z/OS 3.2 IBM Education Assistant

Solution Name: Transaction Execution Removal

Solution Element(s): BCP Supervisor and SLIP/PER

July 2025

Agenda

- Trademarks
- Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Upgrade & Coexistence Considerations
- Installation & Configuration
- Summary
- Appendix

Trademarks

- See URL http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Objectives

- Transactional execution is being removed from zArchitecture
- z17 does not support non-constrained transactions (TBEGIN)
  - z16 is the last machine to support non-constrained transactions
  - Non-constrained transactions naturally have a fallback path, and in z17 TBEGIN will always abort, and the fallback path will always be taken
- zNext will not support constrained transactions (TBEGINC)
  - z17 is intended to be the last machine to support constrained transactions
  - TBEGINC will not work on zNext
  - Constrained transactions have no fallback path
  - Constrained transactions allow for serialization of up to 4 distinct octowords of storage
  - Alternate serialization mechanisms must be provided with z17 to replace TBEGINC
  - z17 has support for new PLO operations that serialize up to 4 distinct octowords of storage
- Learn about z17 PLO updates
- Learn about the SLIP TEND PER event

Overview

- Who (Audience)
  - z/OS installations upgrading to z17, zNext or zFuture
  - Application programmers who own z/OS applications that invoke constrained transactions (TBEGINC)
- What (Solution)
  - Enable software on z/OS to dual path constrained transaction and alternatives to support current and future machines
    - New PLO operands on z17
    - The SLIP TEND support to identify transaction end events
- Wow (Benefit / Value, Need Addressed)
  - Clients, Vendors and IBM can prepare z/OS now for zNext and zFuture (without constrained transactions)

Usage & Invocation – PLO Extension Facility

- Updates to the PLO instruction are available in z17
  - Operations:
    - Fetch
    - Store
    - Compare and swap
    - Compare and swap and store
  - For 1-4 operands
    - Single (1)
    - Double (2)
    - Tripple (3)
    - Quadruple (4)
  - Of various sizes
    - Word (4 bytes)
    - Double Word (8 bytes)
    - Quad Word (16 bytes)
    - Octo Word (32 bytes)
  - See the upcoming version of Principles of Operation for details

Migration Guide for TBEGINC, z/OS 2.4+

- Suppose your program runs only on z/OS 2.4 or later so constrained transactions are known to exist (or were known, but now PLO extensions will exist on z17)
- If PSATXC is on or CVTTXC is on
  - check either, there is no need to check both
  - take the existing constrained transaction path
- Else take an alternate path
  - This can use functions of the PLO Extensions facility
  - Or alternate serialization mechanism

Migration Guide for TBEGINC, z/OS 2.3 or earlier

- Suppose your program might need to run on releases older than z/OS 2.4 so already has a limited function or “old” alternate approach when constrained transactions are not available.
- The facility bit for PLO Extensions comes into play
- If constrained transactions are available
  - Check either PSATXC is on or CVTTXC is on (no need to check both)
  - take the constrained transaction approach
- Else if PLO Extensions facility is available
  - Check facility bit 87, named FACL_zM87 for now, is on
  - then take a new alternate approach that can (obviously) use PLO-e
- Else
  - neither constrained transactions nor PLO Extensions facility is available,
  - take the limited function or “old” alternate approach

Usage & Invocation – SLIP TEND

- TEND is a new SLIP PER trap like ZAD
  - Can be used to identify which programs are using transactional execution
  - When use of the TEND instruction is found, a SLIP PER interrupt is generated
  - Use with suitable filtering options
  - Use with IEAVTSZR action exit routine
    - `SLIP SET,TEND,ACTION=AEXIT,AEXIT=IEAVTSZR,…`
    - More details on the next slide

Usage & Invocation – SLIP TEND (continued 1)

To collect data and get a report of all TEND events, complete the following steps:

- Set up a procedure named IEAVTSZR (see next slide)
- `START IEAVTSZR`, which initializes an area to record the information.
- Run `SETPROG LPA,ADD,DSN=SYS1.LINKLIB,MOD=IEAVTSZE,FIXED`. Alternatively, you could use the FIX system parameter with an `IEAFIXxx` parmlib member to complete this step during IPL.
- Issue the following system command, which sets a SLIP trap named TEN1:
  `SLIP SET,TEND,A=AEXIT,AEXIT=IEAVTSZE,ID=TEN1,PL=50,OK,END`.
  Although A=AEXIT and the AEXIT keyword are not otherwise documented, this specific use is permitted. You can change the PL value, use a different ID and add other SLIP filtering keywords prior to END when setting this SLIP trap.
- Run your programs.
- Run `START IEAVTSZR`. In this instance, the command writes a report for all TEND events to SYSPRINT, and then resets to continue.
- If necessary, run additional iterations of the running programs and the `START IEAVTSZR` command.
- Run `START IEAVTSZR,OP=FREE`. You can skip this step if you are doing an IPL.

Usage & Invocation – SLIP TEND (continued 2)

- IEAVTSZR procedure

```
//*************************************************************
//** SIZE may be nK, nnK, nnnK, nM, nnM, nnnM. It is the amount of      *
//** fixed, 64-bit common storage to be used for individual instruction *
//** data. It is added to the amount needed to capture data for         *
//** every address space.                                              *
//**                                                                   *
//** OP=FREE indicates that you're done                                *
//** OP=DATA indicates                                                 *
//** - Allocate if not yet allocated                                   *
//** - Produce a report based on the current data                      *
//** - After producing a report, clear the data                        *
//** Do not specify STATS. Leave it as YES                             *
//*************************************************************
//IEAVTSZR PROC SIZE=1M,OP=DATA,STATS=YES,SYSOUT=*
//IEAVTSZR EXEC PGM=IEAVTSZR,TIME=1440,REGION=0M,
//      PARM='OP=&OP,SIZE=&SIZE,STATS=&STATS'
//SYSPRINT DD SYSOUT=&SYSOUT
```

Interactions & Dependencies

- Software Dependencies
  - None
- Hardware Dependencies
  - z17 – to use the PLO Extension Facility
  - None for SLIP TEND PER trap
- Exploiters
  - None

Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level: No
- No concerns
  - Intended to allow clients, vendors and IBM prepare for zNext
  - See Migration guide slides for instructions

- Function intended to be rolled down to z/OS 3.1 and 2.5
  - Follow APARs OA66855 (Supervisor) and OA67164 (SLIP) for PTFs when available

Installation & Configuration

- None

Summary

- You should now understand
  - The direction for transactional execution in zArchitecture
    - Non constrained transactions will execute fall back paths on z17
    - Constrained transactions are intended to fail on zNext
    - All applications using constrained transactions must be updated for zNext
  - z17 has PLO Extensions to operate on up to 4 distinct octowords of storage
    - Available as a replacement for constrained transactions
  - TEND is a new SLIP PER trap to help understand which programs are using constrained transactions

Appendix

- Publications
  - z/OS MVS System Commands
  - z/OS MVS Initialization & Tuning Reference
  - z/Architecture Principles of Operation