
GSUK Virtual Conference
= Mainframe@60 : the diamond Anniversary of Digital Dominance
virtual
conference
29 April - 1 May 2025

Tactics To Shrink Your IPL and z/OS Startup Times

David Stephens
Longpela Expertise / CPT Global

29 April 2025
Session 5A

GSUK Virtual Conference
= Mainframe@60 : the diamond Anniversary of Digital Dominance

IPL and z/OS Startup Times

- z/OS startup times I have seen:

	Time Taken (minutes)
Average	20

IPL and z/OS Startup Times

- z/OS startup times I have seen:

	Time Taken (minutes)
Average	20-30
Worst	90

IPL and z/OS Startup Times

- z/OS startup times I have seen:

	Time Taken (minutes)
Best	5
Average	20-30
Worst	90

- In this session, we will look at real-world ways of reducing IPL and system startup times.
- You will also see examples of the good and bad we've seen while working at many sites around the world.

Who is David Stephens

- Lead Systems Programmer at Longpela Expertise. Senior Consultant at CPT Global.
- z/OS systems geek since 1989.
- Mainframe consultant since 2009.
- Worked on many different mainframe sites around the world.

First: What Are We Talking About?

- The time it takes to start z/OS. For this presentation, I will divide it into three parts:

1. IPL (Initial Program Load)
    - Load z/OS nucleus
    - Start MASTER address space

From: Operator initiating IPL on HMC
To: Messages:

```
IOS128I IPL DEVICE: 0AB80  VOLUME: IPL001
IEA370I MASTER CATALOG SELECTED IS CATALOG.MCAT1
```

What Are We Talking About?

- The time it takes to start z/OS. For this presentation, I will divide it into three parts:

1. IPL
2. NIP (Nucleus Initialisation Program)
    - Get z/OS and associated subsystems up (security, network, JES etc.).

From: Messages:

```
IEA370I MASTER CATALOG SELECTED IS CATALOG.MCAT1
IEE252I MEMBER IEASYS00 FOUND IN SYS1.PARMLIB
```

To: Messages:

```
IKT007I TCAS ACCEPTING LOGONS
IKT005I TCAS IS INITIALIZED
```

What Are We Talking About?

- The time it takes to start z/OS. For this presentation, I will divide it into three parts:

1. IPL
2. NIP
3. Get applications up.
    - Start resource managers (Db2, Broadcom Datacom, Innovation IAM, Software AG Adabas, IMS etc.)
    - Start transaction managers (CICS, IMS, WASz etc.)
    - Start middleware (Zowe, MQ, z/OS Connect, IMS Connect, CICS TG, ibi IWay Connectors etc.)
    - Ready for batch.

- We talk about all three in this presentation.

An Example From z/OS Syslog

- Consider this z/OS production syslog output from the beginning of an IPL.

```
03:47:17.06          00000290  IEA101A SPECIFY SYSTEM PARAMETERS FOR z/OS 02.04.00 HBB77C0
03:47:38.29          00000290  IEE600I REPLY TO 00 IS;PROG=08
03:47:38.30          00000290  IEE252I MEMBER IEASYS00 FOUND IN SYS1.PARMLIB
```

Message asking operator for any parameter changes.

- LOAD parameter:

XXXXYYAZ

IMSI character set to A, P, S or T

Idea 1. Don’t Touch That Keyboard!

- Consider this z/OS production syslog output from the beginning of an IPL.

21 second delay

```
03:47:17.06          00000290  IEA101A SPECIFY SYSTEM PARAMETERS FOR z/OS 02.04.00 HBB77C0
03:47:38.29          00000290  IEE600I REPLY TO 00 IS;PROG=08
03:47:38.30          00000290  IEE252I MEMBER IEASYS00 FOUND IN SYS1.PARMLIB
```

- A 21 second delay while we wait for the operator to reply to the IEA101A command.

- Most sites I see do NOT require the IEA101A message.

Sidebar: z/OS Syslog

- The z/OS syslog (or Operlog) is the primary tool when looking at IPL and startup times.
- Today, includes messages from the NIP processing on.

```
X 0000000 SYSP     25040 01:07:46.35 SYSLOG   00000000  IEE042I SYSTEM LOG DATA SET INITIALIZED
NC0000000 SYSP     25040 01:06:36.66 INTERNAL 00000290  CONTROL M,UEXIT=N IEAVN701 - INTERNALLY ISSUED K M
N 0000000 SYSP     25040 01:06:20.96          00000290  IEA371I SYS1.IPLPARM ON DEVICE CB10 SELECTED FOR IPL PARAMETERS
N 0000000 SYSP     25040 01:06:20.96          00000290  IEA246I LOAD   ID 01 SELECTED
N 0000000 SYSP     25040 01:06:20.96          00000290  IEA246I NUCLST ID 01 SELECTED
N 0000000 SYSP     25040 01:06:20.96          00000290  IEA519I IODF DSN = SYS1.IODF01
N 0000000 SYSP     25040 01:06:20.96          00000290  IEA520I CONFIGURATION ID = SYS1    . IODF DEVICE NUMBER = CB10
N 0000000 SYSP     25040 01:06:20.96          00000290  IEA091I NUCLEUS 1 SELECTED
N 0000000 SYSP     25040 01:06:24.84          00000290  IOS128I IPL DEVICE: 0AB20  VOLUME: SYSRS1
N 0000000 SYSP     25040 01:06:24.84          00000290  IEA370I MASTER CATALOG SELECTED IS CATALOG.MCAT1
M 0000000 SYSP     25040 01:06:24.87          00000290  IEA009I SYMBOLIC DEFINITIONS WILL BE READ FROM: 011
E                                         011 00000290          IEASYM00
N 4000000 SYSP     25040 01:06:24.88          00000290  IEE252I MEMBER IEASYM00 FOUND IN SYS1.PARMLIB
N 4000000 SYSP     25040 01:06:24.88          00000290 *IEA247I USING IEASYS00 FOR z/OS 02.04.00 HBB77C0
N 4000000 SYSP     25040 01:06:24.89          00000290  IEE252I MEMBER IEASYS00 FOUND IN SYS1.PARMLIB
M 0000000 SYSP     25040 01:06:24.89          00000290  IEA007I STATIC SYSTEM SYMBOL VALUES 016
D                                         016 00000290   &SYSALVL.          = "2"
D                                         016 00000290   &SYSCLONE.         = "P1"
D                                         016 00000290   &SYSNAME.          = "SYSP"
D                                         016 00000290   &SYSOSLVL.         = "Z1020400"
D                                         016 00000290   &SYSPLEX.          = "PRODPLX1"
D                                         016 00000290   &SYSR1.            = "SYSRS1"
D                                         016 00000290   &CNMNETID.         = "PRDNET"
```

Timestamps to one-hundredth of a second.

End of IPL

NIP Starts

Idea 1. Don’t Touch That Keyboard!

- Another example early in the IPL:

Two units with the same volume serial number. Which to use?

```
03:46:25.89          00080290 *IEA213A DUPLICATE VOLUME 'VOL001' FOUND ON DEVICES 7388 AND FB88.
03:47:03.08          00000290  IEE600I REPLY TO 00 IS;7388
03:46:25.91          00000290 *IEA213A REPLY DEVICE NUMBER WHICH IS TO REMAIN OFFLINE
03:47:03.09          00000290  IEA313I DEVICE 7388 DISMOUNTED
03:47:03.10          00080290 *IEA213A DUPLICATE VOLUME 'VOL002' FOUND ON DEVICES 7389 AND FB89.
03:47:16.93          00000290  IEE600I REPLY TO 00 IS;7389
03:47:03.10          00000290 *IEA213A REPLY DEVICE NUMBER WHICH IS TO REMAIN OFFLINE
03:47:16.94          00000290  IEA313I DEVICE 7389 DISMOUNTED
```

Idea 1. Don’t Touch That Keyboard!

- Another example early in the IPL:

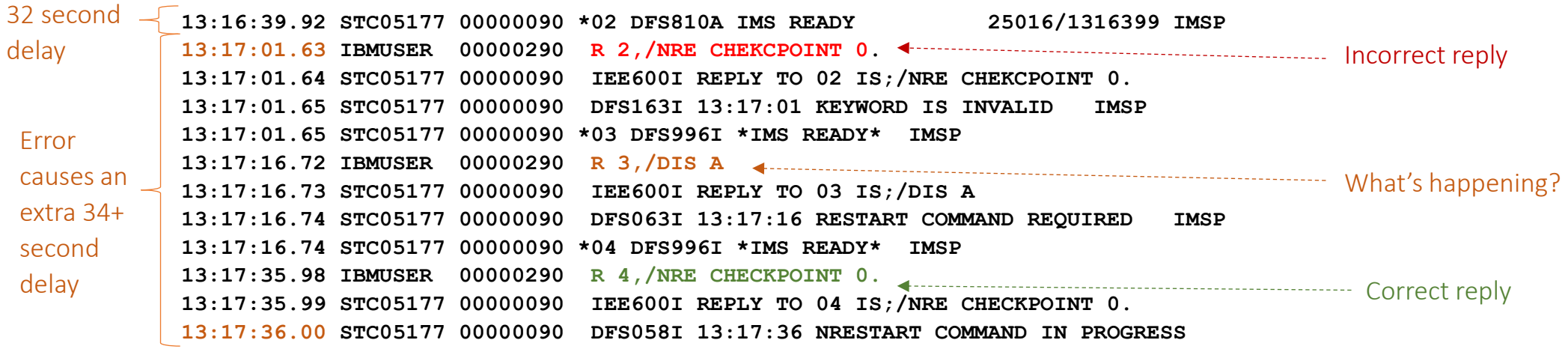
51 second delay

```
03:46:25.89          00080290 *IEA213A DUPLICATE VOLUME 'VOL001' FOUND ON DEVICES 6358 AND F128.
03:47:03.08          00000290  IEE600I REPLY TO 00 IS;6358
03:46:25.91          00000290 *IEA213A REPLY DEVICE NUMBER WHICH IS TO REMAIN OFFLINE
03:47:03.09          00000290  IEA313I DEVICE 6358 DISMOUNTED
03:47:03.10          00080290 *IEA213A DUPLICATE VOLUME 'VOL002' FOUND ON DEVICES 6359 AND F129.
03:47:16.93          00000290  IEE600I REPLY TO 00 IS;6359
03:47:03.10          00000290 *IEA213A REPLY DEVICE NUMBER WHICH IS TO REMAIN OFFLINE
03:47:16.94          00000290  IEA313I DEVICE 6359 DISMOUNTED
```

- Almost one minute delay. Easily resolved by changing I/O config.
- Most sites do not have these messages.
- Every time an operator must enter a command or reply to a message, introduces a delay between 3 seconds and minutes.

Idea 1. Don't Touch That Keyboard!

- Operator intervention introduces the possibility of human error.



- Problem could be worse (e.g. wrong response to IEA101A or IEA213A message seen earlier).

Idea 2. Leave It To Automation

The JES $VS command issues a z/OS console command.
These can be added to the end of JES parms to issue these commands when JES has started.

- Almost all sites have an automation package with features to perform an IPL from start to finish.
- CPT often seen sites that don't fully use this package.
- Example: often see sites using JES $VS commands in JES parms.

```
03:49:31.30 STC06338 00000080  $HASP120 INTRDR $VS,'S OAM' FROM STC06338
03:49:31.30 STC06338 00000080  $HASP120 INTRDR $VS,'S TSO' FROM STC06338
03:49:31.30 STC06338 00000080  $HASP120 INTRDR $VS,'S PRD1STC' FROM STC06338
03:49:31.30 STC06338 00000080  $HASP120 INTRDR $VS,'START APPC,SUB=MSTR' FROM STC06338
03:49:31.30 STC06338 00000080  $HASP120 INTRDR $VS,'START ASCH,SUB=MSTR' FROM STC06338
03:49:31.30 STC06338 00000080  $HASP120 INTRDR $VS,'S HZSPROC' FROM STC06338
```

Don't need JES (SUB=MSTR).
Could be started earlier

z/OS Health Checker can be started earlier from PARMLIB

Idea 2. Leave It To Automation

- Some sites have their own ‘tool’ to start applications.
- Examples:
  - IBM zPDT Application Starter
  - CBTTAPE File 623
  - Job/STC issuing $VS commands.
- Fine only if no automation software. Use one automation solution the best.

Idea 2. Leave It To Automation

- Configured properly, automation software will start something as soon as everything it needs is ready (e.g. CICS should start after CICS CMAS has started).
- Prevents issues like:

```
*IGW456I SMSVSAM ADDRESS SPACE INITIALIZATION IS WAITING FOR SMS ADDRESS SPACE TO BE MARKED AVAILABLE.
*IGW456I SMSVSAM ADDRESS SPACE INITIALIZATION IS WAITING FOR DFSMS COMPRESSION SERVICES TO BE MARKED AVAILABLE.
*BPXP022E ONE OR MORE JOBS ARE WAITING FOR UNIX SYSTEM SERVICES  AVAILABILITY.
IGD033I JOB ACF2 IS WAITING FOR THE SMS ADDRESS SPACE TO INITIALIZE
*BPXP006E OMVS IS WAITING FOR JOB ENTRY SUBSYSTEM INITIALIZATION
```

- These message may (or may not) add to z/OS startup time.
- Such error messages may also slow down problem resolution if there is an z/OS startup related error.

Idea 2. Leave It To Automation

- Example:
  - 03:46: IPL started
  - 03:49: TSO up
  - 03:49: Db2 DBP1 started
  - 04:12: MQP1 started
  - 04:12: CICS region started
  - 04:12: CICS region ready.

26 minute z/OS startup

Idea 2. Leave It To Automation

- Example:
  - 03:46: IPL started
  - 03:49: TSO up
  - 03:49: Db2 DBP1 started
  - 04:12: MQP1 started
  - 04:12: First CICS region started
  - 04:12: First CICS region ready.

IPL and NIP takes 3 minutes.

Db2 started immediately by automation

Idea 2. Leave It To Automation

- Example:
  - 03:46: IPL started
  - 03:49: TSO up
  - 03:49: Db2 DBP1 started
  - 04:12: MQP1 started
  - 04:12: First CICS region started
  - 04:12: First CICS region ready.

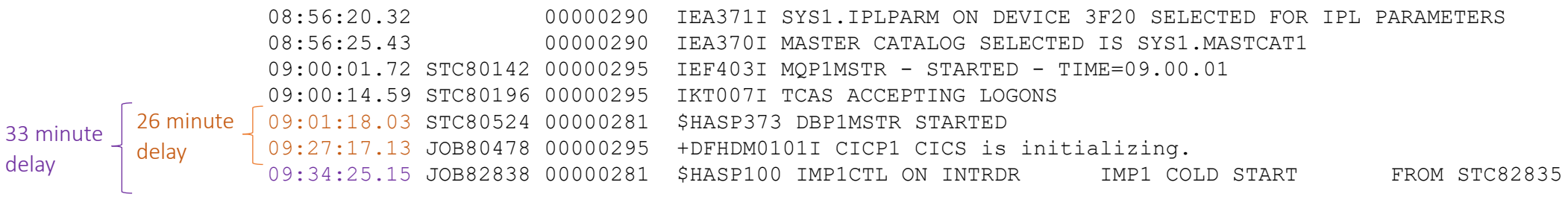
23 minute delay

MQ and CICS startup not performed by automation.

- Site did not use automation to start MQ or CICS: manually performed.
- Introduces a 23-minute delay in z/OS startup.

Idea 2. Leave It To Automation

- Many sites can reduce startup times by ‘tuning’ their automation rules (reviewing dependencies, messages, triggers).



- Example: CICS and IMS controlled by automation, but not started for 26/33 minutes after resource managers started.

Idea 2. Leave It To Automation

- Using automation, IPL times can be FAST.
- Leveraging and tuning automation startup will provide the biggest reduction in z/OS startup times.

Sample Startup Times:
(your times may be different)

System	Start Time (secs)
CICS AOR	25
IMS TM/DB Cold	19
VTAM	15
TCPIP	12
Db2 (MSTR+IRLM+DBM1+DIST)	12
MQ (MSTR+CHIN)	9

Most subsystems can startup in seconds

Sidebar: MTTR Redbook

- This presentation doesn't go into technical detail, nor cover everything..
- An excellent reference is the IBM Redbook MTTR.
- Written in 2010, the information is dated, and this book is now archived by IBM, but much of the content is still relevant.
- Good news: I won't repeat the book. I will talk about what we're seeing in the field, and things not included in the book.

Idea 3. Do Less

The z/OS PARMLIB COMMNDxx member includes z/OS commands to be issued once z/OS is ready.

- Many sites perform processing during IPL and startup that they don't need.
- Example: one site had 85 commands in COMMNDxx, including:
  - `2 x V XXXX-YYYY,AS,ON`
  - `16 x V PATH(AAAA,BB),OFFLINE`
  - `SET MPF=00`
  - `S DEALLOC`
  - `TRACE MT,256K`
  - `SET SMS=00`
  - `19 x Display commands`

Idea 3. Do Less

The z/OS PARMLIB COMMNDxx member includes z/OS commands to be issued once z/OS is ready.

- Many sites perform processing during IPL and startup that they don't need.
- Example: one site had 85 commands in COMMNDxx, including:
  - `2 x V XXXX-YYYY,AS,ON`
  - `16 x V PATH(AAAA,BB),OFFLINE`
  - `SET MPF=00`
  - `S DEALLOC`
  - `TRACE MT,256K`
  - `SET SMS=00`
  - `19 x Display commands`

Not needed with correct I/O config.

Idea 3. Do Less

The z/OS PARMLIB COMMNDxx member includes z/OS commands to be issued once z/OS is ready.

- Many sites perform processing during IPL and startup that they don't need.
- Example: one site had 85 commands in COMMNDxx, including:
  - `2 x V XXXX-YYYY,AS,ON`
  - `16 x V PATH(AAAA,BB),OFFLINE`
  - `SET MPF=00`
  - `S DEALLOC`
  - `TRACE MT,256K`
  - `SET SMS=00`
  - `19 x Display commands`

Reloading MPF rules: reloading existing rules (MPF member specified in IEASYSxx).

Idea 3. Do Less

The z/OS PARMLIB COMMNDxx member includes z/OS commands to be issued once z/OS is ready.

- Many sites perform processing during IPL and startup that they don't need.
- Example: one site had 85 commands in COMMNDxx, including:
  - `2 x V XXXX-YYYY,AS,ON`
  - `16 x V PATH(AAAA,BB),OFFLINE`
  - `SET MPF=00`
  - `S DEALLOC` ← STC running IEFBR14. Used to be used when varying devices offline. Not needed today.
  - `TRACE MT,256K`
  - `SET SMS=00`
  - `19 x Display commands`

Idea 3. Do Less

The z/OS PARMLIB COMMNDxx member includes z/OS commands to be issued once z/OS is ready.

- Many sites perform processing during IPL and startup that they don't need.
- Example: one site had 85 commands in COMMNDxx, including:
  - `2 x V XXXX-YYYY,AS,ON`
  - `16 x V PATH(AAAA,BB),OFFLINE`
  - `SET MPF=00`
  - `S DEALLOC`
  - `TRACE MT,256K` ◄---- Changing master trace size (can be set in PARMLIB SCHEDxx)
  - `SET SMS=00`
  - `19 x Display commands`

Idea 3. Do Less

The z/OS PARMLIB COMMNDxx member includes z/OS commands to be issued once z/OS is ready.

- Many sites perform processing during IPL and startup that they don't need.
- Example: one site had 85 commands in COMMNDxx, including:
  - `2 x V XXXX-YYYY,AS,ON`
  - `16 x V PATH(AAAA,BB),OFFLINE`
  - `SET MPF=00`
  - `S DEALLOC`
  - `TRACE MT,256K`
  - `SET SMS=00` ◄---------- Reloading SMS config (set in PARMLIB IGDSMSxx).
  - `19 x Display commands`

Idea 3. Do Less

The z/OS PARMLIB COMMNDxx member includes z/OS commands to be issued once z/OS is ready.

- Many sites perform processing during IPL and startup that they don't need.
- Example: one site had 85 commands in COMMNDxx, including:
  - `2 x V XXXX-YYYY,AS,ON`
  - `16 x V PATH(AAAA,BB),OFFLINE`
  - `SET MPF=00`
  - `S DEALLOC`
  - `TRACE MT,256K`
  - `SET SMS=00`
  - `19 x Display commands`

Having information in syslog is handy, but some display commands take a lot of resources.

- These all add processing during IPL, but aren't needed.

Idea 3. Do Less

z/OS tries to ‘mount’ DASD during IPL. If the DASD unit isn’t initialised, it can’t be mounted.

- Many sites have unused resources that slow down restart.
- Example: uninitialized DASD:

1.6 second delay.

```
03:46:24.26          00000290  IEA311I UNLABELED DASD ON 6C40. UNIT PUT OFFLINE
03:46:24.29          00000290  IEA311I UNLABELED DASD ON 6C60. UNIT PUT OFFLINE
03:46:24.30          00000290  IEA311I UNLABELED DASD ON 6C20. UNIT PUT OFFLINE
03:46:24.32          00000290  IEA311I UNLABELED DASD ON 6C80. UNIT PUT OFFLINE
03:46:24.33          00000290  IEA311I UNLABELED DASD ON 6CA0. UNIT PUT OFFLINE
03:46:24.35          00000290  IEA311I UNLABELED DASD ON 6CC0. UNIT PUT OFFLINE
03:46:24.36          00000290  IEA311I UNLABELED DASD ON 6C00. UNIT PUT OFFLINE
03:46:24.37          00000290  IEA311I UNLABELED DASD ON A97E. UNIT PUT OFFLINE
03:46:24.39          00000290  IEA311I UNLABELED DASD ON A95E. UNIT PUT OFFLINE
03:46:24.40          00000290  IEA311I UNLABELED DASD ON A99E. UNIT PUT OFFLINE
03:46:24.41          00000290  IEA311I UNLABELED DASD ON A93E. UNIT PUT OFFLINE
03:46:24.42          00000290  IEA311I UNLABELED DASD ON A9BE. UNIT PUT OFFLINE
03:46:24.43          00000290  IEA311I UNLABELED DASD ON A91E. UNIT PUT OFFLINE
03:46:24.45          00000290  IEA311I UNLABELED DASD ON A9DE. UNIT PUT OFFLINE
03:46:24.46          00000290  IEA311I UNLABELED DASD ON F9BE. UNIT PUT OFFLINE
03:46:24.47          00000290  IEA311I UNLABELED DASD ON F93E. UNIT PUT OFFLINE
03:46:24.49          00000290  IEA311I UNLABELED DASD ON F97E. UNIT PUT OFFLINE
03:46:24.50          00000290  IEA311I UNLABELED DASD ON F99E. UNIT PUT OFFLINE
03:46:24.51          00000290  IEA311I UNLABELED DASD ON F95E. UNIT PUT OFFLINE
03:46:24.52          00000290  IEA311I UNLABELED DASD ON F9DE. UNIT PUT OFFLINE
03:46:24.54          00000290  IEA311I UNLABELED DASD ON F91E. UNIT PUT OFFLINE
03:46:24.55          00000290  IEA311I UNLABELED DASD ON 6C41. UNIT PUT OFFLINE
03:46:24.56          00000290  IEA311I UNLABELED DASD ON 6C21. UNIT PUT OFFLINE
03:46:24.58          00000290  IEA311I UNLABELED DASD ON 6C61. UNIT PUT OFFLINE
03:46:24.59          00000290  IEA311I UNLABELED DASD ON 6CA1. UNIT PUT OFFLINE
03:46:24.60          00000290  IEA311I UNLABELED DASD ON 6C81. UNIT PUT OFFLINE
03:46:24.61          00000290  IEA311I UNLABELED DASD ON 6C01. UNIT PUT OFFLINE
03:46:24.62          00000290  IEA311I UNLABELED DASD ON 6CC1. UNIT PUT OFFLINE
03:46:24.64          00000290  IEA311I UNLABELED DASD ON A95F. UNIT PUT OFFLINE
03:46:24.65          00000290  IEA311I UNLABELED DASD ON A9BF. UNIT PUT OFFLINE
03:46:24.66          00000290  IEA311I UNLABELED DASD ON A93F. UNIT PUT OFFLINE
03:46:24.67          00000290  IEA656I EXCESSIVE NUMBER OF UNLABELED DASD FOUND - IEA311I SUPPRESSED
03:46:25.89
```

Idea 3. Do Less – Cleanup Resources

- Many sites have unused resources that slow down restart.
- Example:

3 min 23 second start time

```
23.59.54 STC87176  IEF403I CICSP1 - STARTED - TIME=23.59.54
00.02.15 STC87176  +DFHSI1511I CICSP1 Installing group list CICSP1G.
00.03.01 STC87176  +DFHLG0103I CICSP1 System log (DFHLOG) initialization has started.
00.03.17 STC87176  +DFHSI1517 CICSP1 Control is being given to CICS.
```

- Most CICS regions I see complete cold starts in less than 30 seconds.

- Old resources not removed (programs, files).
- Autoinstall not used (programs, terminals, Db2 resources)

Sidebar: Startup and Shutdown Messages

Look through this presentation for more messages I use.

- There is no single resource with the best z/OS syslog messages showing startup or shutdown. IBM MTTR Redbook has some.
- Some I have used:

CICS

```
23.59.54 STC87176  IEF403I CICSP1 - STARTED - TIME=23.59.54
00.03.17 STC87176  +DFHSI1517 CICSP1 Control is being given to CICS.
```

Db2

```
04:00:30.51 STC08864 00000281  $HASP100 DBP1MSTR ON STCINRDR
04:00:40.99 STC08864 00000090  DSN9022I  -DBP1 DSNYASCP 'START DB2' NORMAL COMPLETION
```

IMS

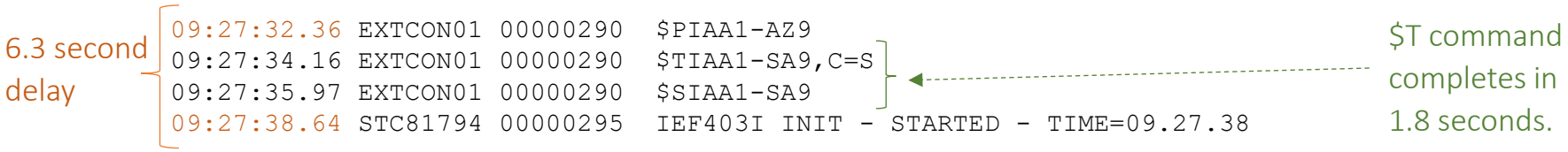
```
13:59:11.57 JOB39088 00000281  $HASP373 IMP1CTL STARTED - INIT SB9  - CLASS S        - SYS SYS1
13:59:23.82 JOB39088 00000090 *028 DFS810A IMS READY        2025074/1359238 IMP1CTL.STEP1    IMP1
```

MQ

```
13:38:19.88 STC38193 00000080  CSQY000I MQP1 IBM WebSphere MQ for z/OS V7.1.0
13:38:21.78 STC38193 00000291  CSQ9022I MQP1 CSQYASCP 'START QMGR' NORMAL COMPLETION
```

Idea 3. Do Less – Review Processing

- I often see sites doing things that may have been needed in the past, but aren't needed any longer.
- Example: this site stops 230+ JES2 initiators, changes the job class, and restarts them.



- Today, JES2 initiators can be changed without stopping them.

Idea 3. Do Less – Beware Long Running Transactions

CICS SMF Type 110 Records can be used to find transactions not syncpointing.

- Scenario: long running CICS transaction(s) performs many Db2 updates without syncpointing.
- CICS fails and restarts.
- CICS is unavailable for several minutes while CICS backs out changes.

- We have seen a few examples where long running transactions have caused issues.

Idea 3. Do Less

- Another example: ACF2 processing.
- 235 obsolete certificates.
- Delayed ACF2 startup a bit.

0.16 second delay

```
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - PBATCH.CERT
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - PBATCH.CERT2
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - PBATCH.CERT3
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - PBATCH.CERT4
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - PBATCH.CERT5
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - PCERT.CERT1
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - CERTAUTH.PRD
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - CERTAUTH.PRD1
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - CERTAUTH.PRD2
13:34:26.97          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - CERTAUTH.PRD3
        (and 200 more)
13:34:27.11          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.CERTA
13:34:27.11          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.CERTB
13:34:27.11          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.CERTC
13:34:27.11          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.CERTD
13:34:27.11          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.CERTE
13:34:27.11          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.CERTF
13:34:27.11          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.CERTG
13:34:27.11          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.CERTH
13:34:27.11          00000090  ACF79468 Certificate RCERT.TCERT1 is expiring within 30 days
13:34:27.12          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.XCERT1
13:34:27.12          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - RCERT.XCERT2
13:34:27.12          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - TN3270P.CERT1
13:34:27.12          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - TN3270P.CERT2
13:34:27.12          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT1.CERT
13:34:27.12          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT1.CERT
13:34:27.12          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT1.CERT2
13:34:27.12          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT1.CERT3
13:34:27.13          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT1.CERT4
13:34:27.13          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT1.CERT5
13:34:27.13          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT1.CERT2
13:34:27.13          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT1.CERT3
13:34:27.13          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT2.CERT2
13:34:27.13          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT2.CERT3
13:34:27.13          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT2.CERT4
13:34:27.13          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - VCERT3.CERT
13:34:27.13          00000090  ACF79464 EXPIRED CERTIFICATE DETECTED - XCERT.CERT
```

Idea 3. Do Less

- How about PARMLIB IEFSSNxx?
  - Statements are executed serially.
  - Often see subsystems defined that are never used.
  - May or may not add significant time to IPL.

0.04 seconds

```
03:48:20.42          00000090  CSQ3111I +MQA1    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.42          00000090  CSQ3110I +MQA1    CSQ3UR00 - SUBSYSTEM MQA1 INITIALIZATION COMPLETE
03:48:20.44          00000090  CSQ3111I +MQA2    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.44          00000090  CSQ3110I +MQA2    CSQ3UR00 - SUBSYSTEM MQA2 INITIALIZATION COMPLETE
03:48:20.44          00000090  CSQ3111I +MQA3    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.44          00000090  CSQ3110I +MQA3    CSQ3UR00 - SUBSYSTEM MQA3 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQA5    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQA5    CSQ3UR00 - SUBSYSTEM MQA5 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQA6    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQA6    CSQ3UR00 - SUBSYSTEM MQA6 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQA7    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQA7    CSQ3UR00 - SUBSYSTEM MQA7 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQA8    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQA8    CSQ3UR00 - SUBSYSTEM MQA8 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQA9    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQA9    CSQ3UR00 - SUBSYSTEM MQA9 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ11    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ11    CSQ3UR00 - SUBSYSTEM MQ11 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ12    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ12    CSQ3UR00 - SUBSYSTEM MQ12 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ15    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ15    CSQ3UR00 - SUBSYSTEM MQ15 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ16    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ16    CSQ3UR00 - SUBSYSTEM MQ16 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ17    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ17    CSQ3UR00 - SUBSYSTEM MQ17 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ18    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ18    CSQ3UR00 - SUBSYSTEM MQ18 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ19    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ19    CSQ3UR00 - SUBSYSTEM MQ19 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ20    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ20    CSQ3UR00 - SUBSYSTEM MQ20 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ21    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ21    CSQ3UR00 - SUBSYSTEM MQ21 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ22    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ22    CSQ3UR00 - SUBSYSTEM MQ22 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ26    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ26    CSQ3UR00 - SUBSYSTEM MQ26 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ27    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ27    CSQ3UR00 - SUBSYSTEM MQ27 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ31    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ31    CSQ3UR00 - SUBSYSTEM MQ31 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ32    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ32    CSQ3UR00 - SUBSYSTEM MQ32 INITIALIZATION COMPLETE
03:48:20.46          00000090  CSQ3111I +MQ41    CSQ3UR00 - EARLY PROCESSING PROGRAM IS V9.3.0    LEVEL
03:48:20.46          00000090  CSQ3110I +MQ41    CSQ3UR00 - SUBSYSTEM MQ41 INITIALIZATION COMPLETE
```

Idea 4. Start Early

- Starting processing as early as possible will reduce IPL and z/OS startup times.
- Tuning automation to start things early is obvious.
- A less obvious option: PARMLIB IEASYSxx:
  - `HZSPROC=xxxx`
  - `CSFPROC=xxxx`

  z/OS will automatically start z/OS Health Checker and ICSF as soon as possible.

- Many sites still start these in COMMNDxx or using automation.

Idea 4. Start Early - Parallel Processing

- Many sites serialise or delay their startups.
- Today with IPL boost, most sites will have many CPs to share the workload after IPL phase.
- Example: one site starting 10 CICS and 1 Db2 in 2.5 minutes.

Many CICS started around the same time.

2.5 minutes to start 10 CICS + 1 Db2

```
01:06:24.84          00000290 IEA370I MASTER CATALOG SELECTED IS CATALOG.MCAT
01:08:08.64 STC00059 00000291 IKT007I TCAS ACCEPTING LOGONS
01:09:29.71 STC00408 00000291 +DFHDM0101I CICSDT1G CICS is initializing.
01:09:29.71 STC00402 00000291 +DFHDM0101I CICSDD1G CICS is initializing.
01:09:29.72 STC00407 00000291 +DFHDM0101I CICSDD3G CICS is initializing.
01:09:29.75 STC00403 00000291 +DFHDM0101I CICSDD2G CICS is initializing.
01:09:31.62 STC00410 00000291 +DFHDM0101I CICSDT2G CICS is initializing.
01:09:31.66 STC00401 00000291 +DFHDM0101I CICSDA4C CICS is initializing.
01:09:31.68 STC00400 00000291 +DFHDM0101I CICSDA2C CICS is initializing.
01:09:33.34 STC00429 00000291 $HASP373 DDBCMSTR STARTED
01:09:33.67 STC00399 00000291 +DFHDM0101I CICSDA1G CICS is initializing.
01:09:33.70 STC00415 00000291 +DFHDM0101I CICSDT3G CICS is initializing.
01:09:37.66 STC00420 00000291 +DFHDM0101I CICSCJP3 CICS is initializing.
01:10:05.59 STC00420 00000291 +DFHSI1517 CICSCJP3 Control is being given to CICS.
01:10:13.09 STC00429 00000291  DSN9022I  + DSNYASCP 'START DB2' NORMAL COMPLETION
```

Idea 4. Start Early - Parallel Processing

- Many sites serialise or delay their startups.
- Today with IPL boost, most sites will have many CPs to share the workload after IPL phase.
- Example: one site starting 10 CICS and 1 Db2 in 2.5 minutes.

```
01:06:24.84          00000290 IEA370I MASTER CATALOG SELECTED IS CATALOG.MCAT
01:08:08.64 STC00059 00000291 IKT007I TCAS ACCEPTING LOGONS
01:09:29.71 STC00408 00000291 +DFHDM0101I CICSDT1G CICS is initializing.
01:09:29.71 STC00402 00000291 +DFHDM0101I CICSDD1G CICS is initializing.
01:09:29.72 STC00407 00000291 +DFHDM0101I CICSDD3G CICS is initializing.
01:09:29.75 STC00403 00000291 +DFHDM0101I CICSDD2G CICS is initializing.
01:09:31.62 STC00410 00000291 +DFHDM0101I CICSDT2G CICS is initializing.
01:09:31.66 STC00401 00000291 +DFHDM0101I CICSDA4C CICS is initializing.
01:09:31.68 STC00400 00000291 +DFHDM0101I CICSDA2C CICS is initializing.
01:09:33.34 STC00429 00000291 $HASP373 DDBCMSTR STARTED
01:09:33.67 STC00399 00000291 +DFHDM0101I CICSDA1G CICS is initializing.
01:09:33.70 STC00415 00000291 +DFHDM0101I CICSDT3G CICS is initializing.
01:09:37.66 STC00420 00000291 +DFHDM0101I CICSCJP3 CICS is initializing.
01:10:05.59 STC00420 00000291 +DFHSI1517 CICSCJP3 Control is being given to CICS.
01:10:13.09 STC00429 00000291  DSN9022I  + DSNYASCP 'START DB2' NORMAL COMPLETION
```

Db2 started after some CICS – CICS connect to Db2 when it is ready.

Starting many CICS at same time did not delay CICS startup (28 seconds).

Idea 4. Start Early - Parallel Processing

- However, be cautious when adding more processing to be performed at the same time.
- May be other issues.
- Example: too many z/OS commands issued in COMMNDxx.

```
03:48:20.75          00000090  IEE389I MVS COMMAND PROCESSING AVAILABLE
03:48:20.77          00000090  IEE822E COMMANDS ARE AT 80% OF LIMIT IN COMMAND CLASS M2

03:48:20.78          00000090 *IEE806A COMMANDS EXCEED LIMIT IN COMMAND CLASS M2
03:48:20.78          00000090  IEE061I COMMAND FLOODING RELIEVED IN COMMAND CLASS M2
```

Idea 4. Start Early - IPLDATA

- z/OS syslog doesn't give much information about IPL and NIPL processing.
- Good news: z/OS IPLDATA does.
- This information shows how long each part of the IPL took.

0.764 seconds to load the z/OS nucleus

```
IPLDATA STATUS
*** IPL Statistics ***

IEAIPL10  00:00:00.000  ISNIRIM - Read SCPINFO
IEAIPL20  00:00:00.000  Test Block storage to 2G
IEAIPL11  00:00:00.004  Fast FIND initialization
IEAIPL31  00:00:00.000  LOAD service initialization
IEAIPL30  00:00:00.000  Load IPLWTO. Allocate IPL Msg Q
IEAIPL46  00:00:00.296  Read SCHIBs into IPL workspace
IEAIPL49  00:00:00.000  Process Load and Default parameters
IEAIPL50  00:00:00.273  IPL parmlib - process LOADxx and NUCLSTxx
IEAIPL51  00:00:00.000  System architecture
IEAIPL43  00:00:00.002  Find and Open IODF data set
IEAIPL60  00:00:00.000  Read NCRs from IODF
IEAIPL70  00:00:00.022  UIM environment - load CBD and IOS services
IEAIPL71  00:00:00.013  Build DFT for each device
IEAIPL08  00:00:00.000  Read EDT information from IODF
IEAIPL40  00:00:00.013  Read MLTs from nucleus
IEAIPL42  00:00:00.001  Read NMLs from nucleus (IEANynnn modules)
IEAIPL41  00:00:00.156  Read PDS directory entries and CESD records
IEAIPL05  00:00:00.000  Build and sort NUCMAP
IEAIPL02  00:00:00.764  Load nucleus modules
IEAIPL04  00:00:00.005  Allocate PFT and SQA/ESQA
IEAIPL14  00:00:00.000  Build LSQA/ELSQA for Master
IEAIPL09  00:00:00.023  IAXMI - PFT, master RAB,  etc.
IEAIPL07  00:00:00.003  Update AMODE for nucleus resident SVCs
IEAIPL03  00:00:00.003  Build UCBs, ULUT, etc.
IEAIPL18  00:00:00.007  Copy and relocate EDT to ESQA
IEAIPL99  00:00:00.158  Page frame table and cleanup

Total IPL Time:  00:00:01.751
```

Idea 4. Start Early - IPLDATA

- It also includes information about NIP processing.

5.625 seconds to start XCF

```
*** NIP Statistics ***

IEAVNIP0  00:00:00.065  NIP Base
IEAVNIPM  00:00:00.032  Invoke NIP RIMs
        (lots more here)
IEAVNPSL  00:00:00.001  System Logger
IEAVNPF9  00:00:05.635  XCF
IEAVNP33  00:00:00.068  GRS
IEAVNPED  00:00:00.206  PROD
IEAVNPH8  00:00:00.002
IEAVNP26  00:00:01.987  SMS
IEAVNPE5  00:00:01.080  LNKLST
IEAVNPD5  00:00:00.231  Load pageable device support modules
IEAVNP88  00:00:00.054  Allocation move EDT II
IEAVNPCS  00:00:00.000  ICSF
IEAVNPA1  00:00:00.672  CONSOLE
IEAVNPDC  00:00:00.113  WLM
IEAVNP16  00:00:00.776  EXCP appendages
IEAVNP13  00:00:00.010  Prepare NIP/MSI interface
IEAVNP17  00:00:00.000  GTF Monitor Call interface
IEAVNP18  00:00:00.039  PARMLIB Scan Routine interface
IEAVNPFR  00:00:00.044  Function Registry
IEAVNPGT  00:00:00.001  Generalized Tracker
IEAVNPF2  00:00:00.037  Process IOS=
IEAVNP15  00:00:00.227  Process VATLST
IEAVNPRR  00:00:00.000  RRS
IEAVNPOE  00:00:00.141  USS
IEAVNPSC  00:00:00.000  SDC
IEAVNPLE  00:00:00.042  System LE
IEAVNPUN  00:00:00.113  Unicode
IEAVNPXL  00:00:00.020  zXML Parser
IEAVNPCI  00:00:00.023  IQP
IEAVNPHC  00:00:00.000  IBM Health Checker for z/OS
IEAVNPE7  00:00:00.020  Service Processor Interface CTRACE
IEAVNP1B  00:00:00.060  Close catalog
IEAVNIPX  00:00:00.000  Nip final cleanup

Total NIP Time:  00:00:42.718
```

Idea 4. Start Early - IPLDATA

- The IEAVIPL section has more NIP type information.

0.125 seconds to start the CONSOLE address space.

```
*** IEEVIPL Statistics ***

IEEMB845  00:00:00.000  CSCB Chain Manipulation
IEETRACE  00:00:00.000  Master trace
ISNMSI    00:00:01.152  SPI
UCMPECBM  00:00:00.125  CONSOLE address space
ENFPC005  00:00:00.000  CONSOLE ready ENF
IEFSCHIN  00:00:00.021  IEFSCHAS address space
IEFJSINT  00:00:00.012  Subsystem interface
IEFSJLOD  00:00:00.013  JESCT
IAZINIT   00:00:00.019  JESXCF address space
IAZFSII   00:00:00.055  FSI trace
IEFQBINT  00:00:00.010  SWA manager
IEFAB4I0  00:00:00.060  ALLOCAS address space

IEEVIPL   00:00:01.471      Uncaptured time:  00:00:00.000
```

Idea 4. Start Early - IPLDATA

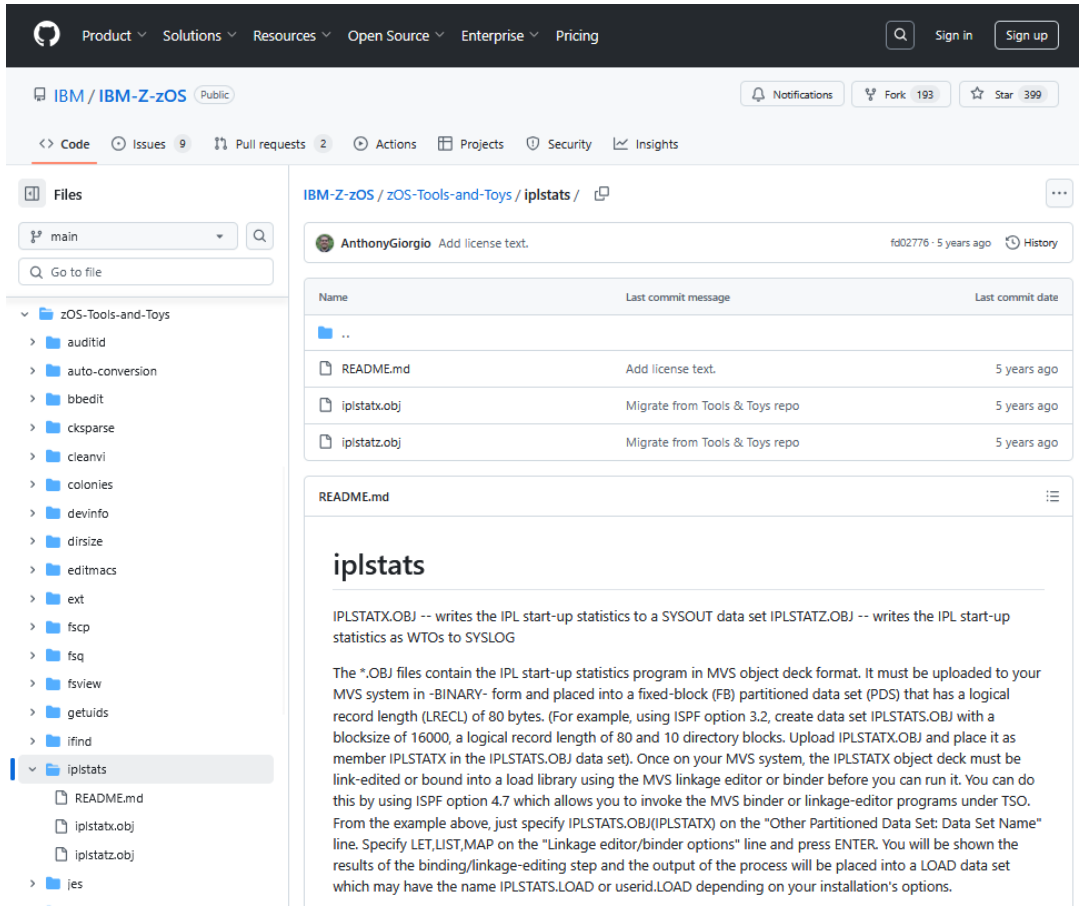
- IEEMB860 section includes z/OS master scheduler information (included in NIP for this presentation).

3.1 seconds to start SMF

```
*** IEEMB860 Statistics ***

IEAVTMSI  00:00:00.019  RTM
ILRTMRLG  00:00:00.522  ASM
IEEVMSI   00:00:00.010  Reconfiguration
IARM8MSI  00:00:00.015  RSM - bring storage online
IECVIOSI  00:00:01.541  IOS dynamic pathing
RACROUTE  00:00:00.000  Initialize Security Environment
ATBINSYS  00:00:00.005  APPC
IKJEFXSR  00:00:00.092  TSO
IXGBLF00  00:00:00.007  Logger
AXRINSTR  00:00:00.009  System REXX
CEAINSTR  00:00:00.017  Common Event Adapter
HWTTMINI  00:00:00.025  HWT Initialization
HWIAMIN1  00:00:00.007  BCPii
COMMNDXX  00:00:00.045  COMMANDxx processing
SMFWAIT   00:00:03.108  SMF
ICHSFI00  00:00:00.052  Security Server
ICHALTSP  00:00:00.618  Security Server
ICHSEC05  00:00:25.922  Security Server
MSIEXIT   00:00:01.059  Cnz_MSIExit Dynamic Exit
IEFJSIN2  00:00:01.352  SSN= subsystem
IEFHB4I2  00:00:00.005  ALLOCAS - UCB scan
CSRINIT   00:00:00.003  Windowing services
FINSHMSI  00:00:00.518  Wait for attached CMDs

IEEMB860  00:00:35.014      Uncaptured time:  00:00:00.053

Total Time:  00:01:20.957
```

Idea 4. Start Early - IPLDATA

- There is a tool provided by IBM in GitHub that can be used to see IPLDATA.
- https://github.com/IBM/IBM-Z-zOS/tree/main/zOS-Tools-and-Toys/iplstats
- But there is an easier way..



Idea 4. Start Early - IPLDATA

Setting the IPCS source to ‘active’ allows you to view information about the currently running system.

- I use IPCS to quicky see this information using the IPCS
  `IPLDATA STATUS`
  command.
- Same information from the IPCS
  `VERBX BLSAIPST`
  command

```
------------------- z/OS 02.04.00 IPCS PRIMARY OPTION MENU
OPTION  ===> 0

   0  DEFAULTS     - Specify default dump and options
   1  BROWSE       - Browse dump data set
```

Step 1: Defaults

```
------------------------- IPCS Default Values -------------------------------
Command ===>

  You may change any of the defaults listed below.  The defaults shown before
  any changes are LOCAL.  Change scope to GLOBAL to display global defaults.

  Scope    ==> LOCAL    (LOCAL, GLOBAL, or BOTH)

  If you change the Source default, IPCS will display the current default
  Address Space for the new source and will ignore any data entered in
  the Address Space field.

  Source   ==> active
  Address Space    ==>
```

Step 2: Set source to ‘active’ (current active system)

```
------------------------- IPCS Subcommand Entry -------------------------------
Enter a free-form IPCS subcommand or a CLIST or REXX exec invocation below:

===> ipldata status


----------------------- IPCS Subcommands and Abbreviations --------------------
ADDDUMP           | DROPDUMP, DROPD    | LISTDUMP, LDMP     | RENUM,     REN
ANALYZE           | DROPMAP,  DROPM    | LISTMAP,  LMAP     | RUNCHAIN, RUNC
```

Step 3: IPLDATA STATUS command

Idea 4. Start Early - IPLDATA

- Comparing two production sites:
- Tuning IPL / NIP for Site 1 may save a few tens of seconds.

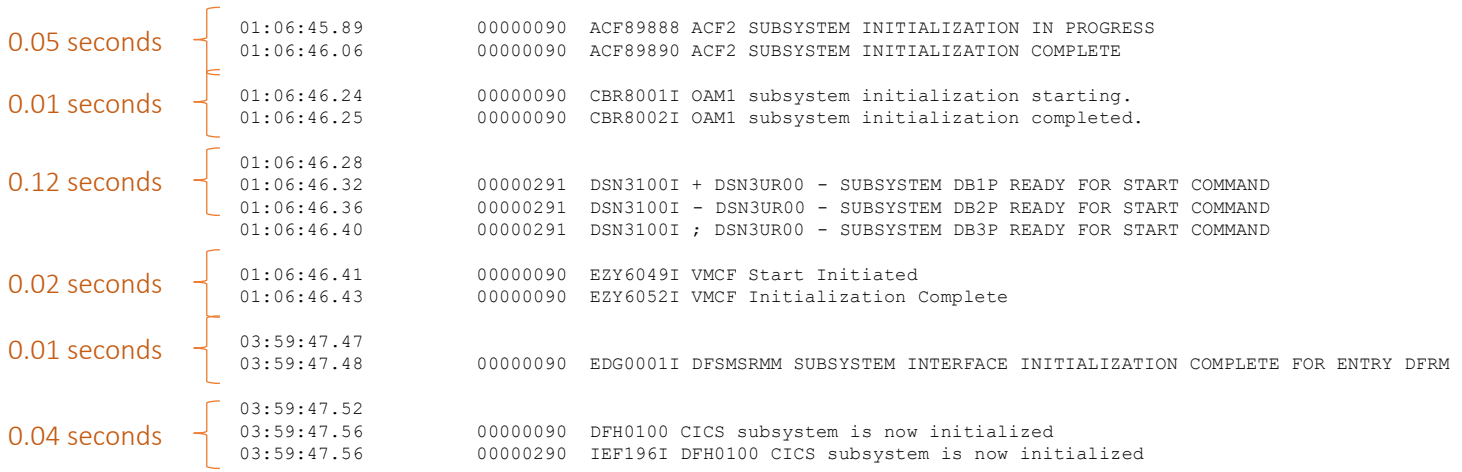
	Site 1	Site 2
IPL Time	1.75	1.93
NIP Time	42.72	15.13
IEEVIPL Time	1.475	0.86
IEEMB860	35.01	9.71
Total	**80.955**	**27.63**

	Site 1	Site 2	Notes
NIP			
IEAVNPA2	13.97	1.74	IOS - Non-DASD UCBs
IEAVNP03	4.24	0.01	Merge and analyse system parameters
IEEMB860			
ICHSEC05	25.92	1.8	Security Server (Site 1 ACF2, Site 2 RACF)

Idea 4. Start Early

Subsystems between the BEGINPARALLEL and ENDPARALLEL statements are executed in parallel. Only supported by some subsystems.

- How about PARMLIB IEFSSNxx?
  - BEGINPARALLEL and ENDPARALLEL statements may shave a second or two off the IEFSSNxx times.
  - But most subsystems start quickly.



Idea 4. Start Early

The earlier you IPL, the earlier your z/OS system is up. Some ways to start your IPL early:

- Tune your z/OS shutdown (another session perhaps).
- No long running batch jobs (that may delay shutdown).
- Define AUTOIPL policy (or similar GDPS features).
- Enable z/OS System Status Detection (SSD) and Sysplex Failure Management (SFM).
- Well trained operators, well documented procedures.

Other Ideas: Eliminate Errors

- I often see errors in startup that you would think would slow down restarts.
- This can be the case, particular if the error results in a WTO.
- Most don't slow down the startup much.
- Resolving many of these may make a small difference.

0.06 second delay.

```
03:59:16.74     00000290  IEA166I VATLST00: NO VOLUME MATCH FOUND FOR AR9O*  ON DEVICE TYPE 3390
03:59:16.74     00000290  IEA166I VATLST00: NO VOLUME MATCH FOUND FOR AM9O*  ON DEVICE TYPE 3390
03:59:16.75     00000290  IEA166I VATLST00: NO VOLUME MATCH FOUND FOR AM9R*  ON DEVICE TYPE 3390
03:59:16.75     00000290  IEA166I VATLST00: NO VOLUME MATCH FOUND FOR AM9W*  ON DEVICE TYPE 3390
03:59:16.80
```

0.03 second delay.

```
01:08:04.66 STC00047 00000090 *IEFTMS7 IPL DATE MORE THAN 8 DAYS FROM LAST IPL
01:08:04.66 STC00047 00000090 *IEFTMS3 PREVIOUS IPL DATE = 2025/026
01:08:04.66 STC00047 00000090 *0411 IEFTMS8 VERIFY DATE = 2025/040   OR 'HELP'
01:08:04.69 NMCTF50K 00000290  R 0411,HIGHDATE
```

No delay

```
01:06:28.82 INTERNAL 00000290  ASA003I SYNTAX ERROR IN PARMLIB MEMBER=DIAG00 ON LINE 13, 079
            POSITION 2: <NON-KEYWORD> WAS SEEN, WHERE ONE OF (CBLOC VSM TRAPS NUCLABEL
            REUSASID AUTOIPL ALLOWUSERKEYCADS FREEMAINEDFRAMES FF31HIGH ASLR <END_OF_FILE>)
            WOULD BE CORRECT. DETECTING MODULE IS IGVDITMS. INPUT LINE:
            MT_ZIIP_MODE=2
01:06:28.82 INTERNAL 00000290  ASA004I PARSING OF PARMLIB MEMBER=DIAG00 081
            CONTINUED AT <END_OF_FILE>, LINE 14.
            DETECTING MODULE IS IGVDITMS. INPUT LINE:
            <END_OF_FILE>
```

Other Ideas: Startup Boost.

Startup boost can give CPU capacity increases and other performance features for up to 60 minutes after IPL.

- Startup boost is enabled by default: all sites we see use it.
- Have not seen or measured any elapsed time differences. But will reduce any CPU contention delays.

```
01:06:28.83  00000290  IEA681I IPL speed boost is active
01:07:04.37  00000090  IEA675I IPL zIIP boost is active with 0 transient zIIP cores
01:07:04.63  00000090  IWM064I BOOST ACTIVATED.
```

IBM System Recovery Boost …

Unleash additional processing capacity using your already-entitled Central Processors and zIIPs during a fixed-duration performance increase known as, "the boost period."

- ✓ *Faster shutdown (planned events only).*
- ✓ *Faster startup (IPL)*
- ✓ *Faster middleware and workload restart*
- ✓ *Faster system recovery and workload execution*
- ✓ *Faster and parallelized GDPS reconfiguration and orchestration actions.*

The boost period can be used twice per IPL:
30-minute boost for shutdown
60-minute boost for startup

1 **Speed Boost**
Enables general-purpose processors on sub-capacity machine models to run at full-capacity speed in the boosting image(s).
Supported by z/OS®, z/TPF, z/VM® & SADMP

2 **zIIP Boost**
Provides additional capacity and parallelism by enabling general-purpose workloads to run on zIIP processors that are available to the boosting image(s).
Supported by z/OS. Requires defined zIIPs

3 **GDPS enhancements**
Increases the speed at which GDPS drives hardware actions, along with the speed of the underlying hardware services.
Supported by z/OS

Summary

Four basic concepts:

1. No manual intervention.
2. Maximise automation.
3. Minimise processing during z/OS startup.
4. Start everything as early as possible.

Summary

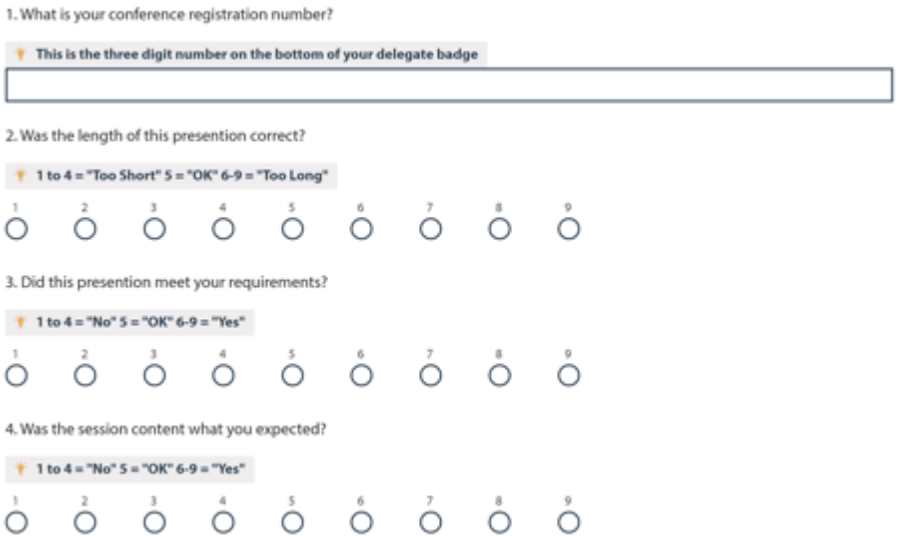
- Most z/OS startup time is after IPL and NIP.
- Looking at manual intervention and automation will provide the biggest reductions in z/OS startup times.

- Today, it is possible to start a z/OS system in 5 minutes: some sites are doing this today.

Category	Time	
IPL	< 2	seconds
NIP	25-70	seconds
The Rest	4-120	minutes

Session feedback

- Submit your feedback at https://conferences.gse.org.uk/2025V/feedback/5C
- Make sure you are signed into MyGSE
- The session code is 5C



Questions?