Technical Report

Introduction to Predictive Cache Statistics

Paul Updike, NetApp
September 2011 | TR-3801

TABLE OF CONTENTS

1 INTRODUCTION TO PREDICTIVE CACHE STATISTICS

NetApp® Predictive Cache Statistics (PCS) offer the ability to emulate large read cache sizes and to measure their effect on system performance. In this way PCS provides a means to approximate the performance gains of adding one or more Flash Cache modules to a system.

PCS is configured in the same manner as Flash Cache and shares the same options for configuration. This guide describes its configuration and use.

2 VERSIONS OF DATA ONTAP THAT INCLUDE PREDICTIVE CACHE STATISTICS

Starting in Data ONTAP® 7.3.2, PCS is capable of simulating Flash Cache size caches in the hundreds of gigabytes. It is available on FAS3000 and FAS6000 systems with at least 4GB of memory per controller. PCS is also available in Data ONTAP Cluster-Mode.

3 THE PCS ANALYSIS PROCESS

You use PCS to determine the size of the read working set in your workload; that is, the amount of data that is being read repeatedly and is flowing through the system. Once you understand this, you can understand the impact of having a cache that would handle a portion of that working set and how your performance would improve by adding a cache of this size.

The process is then to set up and analyze Predictive Cache Statistics centers on the workload and the working set and to capture data about it. The high-level steps are relatively simple:

1. Enable PCS.
2. Allow the representative workload to run.
3. Collect data throughout the process.

This process initially provides information on the “cold” state of the emulated cache. That is, no data is in the cache at the start of the test, and the cache is filled as the workload runs. The best time to observe the emulated cache is once it is filled, or “warmed”, as this will be the point when it enters a steady state. Filling the emulated cache can take a considerable amount of time and depends greatly on the workload.

4 USING PREDICTIVE CACHE STATISTICS

PCS uses the same Data ONTAP counter manager architecture to maintain performance data points. The name of the counter manager object for PCS is the same as that for Flash Cache:

```
ext_cache_obj
```

You can enable PCS in the same way that you enable the actual hardware:

```
>options flexscale.enable on
```

When there is no actual hardware in the system, this command enables PCS. To verify that you now have PCS enabled, you can check the value of the option; it should display `pcs`:

```
>options flexscale.enable

flexscale.enable              pcs
```

To view information in the counter object, you can use the command:

```
>stats show ext_cache_obj
```

The output for this command:

```
ext_cache_obj:ec0:type:SPCS
ext_cache_obj:ec0:blocks:268435456
ext_cache_obj:ec0:size:1024
ext_cache_obj:ec0:usage:0%
ext_cache_obj:ec0:accesses:0
ext_cache_obj:ec0:disk_reads_replaced:0/s
ext_cache_obj:ec0:hit:0/s
ext_cache_obj:ec0:hit_normal_lev0:0/s
ext_cache_obj:ec0:hit_metadata_file:0/s
ext_cache_obj:ec0:hit_directory:0/s
ext_cache_obj:ec0:hit_indirect:0/s
ext_cache_obj:ec0:total_metadata_hits:0/s
ext_cache_obj:ec0:miss:0/s
ext_cache_obj:ec0:miss_metadata_file:0/s
ext_cache_obj:ec0:miss_directory:0/s
ext_cache_obj:ec0:miss_indirect:0/s
ext_cache_obj:ec0:hit_percent:0%
ext_cache_obj:ec0:inserts:0/s
ext_cache_obj:ec0:inserts_normal_lev0:0/s
ext_cache_obj:ec0:inserts_metadata_file:0/s
ext_cache_obj:ec0:inserts_directory:0/s
ext_cache_obj:ec0:inserts_indirect:0/s
ext_cache_obj:ec0:evicts:0/s
ext_cache_obj:ec0:evicts_ref:0/s
ext_cache_obj:ec0:readio_solitary:0/s
ext_cache_obj:ec0:readio_chains:0/s
ext_cache_obj:ec0:readio_blocks:0/s
ext_cache_obj:ec0:readio_max_in_flight:0
ext_cache_obj:ec0:readio_avg_chainlength:0
ext_cache_obj:ec0:readio_avg_latency:0ms
ext_cache_obj:ec0:writeio_solitary:0/s
ext_cache_obj:ec0:writeio_chains:0/s
ext_cache_obj:ec0:writeio_blocks:0/s
```

```
ext_cache_obj:ec0:writeio_max_in_flight:0
ext_cache_obj:ec0:writeio_avg_chainlength:0
ext_cache_obj:ec0:writeio_avg_latency:0ms
ext_cache_obj:ec0:invalidates:0/s
```

As with an actual Flash Cache module,this gives you only a point-in-time view of the data in the counters. For this reason, NetApp recommends using the iterative form of the command with a preset for the PCS counters. This command gives you an output every 5 seconds of per-second data rates for the most critical counters:

```
>stats show -p flexscale-access
```

The output has the following format:

```
Cache                                                  Reads       Writes      Disk Reads
Usage    Hit   Meta   Miss Hit Evict Inval Insert Chain Blocks Chain Blocks  Replaced
     %     /s     /s     /s   %    /s    /s     /s    /s     /s    /s     /s        /s
     0      0      0      0   0     0     0      0     0      0     0      0         0
```

NetApp recommends issuing this command through an SSH connection and logging the output throughout the observation period because you want to capture and observe the peak performance of the system and the cache. This output can also be easily imported into spreadsheet software, graphed, and so on. A second SSH can be used to capture the output of the `sysstat` command. The combination of these two data points, PCS-specific and system data, lets you know what is happening on the system and how the emulated cache that PCS is presenting would handle the workload.

Here are the definitions for the PCS counters:

- **Cache Usage**: How much data is currently stored in the module(s)
- **Hit**: The 4kB disk block cache hit per second
- **Meta**: The 4kB metadata disk block cache hit per second
- **Miss**: The 4kB disk block cache missed per second
- **Hit %**: The percentage of total hit/miss
- **Evict**: The 4kB disk blocks evicted from the cache per second
- **Inval**: The 4kB disk blocks invalidated from the cache per second
- **Insert**: The 4kB disk blocks inserted into the cache per second
- **Reads Chain**: The number of read I/O chains per second
- **Reads Blocks**: The number of 4kB disk blocks read per second
- **Writes Chain**: The number of write I/O chains per second
- **Writes Blocks**: The number of 4kB disk blocks written per second
- **Disk Reads Replaced**: The number of reads that would have gone to disk that were replaced by the cache per second

In this output, the main items to note are Usage, Hit rate, Hit %, and the number of Disk Reads Replaced, because the actual Flash Cache technology can increase the operations per second of the system and increase the hit rate of a Flash Cache as compared to the hit rate for PCS.

Because of this effect, when analyzing PCS data:

1. Usage and Hit % are the first indicators of the benefit of adding a Flash Cache to the system.
2. Hit rate and Disk Reads Replaced can then be analyzed to understand the *minimum* effect that adding Flash Cache would have.

5 CHANGING THE SIZE OF THE PCS CACHE BEING EMULATED

The `flexscale.pcs_size` option displays and sets the size of the PCS emulated cache. By default, this is set to an assumed number of Flash Cache modules that would fit in the system. For example, the default setting for a NetApp FAS3170 is 1024GB. To change the size, you use the `options` command with the new figure. For example, to understand the impact of a single 512GB module in a system, you would change the size in the following way:

```
>options flexscale.pcs_size 512GB
```

To change it back, you would use:

```
>options flexscale.pcs_size 1024GB
```

Note: The simplest way to understand the number of Flash Cache modules required to accelerate the workload is to directly model them by using this option and compare results.

6 CHANGING THE PCS CACHING POLICY

There are three modes of operation available when running PCS. These modes are identical to those available to Flash Cache. To determine which caching policy can best assist the workload, you can set the mode, run your test, and capture data as described earlier, repeating with each mode as necessary.

- **Default mode:**The simplest performance improvement is by caching data as it is accessed from disk. On successive accesses, instead of going to disk, which is higher latency, the data is served from the memory onboard the Flash Cache. This is the default mode. The options for this mode look like this:

  ```
  flexscale.enable              on
  flexscale.lopri_blocks        off
  flexscale.normal_data_blocks  on
  ```

- **Metadata mode:**The second way to improve performance with a Flash Cache is in workloads with a heavy metadata overhead. There are some workloads that require accesses to metadata before application data can be served. In these workloads, placing the metadata in the cache allows low-latency access to the metadata and provides higher-speed access to the application data. Because of the much larger size of the Flash Cache, this mode is more applicable to the original PAM I than to Flash Cache.

  ```
  flexscale.enable              on
  flexscale.lopri_blocks        off
  flexscale.normal_data_blocks  off
  ```

- **Low-priority blocks mode:**A third way to improve performance is through capturing application data that normally would have been forced to disk or not cached at all. These are called low-priority buffers and they exist in a couple of forms:
  - **Write data (flushq blocks):**Normally, writes are buffered in RAM and logged to NVRAM; once committed to disk they are flushed from NVRAM and retain a lower priority in RAM to avoid overrunning the system memory. In other words, recently written data is the first to be ejected. In some workloads, recently written data may be immediately accessed after being written. For these workloads, Flash Cache improves performance by caching recently written blocks in the module rather than flushing them to disk and forcing a disk access for the next read.
  - **Long sequential read blocks:**Long sequential reads can overrun the system memory by overflowing it with a great amount of data that will be accessed only once. By default, Data ONTAP doesnot keep this data in preference to holding data that is more likely to bereused. The large amount of memory space provided by Flash Cache allows sequential reads to potentially be stored without negatively affecting other cached data. If these blocks are accessed again, they will realizethe performance benefit of the module rather than going to disk.

```
flexscale.enable              on
flexscale.lopri_blocks        on
flexscale.normal_data_blocks  on
```

7 DETERMINING THE WORKING SET SIZE OF A WORKLOAD

The Predictive Cache Statistics functionality built into Data ONTAP can help you determine the working set size of a workload; whether Flash Cache will help accelerate the workload; and how many modules will be needed. To determine the working set size by using this option is a simple process:

1. Set the `flexscale.pcs_size` option to your best estimate of the working set.
2. Allow the workload to run completely through. If it is an aggregated workload of many different smaller workloads, choose a representative time period and allow it to run through that period.
3. View the amount of cache that is filled by using the process in section 4, “Using Predictive Cache Statistics.”
4. Calculate the working set based on the % in the cache usage multiplied by the size of the emulated PCS cache (`flexscale.pcs_size`).
5. This should give you the read working set size of the workload, with one caveat. Data ONTAP tries to aggressively store data in the Flash Cache module until it is 70% full. If you note that the cache is at 70% consistently in cache usage, try to size it down and try again. This effect may be filling the cache with extra data (writes, for instance) that isn’t part of the read working set.

8 USING PCS IN DATA ONTAP OPERATING IN CLUSTER-MODE

Predictive Cache Statistics is also available in Data ONTAP systems operating in Cluster-Mode. Caching statistics are gathered at the node-level and must be observed on the node-shell. Once on the node-shell of the desired node, PCS can be used as described in the previous sections.

Enter node shell:

```
node1::> node run -node node1
```

9 SUMMARY

With Predictive Cache Statistics you can investigate the possibility of adding a Flash Cache module to your controller and determine the amount of data that could be served from it. A high hit rate can potentially increase the performance of a system significantly. Because PCS is included in Data ONTAP 7.3.2 and later, you can evaluate running systems with no additional hardware or software required beyond the basic OS.

The process, as described in this paper, is simple: enable PCS, run your workload, and collect PCS and system information while the test occurs. To evaluate the results, review the percentage of disk I/O that could be saved and project the performance improvement.

With this knowledge, you can determine whether a Flash Cache module is applicable to your system and workload.

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

www.netapp.com