

# **Identifying Top CPU Consuming COBOL Programs**



**Igor Cohen**  
**Reid Copeland**  
**Cindy Guo**

Version Date: 19/08/2020

## Contents

Introduction and motivation .....	1
Identifying peak usage times.....	2
Identifying top CPU consuming job-program pairs from peak usage times.....	3
Identifying top CPU consuming modules from job-program pairs .....	6
Identifying top CPU consuming COBOL CSECTs from modules.....	10
Summary .....	12
List of resources .....	13

## Introduction and motivation

This document is for [IBM Enterprise COBOL for z/OS](#) and [IBM Automatic Binary Optimizer \(ABO\) for z/OS](#) clients to help identify their top CPU consuming COBOL programs.

Although your applications may have thousands of COBOL programs, many likely consume very little CPU. Optimizing all your programs using ABO or migrating all of them to the latest compiler can be expensive and time consuming. By targeting the most performance critical sections of your application code for optimization and migration, you can obtain most of the performance improvements of the latest optimizer and compiler technologies but with a much smaller effort.

This document provides a step by step process that starts from a system wide view of peak usage times and drills down to the exact COBOL programs that would deliver the most value from performance improvements.

# Identifying peak usage times

Tools such as [IBM z Batch Network Analyzer \(zBNA\) tool](#), IBM Sub-Capacity Reporting Tool (SCRT) and the z/OS Resource Measurement Facility (RMF) can be used to identify times of the day or days of the month when CPU usage is the highest.

zBNA is a no charge, PC based productivity tool to visualize CPU usage reports from SMF records. This document uses zBNA as the example tool to identify times of peak CPU usage. Once the zBNA client is opened and all required SMF records have been loaded (see zBNA Users Guide for details), you can display the rolling 4-hour average graph by clicking

## Graph > Display Graph: Rolling 4 Hr Avg.

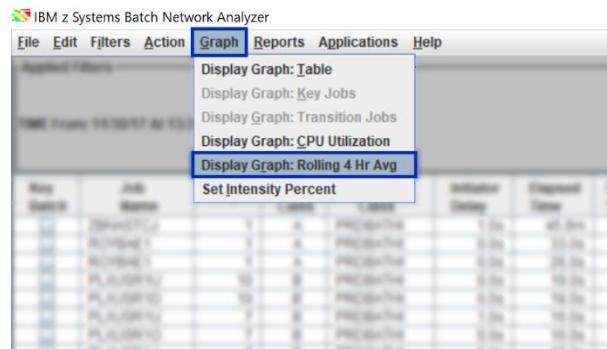


Figure 1a) Screenshot of zBNA showing the Graph tab

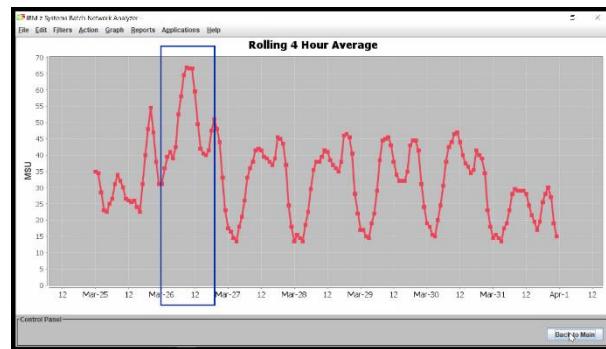


Figure 1b) Rolling 4 Hour Average for March

In this example, the peak usage is on March 26<sup>th</sup>.

Next, use zBNA “**Filters > Set Table Filters...**” to zoom into the day and find the peak usage interval.

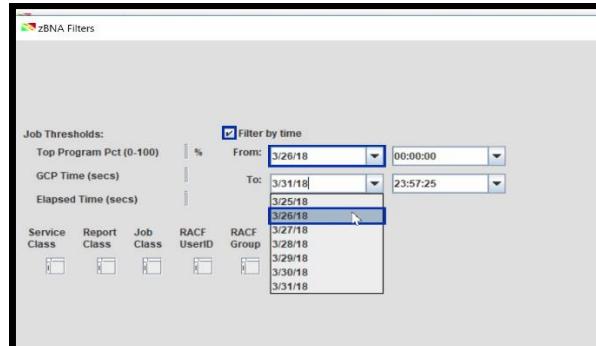


Figure 2a) Screenshot of zBNA showing the Filters page



Figure 2b) Rolling 4 Hour Average on March 26<sup>th</sup>

In this example, the peak usage interval is from 7am to 11am.

# Identifying top CPU consuming job-program pairs from peak usage times

With the peak CPU usage time now identified the top CPU consuming job-program pairs can be listed and extracted from zBNA.

First, add Step Level Records by clicking **File > Add Selected Step Level Records...**. Then use zBNA to save the CPU usage report as a comma-separated values (CSV) file with jobs and steps information by selecting **File > Save as CSV > Save as CSV, Jobs & Steps ...**.

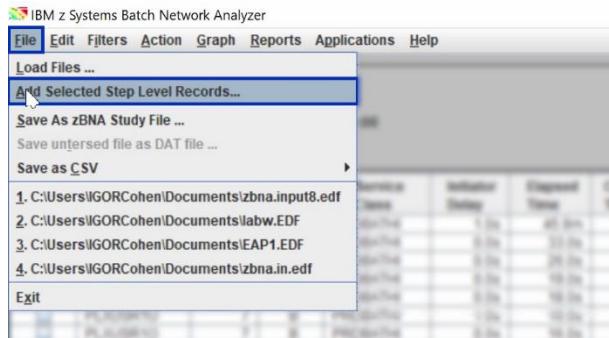


Figure 3a) Screenshot of zBNA showing the File tab

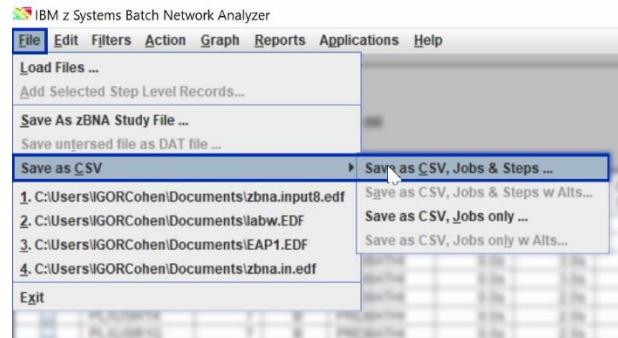


Figure 3b) Screenshot of zBNA showing the File tab

Next, generate the candidate job-program pairs and their cumulative CPU time by editing the CSV file.

**Tip:** For users of zBNA earlier than 2.1, we have developed a tool to help you generate a job list from the CSV file. For more information, please go to <https://www.ibm.com/support/pages/node/742651>

The zBNA 2.1 ‘Top programs’ report can be used to help you generate a job list from the CSV file directly

1. In the CSV file you will see columns labelled “Job Name” and “Program Name”.

Key	Batch	Start Date	End Date	Run Time	Job Name	Proc Name	Program Name	Step Num	Sub Type	Job Class	Initiator	Elapsed	TirAAP	Time	IIP	Cp Time	Init Time	Condition	Interval
2	FALSE	3/26/2018	35:06.4	3/27/2018	30:21.4	CACAS9	D38PGM4	1	JOB	STC	1	86115.08	0	0	0.03	0	0.08		
3	FALSE	3/26/2018	35:06.4	3/27/2018	30:21.4	CACAS9	D38PGM4	1	JOB	STC	1	86115.08	0	0	0.03	0	0.08		
4	FALSE	3/26/2018	47:15.2	3/26/2018	04:18.2	CAIOB1	U269PGM2	1	STEP	C	1	19022.98	0	0	1.48	0	0.335		
5	FALSE	3/26/2018	47:15.2	3/26/2018	47:22.8	CATEST7	40:52.6	2	STEP	C	16	1.455	0	0	0.23	0	0.021		
6	FALSE	3/26/2018	47:29.8	3/26/2018	47:31.3	CACASE4	A	FTP	3	STEP	C	17	1.01	0	0	0.02	0	0.034	
7	FALSE	3/26/2018	47:31.3	3/26/2018	47:32.3	CACASE4	A	FTP	4	STEP	C	18	0.92	0	0	0.02	0	0.049	
8	FALSE	3/26/2018	47:32.3	3/26/2018	47:33.2	CACASE10	A	FTP	5	STEP	C	19	1.02	0	0	0.02	0	0.054	
9	FALSE	3/26/2018	47:33.2	3/26/2018	47:34.2	CAIOB10	A	D29TPGM6	6	STEP	C	20	283.82	0	0	0.04	0	0.117	
10	FALSE	3/26/2018	47:34.3	3/26/2018	34:50.1	CATEST4	A	D289PGM6	7	STEP	C	2856	280.58	0	0	0.02	0	0.280	
11	FALSE	3/26/2018	34:50.1	3/26/2018	39:30.7	CADUM3	A	E119PGM6	8	STEP	C	3137	81.94	0	0	0.03	0	0.306	
12	FALSE	3/26/2018	39:30.7	3/26/2018	40:52.6	CACASE6	A	E161PRG2	9	STEP	C	3219	33.95	0	0	0.04	0	0.570	
13	FALSE	3/26/2018	40:52.6	3/26/2018	41:26.6	CATEST7	A	E194PGM6	10	STEP	C	3252	555.69	0	0	0.04	0	0.652	
14	FALSE	3/26/2018	41:26.6	3/26/2018	50:42.7	CATEST7	A	G239PRG9	11	STEP	C	3808	637.4	0	0	0.07	0	0.705	
15	FALSE	3/26/2018	50:42.7	3/26/2018	50:43.7	CACASE7	A	E129PGM6	12	STEP	C	4446	626.96	0	0	0.07	0	0.573	
16	FALSE	3/26/2018	01:19.7	3/26/2018	11:46.6	CATEST7	A	E129PGM6	13	STEP	C	5073	373.49	0	0	0	0	0.056	
17	FALSE	3/26/2018	11:46.6	3/26/2018	12:00.3	CADUM3	A	E229PRG4	14	STEP	C	5073	585.48	0	0	0.00	0	0.651	
18	FALSE	3/26/2018	14:01.0	3/26/2018	33:34.8	CADUM4	A	E259PGM6	15	STEP	C	9407	580.75	0	0	0.03	0	0.685	
19	FALSE	3/26/2018	24:01.0	3/26/2018	33:34.8	CADUM4	A	E269PRG4	16	STEP	C	9998	69.39	0	0	0.04	0	0.339	
20	FALSE	3/26/2018	33:34.8	3/26/2018	34:51.2	CATEST8	A	G26TPRG9	17	STEP	C	10057	1.07	0	0	0.02	0	0.056	
21	FALSE	3/26/2018	34:51.2	3/26/2018	34:52.2	CADUM2	A	E229PRG3	18	STEP	C	10058	14.67	0	0	0.04	0	0.019	
22	FALSE	3/26/2018	34:52.2	3/26/2018	35:05.9	CATEST9	A	SORT	19	STEP	C	10073	0.66	0	0	0.02	0	0.075	
23	FALSE	3/26/2018	35:05.9	3/26/2018	35:07.6	CAIOB8	A	SORT	20	STEP	C	10073	0.41	0	0	0.02	0	0.097	
24	FALSE	3/26/2018	35:07.6	3/26/2018	35:08.0	CADUM9	A	E79PRG10	21	STEP	C	10074	2.48	0	0	0.04	0	0.064	
25	FALSE	3/26/2018	35:08.0	3/26/2018	35:10.5	CAIOB8	A	E79PRG10	22	STEP	C	10076	20.12	0	0	0.03	0	0.280	
26	FALSE	3/26/2018	35:10.5	3/26/2018	36:30	CADUM4	A	E79PRG10											

Figure 4 Jobs and Steps information shown in CSV format

Not all jobs listed here contain COBOL. You can remove known non-COBOL programs, such as, FTP, IEBCOPY, SORT, etc., tasks in job class STC, and TSO clients in job class TSU. The specific job class names may be different if you have customized them from the IBM defaults.

Key	Batch	Start Date	Job Name	Program Name	Step Num	Sub Type	Job Class	Initiator	Dclapsed Tir	CPU Time	Queue
2	FALSE	3/26/2018	TETEST2	I147PRG9	1	JOB	TSU	0	23494.38	0.52	
3	FALSE	3/26/2018	TEJOB4	I147PRG9	1	STEP	TSU	0	23494.38	0.52	
4	FALSE	3/26/2018	DMCASE8	I176PRG7	1	JOB	TSU	0	25231.56	0.29	
5	FALSE	3/26/2018	DMCASE3	I176PRG7	1	STEP	TSU	0	25231.55	0.29	
6	FALSE	3/26/2018	OTTEST5		1	JOB	TSU	0	42423.11	0.26	
7	FALSE	3/26/2018	OTJOB3	I149PGM8	1	STEP	TSU	0	42423.09	0.26	
8	FALSE	3/26/2018	CICASE9		1	JOB	STC	1	86115.08	7627.22	Z53
9	FALSE	3/26/2018	CIDUM4	D38PGM4	1	STEP	STC	1	86115.08	7627.22	Z53
10	FALSE	3/25/2018	CTEST9	D249PGM4	1	JOB	STC	0	158420.5	1332.75	365
11	FALSE	3/25/2018	DBCASE4	D249PGM4	1	STEP	STC	0	158420.5	1332.75	365
12	FALSE	3/25/2018	DBCASE8	D149PRG2	1	JOB	STC	0	74420.15	798.06	242
13	FALSE	3/25/2018	CUW98	D149PRG2	1	STEP	STC	0	74420.15	798.06	242
14	FALSE	3/26/2018	CUW97	I201PRG7	2	JOB	STC	0	86398.9	27.38	Z53
15	FALSE	3/26/2018	CIDUM4	S118PGM7	1	STEP	STC	0	0	0.01	
16	FALSE	3/26/2018	CIDUM4	D67PRG7	2	STEP	STC	0	86398.9	27.37	Z53
17	FALSE	3/29/2018	DBCASE4	I101PRG7	1	JOB	STC	0	511541.2	18.73	Z12
18	FALSE	3/25/2018	DBCASE2	D226PGM4	1	STEP	STC	0	511541.2	18.73	Z12
19	FALSE	3/25/2018	DBTEST3	I101PRG7	1	JOB	STC	1	531871.6	15.42	Z12
20	FALSE	3/25/2018	DBTEST1	D133PGM4	1	STEP	STC	1	531871.6	15.42	Z12
21	FALSE	3/26/2018	DBTEST9		1	JOB	STC	0	211.99	7.5	Z12
22	FALSE	3/26/2018	DUJOB4	S111PGM2	1	STEP	STC	0	211.99	7.5	Z12
23	FALSE	3/26/2018	NGCASE7	I161PRG2	1	JOB	STC	0	50.45	3.87	
24	FALSE	3/26/2018	NGCASE6	N279PRG5	1	STEP	STC	0	50.44	3.87	
25	FALSE	3/26/2018	DBDUM4	I176PRG6	1	JOB	STC	1	64800.30	1.00	Z12
26	FALSE	3/26/2018	NDACF2	I1230PRG4	1	STEP	STC	1	64800.20	1.00	Z12

Figure 5a) Jobs in Jobs Class TSU and STC are highlighted

Key	Batch	Start Date	Job Name	Program Name	Step Num	Sub Type	Job Class	Initiator	Dclapsed Tir	CPU Time	Queue
138	Key	Batch	Start Date	Job Name	Program Name	Step Num	Sub Type	Job Class	Initiator	Dclapsed Tir	CPU Time
139	138	3/26/2018	CACASE8	A	I147PRG9	2	STEP	STC	16	1.46	0.05
140	139	3/26/2018	CACASE4	A	I147PRG9	3	STEP	STC	17	1.01	0.05
141	140	3/26/2018	CACASE10	A	I147PRG9	4	STEP	STC	18	0.92	0.05
142	141	3/26/2018	CAOJB10	A	I147PRG9	5	STEP	STC	19	1.02	0.05
143	142	3/26/2018	CAEST15	A	I147PRG9	34	STEP	STC	18998	2.08	0.09
144	143	3/26/2018	CAOJB3	A	I147PRG9	35	STEP	STC	19000	20.26	0.24
145	144	3/26/2018	CACASE3	A	I147PRG9	36	STEP	STC	19020	0.93	0.06
146	145	3/26/2018	CACASE3	A	I147PRG9	37	STEP	STC	19021	1.12	0.07
147	146	3/26/2018	CAOJB3	A	I147PRG9	38	STEP	STC	19022	1	0.07
148	147	3/26/2018	CAOJB17	A	I147PRG9	39	STEP	STC	19023	0.77	0.06
149	148	3/26/2018	MPDUM7	A	I147PRG9	4	STEP	B	2	0.01	0
150	149	3/26/2018	CACASE6	A	I147PRG9	5	STEP	B	3	0	0
151	150	3/26/2018	CAOJB7	A	I147PRG9	6	STEP	B	4	0.72	0.06
152	151	3/26/2018	CACASE7	A	I147PRG9	17	STEP	A	9649	8.53	0.12
153	152	3/26/2018	CACASE7	A	I147PRG9	18	STEP	A	9658	3.9	0.06
154	153	3/26/2018	CAOJB8	A	I147PRG9	19	STEP	A	9662	1.23	0.06
155	154	3/26/2018	CADUM7	A	I147PRG9	20	STEP	A	9663	0.91	0.06
156	155	3/26/2018	CADUM2	A	G267PRG9	17	STEP	C	10057	1.07	0.06
157	156	3/26/2018	SLDUM3	A	I100PRG3	5	STEP	Q	31	0.25	0.02
158	157	3/26/2018	DMCASE5	A	I100PRG8	2	STEP	G	2	0.45	0.02
159	158	3/26/2018	DMOB10	A	I107PGM9	2	STEP	G	7	57.5	0.01
160	159	3/26/2018	DMCASE9	A	I107PGM9	3	STEP	G	4	0.03	0
161	160	3/26/2018	DMOB10	A	I107PGM8	6	STEP	G	2	0.32	0.02
162	161	3/26/2018	DMOB10	A	I110PGM6	4.5	STEP	A	4	0.24	0.07
163	162	3/26/2018	DMOB10	A	I110PGM6	4.5	STEP	A	1	1.74	1.11

Figure 5b) Programs with name FTP are highlighted

- Remove all columns except “Job Name”, “Program Name” and “CPU Time”. For any repeated job-program name pairs, the CPU Time should be added together.

A	B	C
M	N	O
1	Job Name	Program N
2	CACASE1	E104PGM4
3	CACASE10	U40PGM8
4	CACASE3	I154PGM3
5	CACASE6	E119PGM6
6	CACASE7	E161PRG2
7	CACASE8	S290PRG3
8	CACASE9	D236PRG9
9	CADUM1	E176PGM6
10	CADUM10	D97PRG8
11	CADUM10	I173PGM1
12	CADUM10	S210PRG6
13	CADUM2	E31PRG10
14	CADUM2	E50PRG7
15	CADUM2	G267PRG9
16	CADUM3	D269PGM6
17	CADUM3	D269PGM8
18	CADUM4	D74PGM6
19	CADUM4	E252PGM4
20	CADUM4	S168PRG1
21	CADUM5	D257PRG3
22	CADUM5	D88PRG1
23	CADUM5	I178PRG4
24	CADUM5	I255PGM8

Figure 6 Job-program pair with repeated entry

3. Sort the job-program list by “CPU Time” column from largest to smallest to see the top contributors.

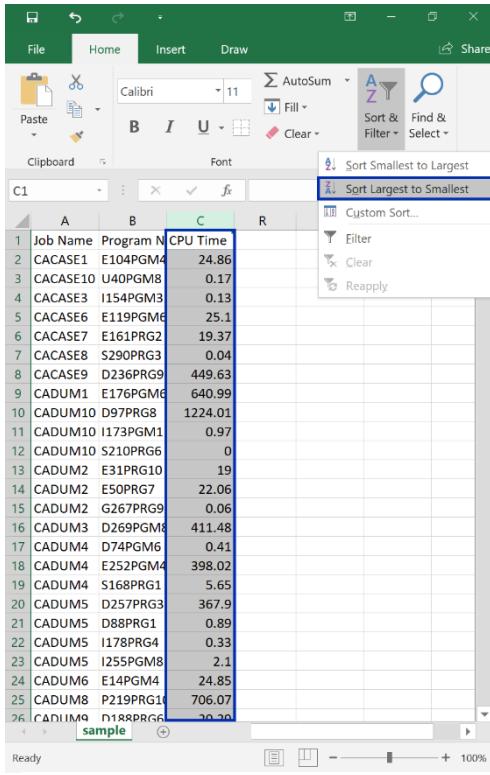


Figure 7a) Sort the final list based on CPU time

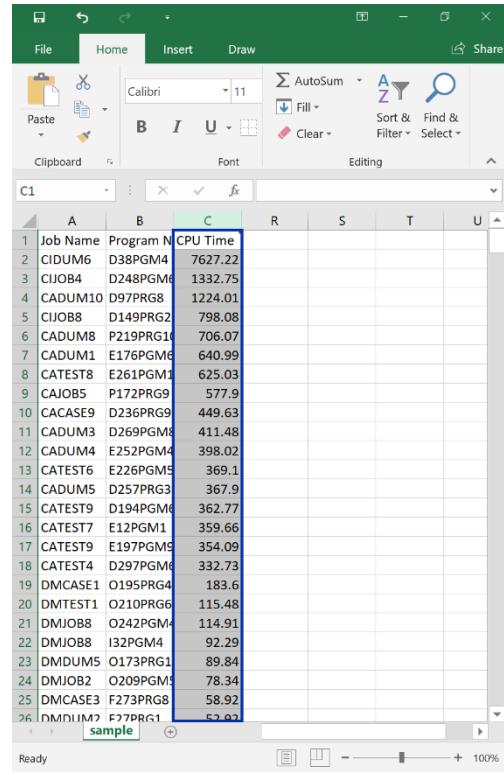


Figure 7b) Final job-program pairs with most CPU time

Now you have your top contributor job-program list sorted from most to least CPU time used. This list, however, may contain thousands of programs. To focus on the job-program pairs that contribute the bulk of the CPU time, compute the cumulative percentage of CPU time taken at each point in the list.

In the example above the first 15 candidates take 90.65% of the CPU time for all chosen candidates and 41.80% of the CPU time across all job-program pairs (that is, including all the non-COBOL programs previously filtered out). Therefore, it is sufficient to focus on the top 15 candidates from this point on.

Generate a CSV file containing the top job-program pairs you have chosen and copy it to your mainframe as the input for the next step.

```
CIDUM6,D38PGM4
CIJOB4,D248PGM6
CADUM10,D97PRG8
CIJOB8,D149PRG2
CADUM8,P219PRG11
CADUM1,E176PGM6
CATEST8,E261PGM1
CAJOB5,P172PRG9
...

```

Figure 8 Example file to be copied using FTP to your mainframe

# Identifying top CPU consuming modules from job-program pairs

Now it's time to find the top CPU consuming modules used by the selected job-program pairs. This step requires a performance analysis tool such as [IBM Application Performance Analyzer for z/OS \(APA\)](#). If you do not have access to APA or any performance analysis tools, ABO includes a tool, called Run Time Instrumentation (RTI) Profiler, that can help determine the modules that are executed the most while your application is running. For more information on RTI Profiler, please go to [ABO's User Guide](#).

The following steps outline how you can identify all modules used by the top contributing jobs and programs using APA.

**Tip:** We have provided REXX script samples to help with this process for those who use APA. For more information, please go to <https://www.ibm.com/support/pages/node/742651>

1. Create an APA batch measurement request using the top contributor job-program list and run this batch job before the peak usage time you have previously identified.
  - Use `JOBNAME` parameter to specify the job name and `STEP` parameter to specify the corresponding program name of a single job-program pair.
  - Use the `RUNAGAIN` parameter for continuous monitoring of repeated job-programs.

```
//CAZBATCH EXEC PGM=CAZBATCH,PARM='STCID=CAZO'  
//STEPLIB DD DISP=SHR,DSN=APA.APA14G.SCAZAUTH  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
NEW JOBNAME=CIDUM6 STEP=(,D38PGM4) ACTIVE=NO  
    RUNAGAIN=(180,FOR=99) RUNTOEOS=YES ;  
NEW JOBNAME=CIJOB4 STEP=(,D248PGM6) ACTIVE=NO  
    RUNAGAIN=(180,FOR=99) RUNTOEOS=YES ;  
NEW JOBNAME=CADUM10 STEP=(,D97PRG8) ACTIVE=NO  
    RUNAGAIN=(180,FOR=99) RUNTOEOS=YES ;  
NEW JOBNAME=CIJOB8 STEP=(,D149PRG2) ACTIVE=NO  
    RUNAGAIN=(180,FOR=99) RUNTOEOS=YES ;  
NEW JOBNAME=CADUM8 STEP=(,P219PRG1) ACTIVE=NO  
    RUNAGAIN=(180,FOR=99) RUNTOEOS=YES ;  
NEW JOBNAME=CADUM1 STEP=(,E176PGM6) ACTIVE=NO  
    RUNAGAIN=(180,FOR=99) RUNTOEOS=YES ;  
NEW JOBNAME=CATEST8 STEP=(,E261PGM1) ACTIVE=NO  
    RUNAGAIN=(180,FOR=99) RUNTOEOS=YES ;  
NEW JOBNAME=CAJOB5 STEP=(,P172PRG9) ACTIVE=NO  
    RUNAGAIN=(180,FOR=99) RUNTOEOS=YES ;  
/*
```

2. Create two APA batch jobs to print the following reports and run these jobs after the peak usage time you have previously identified:
  - Batch job 1: To print the S03 “Load Module Summary” report
  - Batch job 2: To print the C08 “CPU Usage Referred Attribution” and C02 “CPU Usage by Module” reports

S03 report creation job sample is shown below.

```
//CAZBATCH EXEC PGM=CAZBATCH,PARM='STCID=CAZ0'  
//CAZPRNT1 EXEC PGM=CAZPRINT  
//STEPLIB DD DISP=SHR,DSN=APA.APA14G.SCAZAUTH  
//SFILE01 DD DISP=SHR,DSN=APA.YOURID.R0708.CIDUM6.SF  
//SFILE02 DD DISP=SHR,DSN=APA.YOURID.R0709.CIJOB4.SF  
//SFILE03 DD DISP=SHR,DSN=APA.YOURID.R0710.CADUM10.SF  
//SFILE04 DD DISP=SHR,DSN=APA.YOURID.R0711.CIJOB8.SF  
//SFILE05 DD DISP=SHR,DSN=APA.YOURID.R0712.CADUM8.SF  
//SFILE06 DD DISP=SHR,DSN=APA.YOURID.R0713.CADUM1.SF  
//SFILE07 DD DISP=SHR,DSN=APA.YOURID.R0714.CATEST8.SF  
//SFILE08 DD DISP=SHR,DSN=APA.YOURID.R0715.CAJOB5.SF  
//CAZLOG DD SYSOUT=*  
//PRINT1 DD DSN=&SYSUID..APA.PRINTS03, UNIT=SYSDA, /* print output */  
// SPACE=(CYL,(5,5),RLSE),DCB=(LRECL=255,RECFM=VB,BLKSIZE=0),DISP=(,CATLG)  
//CAZCTL DD *  
 PROFILE 01 INPUT=SFILE01  
 SECTION S03 PROFILE=01  
 PROFILE 02 INPUT=SFILE02  
 SECTION S03 PROFILE=02  
 PROFILE 03 INPUT=SFILE03  
 SECTION S03 PROFILE=03  
 ...  
 PROFILE 07 INPUT=SFILE07  
 SECTION S03 PROFILE=07  
 PROFILE 08 INPUT=SFILE08  
 SECTION S03 PROFILE=08  
 PRINT DDNAME=PRINT1
```

C08/C02 combined report creation job sample is shown below.

```
//CAZBATCH EXEC PGM=CAZBATCH,PARM='STCID=CAZ0'  
//CAZPRNT1 EXEC PGM=CAZPRINT  
//STEPLIB DD DISP=SHR,DSN=APA.APA14G.SCAZAUTH  
//SFILE01 DD DISP=SHR,DSN=APA.YOURID.R0708.CIDUM6.SF  
//SFILE02 DD DISP=SHR,DSN=APA.YOURID.R0709.CIJOB4.SF  
//SFILE03 DD DISP=SHR,DSN=APA.YOURID.R0710.CADUM10.SF  
//SFILE04 DD DISP=SHR,DSN=APA.YOURID.R0711.CIJOB8.SF  
//SFILE05 DD DISP=SHR,DSN=APA.YOURID.R0712.CADUM8.SF  
//SFILE06 DD DISP=SHR,DSN=APA.YOURID.R0713.CADUM1.SF  
//SFILE07 DD DISP=SHR,DSN=APA.YOURID.R0714.CATEST8.SF  
//SFILE08 DD DISP=SHR,DSN=APA.YOURID.R0715.CAJOB5.SF  
//CAZLOG DD SYSOUT=*  
//PRINT1 DD DSN=&SYSUID..APA.PRINTC08, UNIT=SYSDA, /* print output */  
// SPACE=(CYL,(5,5),RLSE),DCB=(LRECL=255,RECFM=VB,BLKSIZE=0),DISP=(,CATLG)  
//CAZCTL DD *  
 PROFILE 01 INPUT=SFILE01  
 SECTION C08 PROFILE=01  
 SECTION C02 PROFILE=01  
 PROFILE 02 INPUT=SFILE02  
 SECTION C08 PROFILE=02  
 SECTION C02 PROFILE=02  
 ...  
 PROFILE 10 INPUT=SFILE10  
 SECTION C08 PROFILE=10  
 SECTION C02 PROFILE=10  
  
 PRINT DDNAME=PRINT1
```

The S03 report shows the name and location for all statically and dynamically called programs and displays each module's location within a Load Library. Below is a sample report for job CADUM8.

S03: Load Module Summary (00712/CADUM8)							
Module	Locn	Address	Count	Size(bytes)	Attributes	DDName	Load Library
CALLEE1	JPA	268DAC48	1	5,048		STEPLIB	HLQ.SAMPLE.COBPGM.INLOAD
C612GHP1	JPA	268FAC64	1	68,048		STEPLIB	HLQ.SAMPLE.COBPGM.INLOAD
CALLEE2	JPA	268DCC48	1	5,048		STEPLIB	HLQ.SAMPLE.COBPGM.INLOAD
PGNSWJ13	JPA	268FBC64	1	38,048		STEPLIB	HLQ.SAMPLE.COBPGM.INLOAD
CALLEE3	JPA	268DEC48	1	5,048		STEPLIB	HLQ.SAMPLE.COBPGM.INLOAD
CEEINIT	JPA	00007D88	1	45,688	RU RN	STEPLIB	TSCTEST.CEEZ220.SCEERUN
CEEPLPKA	JPA	26611780	1	2,181,248	RU RN	STEPLIB	TSCTEST.CEEZ220.SCEERUN
IEAVEWAT	NUC	00FF2500	1	4,556			
IEWFETCH	NUC	0165CE20	1	7,632			
IGWCCA00	PLPA	0A57B000	1	177,824			
IGZCEV5	JPA	26826AB0	1	17,744	RU RN	STEPLIB	TSCTEST.CEEZ220.SCEERUN
IGZCPAC	JPA	2682B8E0	1	448,288	RU RN	STEPLIB	TSCTEST.CEEZ220.SCEERUN
I419PRGB	JPA	266008B8	1	5,960		STEPLIB	HLQ.SAMPLE.COBPGM.INLOAD

The C08 report displays each module's relative contribution to overall job CPU.

C08: CPU Usage Referred Attribution (00712/CADUM8)			
Name	Description	Percent of CPU Time *	10.00% +/-3.9%
C612GHP1	Application Program	30.67 =====	*.....1....2....3....4....5....6....7....8....9....*
> C612GHP1	CSECT in C612GHP1	30.67 =====	
> 00074C	Attribution Offset in C612GHP1	30.67 =====	
> IGZCPAC	COBPACK	30.67 *****	
> IGZCXDI	Double precision division	29.29 *****	
> IGZCXMU	Double precision multiplication	1.38 *	
CALLEE1	Application Program	23.15 =====	
> CALLEE1	CSECT in CALLEE1	23.15 =====	
> 00050C	Attribution Offset in CALLEE1	23.15 =====	

The C02 report displays a module's relative contribution to overall job CPU time as in C08. But, a module with low system services activity can be in both C02 and C08, or in C02 report only.

C02: CPU Usage by Module (00712/CADUM8)

Name	Description	Percent of CPU Time * 10.00% +/- 6.3%
C612GHP1	Application Program	100.00 ****
> C612GHP1	CSECT in C612GHP1	100.00 ****

3. From the C02 and C08 reports for each job-program pair, take the line for each module with “Application” in its description and create the combined report shown below that lists the modules that potentially could be COBOL programs.

I419PRGB	Application Program	22.00 =====
C612GHP1	Application Program	99.46 =====
PGNSWJ13	Application Program	30.67 =====
CALLEE1	Application Program	23.15 =====
CALLEE3	Application Program	23.15 =====
CALLEE2	Application Program	23.00 =====
CALLEE6	Application Program	90.24 =====
CALLEE4	Application Program	2.43 =
CALLEE5	Application Program	5.62 ==
I210017A	Application Program	0.60
C612GHP1	Application	100.00 ****
I419PRGB	Application	7.40 ***

Modules that contribute very little to the overall time (e.g. CALLEE4 at 2.43% and I210017A at 0.60%) can be excluded if desired as collectively they take only about 3% of the overall time.

Next, cross-reference the module list above against the S03 location information to determine the fully qualified location to be scanned for COBOL CSECTs. For example, CALLEE5 is located at HLQ.SAMPLE.COBPGM.INLOAD(CALLEE5).

# Identifying top CPU consuming COBOL CSECTs from modules

The top CPU contributing modules may or may not contain COBOL CSECTs (compiled programs). There are several tools available to scan your modules to identify the COBOL CSECTs.

IBM File Manager for z/OS can be used to identify the COBOL CSECTs and the build compiler version in each module. The File Manager's "View Load Module" function can be used from ISPF on-line, TSO or REXX exec to display compiler information.

Address	CSECT name	Type	Size	Class	AMODE	RMODE	Date	Compiler	1
*	*	*	*	*	*	*	*	*	*
<---+-->	<----+---10---+-->	<-->	<---+-->	<----+---->	<--->	<--->	<---+-->	<----+----10---+----2----+-->	
0000000	CALLEE5	SD	0000CC8	B_TEXT	MIN	ANY	2010.027	Enterpr.COBOL for z/OS V4R2	

ABO contains a built-in SCAN=Y mode that can also display the CSECTs eligible for optimization. List the modules to be scanned as input to ABO using the SCAN=Y option as shown.

```
//SYSIN DD *
SCAN=Y
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(I419PRGB)
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(PGNSWJ13)
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE1)
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE2)
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE3)
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(C612GHP1)
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE5)
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE6)
```

ABO will display the eligible COBOL CSECTs in each module. In this example, module I419PRGB contains no COBOL CSECTs that are eligible for optimization, so it can be removed from the final candidate list of modules to optimize with ABO.

```
5697-AB1 IBM Automatic Binary Optimizer for z/OS 1.3.0

===== Monday Oct 20 2018 ======
12:12:39 Optimizer build level: tr_r17_binopt_20180924_141188 (Sept 24 2018 14:05:42)
12:12:39 Processing HLQ.SAMPLE.COBPGM.INLOAD, member CALLEE5
Language ID Records:
  id 5655S7100 v42 m00 2010027 resident CALLEE5
  Enterprise COBOL V4: start=0x0, length=3.20 (kBytes)
    Signature information bytes:
      a0487d4c 20000000 00880100 00000040
      08000000 000000 000008004 1400
12:12:39 Processing HLQ.SAMPLE.COBPGM.INLOAD, member I419PRGB
Language ID Records:
  id 569623400 v01 m06 2018005 resident I419PRGB
  BOZ4101: No applicable COBOL code section found
...
12:12:40 Exiting with return code: 0
```

With the non-COBOL module I419PRGB removed the final list of the top contributing COBOL modules is below.

```
HLQ.SAMPLE.COBPGM.INLOAD(PGNSWJ13)
HLQ.SAMPLE.COBPGM.INLOAD(CALLEE1)
HLQ.SAMPLE.COBPGM.INLOAD(CALLEE2)
HLQ.SAMPLE.COBPGM.INLOAD(CALLEE3)
HLQ.SAMPLE.COBPGM.INLOAD(C612GHP1)
HLQ.SAMPLE.COBPGM.INLOAD(CALLEE5)
HLQ.SAMPLE.COBPGM.INLOAD(CALLEE6)
```

This list, in the format shown below, is ready to be used as input to ABO to optimize these modules to the specified OUT=DD:SYSBOUT location.

```
//SYSIN DD *
ARCH=12
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(PGNSWJ13) OUT=DD:SYSBOUT
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE1) OUT=DD:SYSBOUT
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE2) OUT=DD:SYSBOUT
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE3) OUT=DD:SYSBOUT
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(C612GHP1) OUT=DD:SYSBOUT
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE5) OUT=DD:SYSBOUT
BOPT IN=HLQ.SAMPLE.COBPGM.INLOAD(CALLEE6) OUT=DD:SYSBOUT
```

If instead you are looking for to migrate your build compiler to the latest version of Enterprise COBOL, locate the corresponding source files for the COBOL CSECTs. Many source code management tools allow you to map modules and CSECTs back to the corresponding source file locations.

## Summary

Migrating to the latest version of the Enterprise COBOL compiler or optimizing existing COBOL modules with ABO is an effective way to reduce CPU consumption, lower operating costs and shorten batch windows. Your applications might contain thousands of program modules; therefore, it is advantageous to target top CPU contributing modules for migration or optimization first. This document provides a step-by-step guide to help identify top CPU consuming COBOL modules in your applications. The process is broken down into four major steps: 1) identify peak usage time, 2) identify top CPU consuming job-program pairs, 3) identify top CPU consuming modules, and 4) identify COBOL CSECTs from these modules.

## List of resources

How to Identify Top CPU Consuming COBOL Modules video:  
<https://ibm.biz/cobol-top-modules-video>

IBM Z Identify Top COBOL Module Tools:  
<https://www.ibm.com/support/pages/node/742651>

IBM Enterprise COBOL for z/OS:  
<https://www.ibm.com/products/cobol-compiler-zos>

IBM Automatic Binary Optimizer for z/OS:  
<https://www.ibm.com/products/automatic-binary-optimizer-zos>

IBM Application Performance Analyzer for z/OS (APA):  
<https://www.ibm.com/us-en/marketplace/application-performance-analyzer>

IBM File Manager for z/OS:  
<https://www.ibm.com/us-en/marketplace/file-manager-for-zsystems>

IBM System z Batch Network Analyzer (zBNA):  
<http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS5132>