SMF Recording Options to Improve Your Performance Analysis

Scott Chapman

Enterprise Performance Strategies, Inc.

Scott.chapman@EPStrategies.com

Contact, Copyright, and Trademarks

Questions?

Send email to performance.questions@EPStrategies.com, or visit our website at https://www.epstrategies.com or http://www.pivotor.com.

Trademarks:
Enterprise Performance Strategies, Inc. presentation materials contain trademarks and registered trademarks of several companies.

The following are trademarks of Enterprise Performance Strategies, Inc.: **Health Check®, Reductions®, Pivotor®**

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries: IBM®, z/OS®, zSeries®, WebSphere®, CICS®, DB2®, S390®, WebSphere Application Server®, and many others.

Other trademarks and registered trademarks may exist in this presentation

SMF Recommendations Agenda

- Background

- Intervals

- Compression

- Record Types

SMF Background / Why do we care?

- SMF = System Management Facility (note: not "Measurement")
- Records much data about the system that's very important
  - Certain data required to be submitted to IBM for workload license pricing
  - Security records and certain activity records very important/interesting to the security auditors
  - Of course all the performance measurements are critical for managing the performance and capacity of your system
- Managing your SMF data is a critical part of systems management
- SMF data can be very voluminous
  - In particular, transactional records (101, 110, 116, 120) can be very large
  - Note difference between record counts and byte counts!
  - Key point: be sure to record useful information and what is too large is probably different today than it was 20 years ago
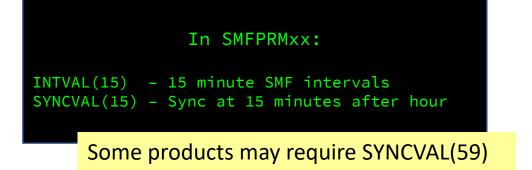
SMF Intervals

(And keeping them in sync!)

Intervals and Synchronization

- Best Practice: use RMF and SMF intervals no greater than 15 minutes and sync them
  - This makes it easier to correlate data
  - Longer intervals can hide peaks that may be important
  - If you have short RMF intervals (<= 5 minutes) then maybe you don't want SMF same as RMF
    - Depends on how many SMF 30 interval records you're really ok with generating
    - If you have really short RMF intervals (e.g. <5 minutes) consider whether that's still needed given the data in the 98s and 99s

```
              In SMFPRMxx:


INTVAL(15)  - 15 minute SMF intervals
SYNCVAL(15) - Sync at 15 minutes after hour
```

Some products may require SYNCVAL(59)

```
                 In ERBRMFxx:
SYNC(SMF) - Sync with and use SMF intervals

                 In CMFCPMxx:
On REPORT statement:
… SYNC=SMF
```

SMF Interval Check

2020-01-16

● SYSA
● SYSB

This shows the smf30 interval records per interval where the interval is the duration and minute of the hour.

This is what this report should look like!

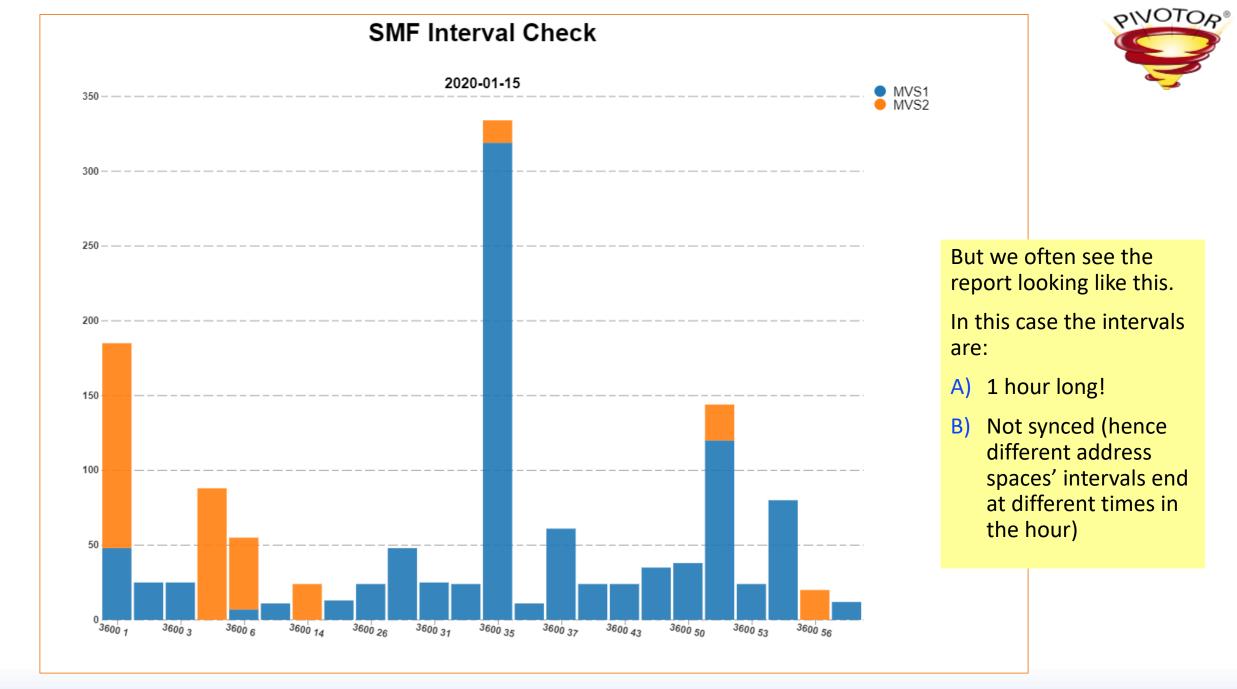# SMF Interval Check

2020-01-15

- ● MVS1
- ● MVS2

But we often see the report looking like this.

In this case the intervals are:

A) 1 hour long!

B) Not synced (hence different address spaces' intervals end at different times in the hour)

Key Sync Problem

- There are usually subsystem-specific options in your SMFPRMxx and those need to be set correctly too.
  - Sometimes there will be a different INTERVAL set there or NOINTERVAL
  - Default is NOINTERVAL which (I think) overrides the global interval
  - Easy answer: specify INTERVAL(SMF,SYNC) on the SYS and SUBSYS statements

```
SYNCVAL(00)                          /* SYNCRONIZE ON THE HOUR          */
INTVAL(15)                           /* STANDARD RECORDING INTERVAL     */
…
SYS(TYPE(0:125,127:255),INTERVAL(SMF,SYNC),DETAIL)
SUBSYS(STC,EXITS(IEFU29,IEFU83,IEFU84,IEFUJP,IEFUSI,IEFUSO),INTERVAL(SMF,SYNC))
```

If you aren't syncing your SMF intervals, you won't get new interval records coming in/out of system recovery boost, making those records that include boost periods problematic!

SMF Compression

Compression on z/OS

- CSRCESRV Macro
  - Very light-weight run length encoding (e.g. eliminate repeated zeros)
  - Consumes very little CPU

- CSRCMPSC Macro
  - Dictionary based compression (better general compression)
  - Hardware-assisted but still consumes some CPU time

- zEDC
  - Standard RFC 1951 DEFLATE ("gzip") compression offloaded to specific hardware
  - zEDC cards for machines before z15
  - On-chip Nest Accelerator Unit on z15
  - May consume very little CPU (except "sync" executions on z15)

Compression for SMF

- DB2 & CICS can optionally compress SMF records with CSRCESRV
  - CICS does this by default, DB2 requires enabling SMFCOMP
  - There's often a lot of 0s in this data so this can be quite helpful!
  - Do this even if doing one of the below as well!
- If dumping MANx datasets to sequential disk files, SMS can compress the QSAM files with either CSRCMPSC or zEDC
- If using SMF Logstreams, can use zEDC to compress the logstreams
- In general, SMF data can compress quite well
  - 5 to 1 or more may be possible

SMF Record Types

What to keep, what to not keep

SMF Record Types

- Historically, SMF record types are 0-255
  - 0-127 reserved for IBM and have fixed meanings
  - 128-255 for everybody else and may have varied meanings (e.g. data in record type 200 depends on which software product wrote it)
- New in z/OS 2.3 is "SMF record type constraint relief" (extended header format)
  - I.E. SMF record type can now go to 2047
  - Reserved for IBM: 0-127 and 1152-2047
  - Other products: 128-1151
  - New header format (including timezone offset!)
  - Existing IBM records not changing
  - Extended header records do not drive exits IEFU83, IEFU84, IEFU85 (IEFU86 is)
  - JES 2 now has record #1153 ("Usage and Performance Data")

SMF Data volume

- Size of the SMF data is often a concern

- Sites sometimes disable recording of certain records to help cut down on the amount of data
  - This may only be a good idea for a limited number of records (discussed later)

- Largest volume of data typically comes from "transactional" records
  - DB2: SMF 101
  - CICS: SMF 110
  - WebSphere Application Server: SMF 120
  - MQ: SMF 116 (rarely seen)
  - TCP/IP: SMF119

- Remember to compress where you can

These records can quite literally be over 90% of the total SMF data volume

SMF Flood Parameters

- Flood options in SMFPRMxx can help you determine when there are spikes of SMF data
  - Can optionally drop records while in a flood state
  - Or maybe can use automation to take some action on a flood state
- In example below, flood is detected if 999 type 101 have been generated per 1 second for 30 seconds and is disabled when < 999 records have been generated in 5 seconds
- Use FLDSTATS output from IFASMFDP to help figure out what good settings are for you

```
                In SMFPRMxx:

FLOOD(ON)

FLOODPOL(TYPE(101),
   RECTHRESH(999),
   INTVLTIME(10),
   MAXHIGHINTS(30),
   ENDINTVL(50),
   ACTION(MSG))
```

Batch job, STC, TSO, etc. details

Some Useful Options for Type 30s

Batch job, STC, TSO, etc. details

- **DDCONS(<u>YES</u>|NO)**
  - YES = consolidate duplicate EXCP entries in the type 30 records, which can save space, but may also elongate the shutdown of long-running jobs
  - NO = bypass the consolidation, possibly reducing job completion time

- **NODETAIL (Specified on subsys parameter in SMFPRMxx)**
  - Exclude EXCP sections from SMF 30.4 and 30.5 for STCs (but not batch or TSO)
  - Consider for STCs to speed up STC shutdown then allow DDCONS(YES)

- **EMPTYEXCPSEC(<u>NOSUPPRESS</u>|SUPPRESS)**
  - With the default, the SMF 30s have empty EXCP sections for each SMS candidate volume that is *not* allocated to the DD statement and for non-dataset allocations like DD DUMMY

```
         In SMFPRMxx:

EMPTYEXCPSEC(SUPPRESS)
DDCONS(YES)
SUBSYS(STC,NODETAIL)
```

Avoiding lots of 30s for OMVS work

- **_BPX_SHAREAS=(YES|<u>NO</u>|MUST)**
  - When set to "YES" spawned child processes are first attempted to run in the same address space as the login shell. Yes is recommended both for performance and also cuts down on the number of SMF type 30 records.
  - "MUST" says if the child process can't be created as a subtask, the request fails.
  - There may be situations where "NO" must be used, see:

    https://www.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.bpxb100/spn.htm
  - Useful if you have users / processes running lots of shell scripts spawning lots of tasks
- **_BPX_SPAWN_SCRIPT is similar (setting to YES is preferred)**
- **_BPX_BATCH_SPAWN is similar (SPAWN is preferred)**

```
        In /etc/profile or $HOME/.profile

export _BPX_SHAREAS=YES
export _BPX_SPAWN_SCRIPT=YES
export _BPX_BATCH_SPAWN=SPAWN
```

SMF 30 Instruction Counter

- SMFPRMxx option SMF30COUNT enables the SMF 30 Counter Data Section
  - Default is NOSMF30COUNT

- The idea for these counters was that while CPU time is variable due to things like cache contention, the number of instructions being executed should be stable, so maybe that would be a better measurement to use

- Except it ends up not being stable
  - CPU timers subtract out interrupt handling time
  - There's no similar mechanism for backing out interrupt handling instructions
  - So the instruction counts are potentially much more variable than CPU time

- So even though section is relatively small, why bother?
  - Recommendation: only enable if you're trying to find SIIS culprits (and it might not be useful even then)

```
In SMFPRMxx:

SMF30COUNT
```

Recording SMF

- SMF Records can be excluded or included in SMFPRMxx with the TYPE and/or NOTYPE options
  - In general: record everything, disable that which is truly not valuable and/or too voluminous to record
  - I.E. use NOTYPE instead of TYPE so that as new data is added you get it without having to change SMFPRM
  - (Although some new records require activation elsewhere)

> Just an example, may be incomplete

```
       In SMFPRMxx:

SYS(NOTYPE(4,5,20,34,35,40,
       92(10,11),99(13)),
   INTERVAL(SMF,SYNC),
   EXITS(IEFUJI,IEFUSI…),
   DETAIL)
SUBSYS(STC,NODETAIL)
```

Potential records to exclude

- Certain 99 subtypes (discussed later)

- Obsolete records: 4, 5, 20, 34, 35, 40
  - "**Note:** IBM recommends that you use record type 30 rather than record types 4, 5, 20, 34, 35, and 40."

- SMF 113 subtype 2 are obsolete, use 113 subtype 1 instead (low volume)

- Type 118 TCP/IP records: use 119 instead

- File System Activity: 92, subtypes 10, 11
  - 10 & 11 written on file open/close, there can be a whole lot of those!!
  - Some of the other 1-17 subtypes may be worth considering whether you really need
  - The new subtypes 50-59 are event or interval data that may be useful and may not be too voluminous

```
In IOEPRMxx:

smfrecording=ON
     or
smfrecording=ON,60
```

Records to Include (that maybe you aren't)

- New 98 High-frequency throughput statistics
  - IBM recommendation is to record on 5 second interval
    - Can use 5, 10, 15, 20, 30 or 60 seconds
    - 5 second interval is about 400MB-500MB/system/day

```
In SMFPRMxx:

HFTSINTVL(15)
```

- SMF 99 SRM/WLM details
  - Our minimum recommended subtypes: 6, 10, 11, 12, 14
  - Subtype 1, 2, and 3 can be quite useful, but can be more voluminous
  - Pivotor customers: send them if you're collecting them!
  - Subtype 13 is fairly voluminous and is undocumented "IBM use only"

- SMF 113 - HIS
  - Most sites have enabled this, but if you haven't: do so now

> None of these records represent data you'll look at every day, but it's nice to have them available when you need them!

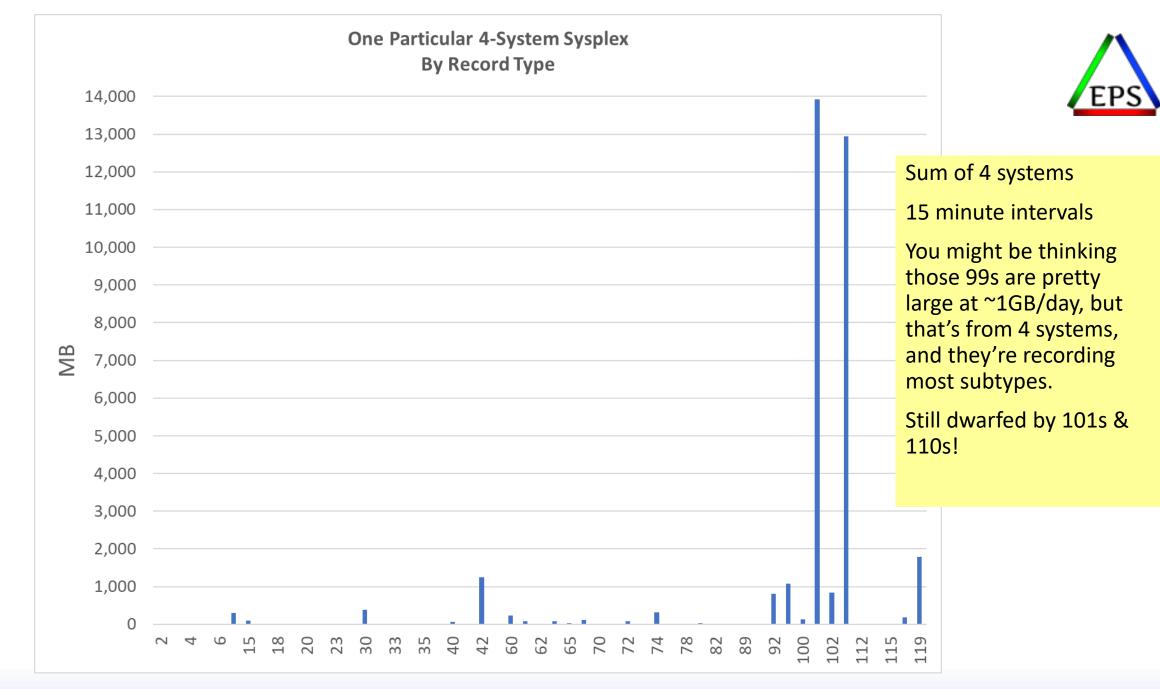| RECORD TYPE	RECORDS READ	PERCENT OF TOTAL	AVG. RECORD LENGTH	MIN. RECORD LENGTH	MAX. RECORD LENGTH	RECORDS WRITTEN
2	31	.00 %	18.00	18	18	1
3	31	.00 %	18.00	18	18	1
4	51,694	.12 %	215.00	215	215	0
5	10,132	.02 %	150.24	145	159	0
6	1,724	.00 %	517.75	3		0
14	630,668	1.49 %	486.05			0
15	201,376	.47 %	457.82	432	661	0
17	123,087	.29 %	100.00	100	108	
18	661	.00 %	144.00	144	144	
19	4,260	.01 %	132.00	132	132	
20	11,137	.03 %	96.11	91	105	
21	2,587	.01 %	104.00	104	104	
23	192	.00 %	7,770.00	7,7		
26	13,585	.03 %	520.42	5		
30	189,245	.45 %	2,076.99	480	32,748	188
32	1,508	.00 %	366.37	268	604	
33	959	.00 %	405.00	405	405	
34	350	.00 %	215.00	215	215	
35	350	.00 %	151.00	1		
36	10	.00 %	214.00	2		0
40	752,618	1.77 %	79.68	7	3,008	0
41	2,273	.01 %	190.82	146	412	0
42	1,334,921	3.15 %	930.24	172	32,748	0
57	11,478	.03 %	120.28	116	230	0

201,376*457.82=
92,193,960

189,245*2,076.99=
393,059,972

752,618*79.68=
59,968,602

IFASMFDP report

Import to Excel and multiply record count by average record length.

Percent records can be misleading in terms of the total space consumed by the records

One Particular 4-System Sysplex
By Record Type

Sum of 4 systems

15 minute intervals

You might be thinking those 99s are pretty large at ~1GB/day, but that's from 4 systems, and they're recording most subtypes.

Still dwarfed by 101s & 110s!

SubType	%Recs	%Bytes	Recs	Bytes
subtype 002_000	0.00	0.00	1	18
subtype 003_000	0.00	0.00	1	18
subtype 030_001	3.96	0.72	29,197	14,247,572
subtype 030_002	3.47	2.95	25,585	58,313,648
subtype 030_003	32.37	16.62	238,553	328,737,984
subtype 030_004	32.43	17.15	238,934	339,277,376
subtype 030_005	4.03	4.13	29,728	81,644,920
subtype 030_006	0.36	0.16	2,688	3,214,848
subtype 070_001	0.03	0.07	192	1,389,312
subtype 070_002	0.03	0.00	192	33,024
subtype 071_001	0.03	0.02	192	444,672
subtype 072_003	2.61	1.77	19,200	35,055,360
subtype 072_004	0.03	0.13	192	2,575,872
subtype 072_005	0.03	0.15	192	2,925,152
subtype 073_001	0.03	0.29	192	5,647,488
subtype 074_001	1.51	17.89	11,136	353,949,696
subtype 074_002	0.03	0.24	192	4,706,960
subtype 074_003	0.03	0.00	192	87,552
subtype 074_004	0.05	0.08	384	1,666,560
subtype 074_005	1.54	16.38	11,328	324,006,144
subtype 074_006	0.03	0.00	192	69,888
subtype 074_010	0.03	0.00	192	59,136
subtype 075_001	0.21	0.02	1,536	417,792
subtype 077_001	0.03	0.18	192	3,636,168
subtype 078_002	0.03	0.02	192	362,496
subtype 078_003	0.03	0.25	192	4,928,128
subtype 088_001	0.98	0.11	7,228	2,226,224
subtype 088_011	0.03	0.00	192	30,912
subtype 089_001	0.03	0.02	242	416,528
subtype 089_002	0.03	0.04	192	879,816
subtype 098_001	1.77	14.72	13,009	291,255,776
subtype 099_006	2.34	2.16	17,206	42,799,404
subtype 099_012	11.66	3.60	85,918	71,140,104
subtype 099_014	0.08	0.01	574	258,300
subtype 113_001	0.10	0.05	768	993,792
subtype 113_002	0.10	0.05	768	998,400

Rec Type	%Recs	%Bytes	Recs	Bytes
type 002	0.00	0.00	1	18
type 003	0.00	0.00	1	18
type 030	76.63	41.72	564,685	825,436,352
type 070	0.05	0.07	384	1,422,336
type 071	0.03	0.02	192	444,672
type 072	2.66	2.05	19,584	40,556,384
type 073	0.03	0.29	192	5,647,488
type 074	3.20	34.60	23,616	684,545,920
type 075	0.21	0.02	1,536	417,792
type 077	0.03	0.18	192	3,636,168
type 078	0.05	0.27	384	5,290,624
type 088	1.01	0.11	7,420	2,257,136
type 089	0.06	0.07	434	1,296,344
type 098	1.77	14.72	13,009	291,255,776
type 099	14.07	5.77	103,698	114,197,808
type 113	0.21	0.10	1,536	1,992,192

We get data from a lot of systems so I thought I'd take a sampling from some of our customers to see if the sizes are still showing as we expect.

This is from 2 systems, so we need to normalize to 1 system.

Type	Cust A Bytes/sys	Cust B Bytes/sys	Cust C Bytes/sys	Cust D Bytes/sys	Cust E Bytes/sys
Systems	2	8	1	4	9
98 interval	15	20	15	60	20
30	412,718,176	57,665,260	210,346,656	188,307,344	196,007,964
70	711,168	1,023,072	820,608	642,100	3,221,172
71	222,336	235,392	235,392	240,130	254,408
72	20,278,192	33,056,360	16,300,624	8,708,248	32,915,175
73	2,823,744	2,982,288	2,853,120	2,754,086	2,969,704
74	342,272,960	198,135,728	375,440,320	257,345,584	104,521,963
75	208,896	199,104	208,896	271,252	753,984
77	1,818,084		544,728		
78	2,645,312	1,835,606	2,477,792	2,013,796	3,258,829
79		244,324,544			
88	1,128,568	599,280	1,869,560	603,372	9,738,142
89	648,172	700,633			
98	145,627,888	84,730,712	152,343,584	57,914,608	140,806,201
99	57,098,904	72,508,176	1,718,093,696	71,055,984	1,275,147,492
113	996,096	1,621,023	3,375,360	1,913,172	2,745,704
115		820,501		877,396	2,858,227
119		205,897,776		136,401,200	

Sampling of common record types sent to us as daily transmissions from a few customers, 24 systems total.

30s have a fair bit of variability

74s also variable, depending on CF and I/O configuration

98s somewhat similar

99s seem to break down into 2 groups

	Cust A	Cust B	Cust C	Cust D	Cust E
Systems	2	8	1	4	9
Sub type	Bytes/sys	Bytes/sys	Bytes/sys	Bytes/sys	Bytes/sys
099_001			131,290,128		161,874,731
099_002			1,283,621,248		770,111,317
099_003			65,775,896		48,567,328
099_004					11,800,446
099_005					171,161
099_006	21,399,702	30,255,704	32,407,776	26,364,136	24,012,363
099_008			9,150,988		26,285,536
099_011		937,440	586,656	726,091	1,434,810
099_012	35,570,052	41,165,216	45,495,912	43,816,436	47,134,475
099_013			149,571,024		183,473,038
099_014	129,150	149,817	194,112	141,184	282,273

Now we see why Customers C and E have so much larger 99s: they included additional subtypes including the useless subtype 13.

Summary

- Sync your RMF and SMF intervals and don't use intervals longer than 15 minutes

- Check whether you can save some space on the type 30s with the right parameters

- Enable DB2 & CICS record compression

- Compressing SMF data on disk can make it easier to manage, and you don't necessarily need zEDC to compress sequential datasets

- Don't need to record obsolete/duplicative records

- Do record 98s and selected 99 subtypes
  - If you're a Pivotor customer include them on your daily data submissions!

Thanks!
Questions?

Email: scott.chapman@epstrategies.com