Measuring IMS Transactions
(Part 1)

Giuseppe Giacomodonato – EPV Technologies
Fabio Massimo Ottaviani – EPV Technologies
Enzo Rossi – EPV Technologies

November 2013

1 Introduction

After almost half a century, IMS (Information Management System) is still alive and in good health. According to IBM, more than 75% of the world’s Top 25 companies listed in the Fortune 100 entrust IMS to run their business.
So being able to measure and tune IMS applications is still an essential skill for every z/OS performance analyst.

From the measurement point of view, IMS is a kind of exception in the z/OS world; unlike all the other subsystems, IMS doesn’t write any SMF records. All the relevant events are mapped to a specific log record number and written to the IMS log.

For many years IMS didn’t provide a specific IMS log record, collecting all the performance related information. This information was spread across many different log records (e.g. x’07’ for program termination and x’08’ for program schedule) which had to be combined, by using a fairly complex algorithm, to analyse application throughput, response time and resource utilisation.

This has been one of the reasons for the success of the BMC Mainview for IMS (a.k.a. IMF) product. From the beginning Mainview for IMS has written its own IMS log records which are designed to analyse application performance. The most important of them is the “Transaction log record”, identified by the log record number x’FA’ (FA in the following), which provides all the details you could wish, about each IMS transaction execution, in a simple and straightforward way.

From IMS V10, IBM finally decided to provide an IMS log record specifically designed to collect performance information: the x’56FA’ log record (56FA in the following).

In this presentation we will discuss what you have to do in order to produce, manage and analyse both 56FA and FA records.
We will also present a case study where we will compare the values of the most important performance metrics collected in these records.

Finally we will briefly discuss the new metrics available to analyse zAAP/zIIP usage of IMS applications.
2 Producing 56FA log records

Since IMS V10, statistical data about transactions are logged in the 56FA log record. Unlike the old log records many customers used in the past (01, 07, 08, etc.), this log record is specifically designed to provide statistical information about transactions performance.

IMS will write one record for each unit of recovery (work done by an application program between sync points) for non-message-driven applications and for SNGL mode transactions; IMS will write one record for each processed message for MULT mode transactions of message-driven applications.

You can enable or disable 56FA logging globally during system definition by specifying a new parameter, TRANSTAT (Y/N, where N is the default in IMS V11, V12 and V13), in the Diagnostics Statistics section of the new DFSDFxxx PROCLIB member. This setting applies to any transactions and application programs that are created with the system definition process.

You can enable or disable 56FA logging on a program basis for non-message driven applications, and on a transaction-by-transaction basis for message-driven applications by using the appropriate IMS UPDATE and CREATE commands (TRANSTAT=N is also the default value in these commands).

3 Producing FA log records

The BMC Mainview for IMS product collects and records event data about IMS transaction executions into the FA records of the IMS log. A specific Mainview for IMS component is responsible for data collection: the Event Collector; you can customise it by setting data collection parameters.

A complete discussion of these parameters is beyond the scope of this paper so we will only focus on two of them:\n
- **BMP** = (YES | NO | NOCPU);
  - YES (default); BMP and JBP (Java Batch Processing)\(^2\) transaction and program activity data are collected;
  - NO; no data are collected for BMP and JBP activity;
  - NOCPU; BMP and JBP transaction and program activity data, but not CPU time usage, are collected.
- **CPU** = (DEP | DEPPGM | DEPDB2 | ALL | NONE)
  - DEP (default); transaction CPU time is collected from dependent regions only; it is recorded as application, DL/I, and DB2 time; most chargeable CPU time is collected; CPU fields for CONTROL and DLISAS are zero;
  - DEPPGM; transaction CPU time is collected from dependent regions only; it is recorded as application time and includes DL/I and DB2 time; most chargeable CPU time is collected; CPU fields for CONTROL and DLISAS are zero;
  - DEPDB2; transaction CPU time is collected from dependent regions only; it is recorded as application and DB2 time; the DL/I CPU time is included in the application CPU time. CPU fields for CONTROL and DLISAS are zero;

\(^1\) For more details see “Mainview for IMS Offline – Customization and Utilities Guide”.
\(^2\) JBP regions are similar to non-message-driven BMP applications.
EPV Technologies

- ALL; all transaction CPU times from dependent, CONTROL and DLISAS regions are collected;
- NONE; transaction CPU time is not collected.

As always happens, you have a tradeoff between overhead and information. BMC default values are normally a good compromise. We strongly suggest you use them.

4 Managing log records

The DFSERA10 utility allows you to easily select only the record types you need from the IMS log. The JCL example in Figure 1 will select both the FA and 56FA records from a SLDS (System Log Data Set) and copy them to the output file referenced by the SYSUT4 DD.

```
//SELIMS   EXEC PGM=DFSERA10
//STEP1B   DD DSN=RESLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*  
//SYSUT1   DD DSN=SLDS,  
  //         DISP=(OLD, PASS),UNIT=TAPE
//SYSUT4   DD DSN=OUTPUT_FILE,DISP=(,CATLG,DELETE),  
  //         UNIT=SYSDA,  
  //         SPACE=(CYL,(100,100),RLSE),  
  //         DCB=(RECFM=VB,LRECL=30970,BLKSIZE=30974)
//SYSIN    DD *
CONTROL  CNTL STOPAFT=EOF
OPTION COPY OFFSET=5,FLDLEN=2,VALUE=56FA,COND=E
OPTION COPY OFFSET=5,FLDLEN=1,VALUE=FA,COND=E
END
```

**Figure 1**

If you don’t want to risk losing records, remember to specify STOPAFT=EOF in the CONTROL statement; by default only the first 16,777,215 records are processed.

Some of the tools available to analyse IMS log records run outside of the mainframe (see next chapter) so you may need to transfer the file including the selected IMS log records to an external platform (e.g. Windows, Linux). When moving VBS and VB files from the mainframe to distributed platforms they have to be masked as undefined to avoid FTP stripping the control blocks out, making the files unreadable.

If you use the standard IBM FTP you can do that directly in the FTP step, otherwise you need an additional IEBGENER step to just modify the dataset DSCB without having to copy the file. This technique requires negligible time and resources. Here is an example of the JCL you can use.

```
//VB2U  EXEC PGM=IEBGENER 
//SYSIN  DD DUMMY  
//SYSPRINT DD SYSOUT=*  
//SYSUT2   DD DSN=OUTPUT_FILE,  
  //         DISP=MOD,DCB=RECFM=U  
//SYSUT1   DD DSN=NULLFILE,DCB=* .SYSUT2
```

**Figure 2**

3 Words in italic have to be substituted with the appropriate file names.
4 Default value in IMS V11, V12 and V13.
5 Analysing log records

All the IBM log records layout, including 56FA, can be obtained by assembling the ILOGREC macro as shown in the JCL example in Figure 3. A snapshot of the resulting 56FA DSECT is provided in Figure 4.

```plaintext
//DSECT  PROC
//HASM    EXEC PGM=IEV90,PARM='NODECK,NOXREF,LIST,NORLD,NOOBJECT'
//SYSLIB  DD DISP=SHR,DSN=IMS_prefix.SDFSMAC
//SYSUT1  DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSUT2  DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSUT3  DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSPRINT DD SYSOUT=*  PEND
//D1      EXEC DSECT
//SYSIN   DD *
ILOGREC  RECID=56
END
```

Figure 3

```
241+ ORG TPCPDATA  DATA PART  SU10FTB1  02-DFSET
242+ TPPFXLEN  DC H'0'  LENGTH OF PREFIX  SU10FTB1  02-DFSET
243+ TPIMSVER  DC X'0'  IMS VERSION (E.g., V10 = X'10')  SU10FTB1  02-DFSET
244+ TPIMSREL  DC X'0'  IMS RELEASE (E.g., R1 = X'10')  SU10FTB1  02-DFSET
245+ TPRECVER  DC H'0'  RECORD VERSION  SU10FTB1  02-DFSET
246+ TPRECV1  EQU X'0001'  - VERSION 1  SU10FTB1  02-DFSET
247+ TPREVCVR  TPRECV1  - CURRENT VERSION OF RECORD  SU10FTB1  02-DFSET
248+ TPJOBN   DC CL8' '  JOB NAME  SU10FTB1  02-DFSET
249+ TPSTEPN  DC CL8' '  STEP NAME  SU10FTB1  02-DFSET
250+ TPLTERM  DC CL8' '  INPUT LTERM  SU10FTB1  02-DFSET
251+ TPNWID   DC CL8' '  NETWORK ID OF MESSAGE FROM APPC  SU10FTB1  02-DFSET
252+ TPUNAME  DC CL8' '  LU NAME OF MESSAGE FROM APPC  SU10FTB1  02-DFSET
253+ TPPGMNM  DC CL8' '  PROGRAM NAME  SU10FTB1  02-DFSET
254+ TPTYPE   DC H'0'  PROGRAM TYPE  SU10FTB1  02-DFSET
255+ TPCIC    DC H'0'  TRAN CLASS  SU10FTB1  02-DFSET
256+ TPRTY    DC X'0'  TRAN PRIORITY  SU10FTB1  02-DFSET
257+ TPTYPE   DC X'0'  PROGRAM TYPE  SU10FTB1  02-DFSET
258+ TPRGTM   EQU X'0'  - MODE=MULT TRANSACTION  SU10FTB1  02-DFSET
259+ TPWFI    EQU X'40'  - WFI TRANSACTION OR WFI REGION  SU10FTB1  02-DFSET
260+ TPJAVA   EQU X'20'  - JAVA DEPENDENT REGION  SU10FTB1  02-DFSET
261+ TPCIC    EQU X'10'  - CPI-C REGION  SU10FTB1  02-DFSET
262+ TPFFP    EQU X'08'  - IPP REGION  SU10FTB1  02-DFSET
263+ TPDBT    EQU X'04'  - DBCTL THREAD  SU10FTB1  02-DFSET
264+ TPBMP    EQU X'02'  - BMP REGION  SU10FTB1  02-DFSET
265+ TPJBP    EQU X'22'  - JBP REGION  SU10FTB1  02-DFSET
266+ TPMPP    EQU X'01'  - MPP REGION  SU10FTB1  02-DFSET
267+ TPJMP    EQU X'21'  - JMP REGION  SU10FTB1  02-DFSET
268+ DC CL8' '  RESERVED  SU10FTB1  02-DFSET
269+ TPCPCD   DC F'0'  COMPLETION CODE (ABEND CODE)  SU10FTB1  02-DFSET
270+ DC F'0'  RESERVED  SU10FTB1  02-DFSET
271+ TPJPM    DC D'0'  CPU EXECUTION TIME IN TOD CLOCK FORMAT  SU10FTB1  02-DFSET
272+ TPJPM    DC X'00'  START TIME OF UOR (STCKE)  SU10FTB1  02-DFSET
273+ TPJPM    DC X'02'  START UTC DATE/TIME OF UOR  SU10FTB1  02-DFSET
274+ TPJPM    DC X'22'  END TIME OF UOR (STCKE)  SU10FTB1  02-DFSET
275+ DC F'0'  RESERVED  SU10FTB1  02-DFSET
276+ DC F'0'  RESERVED  SU10FTB1  02-DFSET
277+ TPJPM    DC X'22'  END TIME OF UOR (STCKE)  SU10FTB1  02-DFSET
278+ TPJPM    DC X'01'  START UTC DATE/TIME OF UOR  SU10FTB1  02-DFSET
279+ DC F'0'  RESERVED  SU10FTB1  02-DFSET
280+ DC F'0'  RESERVED  SU10FTB1  02-DFSET
281+ TPJPM    DC X'00'  START TIME OF UOR (STCKE)  SU10FTB1  02-DFSET
282+ TPJPM    DC X'01'  END UTC DATE/TIME OF UOR  SU10FTB1  02-DFSET
283+ TPJPM    DC X'22'  END TIME OF UOR (STCKE)  SU10FTB1  02-DFSET
284+ DC F'0'  RESERVED  SU10FTB1  02-DFSET
285+ DC F'0'  RESERVED  SU10FTB1  02-DFSET
286+ TPJPM    DC X'00'  START TIME OF UOR (STCKE)  SU10FTB1  02-DFSET
287+ TPJPM    DC X'01'  END UTC DATE/TIME OF UOR  SU10FTB1  02-DFSET
288+ TPJPM    DC X'22'  END TIME OF UOR (STCKE)  SU10FTB1  02-DFSET
289+ DC F'0'  RESERVED  SU10FTB1  02-DFSET
```

Figure 4
BMC FA record layout is described in the Appendix A of “Mainview for IMS Offline – Customization and Utilities Guide”.
FA record DSECTs are also provided in the IMETRN and IMETRNX members of the BBSAMP library.

Based on the above information you could write your own program to analyse IMS log records but it is probably more convenient for you to use one of the existing tools.

As a final note, it is worth considering that your IMS environments may easily produce millions of 56FA or FA log records. Interpreting and analysing them on a non-mainframe platform, by using tools such as EPV zParser and EPV SMF2XL, could provide consistent savings in terms of software license costs and hardware resources.

*In the second part of this paper we will present a case study where we will compare the values of the most important performance metrics collected in 56FA and FA records.*