Discerning scrub based on multi-probabilistic approach

# Abstract & Problem:

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 missing 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.

There are lot of storage solutions where they focus on correcting the data, and scrub function is periodically called to scrub a portion of a component's on-disk data structures. These storage solutions bring the scrub functionality to much higher level (application level) and create minute scrubbing threads which are scheduled for a fraction of time. The work performed by the scrub routine is a relatively small portion of work (on the order of 1sec or so). Scrub functions for various components are called in sequence in order to interleave scrubbing. If a single scrub routine takes an inordinate amount of time, then it can starve the other scrub routines.

# Proposed Solution

To overcome the impact from this issue in any Server and minimize the burden of scrub, we can only scrub the disk for which the operation is really required. We identify this using a learning framework based on multi-probabilistic approach to pin-point specific disks for scrubbing. The method is machine learning agnostic and translates to a binary classification task where we proactively forecast (n-days ahead) the state of a disk into two categories:

* Drives having more likely to have issues (we called them ‘**concern**’ drives)
* Drives with no issue (we call them ‘**no-concern**’ drives)

We create a set of above categories and quantify ***“how much” concern/no-concern*** is it across the entire storage pool based on prediction’s confidence. *This metric 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 failures. 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)
  + Prioritize the scrubbing of disks that are likely to have failures over the disks that have low likelihood of failure.
* Schedule scrubbing of selected disks ***when the system is idle*** (system workload prediction)
  + This can be taken care by system load predictor engine

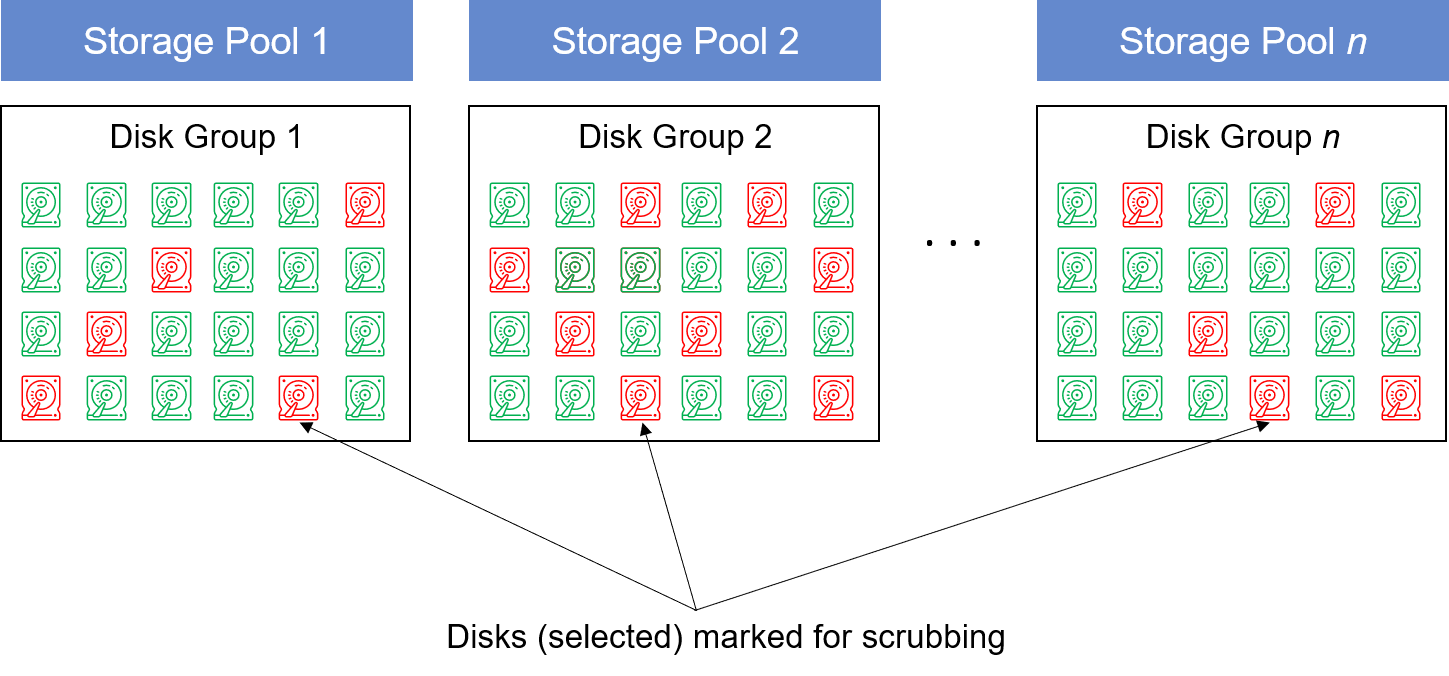


Figure 1: Disks selected for scrubbing based on quantification of “how-bad” the Disks are

A high-level overview of proposed method is shown below. It uses multi-probabilistic approach (VennABER Predictor with KNN or SVM classifier) and sends input to scrubbing scheduler (for selected disks which may potentially have failures) during system idle time.

A screenshot of a cell phone

Description automatically generated

Figure 2: Framework of smart disk scrubbing

|  |  |
| --- | --- |
| Parameter | Description |
| SMART | Self-Monitoring, Analysis and Reporting Technology (specific to disk drive manufacturers) |
| BMS | Background media scan |
| Disk I/O | Collected from system statistics |
| Concern/No-concern Predictor | Identifies the disks which needs scrub operation and quantifies how much scrubbing is require for specific disks |
| I/O latency spike | An unexpected delay in I/O access |
| BER | Bit Error Rate |

Table 1: Terms used in smart disk scrubbing framework

## 3.1 Selection of disk drives

We consider the fact that scrubbing a disk will be an overhead to the system, and so we focus on identifying the disks in the system which are of ‘**concern**’ or may become ‘**concern**’ drives in near future; and we only select those disks for scrubbing (as it will reduce the number of disks meant for scrubbing).

Usually, to identify a concern or no-concern disk there are many state-of-art methods which storage vendors are using. Few have been using statistical, threshold based, machine learning and deep learning methods. These methods provide a binary output to the end user; however, with these existing methods of classification using machine learning, a user won’t be able to quantify “how” concern or no-concern is the disk drive.

Let’s try to understand it this way –

* We have 100 Disks
* “concern”: 5 Disks 🡨 Mark for scrubbing
* “no concern”: 95 Disks
  + Low probability ‘no-concern’ – 20 Disks  🡨Mark for scrubbing

Using our method, we will only scrub 25 (20+5) Disks (frequency of scrubbing can be set by user for these 25 Disks). Existing method will scrub all 100 Disks on certain interval frequency.

We use a multi-probabilistic approach to assign a probabilistic boundary for the prediction. The method uses VennABER predictor along with “any” machine learning classifier (here we use K-Nearest Neighbors; SVM can also be used).

### Mapping error rates:

Most of the advanced HDD gives **Data Unit Read** and **Data Unit Write.** This gives us the information on the read/write unit performed on a single HDD. Information is easily available in SMART pages.

Data Unit Read: 0x746da4 of 512k bytes.

Data Unit Written: 0x2d0 of 512k bytes.

Additional error log gives us the information on error occurring on an HDD.

error\_count : 18446744069579341823

sqid : 65535

cmdid : 0x1

status\_field : 0x2(INVALID\_OPCODE)

parm\_err\_loc : 0xffff

lba : 0xffffffff00040003

nsid : 0xffffffff

vs : 255

If for a read/write numbers more errors are coming so that HDD will be selected for scrub operation.

## Bytes Written + Bytes Read/RW Weight >= 1/BER

This will generate **BER\_metric.** It is an important parameter in learning algorithm.

## 3.2 Experiment results

We use HDD SMART dataset for Seagate disk drive and transform the dataset. 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

From the dataset obtained in step 1, use the relevant parameters:

BER\_metric, *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*



**Output**: Multi-probabilistic prediction for disk drive



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Disks | Actual | Predicted | P(lower) | P(upper) |
| DISK001 | 1 | 1 | 0.875 | 0.916667 |
| DISK002 | 1 | 1 | 0.495646 | 0.697097 |
| DISK003 | 1 | 1 | 0.988568 | 0.88924 |
| DISK004 | 1 | 1 | 0.588568 | 0.78924 |
| DISK005 | 1 | 1 | 0.891342 | 0.997097 |
| DISK006 | 0 | 0 | 0.981132 | 1 |
| DISK007 | 1 | 1 | 0.988568 | 0.99924 |
| DISK008 | 1 | 1 | 0.989621 | 1 |
| DISK009 | 0 | 0 | 0.288568 | 0.58924 |
| DISK010 | 1 | 1 | 0.795646 | 0.897097 |

We see the column “Actual” has two labels 0 (‘**concern**’) and 1 (‘**no-concern**’). “Predicted” column is obtained by using the VennABER predictor and we have two more columns ***P***(lower) and ***P***(upper), they are the probability of prediction and respective boundaries. Smaller the boundary, better is the prediction quality.

Let’s take an example where we want to predict the (‘**concern**’ or ‘**no-concern**’) of 10 disk drives (as shown above) and obtain prediction outcome.

How do we interpret the quality of prediction for selecting the disks for scrubbing?

* We can have Disks classified as ‘**no-concern**’ (label = 1) with lower probability distribution.
  + Here, the user can set a threshold that even if the disk is of no-concern, and the probability of prediction is too low (as compared to threshold), then we will mark this ‘no-concern’ disk for scrubbing (e.g., DISK002). This can be based on system utilization or other factors.
* Mark all “concern” drives for scrubbing

Group the disks in three categories – Best, Medium and Poor (based on Probability value, but no-concern disks selected for scrubbing will always be put in ‘best’ bucket)

* A list of disks marked for scrubbing will be sent to scrubbing scheduler. This scrubbing scheduler will run when the system is idle or when the system is not too busy.
* 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.

A picture containing clock

Description automatically generated

Figure 3: Probabilistic boundary value for prediction (12000 data points)

A close up of a logo

Description automatically generated

Figure 4: Probability distribution for upper limit

A close up of a logo

Description automatically generated

Figure 5: Probability distribution for lower limit

We notice in Figure 4 and 5 that there are few probability values which are very less for upper and lower boundary. If the **“no-concern”** disks are falling in this range (lower values), then they are also marked for scrubbing. The uniqueness is that we also identify the **“no-concern”** disks which will need scrubbing

# Non-obviousness

|  |  |  |
| --- | --- | --- |
|  | **Prior Art** | **Inventive Step** |
| Method | Statistical, ML, Deep Learning, Markov Decision Process | 1. Multi Probabilistic Prediction (an online learning framework) 2. Wrapper over any ML/DL/Statistical learning model |
| Motive | Reduces scrub frequency | 1. Optimal (selective) Scrubbing ‘specific’ for Disks which really need scrubbing based on their concern or no-concern 2. We also identify the no-concern disks which needs scrubbing 3. We assign frequency of scrubbing based on probability distribution |

# Novelty

1. **Smart disk scrubbing frequency strategy based on drive scrubbing eligibility prediction’s probability boundaries**
   1. The method is algorithm-agnostic (we can use any ML/DL/Statistical learning based on acceptable time complexity)
   2. Wrap multi-probabilistic prediction framework on any ML and get probabilistic boundary for each prediction (confidence of each prediction for particular ‘label’)
   3. Translate confidence of prediction to quantify” concern” and ”**no-concern**” category of the drive
      1. We also identify the disks classified as “no-concern” and quantify ‘how much’ it belongs to ‘no-concern’. According to the assigned score we decide if scrubbing is required or not.
   4. Use disk I/O latency spike to identify ‘concern’ drives
      1. One of the best indicators of imminent media failure is a spike in I/O latency of a drive.
      2. If for some reason, an I/O takes longer than usual to complete on a drive, it is usually an indication that drive may be doing error recovery internally to service the I/O and hence is an indication of an imminent media failure (not the full drive usually, just a few LBAs).
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 extends the use case for this disclosure)

# 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: **Extended Retention and 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 drives. It helps detect and correct sector errors in advance
* Better performance: only schedule disk scrubbing when predicting system workload 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