# Run Recoup on fenced I-streams

2022 TPF Users Group Conference
March 27-30, Dallas, TX
Systems Control Program

Michael Shershin

IBM Z

IBM

# Agenda

- PJ46401 - Use of fenced I-streams
- PJ46401 - IOPS control
- PJ46401 - Other changes
- PJ46563 – Recoup profile option for low priority

PJ46401 was released in October 2021
PJ46563 was released in September 2021

# Problem Statement

Typically, Recoup has a time window when it can run.  If Recoup exceeds the time window, other work is impacted.  As databases grow larger, it becomes difficult to keep Recoup run time within the allowed time window.

# Pain Points

A balancing act is performed when Recoup is run.

- Transactional work should have higher priority. If there is not enough capacity, Recoup slows down and takes longer to complete.
- Other utilities that are required for the business are usually not run while Recoup is running. Historically, there is concern that running multiple utilities together will cause system problems.
- If Recoup does not complete in its allotted time, the ability to run other utilities is impacted. This might cause business impacts.
- As the size of databases grow, will Recoup be able to complete in its allotted time?

# Technical Details – Terminology review

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

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

**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.

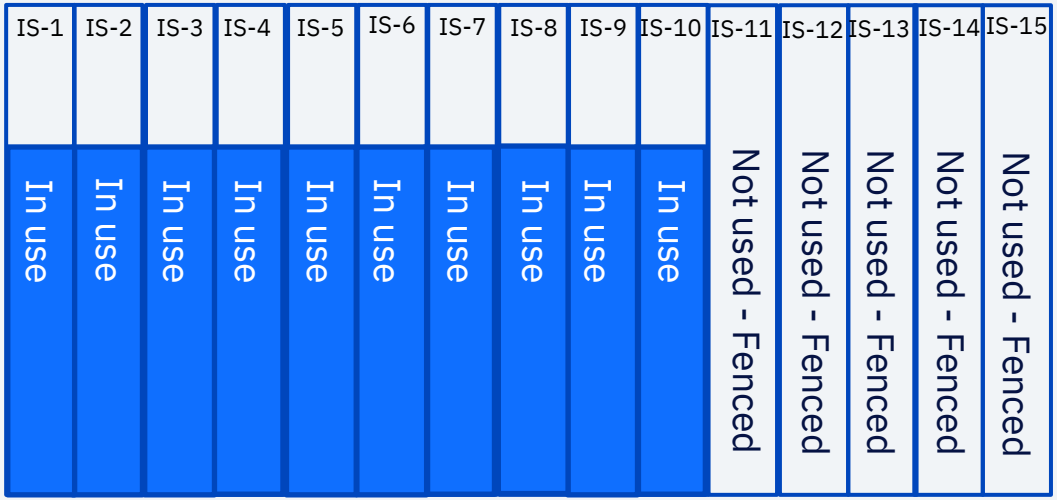
# As-Is: Recoup

Recoup work distribution

- Recoup shares processing power of in-use I-streams with transactional work.
- Recoup primary ECB runs on IS-1.
- Recoup child ECBs are load balanced across all in-use I-streams.

# As-Is: Example configuration

z/TPF LPAR




I-stream cap = 10
In-use I-streams (ISTUSEIS) = 10
Active I-streams (ISTACTIS) = 15
Fenced I-streams = IS-11 through IS-15

- **Transactions** run on in use I-streams: IS-1 through IS-10
- **Recoup ECBs** run on in use I-streams

# To-Be: Recoup

When Recoup is run as low priority, fenced I-streams are used

- Dynamic CPU must be in use
  - An I-stream CAP must be set and one or more fenced I-streams must exist
- Up to 4 fenced I-streams will be used
- No additional cost to you

Recoup can be run as low priority without fenced I-streams

- Allows Recoup to be run at any time during the day
- No impact on transactional work
- Work distributed across in-use I-streams but transactional work has priority

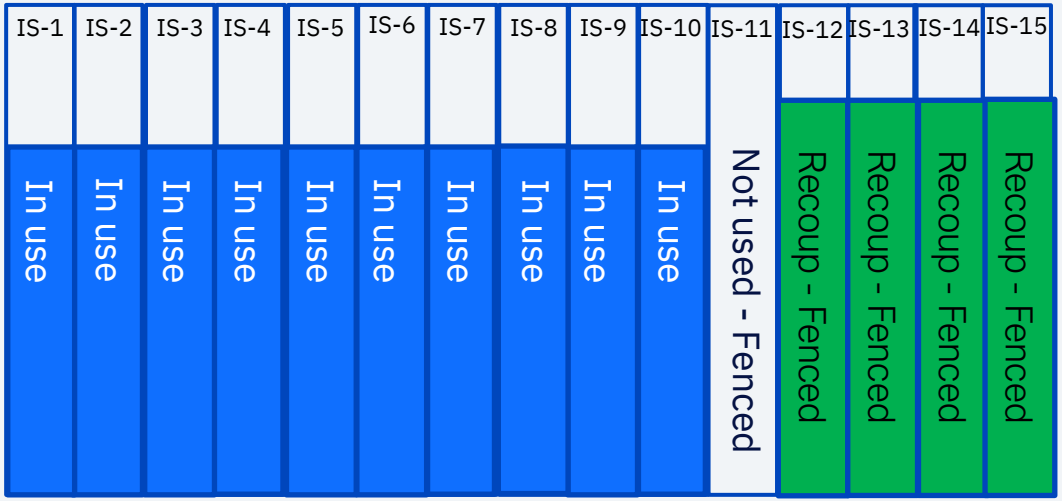
# To-Be: Recoup

Recoup work distribution when fenced I-streams are used

- Recoup primary ECB runs on IS-1.
- Recoup child ECBs are automatically routed to one of four fenced I-streams.
  - Drive the four fenced I-streams as hard as possible.
  - Utilization on in-use I-streams will likely be different than utilization on the four fenced I-streams.
- If the four fenced I-streams are running at 100% utilization, additional Recoup child ECBs are load balanced across in-use I-streams.

# To-Be: Example configuration with Recoup using fenced I-streams

z/TPF LPAR




I-stream cap = 10
In-use I-streams (ISTUSEIS) = 10
Active I-streams (ISTACTIS) = 15
Fenced I-streams = IS-11 through IS-15
Recoup use = IS-12 through IS-15

Number of Recoup ECBs = 200

- **Transactions** run on in use I-streams: IS-1 through IS-10
- **Recoup primary ECB** runs on IS-1
- **Recoup child ECBs** run on fenced I-streams

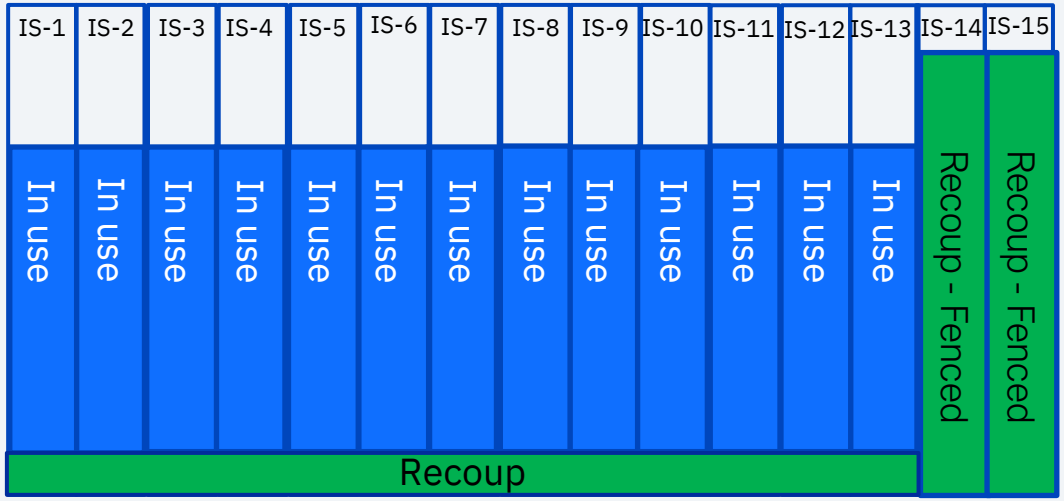
# Technical Details

Why require Recoup to run as low priority?

- If the I-stream cap changes and a fenced I-stream that is currently doing Recoup work is now an in-use I-stream, transactional work must be higher priority.
  - Purpose of changing the I-stream cap is to get more capacity for transactional work.
  - By having Recoup as low priority, Recoup will allow transactional work to use the added capacity immediately.
- Even as low priority, all Recoup work has the same priority when running on fenced I-streams.

# To-Be: Example configuration with Recoup using fenced I-streams after I-stream cap change

z/TPF LPAR




**I-stream cap = 13 (was 10)**
In-use I-streams (ISTUSEIS) = 13
Active I-streams (ISTACTIS) = 15
Fenced I-streams = IS-14 through IS-15
Recoup use = IS-14 through IS-15

Number of Recoup ECBs = 200

- **Transactions** run on in use I-streams: IS-1 through IS-13
- **Recoup primary ECB** runs on IS-1
- **Recoup child ECBs** run on fenced I-streams and overflow to in-use I-streams

# To-Be: Recoup

## Overflow from fenced I-streams to in-use I-streams

- If the Recoup child ECBs overflow from fenced I-streams to in-use I-streams, transactional ECBs will be higher priority and they will not be impacted by the Recoup child ECBs.
- The intent of overflowing to in-use I-streams is to exploit unused capacity on in-use I-streams.
  - If in-use I-streams are not running at 100%, spare cycles are available.
  - These cycles are highly perishable.
  - Best to use spare cycles as long as there is no impact to transactions.
  - No additional software licensing costs to run Recoup on in-use I-streams if SCRT is used.

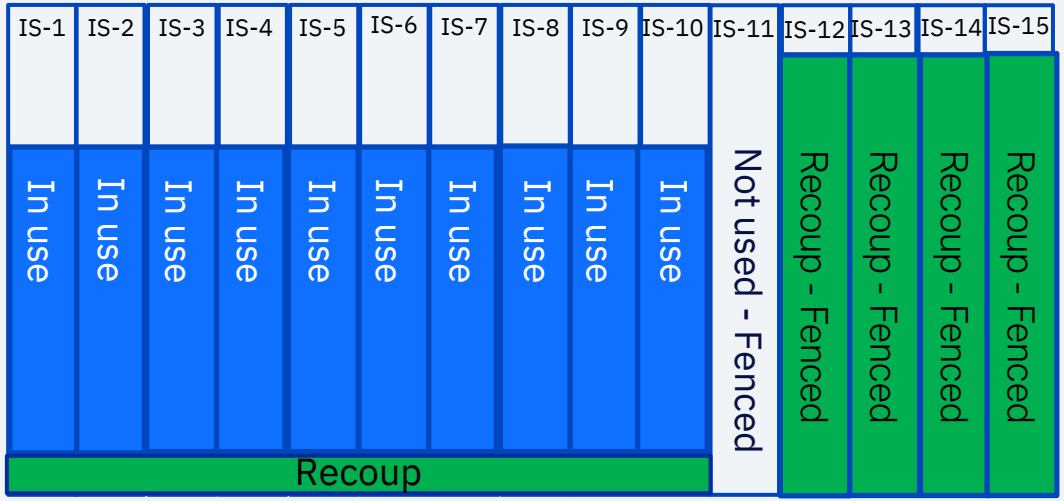
# Technical Details

Number of Recoup ECBs allowed can be used to control fenced I-stream utilization and amount of overflow to in-use I-streams

- ZRECP LEVEL-number_of_ecbs
- GROUP macro has optional ECB parameter for the maximum number of ECBs allowed for this GROUP.

# To-Be: Example configuration with Recoup using fenced I-streams after increase in number of Recoup ECBs

z/TPF LPAR




I-stream cap = 10
In-use I-streams (ISTUSEIS) = 10
Active I-streams (ISTACTIS) = 15
Fenced I-streams = IS-11 through IS-15
Recoup use = IS-12 through IS-15

Number of Recoup ECBs = 250

- **Transactions** run on in use I-streams: IS-1 through IS-10
- **Recoup primary ECB** runs on IS-1
- **Recoup child ECBs** run on fenced I-streams and overflow to in-use I-streams

# Technical Details

ZSTAT U status column has SF1 (system use + fenced stage 1) for fenced I-streams that are being used for Recoup.

```
CSMP0097I 09.48.04 CPU-B SS-BSS  SSU-HPN  IS-01
STAT0039I 09.48.04 SYSTEM UTILIZATION DISPLAY
STATIC POWER SAVE MODE - SYSTEM AT 100 PERCENT
SPEED BOOST IS NOT ACTIVE
I-STREAM BOOST IS NOT ACTIVE
NUM  ADR  UTIL/ ADJ  CROSS READY INPUT   VCT SUSPD DEFER ACT-ECB S   PSU  LPUU
IS- 1 00  72.8/ 72.5     0     0     0     0     0     0      31  U  72.6   .0
IS- 2 01  71.0/ 70.8     1     0     0     0     0     0      27  U  71.0   .0
IS- 3 02  70.6/ 70.4    12     0     0     0     0     0      22 CU  70.6   .0
IS- 4 03    .2/   .0     0     0     0     0     0     0       1 F2   .2   .0
IS- 5 04  94.3/ 94.1     0     0     0     0     0     1      11 SF1 95.1 93.4
IS- 6 05  95.0/ 94.8     0     0     0     0     0     0      15 SF1 95.4 94.2
IS- 7 06  95.0/ 94.8     0     0     0     0     0     4      10 SF1 95.0 94.0
IS- 8 07  94.9/ 94.7     1     0     0     0     0     0      16 SF1 95.3 94.0
END OF DISPLAY+
```

# Technical Details

What Recoup commands can use fenced I-streams?

- All ZRECP commands that are started as low priority and create child ECBs.
  - ZRECP RECALL
  - ZRECP RESUME
  - ZRECP PROCEED
  - ZRECP DUMP
  - ZRECP LOST

# Agenda

PJ46401 - Use of fenced I-streams

PJ46401 - IOPS control

PJ46401 - Other changes

PJ46563 – Recoup profile option for low priority

PJ46401 was released in October 2021
PJ46563 was released in September 2021

# Pain Points

- DASD I/Os are naturally limited when Recoup is running on in use I-streams. Recoup shares in use I-streams with transactional workload. Available CPU time to allow Recoup to run is limited and this limits Recoup DASD I/O requests.
- DASD I/Os can increase substantially when Recoup is running on fenced I-streams. Recoup will get all the CPU time on fenced I-streams.
- Can the DASD subsystem handle the added load when Recoup is running on fenced I-streams?

# Technical Details – Terminology review

**IOPS** – input / output (I/O) operations per second.  These are physical operations that are NOT satisfied out of VFA.

# Technical Details

New API has been created to control IOPS for real time DASD devices.

- Only applies to IOPS for DEVA, DEVB, DEVC, and DEVD DASD modules.
- API:

```
#include <tpf/tpfapi.h>
void tpf_dasd_iops_limit(struct iops_limit *iops_info);
```

- API returns:
  - Indication of whether IOPS are above the allowed limit
  - If IOPS are above the allowed limit, the number of microseconds to wait before re-trying the API.

# Technical Details – API example

```
#include <tpf/c_iops.h>
#include <tpf/tpfapi.h>
#include <time.h>

  struct iops_limit  iops_info;
  struct iops_limit *my_iops_limit;

  my_iops_limit = &iops_info;

  tpf_dasd_iops_limit(my_iops_limit);

  while (my_iops_limit->limit_flag) {
    usleep((unsigned int)my_iops_limit->microseconds);
    tpf_dasd_iops_limit(my_iops_limit);
  }
```

# Technical Details – API usage

- Do not use the tpf_dasd_iops_limit() API in transactional work
- Intended for utilities that can drive a large amount of physical DASD I/O
- Where to put the API
  - Do not put the API before every FINWC or FIWHC
  - Best to put the API at same place where existing LODIC checks are made
    - Add API to parent ECB when decision is made to create a child ECB
- IBM support is using the API
  - Recoup
  - ZPOOL INIT

# Technical Details – IOPS limit

## Single DASD IOPS limit

- Only used when API tpf_dasd_iops_limit() is called
- Applies to IOPS on the processor
  - If IOPS limit is 500,000 and a check is done on CPU-B, the API checks for IOPS done on CPU-B
- One value is used by all processors in the complex (i.e. processor shared)
  - If IOPS limit is 500,000, the limit is 500,000 on each processor in the complex
- DASD IOPS limit is kept in format 2 global IGSHARED
  - IGSHARED is processor shared, I-stream shared, and synchronizable

# Technical Details – IOPS limit

## Single DASD IOPS limit

- The DASD IOPS limit should be a percentage of the total capacity of your DASD I/O subsystem
  - Intend is to prevent utilities from using all DASD I/O capacity
- To display the DASD IOPS limit do: ZSONS DISPLAY IOPS
- To alter the DASD IOPS limit do: ZSONS ALTER IOPS LIMIT-*maxiops_value*
  - DASD IOPS are unlimited when maxiops_value is zero.

# Technical Details – ZSONS example

Display and change the IOPS limit.

```
ZSONS DISPLAY IOPS

CSMP0097I 09.39.27 CPU-B SS-BSS  SSU-HPN  IS-01
SONS0066I 09.39.27 MAXIMUM PHYSICAL DASD IOPS         0+



ZSONS ALTER IOPS LIMIT-200000

CSMP0097I 09.39.27 CPU-B SS-BSS  SSU-HPN  IS-01
SONS0066I 09.39.27 MAXIMUM PHYSICAL DASD IOPS    200000+
```

# Technical Details – ZSTAT example

ZSTAT IOPS shows the number of IOPS for the last second.

```
ZSTAT IOPS

CSMP0097I 09.57.10 CPU-B SS-BSS  SSU-HPN  IS-01
STAT0062I 09.57.10 DASD IOPS DISPLAY

SUBSYSTEM         TOTAL         DEVA         DEVB         DEVC         DEVD
BSS               48388        41435         6953            0            0
END OF DISPLAY+
```

# Agenda

PJ46401 - Use of fenced I-streams

PJ46401 - IOPS control

PJ46401 - Other changes

PJ46563 – Recoup profile option for low priority

PJ46401 was released in October 2021
PJ46563 was released in September 2021

# Technical Details – other changes

Updates made to use overlapping I/O in pseudo directory handling

- ZPOOL INIT (brv2.asm) uses multiple ECBs
- ZRECP DUMP (bkc1.asm) uses multiple DECBs
- Initialize Recoup pseudo directories (brv0.asm) uses multiple DECBs
- Recoup GFS activity and pseudo directory count hander (bcp6.asm) uses multiple ECBs.

# Agenda

PJ46401 - Use of fenced I-streams

PJ46401 - IOPS control

PJ46401 - Other changes

PJ46563 – Recoup profile option for low priority

PJ46401 was released in October 2021
PJ46563 was released in September 2021

# Technical Details – PJ46563

Support added new ZRECP PROFILE option for low priority

- When the profile option is set to use low priority, there is no need to prefix the commands with –LP/ in order to run as low priority.
- Command: ZRECP PROFILE PRIORITY-LOW
- The following Recoup commands use the profile setting.
  - ZRECP CONTINUE ZRECP DUMP ZRECP LOST
  - ZRECP PROC ZRECP PROCEED ZRECP RECALL
  - ZRECP RESTART ZRECP RESUME ZRECP START

# Technical Details – ZRECP PROFILE example

```
ZRECP PROFILE PRIORITY-LOW
CSMP0097I 10.54.28 CPU-B SS-BSS  SSU-HPN  IS-01
RECP0856I 10.54.28 RECOUP OPTIONS
CURRENT OPTIONS                  LAST OPTIONS

DIRTIM    - 00500                DIRTIM    - 00500
ERRLOG    - BOTH                 ERRLOG    - BOTH
ADR       - NO                   ADR       - NO  _
ADRNUM    - 0032                 ADRNUM    - 0032
FIXERMAX  - 0100                 FIXERMAX  - 0100
...
PRIORITY  - LOW                  PRIORITY  - NORMAL
DFOPTMZ   - DISABLE              DFOPTMZ   - DISABLE
RCP       - NONE                 RCP       - NONE
RCPX      - NONE                 RCPX      - NONE
CURRENT PATH   - /usr/IBM _
LAST PATH      - /usr/IBM

END OF DISPLAY+
```

# Value Statement

Additional processing power can be used for Recoup so that it can complete in much less time.

# Thank you

© Copyright IBM Corporation 2022. All rights reserved. The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. Any statement of direction represents IBM's current intent, is subject to change or withdrawal, and represent only goals and objectives. IBM, the IBM logo, and ibm.com are trademarks of IBM Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at Copyright and trademark information.