

Messen mit Strobe

und

die Verbindung zum Programm

Inhaltsverzeichnis

Batch-Job messen.....	2
Vom CPU-Verbrauch im Programm zum COBOL-Befehl.....	6
Ergänzender Hinweis zum Erstellen des Profils	9

Eine Ausarbeitung von:

cps4it

Ralf Seidler • Stromberger Straße 36A • 55411 Bingen
Fon: +49-6721-992611 • Fax: -49-6721-992613 • Mail: ralf.seidler@cps4it.de
Internet : <http://www.cps4it.de>
Steuernummer: 08/220/2497/3, Finanzamt Bingen, Ust-ID : DE214792185

Batch-Job messen

In diesem Kapitel wird an Hand von Screenshots beschrieben, wie Strobe-Messungen für Batch-Jobs angestoßen werden können, wenn nicht die Sysplex-Version von Strobe aktiviert ist. Wesentlich ist, dass der zu messende Job in der gleichen LPAR (logische Partition) läuft, wie die TSO-Session, in der man angemeldet ist. Ist welchem System man angemeldet ist, sieht man auf dem Primary Panel im ISPF rechts.

Vorbereiten der Main-Karte im Job:

```
//XV8822DX JOB '1N289003010200100000',REXX#J,  
//          CLASS=S,MSGCLASS=Y,  
//          TIME=1440,REGION=0M,  
//          COND=(0,NE)  
//*MAIN SYSTEM=MVSD, ██████████
```

Messen des Jobs über die Funktion 2 (Add Queued). Auf die einzelnen Angaben wird hier nicht eingegangen; sie wurden im Kurs erklärt bzw. können im Strobe-Handbuch nachgelesen werden. (Oder den Autor fragen. ☺)

```
----- Strobe - Add Queued Request ----- Req 62 QUEUED  
COMMAND ==>  
  
JOBNAME   ==> XV8822DX (Jobname or jobname* for pattern match)  
PROGRAM  ==>          (Program or clear both PROGRAM & STEP to input a list)  
STEP     ==> S010.TCAF (Name,num,step.procstep or *ALL for all steps)  
  
SCHEDULE REQUEST ==> N (Y or N; Use Y only when overriding the default)  
AUTO PROFILE CREATION ==> N (Y or N; Use Y only when overriding defaults)  
  
MEASUREMENT SESSION INFORMATION:  
SESSION DURATION   ==> 1 (Estimated time in minutes)  
TARGET SAMPLE SIZE ==> 5000 (Target number of samples)  
  
TSO USERID TO NOTIFY ==> XV8822D (Notify when session completes)  
  
SAMPLE DATASET INFORMATION:  
DATASET NAME PREFIX ==> USER.STROBE VOLUME ==>  
UNIT NAME   ==> SYSDA VOLUME ==> DISP ==> CATLG (CATLG OR KEEP)  
  
SELECT ADDITIONAL PARAMETERS: (Y or N; Use Y only when overriding defaults)  
DATA COLLECTORS ==> N MODULE MAPPING DATA ==> N  
SESSION MANAGEMENT ==> N REQUEST RETENTION ==> N  
OTHER PARAMETERS ==> N SAVE REQUEST IN GROUP ==> N
```


Messungen mit Strobe und die Verbindung zum Programm

Bild für Submit . . .

```
----- Strobe - Select Options for Job Stream ---- JCL generated
OPTION ==> 1

  1 - Submit the job stream for execution
  2 - Write the job stream to a dataset and edit
  3 - Write the job stream to a dataset and submit for execution

END - Cancel

JOB STATEMENT INFORMATION:
  //XV8822DA JOB '1N289003010200100000','R.SEIDLER AL34-188',
  //          CLASS=S,MSGCLASS=Y,
  //          TIME=1440,REGION=0M,NOTIFY=&SYSUID,
  //          COND=(0,NE)

If you select option 2 or 3, specify:
DATASET  NAME ==>
          UNIT ==> SYSDA          VOLUME ==>
```

```
IKJ56250I JOB XV8822DA(JOB01679) SUBMITTED
***
```

```
IAT6108 JOB XV8822DA (JOB01679) ENDED, MAXCC=0000          CN(INTERNAL)
***
```

Anzeige der Datei im Browse – siehe nächstes Kapitel

*Hinweis: Die Dateien XV8822D.** sind für alle lesbar!*

Vom CPU-Verbrauch im Programm zum COBOL-Befehl

Hinweis: Die Dateien XV8822D.** sind für alle lesbar!

In diesem Beispiel ging es um den CPU-Verbrauch im Programm P03N941. In dem Kapitel #IEP findet man die Top-10-Verbraucher.

```

BROWSE          XV8822D.STROBE.P03N941.LANG          Line 00000000 Col 001 080
Command ==>                                         Scroll ==> CSR
***** Top of Data *****
#IEP          ** MOST INTENSIVELY EXECUTED PROCEDURES **

MODULE      SECTION  LINE      PROCEDURE      START      % CPU TIME
NAME        NAME      NUMBER    NAME           LOC        SOLO TOTAL
P03N941     P03N941                    004E40     80.82 96.82
P03N941     P03N941                    004E80     1.28 1.52
QUICKSA0
.SVC        SVC 011          TIME          000280     .26 .31
.SVC        SVC 120          GETMAIN/FREEMAIN .13 .20
.COBLIB     IGZCPAC         COBPACK GENERAL .11 .13
QUICKSA0
BTCHCKP0
QUICKSAM
P03N941     P03N941                    0000C0     .09 .09
P03N941     P03N941                    0003C0     .07 .09
P03N941     P03N941                    000000     .07 .07
P03N941     P03N941                    001680     .07 .07
    
```

Suchen wir die Stelle in der Compileliste. Es ist – wie bei der Dump-Analyse darauf zu achten, dass die Compileliste zum Joblauf passt.

Suchen wir genau die Stelle, die Strobe ausweist:

```

File Edit Edit_Settings Menu Build SCLM Utilities Test Help
*****
VIEW          S0300.TEST.COMPMVS (P03N941) - 01.00          Columns 00001 00072
Command ==>  f 004E40          Scroll ==> CSR
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 1PP 5655-G53 IBM Enterprise COBOL for z/OS 3.4.1
000002 0Invocation parameters:
000003  APOST,DYN,TRUNC(OPT)
000004 -Options in effect:
000005          NOADATA
    
```

Ergebnis (in diesem Fall):

```

No CHARS '004E40' found
    
```

Na ist doch super. Hat der Seidler wieder Mist erzählt? Passt die Compileliste nicht? All das ist zu prüfen. In diesem Fall gilt weder das Eine noch das Andere.

Messungen mit Strobe und die Verbindung zum Programm

Neuer Versuch: Schauen wir mal, ob wir etwas im Umfeld von 004E40 finden.

```
File Edit Edit_Settings Menu Build SCLM Utilities Test Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
VIEW          S0300.TEST.COMPMVS (P03N941) - 01.00          Columns 00001 00072
Command ==>  F 004E4          Scroll ==> CSR
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 1PP 5655-G53 IBM Enterprise COBOL for z/OS 3.4.1
000002 0Invocation parameters:
```

Jep. Da ist was. Aber was?

```
File Edit Edit_Settings Menu Build SCLM Utilities Test Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
VIEW          S0300.TEST.COMPMVS (P03N941) - 01.00          CHARS '004E4' found
Command ==>          Scroll ==> CSR
010547      004E3E  9240 27EB          MVI  2027(2),X'40'          TAB-S1
010548      004E42  D260 27EC 27EB          MVC  2028(97,2),2027(2)    TAB-S1
010549      004E48  9240 284D          MVI  2125(2),X'40'          TAB-S1
010550      004E4C  D20E 284E C004          MVC  2126(15,2),4(12)      TAB-S1
010551      004E52  D207 285D 3A48          MVC  2141(8,2),2632(3)     (BLW=3
010552      004E58  9240 2865          MVI  2149(2),X'40'          TAB-S1
010553      004E5C  9240 2866          MVI  2150(2),X'40'          TAB-S1
010554      004E60  D226 2867 2866          MVC  2151(39,2),2150(2)    TAB-S1
010555      004E66  4110 2780          LA   1,1920(0,2)          TAB-S1
```

Blättern wir mal zurück (PF7), bis wir „vorne“ einen COBOL-Befehl finden.

```
010523 002006 PERFORM
010524 002653 INITIALIZE
010525      004DCE  5820 915C          L    2,348(0,9)          BLW=3
010526      004DD2  9240 2780          MVI  1920(2),X'40'          TAB-S1
010527      004DD6  D20E 2781 C004          MVC  1921(15,2),4(12)      TAB-S1
010528      004DDC  5830 C028          L    3,40(0,12)          CBL=1
010529      004DE0  D205 2790 3A7B          MVC  1936(6,2),2683(3)     (BLW=3
010530      004DE6  4140 000C          LA   4,12(0,0)
010531      004DEA  4040 2796          STH  4,1942(0,2)          (BLW=3
010532      004DEE  4040 2798          STH  4,1944(0,2)          (BLW=3
010533      004DF2  D202 279A 3B06          MVC  1946(3,2),2822(3)     (BLW=3
010534      004DF8  D208 279D 3A04          MVC  1949(9,2),2564(3)     (BLW=3
010535      004DFE  D204 27A6 3A48          MVC  1958(5,2),2632(3)     (BLW=3
010536      004E04  D204 27AB 3A48          MVC  1963(5,2),2632(3)     (BLW=3
010537      004E0A  D204 27B0 3995          MVC  1968(5,2),2453(3)     (BLW=3
010538      004E10  D206 27B5 3888          MVC  1973(7,2),2184(3)     (BLW=3
010539      004E16  9240 27BC          MVI  1980(2),X'40'          (BLW=3
010540      004E1A  9240 27BD          MVI  1981(2),X'40'          TAB-S1
010541      004E1E  D211 27BE C004          MVC  1982(18,2),4(12)      TAB-S1
010542      004E24  D202 27D0 3B06          MVC  2000(3,2),2822(3)     (BLW=3
010543      004E2A  9240 27D3          MVI  2003(2),X'40'          TAB-S1
010544      004E2E  D212 27D4 C004          MVC  2004(19,2),4(12)      TAB-S1
010545      004E34  D202 27E7 3B06          MVC  2023(3,2),2822(3)     (BLW=3
010546      004E3A  9240 27EA          MVI  2026(2),X'40'          TAB-S1
010547      004E3E  9240 27EB          MVI  2027(2),X'40'          TAB-S1
010548      004E42  D260 27EC 27EB          MVC  2028(97,2),2027(2)    TAB-S1
010549      004E48  9240 284D          MVI  2125(2),X'40'          TAB-S1
010550      004E4C  D20E 284E C004          MVC  2126(15,2),4(12)      TAB-S1
010551      004E52  D207 285D 3A48          MVC  2141(8,2),2632(3)     (BLW=3
010552      004E58  9240 2865          MVI  2149(2),X'40'          TAB-S1
```

... und schon haben wir den Übeltäter. Der Befehl steht auf der Zeile 2653.

Messungen mit Strobe und die Verbindung zum Programm

... also suchen wir danach ...

```
File Edit Edit_Settings Menu Build SCLM Utilities Test Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
VIEW          S0300.TEST.COMPMVS(P03N941) - 01.00          Columns 00001 00072
Command ==> f first 002653                               Scroll ==> CSR
```

... und finden – wie behauptet – den Befehl.

```
S0300.TEST.COMPMVS(P03N941) - 01.00          Columns 00004 00075
==>                                         Scroll ==> CSR
LineID  PL SL  ----+*A-1-B--+----2----+----3----+----4----+----5----+--
002647          /
002648          *
002649          init-tab-s1 section.
002650          * == Initialisieren der Zwischenspeicher-Tabelle
002651          * == fuer die Satzartendatei
002652          *
002653          XXX          initialize tab-satz1-X
002654          *Perf. Schleife nur bis i-s1 laufen lassen
002655          XXX          * perform with test before
002656          XXX          * varying i-ts1 from 1 by 1
002657          XXX          * until i-ts1 > i-s1
002658          XXX          * initialize tab-s1-ordbeg (i-ts1)
002659          XXX          * initialize tab-s1-daten (i-ts1)
002660          XXX          * end-perform
002661          *
002662          * == Initialisieren des Zaehlers fuer jeden Ordnung
002663          move zeroes to i-s1
002664          * move zeroes to i-ts1 nicht noetig, da immer
002665          *
002666          continue.
002667          *
002668          init-tab-s1-ex.
```

Der blau geschriebene und gelb hinterlegte Code ist der, den ich dann eingefügt hatte. Das Feld i-s1 wird während des Programmdurchlaufs stets gepflegt (der Code war schon vorhanden).

Ergebnis der Optimierung – wie im Kurs erwähnt – von mehr als 4 Minuten auf unter 1 Sekunden Laufzeit und CPU-Zeit.

Ergänzender Hinweis zum Erstellen des Profils

(Auch der Autor selbst fällt fast immer darauf herein.)

Um die Auflösung der Statement statt mit 64 (hexadezimal 40) Byte mit 4 Byte zu erhalten, muss ich **immer** Folgendes tun:

```
SPECIFY PROFILE REPORT PARAMETERS: (Y or N)
Detail Reports      ==> Y Tailor Reports
```

Y angeben und auf der Folgemaske die 4 Byte Auslösung wählen. Die Datei wurde schon erzeugt und steht unter dem Namen XV8822D.STROBE.P03N941.LANG1.

Das Ergebnis sieht dann so aus (Extrakt aus dem Profil in der Datei **.LANG1, hier nicht aus dem Kapitel #IEP sondern dem Kapitel, in dem alle Programme explizit aufgeführt sind (#PUP?)):

```
MODULE - P03N941
SECTION - P03N941
```

LINE NUMBER	PROCEDURE NAME	START LOC	INTRVL LENGTH	% CPU SOLO	TIME TOTAL	MARGIN .00	OF ERROR 47.00	1.46% 94.00
		000000	5804	.00	.00	-.		
		0016AC	4	.02	.02	.		
...snip...								
LINE NUMBER	PROCEDURE NAME	START LOC	INTRVL LENGTH	% CPU SOLO	TIME TOTAL	MARGIN .00	OF ERROR 47.00	1.46% 94.00
		004C2C	580	.00	.00	-.		
		004E70	4	.62	.77	.		
		004E74	4	2.21	2.69	.+		
		004E78	4	.00	.00	.		
		004E7C	4	78.00	93.36	.*****++++		
		004E80	4	.55	.62	.		
		004E84	4	.73	.90	.		
		004E88	3948	.00	.00	-.		
SECTIO P03N941 TOTALS				82.30	98.57			

Wenn wir jetzt in der Compileliste suchen . . .

```
File Edit Edit Settings Menu Build SCLM Utilities Test Help
#####
VIEW          S0300.TEST.COMPMVS(P03N941) - 01.00          Columns 00001 00072
Command ==> F 004E7C                                     Scroll ==> CSR
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 1PP 5655-G53 IBM Enterprise COBOL for z/OS 3.4.1
000002 0Invocation parameters:
```

Messungen mit Strobe und die Verbindung zum Programm

... finden wir:

010523	002006	PERFORM					
010524	002653	INITIALIZE					
010525	004DCE	5820 915C		L	2,348(0,9)		BLW=3
010526	004DD2	9240 2780		MVI	1920(2),X'40'		TAB-S1
010527	004DD6	D20E 2781 C004		MVC	1921(15,2),4(12)		TAB-S1
010528	004DDC	5830 C028		L	3,40(0,12)		CBL=1
010529	004DE0	D205 2790 3A7B		MVC	1936(6,2),2683(3)		(BLW=3
010530	004DE6	4140 000C		LA	4,12(0,0)		
010531	004DEA	4040 2796		STH	4,1942(0,2)		(BLW=3
...	snip...						
010558	004E72		GN=488	EQU	*		
010559	004E72	5820 915C		L	2,348(0,9)		BLW=3
010560	004E76	D20D 1000 2780		MVC	0(14,1),1920(2)		TAB-S1
010561	1PP 5655-G53	IBM Enterprise COBOL for z/OS 3.4.1					P03N941
010562	0	004E7C D2FF 100E 278E		MVC	14(256,1),1934(2)		TAB-S1
010563	004E82	4110 110E		LA	1,270(0,1)		
010564	004E86	4600 BEA8		BCT	0,3752(0,11)		GN=488
010565	002663	MOVE					

Noch einmal der generelle Hinweis: Es kann durchaus sein, dass das Offset nicht gefunden wird. Dann in der Umgebung suchen.