IBM Z Common Data Provider
&
IBM Z Operations Analytics
Best Practices

Version: September 06, 2019

IBM Z Common Data Provider

Test Environment Information:

There are several parameters that may be tuned for ZCDP by the user on the host system. This will break down the parameters by the individual procedures, and suggest values that, in our test environment, have shown the best CPU performance. The values presented below are for a scenario of 1 LPAR streaming data to 1 subscriber. More information regarding scaling with multiple instances of ZCDP and multiple subscribers are under investigation and will be available at a later date.

Test Environment Details:

Z14 model M02 running z/OS v2.3 with a single 3906.7 processor active and 16G Storage

Adjustable Parameters in the SDE:

- IBM_SDE_INTERVAL
    - Default: 1 MINUTES
    - Recommended: 1 MINUTES or 30 SECONDS
        - 1 MINUTES parameter has usually shown the best CPU performance. There are cases with very high throughput, however, when this would result in buffering too much data at once. If you find that you are filling the buffer in this interval, a SDE interval of 30 SECONDS can alleviate this issue without majorly impacting CPU use.
        - *NOTE: If your SDE collection starts as a system-defined SMF interval is waiting, it is possible that you will not see data for those SMF types until the next system interval is reached. Check your SMFPRMxx members for more information on your system SMF interval – as some records aren't produced off-schedule or by individual event.*
- IBM_RESOURCE
    - Recommended: INMEM
        - Streaming ZCDP using an in-memory resource has some performance improvements over traditional Logstream. If your implementation of ZCDP uses Logstream and you have the ability to use the in-memory resource, it is suggested to do so. For users implementing the MANx datasets via user exits, continue to define IBM_RESOURCE = 'EXIT'.
- ZIIPOFFLOAD
    - Default: NO
        - The zIIP offloading function is disabled by default. If you want to offload a significant portion of the SDE workload, you can change ZIIPOFFLOAD=YES to activate the zIIP offloading function.

- If ZIIPOFFLOAD is not included in PARM of the EXEC statement, you can add it in the PARM parameter as in the following sample:

  ```
  //HBOSMFCL EXEC PGM=HBOPDE,REGION=0M,TIME=1440,
  //         PARM='SHOWINPUT=NO,ZIIPOFFLOAD=YES'
  ```

- By activating the zIIP offloading function, a significant portion of general CPU workload can be offloaded to zIIP, thereby reducing the general CPU utilization.

Adjustable Parameters in the Data Streamer:

- JCL changes:
    - Default/Recommended: trace =n
        - Turning trace on is, as in most cases, not recommended unless needed for diagnosis by IBM support. Turning on trace can heavily impact CPU use.
- Changes in the startup.sh script:
    - Heap size
        - Default: -Xms4g -Xmx4g
            - In most cases, a heap size of 4G should be sufficient for ZCDP. If you are running out of memory, you may be able to resolve the issue by increasing the heap size. In cases such as this, it is good to profile your heap use with java monitoring tools such as VisualVM. If you find that you are never coming close to filling your heap, it is also possible to reduce the heap size to free up storage. In any case, the Xms and Xmx should be set to the same size. Having a Xms that is smaller than the Xmx negatively impacts performance, as the JVM then has to calculate and adjust the heap continuously.

WLM Recommendations:

Considering the nature of the active address spaces for ZCDP, any of the three main process addresses going down would cause the entire process to fail. Because of this, our recommendation would be to set the SDE, Log Forwarder, and Data Streamer as high-priority. Since they are started tasks, either SYSSTC or other high-priority classes are appropriate. The non-z/OSMF configuration address spaces are only used when necessary to make configuration changes and are low-priority. More specific information can be found below.

Default Started Task Name	Description
Group 1: Collector Tasks *This group of tasks needs high-priority since they collect data. SYSSTC is suitable for this group.*	
HBOSMF	ZCDP: System Data Engine
GLAPROC	z/OS Log Forwarder
Group 2: Data Streaming Tasks *This group of tasks needs high-priority since they are started tasks that format and send data to subscribers. SYSSTC or other high-priority classes are suitable for this group.*	
HBODSPRO	ZCDP: Data Streamer
Group 3: Configuration Tasks *This group of tasks can be low-priority since they are only started as needed and run for a short time.*	
HBOCFGA	Configuration Tool Angel Server
HBOCFGT	Configuration Tool Server

Subscribers:

Currently, sending data to Logstash and Splunk subscribers is supported, as well as generic HTTP subscribers. Here are a few guidelines for each:

Logstash Server (General)

- Always set the JVM heap minimum (Xms) and maximum (Xmx) to the same value. Generally, starting with a value of 4GB is sufficient, and profiling the JVM heap utilization can help lower the value if this exceeds your needs.
- If events are backing up, slowly scale up the number of pipeline workers to make more CPU threads available
  - Pipeline batch size can also be increased for more efficient and faster processing but increases memory overhead. If your heap is well tuned, consider increasing the heap size before modifying this setting.
- Disable swapping to improve garbage collection times and increase node stability
  - This can be done temporarily with ***sudo swapoff -a*** or permanently by editing the ***/etc/fstab*** file and commenting out all lines that contain the word “swap”
- The file descriptor limit (ulimit -n) should be set to 65536 or higher.
- The user thread limit (ulimit -u) should be set to at least 4096

IOA-LA - Logstash

- IOA-LA has been tested to ingest up to 2TB/day

Elastic - Logstash

- We have seen ingestion rates of up to 480GB/day with one Elasticsearch node and one Logstash instance. If additional ingestion volume is needed, Elasticsearch should be scaled horizontally to include more nodes.

Subscribers (continued):

Splunk – Data Receiver

- Splunk indexers on reference hardware can generally ingest up to 300GB/day. Further data volumes require multiple indexers and search heads. See Splunk's recommendations on scaling and capacity planning for more information.
  - You should plan your ZCDP configuration to send to multiple Data Receivers, each on their own Splunk indexer if you will exceed 300GB/day.
- System CPU utilization while ingesting 500GB/day through a Data Receiver has shown between 4%-8% on Splunk reference hardware. Due to Splunk's behavior and CPU utilization for ingestion and searches, individual Data Receivers on a Splunk indexer should not negatively impact Splunk performance.

Splunk – HTTP Event Collector (HEC)

- Splunk HEC ingestion is subject to the same throughput limits as described above for Splunk Data Receiver ingestion. This is extremely important to consider when sizing your workload for ingestion, as Key-value formatting increases output record size more than CSV formatting does. Thus, for equivalent workloads, you will not be able to stream as many Key-value records as you can CSV records in a day.
- The magnitude of Key-value inflation depends on the record type, as different types have different numbers of data entries to be paired with keys. Lab testing has shown several record types inflating anywhere from 2-8 times in Key-value format compared to CSV format.
- When configuring the subscriber in your policy, always select the default option of 12 threads unless instructed otherwise by support.
  - You should only need to change this when your workload is large enough to form a bottleneck in the Data Streamer's sender queue. If this happens, the growing bottleneck will eventually cause your Data Streamer's heap to exceed capacity. (For that reason, increasing your heap size will not be a viable solution in this case, as doing so will merely prolong the heap's inevitable saturation.)
  - Our testing has shown very minimal CPU impact of thread allocation alone. Thus, we strongly advise against deliberately decreasing your thread count to achieve performance

    improvement, as this will risk a Data Streamer bottleneck for sufficiently large workload volumes.

- Compared to Splunk Data Receiver ingestion, Splunk HEC ingestion yields level CPU utilization for the SDE and LogForwarder but higher CPU utilization for the Data Streamer. This is explained by the processing overhead associated with the Data Streamer formatting Key-value data, in addition to the resulting inflation of that data to be streamed.
- Using the Data Receiver is our recommended best practice for Splunk ingestion. However, you may consider using HEC if:
    - You are not using IZOA.
    - You deem acceptable the anticipated CPU increase and data inflation in choosing HEC over Data Receiver ingestion.
    - There are limitations on your ability to install or update the Splunk buffered ingestion app.

Measuring ZCDP CPU Utilization:

CPU performance is critical for our products, so knowing how to accurately measure this important. Here are some considerations for measuring the CPU utilization of ZCDP in your environment:

- **Streaming**
  - ZCDP has a startup overhead that is negligible in a long running process, but noticeable if measuring for only a short amount of time
  - If measuring the Data Streamer or Log Forwarder, wait for Java optimization to take place
    - In our testing, this usually stabilized after approximately 30-45 minutes of processing workloads

  - Standard test process:
    1. Start monitor
    2. Start ZCDP addresses
    3. Start workload(s)
    4. Wait approximately 90 minutes
    5. Review results
       - Only use the last 30 minutes of the test for CPU utilization
       - The first hour can be used to understand the JIT optimization, changes over time, and identify other issues
       - If 90 minutes is too long of a test, you can start workloads for approximately 5 minutes, turn them off for another 5 minutes, and restart the workloads. Repeat this process as necessary over a prolonged period of time without restarting any ZCDP address spaces. Java optimization will still occur in this scenario.

- **zIIP Measurement**
  - You can measure the zIIP time of ZCDP from RMF Monitor III.
  - For convenience, you also can check the zIIP time from Display Active Users panel (DA) for Data Streamer, and from Enclaves panel (ENC) for System Data Engine.

- **Batch Loading**
  - **Use Record Format VB, Logical Record Length 32756**
    - **Sample**: *PATHDISP=(KEEP),RECFM=VB,LRECL=32756,FILEDATA=RECORD*
    - This has shown over 70% reduction in CPU time over RECFM=V
    - Batch load sample JCL can be found below

Sample JCL:

Creating a Summary Activity Report:

This JCL can be used to create a Summary Activity Report. To help with sizing your environment, IBM will ask you to provide the output of the Summary Activity Report during peak traffic. The job output from this procedure will give details to how many SMF records of each type have been created since the current SMFPRMxx member was activated, as well as the average length for each record type.

For MANx datasets

*Edit Line 3 with your MANx dataset location, and line 5 with available volume to dump the SMF data to. *NOTE – this will not clear your MANx dataset.**

```
//SMFDUMP1 EXEC PGM=IFASMFDP
//DUMPIN   DD DSN=SYS1.MAN1,DISP=SHR
//DUMPOUT  DD DSN=HLQ.MLQ.MAN1.DUMP(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=3390,VOL=SER=(VOLID1,VOLID2),
//            SPACE=(CYL,(300,150),RLSE),
//            DCB=(RECFM=VBS,LRECL=80,BLKSIZE=6160)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
```

For Logstream

Edit lines 2 and 4 with the instructions from above.

```
//SMFDMPLS EXEC PGM=IFASMFDL,REGION=0M
//DUMPOUT  DD DSN=HLQ.MLQ.DUMP(+1),
//            DISP=(,CATLG,DELETE),
//            UNIT=3390,VOL=SER=(VOLID1,VOLID2),
//            SPACE=(CYL,(1,1),RLSE),
//            DCB=(RECFM=VBS,LRECL=32760,BLKSIZE=0)
//SYSPRINT DD SYSOUT=*
//SYSIN      DD *
 LSNAME(IFASMF.SYSNAME.LSNAME,OPTIONS(DUMP))
 OUTDD(DUMPOUT,TYPE(0:255),START(0000),END(2400))
 DATE(YYYYDDD,YYYYDDD)
 SID(SYS)
/*
```

Sample JCL Continued:

Batch Loading SMF Datasets:

If you need to batch load SMF datasets through ZCDP, use the sample provided below. To run, simply have the Data Streamer running and submit the procedure defined by the following JCL:

```
//BATCHCDP JOB 'ACCOUNTING INFORMATION','GET SMF DS',
// CLASS=A,MSGCLASS=A,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*
//* POINT TO ZCDP LEVEL IN LINE 8
//* POINT TO SDE CONFIG IN LINE 12
//*
//HBOSMFCL EXEC PGM=HBOPDE,REGION=0M,PARM=('SHOWINPUT=NO')
//STEPLIB  DD DISP=SHR,DSN=HLQ.MLQ.SHBOLOAD
//HBOIN    DD *,DLM='/+'
SET IBM_UPDATE_TARGET = 'PORT 51401';
SET IBM_FILE_FORMAT = 'CSV';
//         DD  PATH='/u/CDP/config.sde',
//         PATHDISP=(KEEP),RECFM=VB,LRECL=32756,FILEDATA=RECORD
//         DD *,DLM='/+'
COLLECT SMF
/+
//*
//* DEFINE SMF DATASET IN LINE 22
//*
//HBOLOG   DD DISP=SHR,DSN=HLQ.MLQ.SMFDSN
//HBOOUT   DD SYSOUT=*
//HBODUMP  DD SYSOUT=*
```

IBM Support Must-Gather Documentation:

When reaching out to IBM Support or Performance for questions or issues with ZCDP, the following information should be provided to insure quick response times, and less requests for information.

- Jobs for started procs:
    - Data Streamer, SDE and/or Log Forwarder
- Job output:
    - Data Streamer, SDE and/or Log Forwarder
- Configuration Files:
    - Screenshot of z/OSMF configuration
    - policy file (Data Streamer)
    - config.properties file (Log Forwarder)
    - SDE file (SDE)
- Log outputs/dumps
- SMF Summary Activity Report (see above sample)
- CPC SI output (Command: D M=CPU)
- Environment Details for capacity planning inquiries
    - LPARs
        - SMF Summary Activity Report for each
        - Data intended for collection (specific SMF types, record types, SYSLOG, etc)

Performance Assistance Documentation

If seeking help for performance issues, additional documentation may be required to help in diagnosing issues. RMF CPU, Transaction, and Memory reports targeting ZCDP address spaces help us pinpoint problems and hasten investigation. Please provide RMF reports when available for performance inquiries. When running RMF, make sure oyu also run RMF Monitor II at an interval of 5 minutes (or less), with both ARD and ASD reports enabled.

IBM Z Operations Analytics

Guidelines:

- IBM Z Operations Analytics dashboards are designed to return information from Summary SMF data. Due to search return scaling with endpoints such as Splunk and Kibana, trying to retrieve multiple days' worth of data or more could result in long load times for dashboards. The load time scales linearly with the amount of matching events in an index within the given search timeframe.
  - Splunk example:
    - Dashboard load time, returning 200,000 matches – 3.5 seconds
    - Dashboard load time, returning 2,000,000 matches – 35 seconds
  - Narrowing your search timeframe can greatly improve dashboard load times. It is best to search around a smaller timeframe with a known problem than to search multiple days' data.
- It is recommended to only search one day's worth of ingested data in any dashboard. For a customer ingesting roughly 150,000 summary records per day, searching across this for one LPAR's data should take approximately 2 seconds. Searching the data from all LPARs in our reference environment with approximately 550,000 summary records ingested in one day would take around 10 seconds.
- To determine how many records you are generating daily, you can use the SMF Summary Activity Report. Details on how to generate this report are given above under '**Creating a Summary Activity Report**'
  - You can also use a Summary Activity Report to determine volume of data in bytes for a given period of time
    - Divide the total range of time you want to calculate for in seconds by the total seconds for the Summary Activity Report period of time to get periods/range of time.
    - Multiply the number of written records of a particular SMF type by the above number of periods (round down to the nearest integer as necessary).
    - Multiply the total records by the average record length. This results in total bytes/time range.

Example: 1

Calculate GB/day of SMF30 records

Report Period Time:
19min 28sec
(**1,140** sec)

Desired Time Range:
1 day
(**86,400** sec)

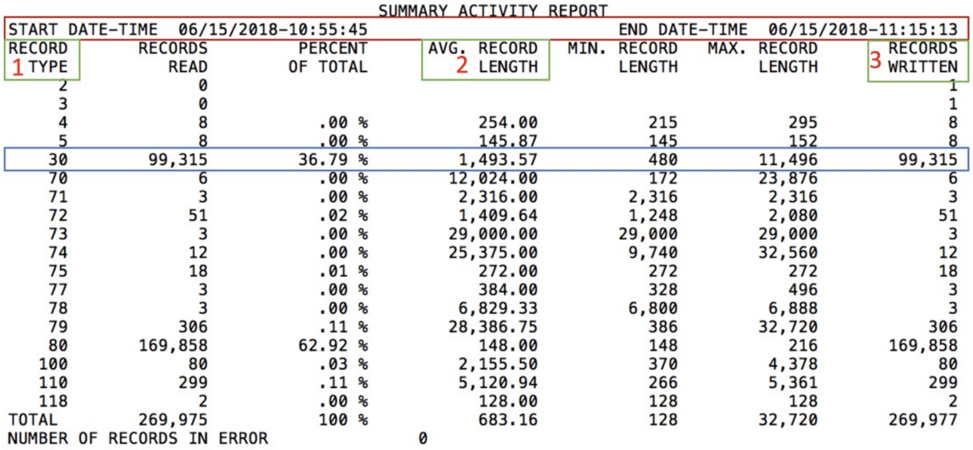
86,400/1,140 =
75.79 periods

3 Records:
99,315

Records per day:
99,315 * 75.79 =
7,527,083 rec/day

2 Avg. Record Length:
1,493.57 bytes

Bytes per Day:
7,527,083 * 1,493.57 =
11,242,225,356.31 bytes/day
or **11.24 GB/day**



```
                                     SUMMARY ACTIVITY REPORT
START DATE-TIME  06/15/2018-10:55:45                         END DATE-TIME  06/15/2018-11:15:13
RECORD       RECORDS         PERCENT      AVG. RECORD   MIN. RECORD   MAX. RECORD       RECORDS
  TYPE          READ        OF TOTAL           LENGTH        LENGTH        LENGTH       WRITTEN
     2             0                                                                          1
     3             0                                                                          1
     4             8           .00 %           254.00           215           295             8
     5             8           .00 %           145.87           145           152             8
    30        99,315         36.79 %         1,493.57           480        11,496        99,315
    70             6           .00 %        12,024.00           172        23,876             6
    71             3           .00 %         2,316.00         2,316         2,316             3
    72            51           .02 %         1,409.64         1,248         2,080            51
    73             3           .00 %        29,000.00        29,000        29,000             3
    74            12           .00 %        25,375.00         9,740        32,560            12
    75            18           .01 %           272.00           272           272            18
    77             3           .00 %           384.00           328           496             3
    78             3           .00 %         6,829.33         6,800         6,888             3
    79           306           .11 %        28,386.75           386        32,720           306
    80       169,858         62.92 %           148.00           148           216       169,858
   100            80           .03 %         2,155.50           370         4,378            80
   110           299           .11 %         5,120.94           266         5,361           299
   118             2           .00 %           128.00           128           128             2
TOTAL        269,975           100 %           683.16           128        32,720       269,977
NUMBER OF RECORDS IN ERROR               0
```