCSI tape alerting subsystem [2], would offer a deeper insight on the state of the tape equipment.

2.1. Metrics

Each tape vendor offers a variety of tape, tape drive and library related values reported to the user through the drive's SCSI interface. Unfortunately, not all vendors report the same values or fully adhere to the T10 specifications [3]. This results in a per-vendor implementation and matching of equivalent metrics attempt, which in turn produces more complex code reporting statistics, potentially not directly interrelated.

Metrics collected can be divided into four main categories: mount general, volume, drive and quality statistics. For each category, the following metrics are collected:

Table 1. SCSI Metrics collected per vendor [4,5]

<table>
<tr><th>IBM</th><th>ORACLE</th></tr>
<tr><td colspan="2">1. Mount general statistics</td></tr>
<tr><td colspan="2">Write Errors (0x02)</td></tr>
<tr><td colspan="2"><i>Total Corrected Write Errors</i>
<i>Total Write Bytes Processed</i>
<i>Total Uncorrected Write Errors Count</i></td></tr>
<tr><td colspan="2">Read Errors (0x03)</td></tr>
<tr><td colspan="2"><i>Total Corrected Read Errors</i>
<i>Total Read Bytes Processed</i>
<i>Total Uncorrected Read Errors Count</i></td></tr>
<tr><td colspan="2">Non-Medium Errors (0x06)</td></tr>
<tr><td colspan="2"><i>Total Non-Medium Errors Count</i></td></tr>
<tr><td colspan="2">2. Volume Statistics</td></tr>
<tr><td colspan="2">Volume Statistics (0x17)</td></tr>
<tr><td colspan="2"><i>Page valid</i></td></tr>
<tr><td colspan="2"><i>Volume Mounts</i></td></tr>
<tr><td colspan="2"><i>Volume Recovered Write Data Errors</i></td></tr>
<tr><td><i>Volume Unrecovered Write Data Errors</i></td><td>-</td></tr>
<tr><td colspan="2"><i>Volume Recovered Read Errors</i></td></tr>
<tr><td><i>Volume Unrecovered Read Errors</i></td><td>-</td></tr>
<tr><td colspan="2"><i>Volume Manufacture Date</i></td></tr>
<tr><td><i>Beginning of Medium Passes</i></td><td>-</td></tr>
<tr><td><i>Middle of Tape Passes</i></td><td>-</td></tr>
<tr><td colspan="2">3. Drive Statistics</td></tr>
<tr><td>Drive Write, Read Forward/Backwards Errors (0x32, 0x34, 0x36)</td><td>Vendor Unique Statistics (0x3D)</td></tr>
<tr><td><i>Data Acquisition Temps</i></td><td><i>Temporary Drive Errors</i></td></tr>
<tr><td><i>Servo Temps</i></td><td><i>Servo Temporaries</i></td></tr>
<tr><td><i>Servo Transients</i></td><td><i>Servo Transient Conditions</i></td></tr>
<tr><td><i>Data Transients</i></td><td><i>Read/Write Transient Conditions</i></td></tr>
<tr><td><i>Total Retries</i></td><td><i>Read/Write Recovery Retry Count</i></td></tr>
<tr><td colspan="2">4. Quality Statistics</td></tr>
<tr><td>Performance Characteristics (Mount and Lifetime) (0x37)</td><td>Vendor Unique Statistics (0x3D)</td></tr>
<tr><td><i>Drive Efficiency</i></td><td>-</td></tr>
<tr><td><i>Media Efficiency</i></td><td><i>Tape Efficiency (TEFF)</i></td></tr>
<tr><td><i>Primary Interface Efficiency Port {0,1} Efficiency</i></td><td>-</td></tr>
<tr><td><i>Library Interface Efficiency</i></td><td></td></tr>
<tr><td><i>Read Performance Efficiency</i></td><td><i>Read Quality Index (RdQI)</i></td></tr>
<tr><td><i>Write Performance Efficiency</i></td><td><i>Write Efficiency (WEFF)</i></td></tr>
</table>

2.2. Interpretation

Starting January 2016, SCSI metrics collection was incorporated into CASTOR, and the values were tunnelled into log files. Their interpretation was postponed until sufficient data was available for analysis, in mid-July 2016. Our study is mainly focused on common metrics across the two vendors.
Concerning mount general statistics, it can be observed that IBM error metrics generally report higher numbers than Oracle's. This does not necessarily mean that the former drives produce more errors; they simply measure more. Additionally, big sessions can saturate these counters.

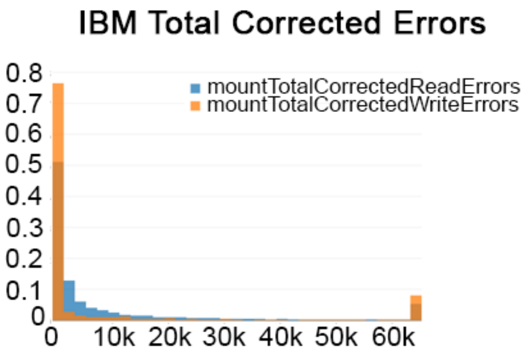


Figure 1. IBM Total Corrected Read/Write Errors statistical distribution

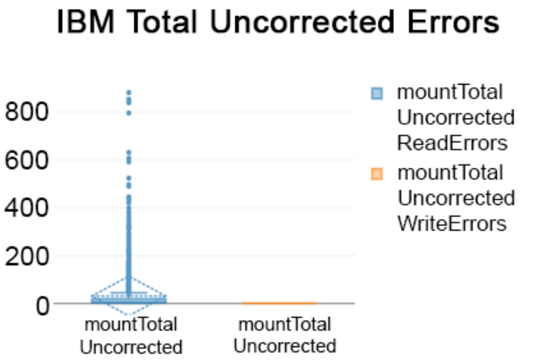


Figure 2. IBM Total Uncorrected Read/Write Errors boxplots

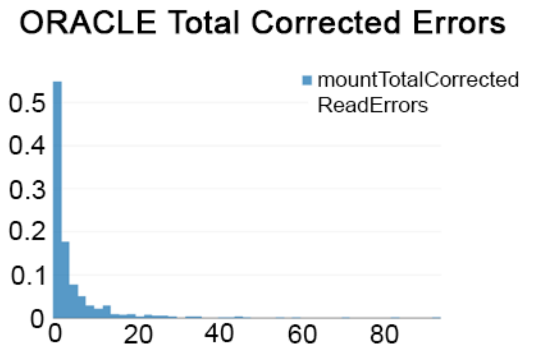


Figure 3. Oracle Total Corrected Read Errors statistical distribution

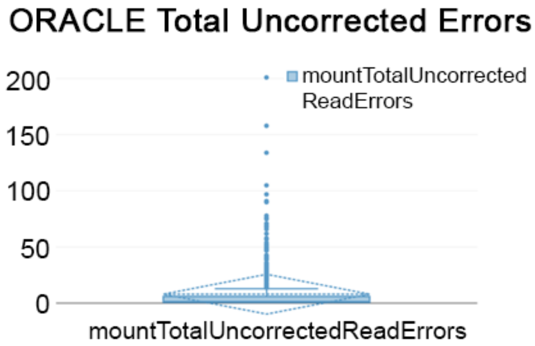


Figure 4. Oracle Total Uncorrected Read Errors boxplot

In the graphs above (see figures 1-4), the statistical distribution of Corrected/Uncorrected, Read and Write errors is depicted per vendor. No Write Errors are shown in Oracle's case due to low write usage during the observation period.
It is interesting to note that no non-zero occurrence of non-medium errors or Oracle Corrected Read Errors was obtained.
Measurements further indicate the following workflow (see figure 5):

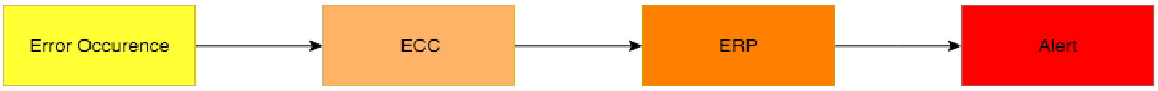


Figure 5. Error to alert workflow
ECC: Error Correcting Code (software), ERP: Error Recovery Processes (may include physical retry)

In more detail, when an error occurs, Error Correcting Code gets initiated on the tape drive to remedy the situation. In case of failure, Error Recovery Procedures are run. On subsequent failure, an unrecovered error gets logged and a tape alert gets raised.

For the following metrics categories, we initially grouped the reported values both by tape and by drive, in order to link potential elevated error metrics with imminent medium or equipment failure. High reported values across different categories do not seem to be related.
We tested volume metrics[1] against real hardware cases, expecting problematic hardware to lead to high error metrics.

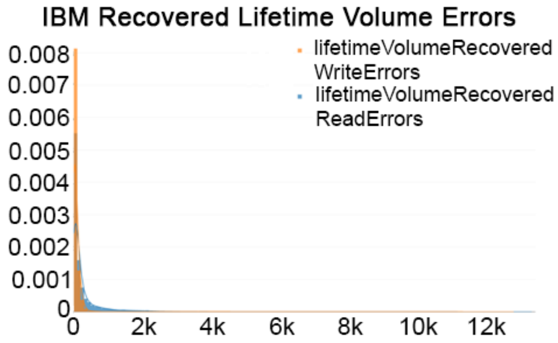


Figure 6. IBM Recovered Lifetime Volume Errors statistical distribution

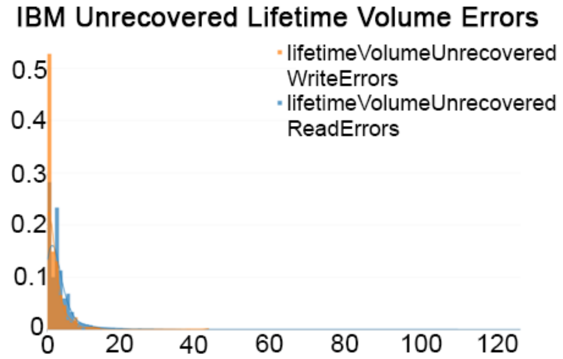


Figure 7. IBM Unrecovered Lifetime Volume Errors statistical distribution

In the graphs above (see figures 6,7), a statistical distribution of non-zero recovered and unrecovered volume read and write errors is shown.

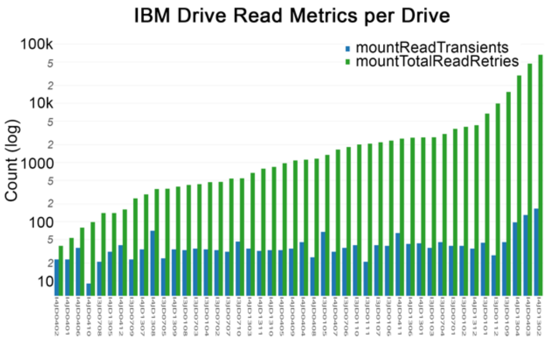


Figure 8. IBM Drive Read Metrics grouped by drive *{mountReadTransients, mountTotalReadRetries}*

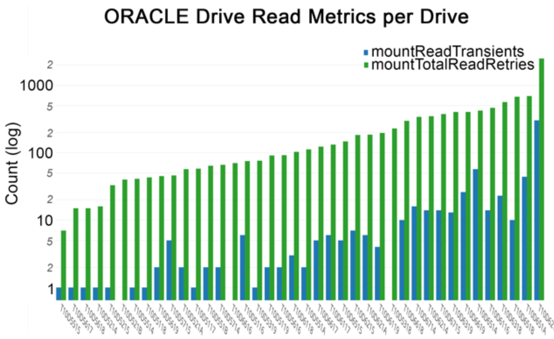


Figure 9. Oracle Drive Read Metrics grouped by drive *{mountReadTransients, mountTotalReadRetries}*

As far as the drive statistics are concerned, in the graphs above (see figures 8,9), the average count of read retries are shown along with transient errors, grouped per tape drive for the two vendors. High values of one metric do not link with high reported values of the others. Moreover, none of the metrics seemed to relate with already raised tape alerts.

Concerning quality statistics, I/O efficiencies were normalized and the most used tapes per operation[2] were selected and analysed for efficiency equivalently. The metrics were grouped by tape and ordered by timestamp. In order to detect potential trends, the data were fitted with 3rd degree polynomial functions.

[1] Volume metrics became available through SCSI Log Page 0x17 from Oracle with the latest firmware and though incorporated into CASTOR, not enough data was available, thus they are not analysed in this study.
[2] Operations refer to read and write sessions.

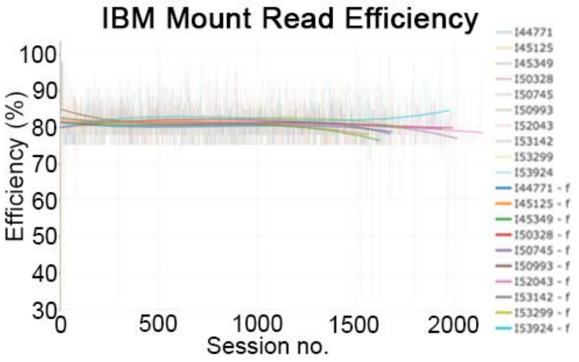


Figure 10. IBM Mount Read Efficiency trends in time per tape for top read-mounted tapes

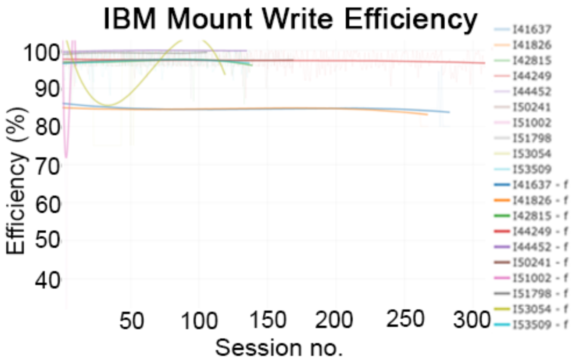


Figure 11. IBM Mount Write Efficiency trends in time per tape for top write-mounted tapes

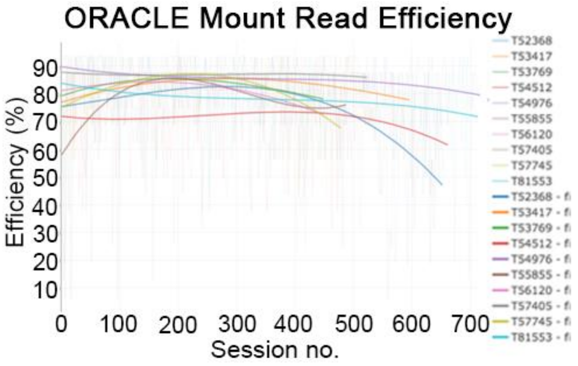


Figure 12. Oracle Mount Read Efficiency trends in time per tape for top read-mounted tapes

As depicted (see figures 10-12), for many tapes' cases a slight downward trend can be noticed, showing that tape condition deteriorates over time. In a larger timeframe, this trend could become more evident.

In terms of imminent failure prediction, we found no statistically significant linkage between the metrics gathered and tape failure alerts.

3. Tape Encryption

Privacy is an omnipresent affair when dealing with data. Encryption is to ensure that only authorized clients who have obtained the encryption key will be able to access the encrypted information.

We opted for implementing encryption on the drive-application level[3] for the following reasons:

- *Speed*: dedicated hardware in the tape drive is responsible for real-time encryption of the data
- *Versatility*: key scope can vary from per library to per block of the tape scope and is externally managed
- *Compression*: data is first compressed and then encrypted by the drive

[3] Application Level Encryption (AME): Key management is handled in application level. Encryption is done on tape drive dedicated hardware.

- *Transparency*: the encryption operation of the data stored on tape is completely transparent to the end user

There are essentially two parts in implementing encryption on tape drives:

- The encryption key management
- The encryption implementation, given the key

3.1. Key Management

Key management is implemented completely outside of CASTOR and is enabled as a plugin. Thus, CASTOR is not tied to one specific implementation of a key manager, nor does it enable encryption by default, but merely imposes a JSON interface to interact with it.

The current key management solution inherits a simplistic approach and consists of three entities: a key store, a table in the VMGR database and a script implementing the logic.

Key Store

A key-value store associating a key ID with a key. Key ID's are versioned, enabling key revocation.

VMGR Database

VMGR is a database keeping information for each tape in CASTOR. We created a new table, associating each tape with a key ID.

Encryption System Backend

CASTOR contains all the SCSI calls to interface with the tape drives. Operations in CASTOR are fulfilled in sessions. Essentially, there are three classes of sessions: read, write and labelling. The latter, which is responsible for writing a label file[4] to each tape, is always done without encryption.

In the backbone of the backend encryption implementation there are two methods, one passing the encryption parameters to the drive and one clearing them out; each of which issues an `sg_io` ioctl to the drive, containing a SPOUT SCSI command payload [3,4,5].

Both vendors implement AES-256 [6] symmetric encryption algorithm for encrypting the data written, following the T10 specification [3].

3.2. Workflow

I/O with encryption support

In the beginning of a session, CASTOR initially clears any encryption data left as a remainder of a potentially failed session. Afterwards, the key management application is called where the Tape's Virtual ID is passed and a parameter on whether to update the encryption state of the tape or not.

The encryption key manager first polls the VMGR database for the key ID associated with the specific tape. If existent, it extracts the key from the key store and returns it to the caller. In case there is no associated key ID with the tape and this is the first write to it, then the application gets the latest version of the key and updates the key ID to the VMGR database, essentially declaring a mapping of data written on a specific tape and a key ID. If it is not the first write, which means the tape contains non-encrypted data, encryption is disabled.

Key revocation process

In the case some key needs to get revoked, we move all the tapes associated with it to a new tape pool, then create a new version of the pair key ID – key and rewrite the data with the new key.

3.3. I/O Benchmarks

Methodology

One key aspect of opting for integrating encryption with CASTOR was the lightweight performance penalty associated with it. For this reason, we measured the performance under the following parameters:

- Two drive types: {IBM TS1150, Oracle T10000D}
- Two file sizes: {small: 5.2MB, big: 5.2GB}

[4] A label file is a CASTOR specific file containing information about the tape.

- Two operations: {write, read}
- Two modes: {non-encrypted,encrypted}
- 200 iterations of each operation

The I/Os were performed with the dd UNIX command-line utility [7], whose performance output was captured and interpreted.

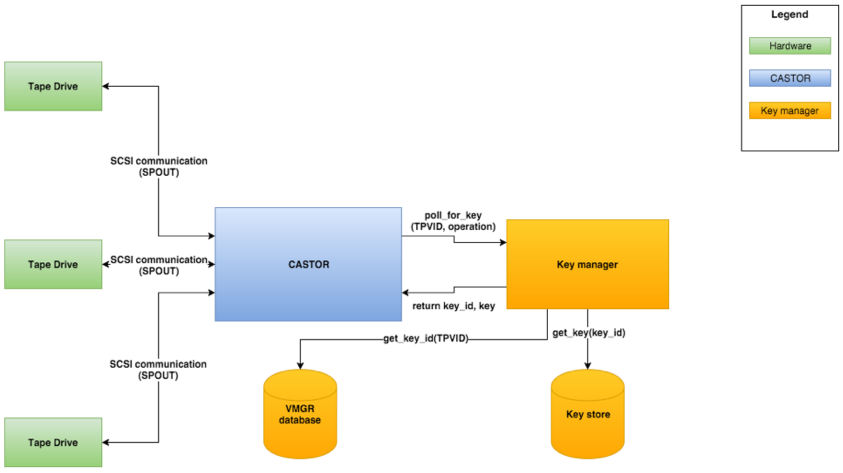
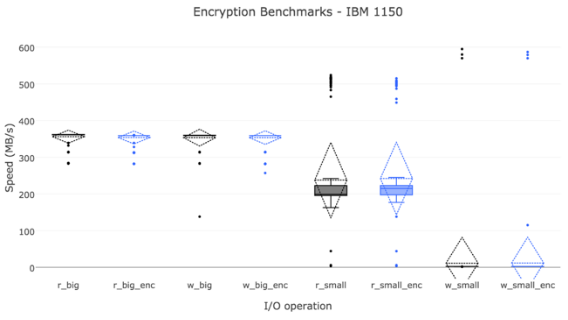


Figure 13. Encryption architecture in CASTOR

Results

In neither vendor's case (see figures 14, 15), could we observe a statistically significant performance drop between non-encrypted and encrypted operations.

It should be highlighted that outliers - more often present in small file sizes - are omitted from our study as they warp the experiment space without adding value, being present for both encrypted and unencrypted operations.



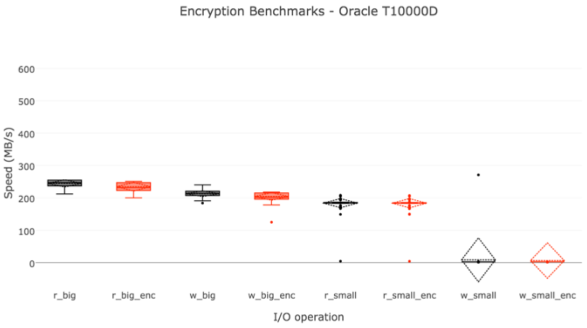


Figure 14. IBM Encryption benchmarks for each operation **Figure 15.** Oracle Encryption benchmarks for each operation

4. Conclusions and Outlook

4.1. Assessment of SCSI Metrics

The outcome of our research on failure prediction based on data gathered from the tape drives' SCSI interfaces was generally non-conclusive. We did not manage to correlate high error values with tape or tape drive failures, nor to predict the latter.

Moreover, the different implementations between vendors on the backend of these metrics further aggravates the results, rendering it more difficult to find a prediction model.

4.2. Encryption status

Encryption has already been integrated to production, deployed across the whole CERN's tape infrastructure and enabled for specific tape pools[5] containing sensitive data. No major issues or performance drops have been observed so far on either vendor.

On the other side of the coin, we feel entitled to raise the following two concerns:

First, the public availability of the encryption algorithm implementation. The security of the data should not be based on the secrecy of the algorithm implementation, but on the secrecy of the key [8]. Moreover, security through obscurity could lead to algorithms with serious vulnerabilities, given that the code is exposed to less people for scrutiny.

Secondly, given the computational power advances (doubling every 18 months [9]), and in combination with the fact that data on cold storage are almost never deleted, one secure algorithm today may not guarantee privacy in the future. Last, recent advances in quantum computing could render AES non-secure[6]. This means there could be scenarios where one could obtain encrypted tape data today and be able to decrypt them without having the key in certain years from now.

4.3. Future work and extensions

As far as SCSI monitoring is concerned, while failure prediction based on gathered metrics failed to produce meaningful results, further investigation might indicate deeper latent connections that prove the causality of failures. By correlating metrics with specific tape sections touched per session, a map of performance/error count and tape partitions could be created, potentially revealing problematic parts of the tape. Moreover, a feedback mechanism for evaluating the metrics gathered could be created by employing supervised machine learning techniques based on historical data, in order to predict failure based on SCSI metrics gathered as features.

On the encryption side, the feedback has been positive. The next steps include making our key management system more robust by being self-contained in its own database abolishing interdependencies with CASTOR. This way, the key manager would be able to be plugged-in in different systems.

References

[1] CASTOR homepage http://cern.ch/castor
[2] Experiences and Challenges running CERN's High-Capacity Tape Archive https://indico.cern.ch/event/304944/contributions/1672655/attachments/578843/797036/CHEP-2015-CERN-Tape-Archive.pdf
[3] SCSI T10 Standards http://www.t10.org/pubs.htm
[4] IBM System Storage Tape Drive 3592 SCSI Reference
[5] StorageTek T10000 Tape Drive Fibre Channel Interface Reference Manual
[6] Daemen J and Rijmen V 2003 AES Proposal: Rijndael http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf
[7] dd command-line utility https://www.gnu.org/software/coreutils/manual/html_node/dd-invocation.html
[8] Kerckhoffs A 1883 A La cryptographie militaire *Journal des sciences militaires*
[9] Moore G E 1965 Cramming more components onto integrated circuits http://www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf
[10] Grover L 1996 A fast quantum mechanical algorithm for database search *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, arXiv:quant-ph/9605043

[5] Logical groups of tapes inside CASTOR

[6] Grover's quantum algorithm [10]: Finds with high probability the unique input to a black box function that produces a particular output value, using just $O(\sqrt{\frac{1}{2}})$ evaluations of the function, where N is the size of the function's domain. AES-256 is still secure.