Rule DAS160: MAJOR CAUSE OF I/O DELAY WAS MISSED CACHE READ HITS

Finding: CPExpert determined that missed cache read hits was the major cause of delay in DASD response for the device.

Impact: This finding may have a MEDIUM IMPACT or HIGH IMPACT on the performance of the device and on the performance of other volumes attached to the cache controller.

Logic flow: The following rules cause this rule to be invoked:

DAS100: Volume with the worst overall performance

Discussion: Cache controllers offer the potential of **significant** reduction in I/O response times. Cache controllers operate by maintaining some amount of data from the volumes being cached in high-speed memory. The following summarizes the cache controller operation and performance considerations:

- When an I/O read operation is executed for data from a cached volume, the controller determines whether the required data resides in the cache.

  - If the data **does** reside in the cache, then the data is immediately provided. This situation is titled a "read hit" operation. A read hit eliminates the requirement for interaction with the volume (seek operation is potentially eliminated and rotational position sensing is eliminated in all cases). The cost of the read operation is the small amount of controller time involved in determining whether the data resides in the cache and the data transfer time. The I/O read operation savings resulting from a read hit can be 90% or more of the normal read I/O time!

  - If the data does **not** reside in the cache, then the data is retrieved from the volume in the normal read process (i.e., potentially a seek, a set sector, a search, and data transfer). This situation is titled a "read miss" operation. The cost of this read operation is the normal read I/O cost, plus the controller time determining whether the data resided in cache.

    Additionally, upon a read miss, the cache controller will transfer all data from one or more tracks from the volume into cache (the number of tracks transferred depends upon the controller model).

This process is called "staging" of the data, and is done with the idea that if data is read from a file, it is likely that the next sequential records in the file will be read soon. The controller stages these records into cache so that they will be in the cache when a subsequent read operation occurs.

Depending upon the controller model, the controller may be busy during the staging of data. Regardless of the controller model, the device is busy during the staging operation.

- When an I/O write operation is executed for data from a cached volume, the controller determines whether the required data resides in the cache just is it does for a read I/O operation.

  - If the data **does** reside in the cache, then the data is transferred to the cache and the data is transferred from the cache to the volume. This situation is titled a "write hit" operation. A write hit requires transfer of data to the cache and transfer of data to the device in the normal write process (i.e., potentially a seek, a set sector, a search, and data transfer). The cost of this write operation is the channel time to transfer the data to cache, plus the controller time determining whether the data resided in cache.

  - If the data does **not** reside in the cache, then the data is written to the volume in the normal write process (i.e., potentially a seek, a set sector, a search, and data transfer). This situation is titled a "write miss" operation. The cost of this write operation is the normal write I/O cost, plus the controller time determining whether the data resided in cache.

From the above brief discussion, it is clear that the savings from cache controller occurs when the read I/O operations find the required data in the cache. Performance is significantly improved in this situation. Performance suffers because of cache in all other situations. Consequently, it highly desireable to cache volumes which mostly have read hit situations.

In the absence of direct measurements of cache hit/miss operations[1], CPExpert makes some conclusions about the suitability of cache hit/miss based upon the average DISC time of I/O operations to each volume. (Recall that the DISC time is mostly time spent seeking, normal rotational latency, and missed RPS connect.)

- If cache is being used effectively, most I/O operations will be satisfied

[1]Cache hit and miss information is available, for example, from IBM's RMF Cache Reporter.

by read hits and the average DISC time will be quite small. This is because most operations will require no access to the actual device, but will be satisfied from cache. Consequently, if the average DISC time is small, it is clear that the cache is effective.

- If cache is not being used effectively, relatively few I/O operations will be satisfied by read hits, and a relatively large number of I/O operations will require access to the device. These I/O operations will require the normal latency (an average of one-half the rotational time of the device), plus may require seeking to position the arm. Consequently, if the average DISC time is relatively large for cached volumes, one interpretation is that cache is not being effective for the volume.

  CPExpert estimates the effect of missed cache hits. If the DISC time is less than the average latency, CPExpert attributes the DISC time to missed cache hits. If the DISC time is greater than the average latency, CPExpert attributes the average latency to missed cache hits (the remainder of the DISC time is attributed to missed RPS reconnect and seek). If the estimated time attributed to missed cache hits accounts for a majority of the I/O delay time, CPExpert concludes that the major problem with the volume is cache miss operations.

  This analysis approach works well in many situations. However, there is a potential problem with this approach since it is based upon averages. Suppose that a relatively few I/O operations resulted in seeks over a large number of cylinders. The seek time (and thus the DISC time) would be large for these relatively few I/O operations, and these large times would distort the average DISC for all operations. It is possible that most I/O operations to the volume were satisfied by read hits, but the average DISC time would be large because of the long seek operations.

Suggestion: There are several alternatives you should consider:

- If the applications and data contained on the volume are not critical, you should consider moving the volume to a non-cached controller, since it does not appear that the volume significantly benefits from being cached.

- Alternatively, there may be non-critical volumes on the cache controller with data access patterns which cause the cache to be full with their data. These volumes may prevent the cache from adequately responding to the I/O requirements of the volume identified by this rule. You may wish to identify these volumes and move them to a non-cached controller.

- Alternatively, applications accessing the volume may be generating seeks which require movement of the arm over a large number of cylinders. If this is the case, then the actions under Rule DAS110 should be reviewed.

- Alternatively, you may wish to consider acquiring another cached controller if all volumes currently being cached are critical and the I/O requests to the identified volume are **not** being delayed because of long seeks.

