Selective scrubbing based on algorithmic randomness

Contents

[1. Abstract 3](#_Toc98117808)

[2. Motivation 4](#_Toc98117809)

[3. Prior Art 4](#_Toc98117810)

[4. Proposed Solution 4](#_Toc98117811)

[4.1 Framework 4](#_Toc98117812)

[4.2 Health ranking of disk drives 5](#_Toc98117813)

[4.3 Scrub cycle according to health score 6](#_Toc98117814)

[4.4 Disks grouped for Scrub cycle 6](#_Toc98117815)

[4.5 System work load predictor 6](#_Toc98117816)

[4.6 Scrubbing schedule window 8](#_Toc98117817)

[5. Novelty 8](#_Toc98117818)

[6. Non-obviousness 8](#_Toc98117819)

[7. Utility 8](#_Toc98117820)

[8. Advantages over former approach 9](#_Toc98117821)

# Abstract

Disk scrubbing is a process of performing full media pack sweeps across allocated and unallocated disks and if latent medium error is detected then rebuild the corrupted data, which in turn reduces the chance of bad block media detection during host IO activity. However, running scrubbing task for entire population of disks in an array significantly increases the load of the data storage system, and may degrade its performance. Reading data from the disk for the disk scrubbing may also result in wear on the disk. Apart from that if the disk has larger capacity (~ 12TB), it will take comparatively large amount of time for the operation.

To overcome the impact from this issue in Enginuity, we can only scrub the disk for which the operation is really required. We use a learning framework based on Conformal Prediction framework to pin-point specific disks for scrubbing. The method is machine leaning agnostic and translates to a binary classification task where we proactively forecast (n-days ahead) the health of a disk as GOOD and BAD (not healthy). We create a set of BAD drives (not healthy) and quantify ***“how” unhealthy*** is it across the entire storage pool based on prediction’s confidence. This metrics is used to prioritize (selective) scrubbing for the drives.

The proposed method has two-fold advantages. First, the method can be used to forecast disk drive failure in PowerMAX. Second, the quantified output from the forecast engine can be fed as an input for Scrubbing scheduler engine.

* A proactive approach that provides value for business by earlier failure detection and ***selective scrubbing*** 
  + by resource/power savings in data centers for the data scrubbing (e.g. by spinning disks down)
  + Only the required disks need be scrubbed on a priority basis, and the healthy disks can be scrubbed rarely or not at all
* Schedule scrubbing of selected unhealthy disks ***when the system is idle*** (system workload prediction)

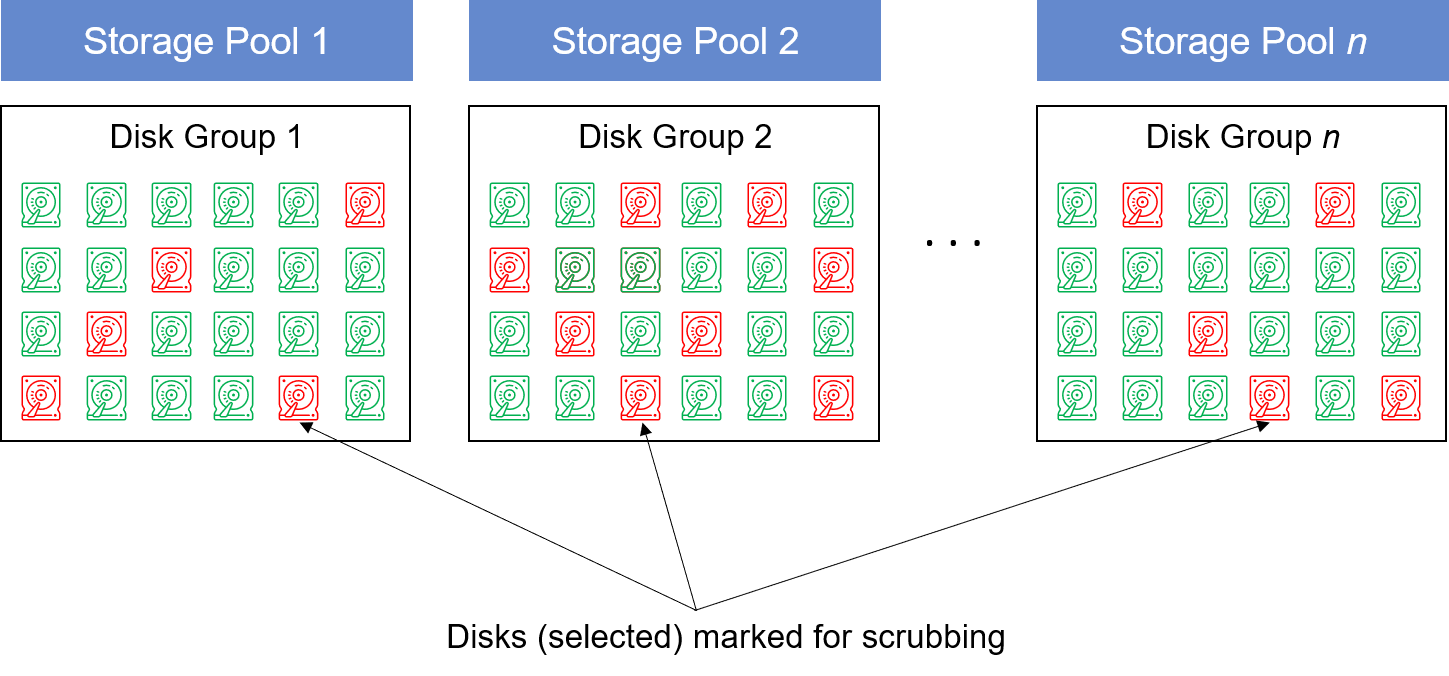


Figure 1: Disks selected for scrubbing based on quantification of “how-bad” are the Disks

# Motivation

After a careful study, we infer those Statistical assessments seem to overcome the limitations of probabilistic approaches and use statistical framework which uses hypothesis testing, algorithmic randomness and transductive inference for quantifying prediction’s output (detailed explanation in Solution section).

# Prior Art

We noticed that Enginuity also scrubs all the disks (Most of existing methods run disk scrubbing operation in a constant cycle, such as once two months, monthly or bi-weekly). This scrubbing frequency not only degrade the system performance, but also is harmful to storage reliability.

* Nutanix performs scrubbing based on cold that (once per day) concurrently across all drives in the cluster so the chance of multiple concurrent failure occurring such as a drive failure and a corrupted extent.
* [US8407191](https://patents.google.com/patent/US8407191): (Dell EMC)Uses “priority-based” scrubbing in a deduplication-based storage.

# Proposed Solution

## Framework

A high-level overview of proposed method is shown below. It uses Conformal Prediction framework and schedules scrubbing for selected disks (which are not healthy) during system idle time.



Figure 2: Framework of smart disk scrubbing

**Method**

1. Obtain SMART data for each disk in the enclosure for a period of 6 months (user defined) and create a dataset having disk state as NORMAL or FAILED
2. From the dataset obtained in step 1, use the relevant parameters like *medium\_err, disk\_reallocated, disk\_tempc, uncorr\_rd\_err, uncorr\_wrt\_err, phy\_err\_other, start\_stop\_count, disk\_busy, log\_cnt, range\_days, power\_on\_hours, reco\_err\_uniq, recov\_err\_uniq, and err\_head*
3. Choose a classification algorithm for classifying the heath of disks n-days ahead (here we have used Random Forest; but it can also be extended to other algorithms like Online SVM, Online SGD classifiers)
   1. Design a non-conformity measure for the choice of algorithm, i.e., Random Forest
   2. Find the p-values for the disks; one assuming the disk is in NORMAL state and other assuming the disk is in FAILED state
   3. Create a prediction set whose p-values satisfies the confidence level
   4. For each disk assign health confidence
4. From step 3.d we will have a set of (NORMAL and FAILED) disk with health score
   1. Reject all the disks with status == FAILED from the list and send an alert to administrator
   2. For all the disks with status == NORMAL, arrange them in descending order
5. Group the disks in three category – Best, Medium and Poor
6. Assign required Scrubbing frequency cycle to each category. For example, for disks with poor heath score, they should be treated with high frequency of scrubbing, and disks with good heath, the scrubbing frequency should be low.

## 4.2 Health ranking of disk drives

Binary classification (ML algorithm) wrapped over conformal predictor gives the status of each disk drive along with confidence associated with that (this is used for ranking disk drives).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sequence | Disk Serial Number | Health Rank | Status |  |
| 1 | Z296Z3KN\_\_\_\_00009330KBD2 | 0.934426 | NORMAL |  |
| 2 | Z296Z3A6\_\_\_\_00009330EHZU | 0.819672 | NORMAL |  |
| 3 | Z296Z5TP\_\_\_\_00009329ZNXU | 0.770492 | NORMAL |  |
| 4 | Z296Z45F\_\_\_\_00009330JWGZ | 0.763187 | NORMAL |  |
| 5 | Z296Z5MQ\_\_\_\_00009330K4FR | 0.754098 | NORMAL |  |
| 6 | Z296Z421\_\_\_\_00009330KB5X | 0.606557 | NORMAL |  |
| 7 | Z296Z4LB\_\_\_\_00009330K3D5 | 0.491803 | NORMAL |  |
| 8 | Z296Z5R8\_\_\_\_00009330EX5L | 0.47541 | NORMAL |  |
| 9 | Z296Z43S\_\_\_\_00009330JX16 | 0.459016 | NORMAL |  |
| 10 | Z296Z3MV\_\_\_\_00009330K6PW | 0.360656 | NORMAL |  |
| 11 | Z296Z5NR\_\_\_\_00009330EJ5E | 0.278689 | NORMAL |  |
| 12 | Z296Z5H8\_\_\_\_00009330EX93 | 0.163934 | NORMAL |  |
| 13 | Z296Z3EA\_\_\_\_00009330JW2D | 0.131148 | NORMAL |  |
| 14 | Z296Z3NM\_\_\_\_00009329JENE | 0.081967 | NORMAL |  |
| **. . .** | **. . .** | **. . .** | **. . .** |  |
| 30 | Z296Y6DW\_\_\_\_00009330JUKS | 0.043547 | NORMAL |  |

Figure 3: Ranking of Disk health based on prediction confidence

## Scrub cycle according to health score

|  |  |  |
| --- | --- | --- |
| Disk Health | Scrub Frequency | Heath Score |
| Best | LOW (CYCLE – A) | 0.9 – 0.7 |
| Medium | MEDIUM (CYCLE - B) | 0.6 – 0.4 |
| Poor | HIGH (CYCLE – C) | 0.3 – 0.1 |

Figure 4: Mapping table for disk heath score and scrubbing frequency

## 4.4 Disks grouped for Scrub cycle

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sequence | Disk Serial Number | Health Score | Status | **Scrub Frequency** |  |
| 1 | *Z296Z3KN\_\_\_\_00009330KBD2* | *0.934426* | *BEST* | CYCLE - A |  |
| 2 | *Z296Z3A6\_\_\_\_00009330EHZU* | *0.819672* | *BEST* | CYCLE - A |  |
| 3 | *Z296Z5TP\_\_\_\_00009329ZNXU* | *0.770492* | *BEST* | CYCLE - A |  |
| 4 | *Z296Z45F\_\_\_\_00009330JWGZ* | *0.763187* | *BEST* | CYCLE - A |  |
| 5 | *Z296Z5MQ\_\_\_\_00009330K4FR* | *0.754098* | *BEST* | CYCLE - A |  |
| 6 | *Z296Z421\_\_\_\_00009330KB5X* | *0.606557* | *MEDIUM* | CYCLE - B |  |
| 7 | *Z296Z4LB\_\_\_\_00009330K3D5* | *0.491803* | *MEDIUM* | CYCLE - B |  |
| 8 | *Z296Z5R8\_\_\_\_00009330EX5L* | *0.47541* | *MEDIUM* | CYCLE - B |  |
| 9 | *Z296Z43S\_\_\_\_00009330JX16* | *0.459016* | *MEDIUM* | CYCLE - B |  |
| 10 | *Z296Z3MV\_\_\_\_00009330K6PW* | *0.360656* | *POOR* | CYCLE - C |  |
| 11 | *Z296Z5NR\_\_\_\_00009330EJ5E* | *0.278689* | *POOR* | CYCLE - C |  |
| 12 | *Z296Z5H8\_\_\_\_00009330EX93* | *0.163934* | *POOR* | CYCLE - C |  |
| 13 | *Z296Z3EA\_\_\_\_00009330JW2D* | *0.131148* | *POOR* | CYCLE - C |  |
| 14 | *Z296Z3NM\_\_\_\_00009329JENE* | *0.081967* | *POOR* | CYCLE - C |  |
| . . . | *. . .* | *. . .* | *. . .* | . . . |  |
| 30 | *Z296Y6DW\_\_\_\_00009330JUKS* | *0.043547* | *POOR* | CYCLE - C |  |

Figure 5: Disks grouped in three categories and assigned scrubbing frequency

## 4.5 System work load predictor

From System we obtain SAR logs (standard logs for system utilization) and parse them as a dataset for predicting n-step ahead system utilization using time series algorithm (probabilistic fuzzy time series).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Time stamp** (epoch) | **CPU Busy %** | **Disk IO %** | **Memory %** | **\*Network %** |
| 1 | 1546292100 | 33.79873 | 21 | 48 | 26 |
| 2 | 1546292400 | 35.43446 | 24 | 52 | 33 |
|  | . . . | . . . | . . . | . . . | . . . |
| 98 | 1546377000 | 37.65735 | 28 | 57 | 12 |

Figure 6: Snapshot of SAR logs from System

A screenshot of a cell phone

Description automatically generated

Figure 7: Workload Prediction and assigning scrubbing cycles in desired system utilization

A picture containing text

Description automatically generated

Figure 8: Multivariate PWFTS for n-step ahead forecasting

In figure 8, the first graph shows the output of n-step ahead time series forecasting (probabilistic fuzzy time series at 2nd hour (interval) from current time.

**X – axis** represents the *system utilization percentage* (0 – 100%) and **Y – axis** shows the *probability distribution of system resources* at t = 0 (2nd time interval).

We select the system utilization for t = 0, where the probability distribution is maximum. So, in this case, for t = 0 (2nd interval) system utilization is 48% and for t = 2 (4th interval), it is 79%.

## 4.6 Scrubbing schedule window

Now we are aware of the system load in next 2nd and 4th hour (interval decided by the administrator), we can schedule the disk scrubbing based on the below mapping table. (For example, we only allow the scrubbing schedule when system utilization is below 50%)

In the above example, at t = 0 (2nd interval) we see that the system utilization is 48%, we can plan to scrub the disks without big impact on business.

|  |  |
| --- | --- |
| **System Utilization %** | **Scrubbing schedule** |
| 0% - 50% | Yes |
| 51 - 100% | No |

Figure 9: Mapping future System utilization and scrubbing schedule

# Novelty

1. **Smart disk scrubbing frequency strategy based on drive health prediction’s confidence**
   1. The method is algorithm-agnostic (we can use any ML/DL/Statistical learning based on acceptable time complexity)
   2. Wrap Conformal Framework on any ML and get uncertainty quantification metrics for each prediction (confidence of each prediction for particular ‘label’)
   3. This “confidence” of prediction is used to assign **Rank** for Disk Heath (for NORMAL disks)
      1. We are the first to create and implement this metrics
2. **Smart disk scrubbing scheduling strategy based on system workload prediction**
   1. Forecast system load in future and schedule disk scrubbing when system is idle (we filed this method at USPTO and here extend the use case for this disclosure)

# Non-obviousness

|  |  |  |
| --- | --- | --- |
|  | **Prior Art** | **Inventive Step** |
| Method | Statistical, ML, Deep Learning, Markov Decision Process | Conformal Prediction (a learning framework)  Wrapper over any ML/DL/Statistical learning model |
| Motive | Reduces scrub frequency | Optimal (selective) Scrubbing ‘specific’ for Disks which really need scrubbing based on their health |
| Scheduling | System run-time busy  Throttling resources for disk scrubbing | Automated scheduling of disk scrubbing based on n-step ahead system load prediction, an incremental improvement of filed patent: US.16/784721 (7 Feb 2020) |

# Utility

Use Case 1: **Large Scale System**

* Select only small set of disks from large scale system and only scrub them
* No need to run scrubbing over all disks in inventory

Use Case 2: **Data Domain Extended Retention / Archive Tier**

* Opportunistic scrubbing only for disks which really need scrubbing
* Avoid system over utilization

# Advantages over former approach

* Storage reliability improvement: more frequent scrubbing on poor healthy drives. It helps detect and correct sector errors in advance
* Better performance: only schedule disk scrubbing when predicting system work load is idle. Compared with primary storage, this method does show its advantages in backup systems
* Intelligent: the scrubbing frequency and schedule is based on the machine learning prediction, which provides intelligence to the solution