PJ44591 - Dynamic CPU Capacity

Michael Shershin
TPF Development lab

Dynamic CPU Capacity

I-stream Cap

Users can handle a sustained increase in workload without needing to take an outage.

HiperDispatch

Users can maximize CPU resources to lower hardware costs.

Low Priority Utilities

Users can selectively run utilities even during peak volumes without impacting real-time transactions.

Additional change

Constraint relief for memory below 2 gig

Before PJ44591

z/TPF uses all general purpose I-streams that are defined to an LPAR.

Users must purchase all general purpose I-streams before z/TPF is IPLed.

If additional capacity is needed, the additional general purpose I-streams need to be obtained; the z/TPF LPAR definition updated to use more I-streams; and z/TPF must be IPLed.

With PJ44591

z/TPF has ability to IPL with more capacity than is currently needed. An I-stream cap can be set. Application work is not given to I-streams above the cap.

On z14 machines when z/TPF is running as a dedicated LPAR, a Capacity on Demand pricing can be used for I-streams above the cap to reduce their cost.

When additional capacity is needed, the I-stream cap can be changed to increase capacity without needing an IPL. If Capacity on Demand pricing is used, additional costs will be incurred based on the number of I-streams and the length of time that the I-streams are needed.

Terminology

Active I-stream – an I-stream that is defined to the LPAR and is available for use by z/TPF.

Field ISTACTIS in the I-stream status table contains the number of active I-streams

In-use I-stream – an active I-stream that application work can be dispatched to.

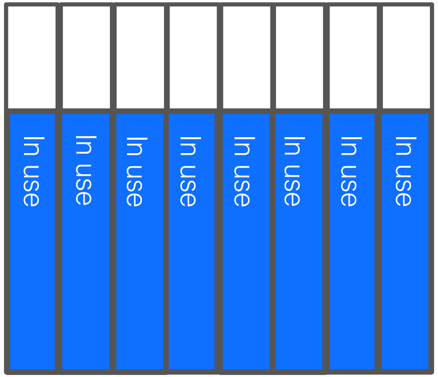
Field ISTUSEIS in the I-stream status table contains the number of in use I-streams

I-stream cap – the highest I-stream number of any I-stream that application work can be dispatched to.

Fenced I-stream – an active I-stream that is not in use and that has an I-stream number greater than the I-stream cap

As-Is Environment

z/TPF LPAR

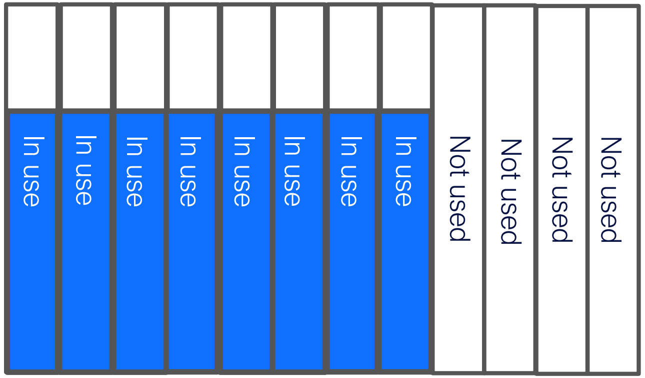


In-use I-streams	1	2	3	4	5	6	7	8
Active I-streams	1	2	3	4	5	6	7	8

In-use I-streams (ISTUSEIS) = 8
Active I-streams (ISTACTIS) = 8

To-Be: Set I-stream cap; Add additional I-streams; IPL z/TPF

z/TPF LPAR

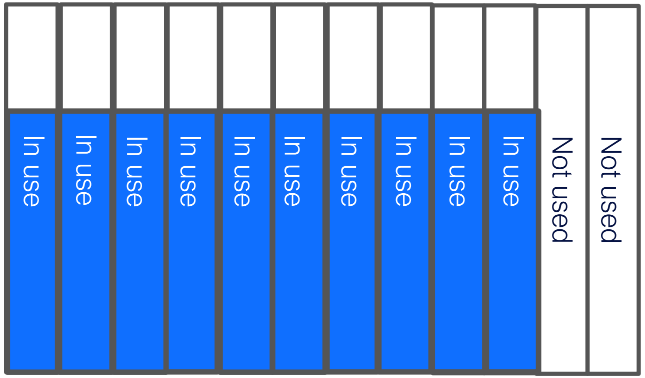


In-use I-streams	1	2	3	4	5	6	7	8				
Fenced I-streams									9	10	11	12
Active I-streams	1	2	3	4	5	6	7	8	9	10	11	12

I-stream cap = 8
In-use I-streams (ISTUSEIS) = 8
Active I-streams (ISTACTIS) = 12

To-be: Additional capacity needed; change I-stream cap

z/TPF LPAR



In-use I-streams	1	2	3	4	5	6	7	8	9	10		
Fenced I-streams											11	12
Active I-streams	1	2	3	4	5	6	7	8	9	10	11	12

I-stream cap = 10
In-use I-streams (ISTUSEIS) = 10
Active I-streams (ISTACTIS) = 12

How to set the I-stream cap?

- Three options
  - Command: ➔ ZMISS SET CAP-xx
  - Utilization configuration file
  - Default

- Order of precedence
  1) Obtain cap from previous ZMISS command
     - Previous ZMISS SET command must have been entered on the current z/TPF logical processor and the current physical machine and the same LPAR
  2) Obtain cap from utilization configuration file
     - Find the entry in the utilization configuration file that matches the current physical machine and LPAR
  3) Use default, which is the number of active I-streams

Example of setting the cap

```
==> ZMISS SET CAP-25
CSMP0097I 17.17.00 CPU-A SS-BSS  SSU-BSS  IS-01
MISS0003I 17.17.00 THE I-STREAM CAP WAS CHANGED FROM 20 TO 25
                   FOR CPU-A MACHINE 004352C7 LPAR TPFP2   .
```

Utilization configuration file

- Comma separated variable (CSV) file that contains control values that might differ based on the physical machine and LPAR
- On z/TPF the file name is: /etc/ibm_utl_cfg.csv
- File can be transferred to z/TPF using FTP or loaded as part of a OLD loadset
- To use updated file enter command: ➔ ZSYSL REFRESH UTILCONFIG
- Contents

```
UTILIZATION_CONFIGURATION_VERSION=2
```

machine_serial_number, LPAR_name, GP_cap, I-stream_floor, I-stream_cap, LODIC_utilization_class

- For example:

```
 UTILIZATION_CONFIGURATION_VERSION=2
114401,TPFP0001,99,2,8,UTLCLS02
*,*,99,2,0,UTLCLS02
```

I-stream cap logging

- Periodically collects status of the I-stream cap and utilization of fenced I-streams
- To enable logging enter command: ➔ ZMISS SET LOG KEEP-days
- Data is put into a binary I-stream cap log file
  - Files are in directory: /IBM/isclog/
  - Data is collected:
    - Cycle to NORM state
    - I-stream cap change
    - Every hour on the hour
  - A unique file exists for each: physical name, LPAR name, z/TPF logical CPUID, and date

I-stream cap report

- Contains I-stream control activities over the period of the report
  - Report can be for the entire loosely coupled complex
  - Report can be for a single z/TPF logical CPUID
- Required if Capacity on Demand pricing is being used

- Generated from the I-stream cap log files
  - To generate a report use command:

    ➔ ZMISS REPORT BEGIN-bdate END-edate FILE-’reportname’
  - Report is a comma separated variable (CSV) file.

Dynamic CPU Capacity

I-stream Cap

Users can handle a sustained increase in workload without needing to take an outage.

HiperDispatch

Users can maximize CPU resources to lower hardware costs.

Low Priority Utilities

Users can selectively run utilities even during peak volumes without impacting real-time transactions.

Additional change

Constraint relief for memory below 2 gig

Problem

Experiencing performance loss when running shared LPAR.

The number of logical CPUs exceeds the number of physical CPUs.

With PJ44591

Use HiperDispatch support so that a z/TPF LPAR can reduce the number of in-use I-streams to the minimum number needed to process the work load at that point in time.

Using a minimum number of I-streams reduces the virtual I-stream to real I-stream (or V to R) ratio.

Also, using a minimum number of I-streams improves memory cache efficiency, which helps the performance of the z/TPF LPAR and the physical machine.

Terminology

I-stream floor – the highest I-stream number of any I-stream that cannot be collapsed by z/TPF.

Parked I-stream – an active I-stream that is not in use and that has an I-stream number less than or equal to the I-stream cap.

Collapsed I-stream – an I-stream that completes the transition from an in-use I-stream to a parked or fenced I-stream.

Terminology

Entitlement – amount of CPU execution time that a shared LPAR is allowed when all shared LPARs are running at 100% busy. Entitlement is based on the weight of a shared LPAR compared to the weight of all shared LPARs.

Whitespace – under utilized capacity in the machine configuration.

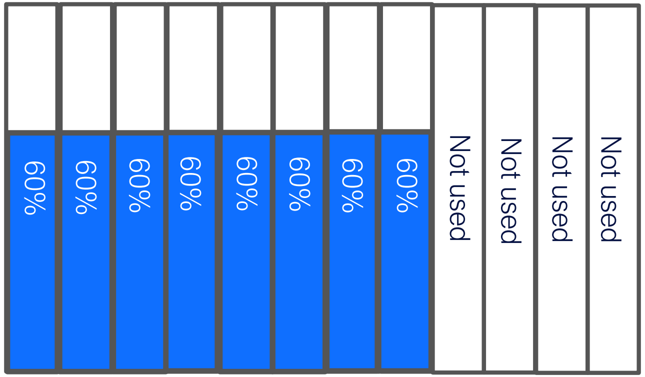
Horizontal polarization – host-CPU resources assigned to a multiprocessing (MP) LPAR configuration is spread evenly across the number of guest configured CPUs. HIPERDISPATCH-NO in z/TPF

Vertical polarization - host-CPU resources assigned to a multiprocessing (MP) LPAR configuration might be unevenly spread across the number of guest configured CPUs. HIPERDISPATCH-YES in z/TPF

As-Is: HiperDispatch-No; Two shared LPARs

z/TPF LPAR 1

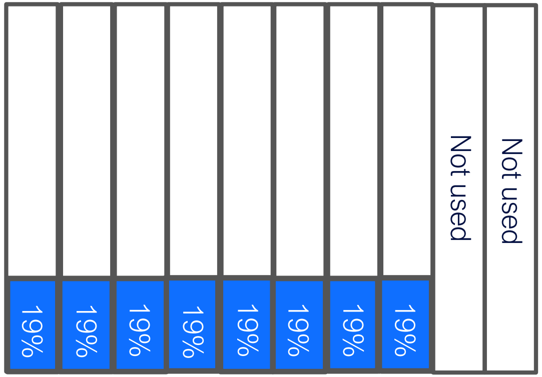
I-stream cap = 8
In-use I-streams (ISTUSEIS) = 8
Active I-streams (ISTACTIS) = 12



In-use I-streams	1	2	3	4	5	6	7	8				
Fenced I-streams									9	10	11	12
Active I-streams	1	2	3	4	5	6	7	8	9	10	11	12

z/TPF LPAR 2

I-stream cap = 8
In-use I-streams (ISTUSEIS) = 8
Active I-streams (ISTACTIS) = 10



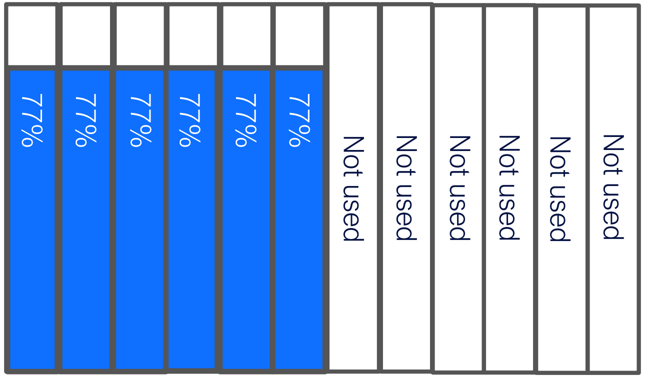
In-use I-streams	1	2	3	4	5	6	7	8		
Fenced I-streams									9	10
Active I-streams	1	2	3	4	5	6	7	8	9	10

Logical CPs in use by shared LPARs = 16
Physical CPs used by shared LPARs = 12

As-Is: HiperDispatch-Yes; Two shared LPARs

z/TPF LPAR 1

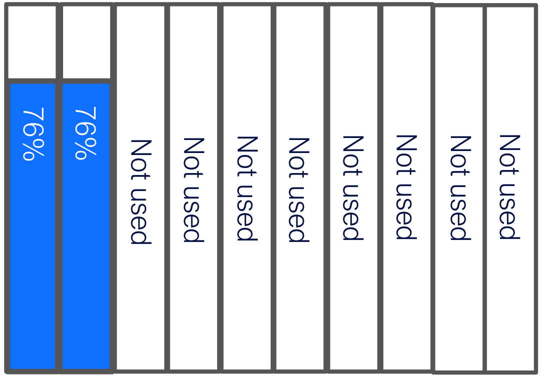
I-stream cap = 8
In-use I-streams (ISTUSEIS) = 6
Active I-streams (ISTACTIS) = 12



In-use I-streams	1	2	3	4	5	6						
Parked I-streams							7	8				
Fenced I-streams									9	10	11	12
Active I-streams	1	2	3	4	5	6	7	8	9	10	11	12

z/TPF LPAR 2

I-stream cap = 8
In-use I-streams (ISTUSEIS) = 2
Active I-streams (ISTACTIS) = 10



In-use I-streams	1	2								
Parked I-streams			3	4	5	6	7	8		
Fenced I-streams									9	10
Active I-streams	1	2	3	4	5	6	7	8	9	10

Logical CPs in use by shared LPARs = 8
Physical CPs used by shared LPARs = 12

HiperDispatch proof point

- Test environment
  - Machine is a z13 – 706
  - 3 LPARs; all 3 LPARs are shared
  - Each LPAR has 6 I-streams
  - V to R ratio is 3:1
- Ran AIR1 driver on all 3 LPARs
  - Internal driver that is used to simulate transactional workload

- Observed 13.6% performance improvement when HiperDispatch was used

How to turn on HiperDispatch?

- Two options
  - Command: → ZMISS SET HIPERDISPATCH-YES
    - Turned on for current processor only
    - Immediately enabled
  - Command: → ZMISS SET DEFAULT HIPERDISPATCH-YES
    - Turned on for all processors
    - Sets indicator in file record
    - At IPL z/TPF will turn on HiperDispatch for the IPLing z/TPF LPAR
      - Only effective if ZMISS SET HIPERDISPATCH was not previously entered for the logical processor on the physical machine using the current LPAR

Example of turning on HiperDispatch

```
==> ZMISS SET HIPERDISPATCH-YES
CSMP0097I 14.00.00 CPU-A SS-BSS  SSU-BSS  IS-01
MISS0006I 14.00.00 THE OVERRIDE HIPERDISPATCH SETTING WAS SET TO YES
                   FOR CPU-A MACHINE 004352C7 LPAR TPFP2   .
```

I-stream expansion / collapse

- Design point:

  Greed is good / LPAR weights are key

- Take as much CPU time as is needed as long as whitespace exists
  - May exceed LPAR entitlement
- Let PR/SM apportion the LPAR time based on LPAR weights
  - PR/SM may give an LPAR less cpu time as whitespace approaches zero.
  - z/TPF may collapse I-streams even if running at high utilization
    - Intent is to improve overall performance for the machine
- LPAR weights need to be considered carefully
  - How do you want the machine to operate at 100% busy

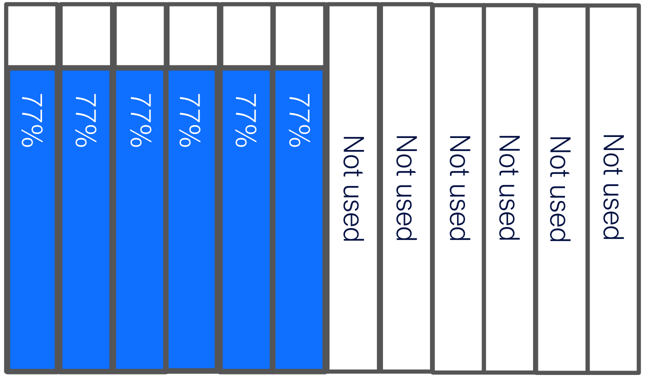
I-stream expansion

- Expand the number of in-use I-streams when:
  - Number of in-use I-streams is less than the I-stream cap
  - When average smoothed utilization over in-use I-streams is greater than or equal to the expansion utilization value
    - Default expansion utilization value is 80%
    - As whitespace approaches 0%, expansion utilization value increases
      - Highest expansion utilization value is 97%
  - The I-stream to be expanded is the highest in-use I-stream plus one
  - Expansion checks are made every second
    - Only one I-stream is expanded in a second

z/TPF LPAR 2 utilization = 83% > 80%; Expand

z/TPF LPAR 1

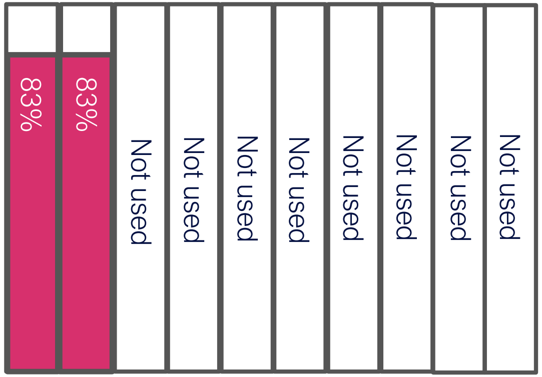
I-stream cap = 8
In-use I-streams (ISTUSEIS) = 6
Active I-streams (ISTACTIS) = 12



In-use I-streams	1	2	3	4	5	6						
Parked I-streams							7	8				
Fenced I-streams									9	10	11	12
Active I-streams	1	2	3	4	5	6	7	8	9	10	11	12

z/TPF LPAR 2

I-stream cap = 8
In-use I-streams (ISTUSEIS) = 2
Active I-streams (ISTACTIS) = 10



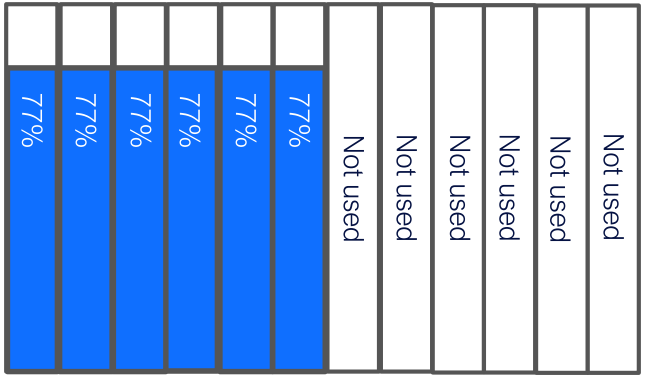
In-use I-streams	1	2								
Parked I-streams			3	4	5	6	7	8		
Fenced I-streams									9	10
Active I-streams	1	2	3	4	5	6	7	8	9	10

Logical CPs in use by shared LPARs = 8
Physical CPs used by shared LPARs = 12

z/TPF LPAR 2 expanded to 3 in-use I-streams

z/TPF LPAR 1

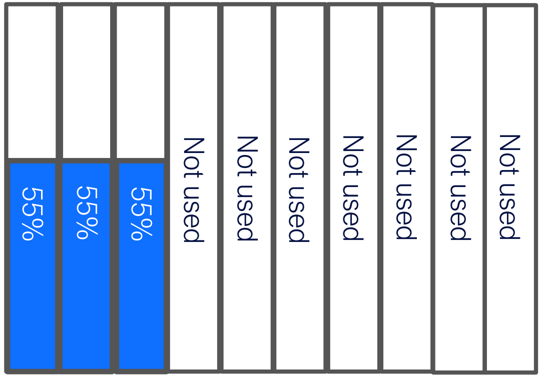
I-stream cap = 8
In-use I-streams (ISTUSEIS) = 6
Active I-streams (ISTACTIS) = 12



In-use I-streams	1	2	3	4	5	6						
Parked I-streams							7	8				
Fenced I-streams									9	10	11	12
Active I-streams	1	2	3	4	5	6	7	8	9	10	11	12

z/TPF LPAR 2

I-stream cap = 8
In-use I-streams (ISTUSEIS) = 3
Active I-streams (ISTACTIS) = 10



In-use I-streams	1	2	3							
Parked I-streams				4	5	6	7	8		
Fenced I-streams									9	10
Active I-streams	1	2	3	4	5	6	7	8	9	10

Logical CPs in use by shared LPARs = 9
Physical CPs used by shared LPARs = 12

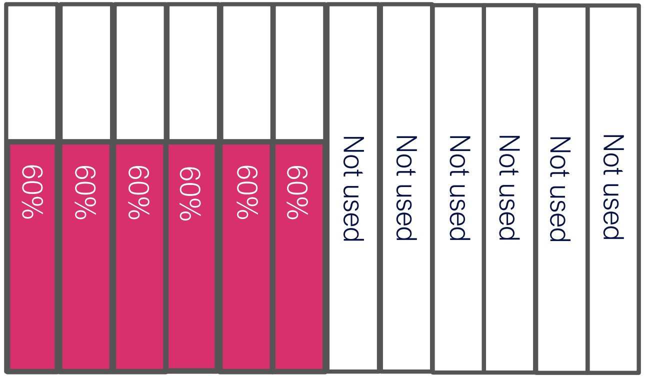
I-stream collapse

- Intent is to be conservative
  - Do not want to collapse an I-stream and have it expand in the next second
- Cannot collapse below the I-stream floor value
- The I-stream to be collapsed is the highest in-use I-stream
- Collapse checks are made every second
  - Only one I-stream is collapsed in a second

z/TPF LPAR 1 utilization drops and stays at 60%; Collapse

z/TPF LPAR 1

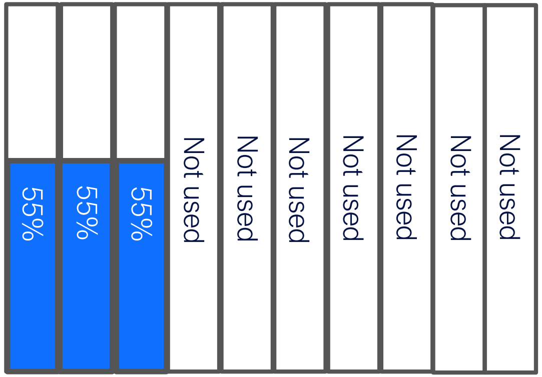
I-stream cap = 8
In-use I-streams (ISTUSEIS) = 6
Active I-streams (ISTACTIS) = 12



In-use I-streams	1	2	3	4	5	6						
Parked I-streams							7	8				
Fenced I-streams									9	10	11	12
Active I-streams	1	2	3	4	5	6	7	8	9	10	11	12

z/TPF LPAR 2

I-stream cap = 8
In-use I-streams (ISTUSEIS) = 3
Active I-streams (ISTACTIS) = 10



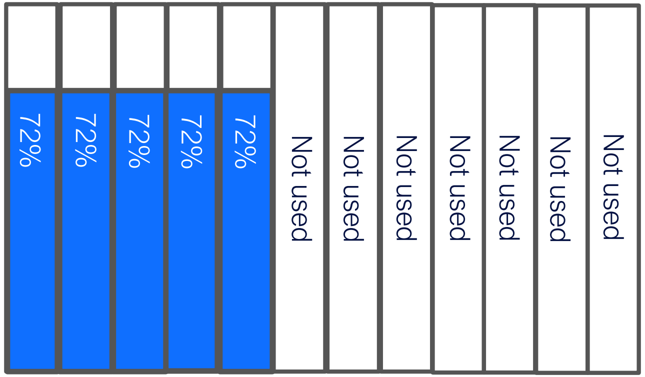
In-use I-streams	1	2	3							
Parked I-streams				4	5	6	7	8		
Fenced I-streams									9	10
Active I-streams	1	2	3	4	5	6	7	8	9	10

Logical CPs in use by shared LPARs = 9
Physical CPs used by shared LPARs = 12

z/TPF LPAR 1 collapsed to 5 in-use I-streams

z/TPF LPAR 1

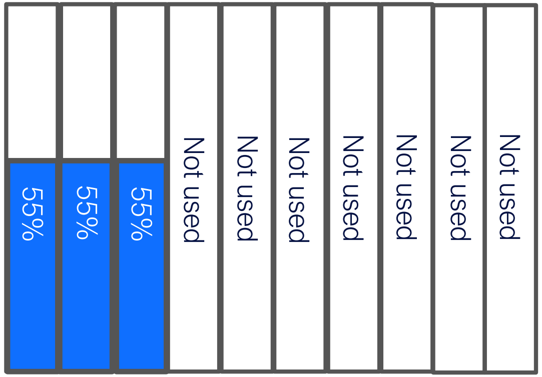
I-stream cap = 8
In-use I-streams (ISTUSEIS) = 5
Active I-streams (ISTACTIS) = 12



In-use I-streams	1	2	3	4	5							
Parked I-streams						6	7	8				
Fenced I-streams									9	10	11	12
Active I-streams	1	2	3	4	5	6	7	8	9	10	11	12

z/TPF LPAR 2

I-stream cap = 8
In-use I-streams (ISTUSEIS) = 3
Active I-streams (ISTACTIS) = 10



In-use I-streams	1	2	3							
Parked I-streams				4	5	6	7	8		
Fenced I-streams									9	10
Active I-streams	1	2	3	4	5	6	7	8	9	10

Logical CPs in use by shared LPARs = 8
Physical CPs used by shared LPARs = 12

Collapse stage 1

- Purpose
  - Transactional workload is not routed to the I-stream
  - ECBs are allowed to process
    - Work running on the I-stream can complete or switch to another I-stream
    - Systems work can update memory tables
      - Global synchronization
- Characteristics
  - Load balancing does not select I-stream
  - No DASD I/O interrupt processing
  - No OSA polling and deblocking
  - No Crypto card operation dequeuing
  - No Coupling facility dequeuing
  - CPU timer external interrupts - 1 every millisecond

Collapse stage 2

- Purpose
  - Minimal work executed
  - ECBs are not allowed to process
- Characteristics
  - Includes stage 1 characteristics
  - CPU timer external interrupts - 1 every second

- When in collapse stage 2, an I-stream will return to collapse stage 1
  - If an ECB is created is created on the I-stream, or
  - If an ECB switches to the I-stream

How to set the I-stream floor?

- Three options
  - Command: ➔ ZMISS SET FLOOR-xx
  - Utilization configuration file
  - Default

- Order of precedence
  1) Obtain floor from previous ZMISS command
     - Previous ZMISS SET command must have been entered on the current z/TPF logical processor and the current physical machine and the same LPAR
  2) Obtain floor from utilization configuration file
     - Find the entry in the utilization configuration file that matches the current physical machine and LPAR
  3) Use default, which is 2 for HPO systems or 1 for non-HPO systems

Cycle to NORM state

- I-stream expansion and collapse is only done in NORM state
- Restart sets in-use I-streams equal to the I-stream cap
- Cycle to NORM sets in-use I-streams equal to the I-stream cap
  - After outage, pent up demand may be waiting to arrive from the network
  - Want to have as much capacity as possible
- After cycle to NORM, collapse is not allowed for a period of time
  - Default time period is 30 seconds
  - Can alter the time period with ZMISS command

    ➔ ZMISS SET WAIT-seconds

Long running ECBs

- How to handle long running ECBs with an I-stream expands or collapses?
- Set the SWITCHABLE attribute (new capability)
- Behavior
  - Switches I-streams when the number of in-use I-streams changes
    - I-stream expands
    - I-stream collapses
  - Switches I-streams every 10 seconds
  - ECB can be switched when the ECB gives up control

- SWITCHABLE ECBs should not use I-stream unique memory area (globals)
- IBM long running ECBs have been updated to set the SWITCHABLE attribute

How to set the SWITCHABLE attribute

Use new API

- EASETC or tpf_easetc()
  - For example:

```
EASETC SWITCHABLE=YES,INHERIT=YES
tpf_easetc(TPF_EASETC_SWITCHABLE,
                 TPF_EASETC_INHERIT_YES+TPF_EASETC_SET_ON);
```

  - INHERIT=YES option causes any child ECBs to have the SWITCHABLE attribute set
  - Once SWITCHABLE attribute is set, switchable behavior is effective until ECB exits or SWITCHABLE attribute is turned off
  - Call API once at the start of a long running ECB

CRETC handling

- CRETC will create ECB on same I-stream where CRETC was executed
- If CRETC ECB is created on a collapsed I-stream, the ECB will be dispatched on IS-1.
- Consideration: Cannot pass address of I-stream unique memory areas as a parameter on a CRETC

Testing HiperDispatch under VM

- HiperDispatch is only available to LPARs running shared PR/SM
- A test option is provided when z/TPF runs as a guest under VM
  - Test collapse
  - Allows:
    - I-stream expansion
    - I-stream collapse
    - Same behavior as if HiperDispatch is in use
  - To use:

    ➔ ZSTRC ALTER TESTCLPS

Example of ZSTAT GPU output message

```
CSMP0097I 17.47.00 CPU-A SS-BSS  SSU-BSS  IS-01
STAT0019I 17.47.00 SYSTEM UTILIZATION DISPLAY
STATIC POWER SAVE MODE - SYSTEM AT 100 PERCENT
          ACTUAL   GP       GP     GP
NAME       UTIL   UTIL    CAP   ENGINES
TPFP2      21.9   13.1    15.0     3 _
NUM  ADR  UTIL/ ADJ  CROSS READY INPUT   VCT SUSPD DEFER ACT-ECB S   PSU  LPUU
IS- 1 00  72.8/ 72.6     1    10     0     0     0     0     41  U   74.3   .0
IS- 2 01  68.4/ 68.1     0     0     0     0     0     0     36  U   69.6   .0
IS- 3 02  67.4/ 67.2     0     0     0     0     0     0     37  U   68.6   .0 _
IS- 4 03  72.3/ 72.0     0     0     1     1     0     0     38  U   73.3   .0
IS- 5 04  69.7/ 69.5     0     0     0     0     0     0     38  U   70.2   .0
IS- 6 05  68.0/ 67.8     1     0     0     0     0     0     29  U   68.4   .0
IS- 7 06    .2/   .0     0     0     0     0     0     0      8  P1     .2   .0 _
...
IS-19 12    .2/   .0     0     0     0     0     0     0      5  P2     .2   .0 _
IS-20 13    .2/   .0     0     0     0     0     0     0      2 CP1     .2   .0
IS-21 14    .2/   .0     0     0     0     0     0     0      3  F1     .2   .0
IS-22 15    .2/   .0     0     0     0     0     0     0      3  F2     .2   .0
END OF DISPLAY
```

Examples of expand and collapse messages

Expand

```
CSMP0097I 17.49.13 CPU-A SS-BSS  SSU-BSS  IS-01
IISC0001I 17.49.13 IS-7 IS IN USE.
 D4076A14 A4A9E700 CURRENT TIME
 D4076A10 E3D2DA00 PREVIOUS TIME THAT IS-7 WAS NOT IN USE
END OF DISPLAY
```

Collapse

```
CSMP0097I 17.52.03 CPU-A SS-BSS  SSU-BSS  IS-01
IISC0002I 17.52.03 IS-7 IS IN COLLAPSE STAGE 1.
 D4076AB6 6D216400 CURRENT TIME
 D4076A14 A4A9E700 PREVIOUS TIME THAT IS-7 WAS EXPANDED
END OF DISPLAY
```

Unauthorized program entered on a collapsed I-stream

```
CSMP0097I 17.53.31 CPU-A SS-BSS  SSU-BSS  IS-10 _
CZIS0001W 17.53.31 AN UNAUTHORIZED PROGRAM WAS ENTERED ON A COLLAPSED I-STREAM
 ENTERED PROGRAM - QMLA
 PROGRAM THAT ENTERED THE UNAUTHORIZED PROGRAM - QMLA
 NUMBER OF UNAUTHORIZED PROGRAMS ENTERED SINCE THE LAST WARNING MESSAGE -    37
END OF DISPLAY
```

Example of ZSTAT LPAR output message

```
==> ZSTAT LPAR
CSMP0097I 13.28.55 CPU-A SS-BSS  SSU-BSS  IS-01
STAT0034I 13.28.55 LPAR UTILIZATION DISPLAY

NUMBER OF LPARS - 50
TOTAL WEIGHT - 3610
TOTAL NUMBER OF PHYSICAL CPUS - 34
TOTAL NUMBER OF CPUS FOR SHARED LPARS - 30
WHITESPACE - 0.0

 _
 LPARNAME  RCPUs  SCPUs  TYPE  WEIGHT   UTIL  UTILVCAP   MACHINE   ENTITLE
 CB88         20     20     S     500    0.0                 0.0       0.0
 LP1           3      3     S      10    0.0                 0.0       0.0
 SVT1CF6       2      2    D
 C05          14     14     S     500   89.9                42.0     303.0 _
 C06          14     14     S     500   89.6                41.8     302.0
 TPFP2        26     26     S     100    7.3                 6.3     228.4
 TPFP3        16     16     S     500    6.3                 3.4      24.4
 TPFP7        11     11     S     500    5.8                 2.1      15.5
END OF DISPLAY+
```

Dynamic CPU Capacity

I-stream Cap

Users can handle a sustained increase in workload without needing to take an outage.

HiperDispatch

Users can maximize CPU resources to lower hardware costs.

Low Priority Utilities

Users can selectively run utilities even during peak volumes without impacting real-time transactions.

Additional change

Constraint relief for memory below 2 gig

Definition

Utility – Batch processing on z/TPF that is not interactive with an end user. Typically, a utility is started by the operator or it is started at a specific time (time initiated). In addition, a utility typically does more work than a transactional request.

As-Is Scenario

Utilities previously ran during off peak times. As business has expanded across multiple continents there are no multi-hour “low traffic periods” anymore.

Now some utilities are forced to run during high traffic periods. To prevent these utilities from impacting transactional workloads, additional CPU capacity must be installed. This additional capacity increases costs.

As-Is: Utilities

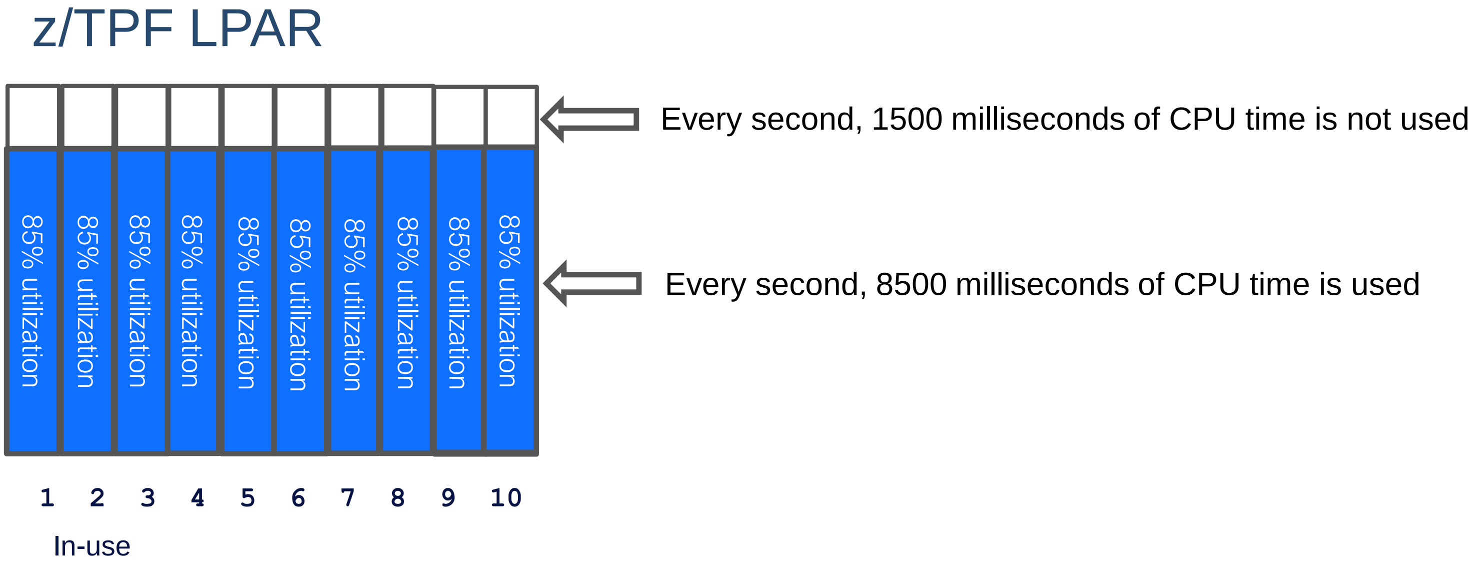
Utilities execute with the same priority as transactional work

- ECBs execute as soon as they are created
- VFA forces an ECB to give up control after 50 VFA accesses
- After DASD I/O completion ECBs are dispatched from the ready list
  - Utilities ECBs are dispatched when they are first on the ready list even if transactional ECBs are on the ready list

Controls used by utilities

- LODIC Resource checks
- LODIC Utilization checks
- DLAYC priorities the ECB less than current work and the same as new work
- DEFRC priorities the ECB lower than current work and lower than new work

Unused CPU execution time is highly perishable.



To-Be Scenario

Utilities can be marked as “low priority”. If CPU resources become constrained, these utilities will run at a slower rate so that transactional workloads will not be impacted.

Utilities can be run at any time of day or night without purchasing additional CPU capacity.

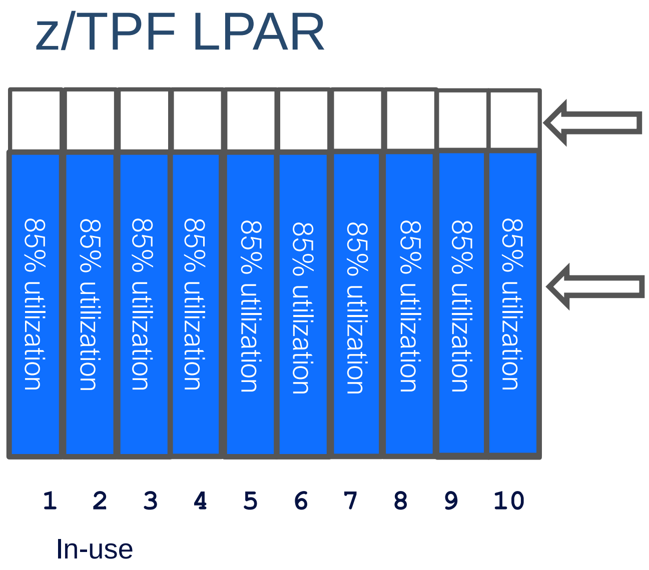
Low priority behavior differences

	Normal Priority	Low Priority
ECB dispatch after DASD I/O is complete	Use ready list	Use deferred list
VFA trip counter (CE1VCT) initial value	500	50

CPU loop review

- Cross list – dispatch entries between I-streams
- Ready list – dispatch existing work after doing I/O
- Input list – dispatch new work
- Deferred list – dispatch work that has been delayed
- Time Available Supervisor (TAS) on main I-Stream only
- Idle routine

Low priority utilities to run in currently unused space



Low priority utilities run in this space
Because ECBs are dispatched from deferred list
Reduce utility impact on response time for transactional ECBs

Transactional workload and normal priority utilities run in this space

Low priority utilities proof point

Use dedicated LPAR with 1 I-stream	**AIR1 with no utility**	**AIR1 + I/O intensive driver as low priority**	**AIR1 + I/O intensive driver as normal priority**
Mean utilization	81.6%	99.4%	99.7%
Mean AIR1 weighted message rate	5988	5989	3719

AIR1 is an internal driver that is used to simulate transactional workload

Become a low priority utility

New API to set LOWPRIORITY ECB attribute

- EASETC or tpf_easetc()
  - For example:

```
EASETC LOWPRIORITY=YES,INHERIT=YES
tpf_easetc(TPF_EASETC_LOWPRIORITY,
                    TPF_EASETC_INHERIT_YES+TPF_EASETC_SET_ON);
```

  - INHERIT=YES option causes any child ECBs to have the LOWPRIORITY attribute set
  - Once LOWPRIORITY attribute is set, low priority behavior is effective until ECB exits or LOWPRIORITY attribute is turned off

Become a low priority utility

New SMP prefix

– Use of –LP/ prefix will turn on LOWPRIORITY attribute with INHERIT=YES for authorized Z commands

For example, start PDU as low priority:

➔ -LP/ZRPDU CREATE

– Use ZFMSG command to manage Z commands that are LOWPRIORITY authorized

For example, authorize PDU verify and roll in to run as low priority

➔ ZFMSG CHG ZDUPD LPA

Prevent low priority work from using TEs

- Use LODIC utilization class
  - LODIC utilization class – TARGET value is used to determine if pseudo utilization is too high
  - LODIC utilization class – GPTARGET value is used to determine if utilization of functions that are not TE eligible are too high

- Ability exists to automatically use a LODIC utilization class when API EASETC LOWPRIORITY=YES is executed
  - Use the utilization configuration file
    - In entry for current processor include existing LODIC utilization class

      ```
      UTILIZATION_CONFIGURATION_VERSION=2
      *,*,99,0,0,UTLCLS01
      ```

    - Refresh the utilization configuration file

      ➔ ZSYSL REFRESH UTILCONFIG

Consider running Recoup as low priority utility

- Reduce Recoup impact to transactional ECBs
- Number of ECBs that recoup uses can be increased
  - If number of ECBs does not change, low priority Recoup may take longer to complete
  - If available ECBs exist, adding ECBs to low priority Recoup will make Recoup run faster and have minimal impact on transactional ECBs

    ➔ ZRECP LEVEL-xxxx
- Use –LP/ prefix to start Recoup

    ➔ -LP/ZRECP START

    ➔ -LP/ZRECP RECALL

    ➔ -LP/ZRECP RESUME

    ➔ -LP/ZRECP PROCEED

    ➔ -LP/ZRECP LOST CREATE

ZSTAT GPU when recoup is running as low priority

```
==> -LP/BSS/ZRECP RECALL
CSMP0097I 06.36.44 CPU-B SS-BSS  SSU-HPN  IS-01
CSMP0099I 06.36.44 010005-B -LP/BSS/ZRECP RECALL+


CSMP0097I 06.43.00 CPU-B SS-BSS  SSU-HPN  IS-01
STAT0019I 06.43.00 SYSTEM UTILIZATION DISPLAY
STATIC POWER SAVE MODE - SYSTEM AT 100 PERCENT
          ACTUAL   GP      GP     GP
NAME       UTIL   UTIL    CAP   ENGINES
TPFP1      99.7   51.8    85.0     3 _
NUM  ADR  UTIL/ ADJ  CROSS READY INPUT   VCT SUSPD DEFER ACT-ECB S   PSU  LPUU
IS- 1 00  99.5/ 99.3     0    17     0     0     0     5     71  U   99.5  5.6
IS- 2 01  99.8/ 99.5     0     1     0     0     0     2     40  U   99.8  5.6
IS- 3 02  99.8/ 99.4     0     0     0     0     0     4     45  U   99.8  5.8 _
IS- 4 03  99.7/ 99.5     2     0     0     0     0     3     44 CU   99.7  3.1
END OF DISPLAY+
```

Dynamic CPU Capacity

I-stream Cap

Users can handle a sustained increase in workload without needing to take an outage.

HiperDispatch

Users can maximize CPU resources to lower hardware costs.

Low Priority Utilities

Users can selectively run utilities even during peak volumes without impacting real-time transactions.

Additional change

Constraint relief for memory below 2 gig

Constraint relief for memory below 2 gig

- Dispatch control records (DCR) are above 4 gig
- Dispatch control lists (DCL) are above 4 gig
  - Can be reasonably large
  - I-stream unique
  - Each work list (cross, ready, input, deferred) has different size
    - Size of DCL for specific work list is based on number of allocated core blocks

Thank you

Michael Shershin

shershin@us.ibm.com