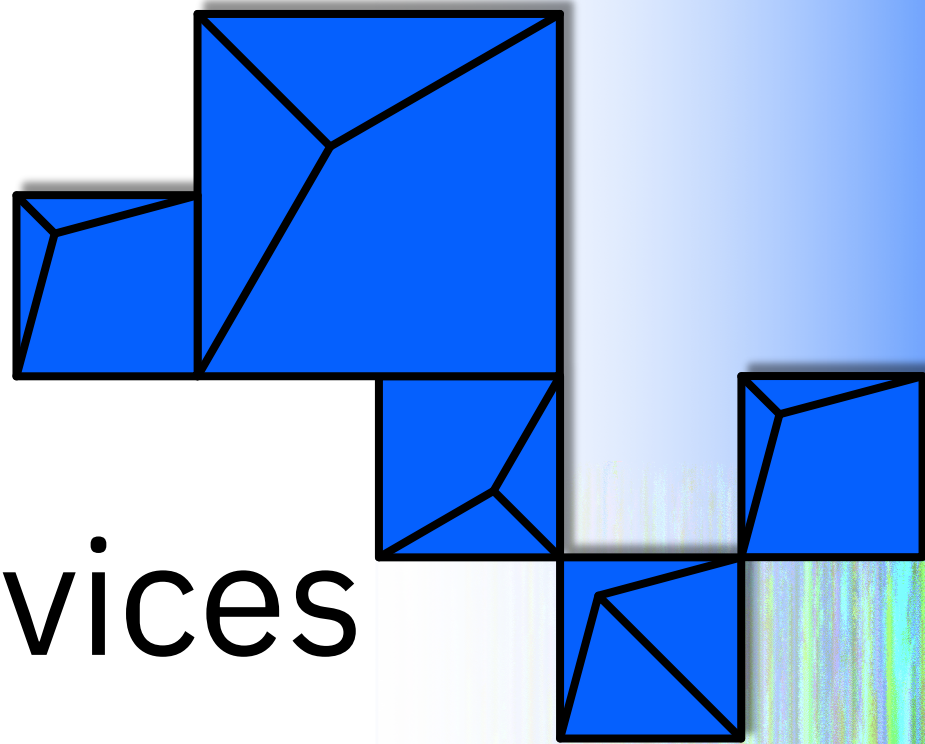# z/TPF System Services Enhancements

Michael Shershin

IBM

# Agenda

**PJ45401 – No impact to IPL time with large VFA**

PJ45598 – Recoverable system heap

PJ45739 – System recovery boost

Improved handling of sick DASD

PJ45930 – CTL-571 due to RIOSC time out

PJ45934 – RIOSC time out enhancement

PJ46019 – VFA changes for Recoup chain chase

Other enhancements

PJ46026 - Manage recovery log as a resource

PJ45025 - Include name-value pairs in dump messages

PJ45959 - Display core enhancements

What's next

Multi-mod copy

**Problem Statement**

# PJ45401 – No impact to IPL time with large VFA

Time needed to IPL can increase significantly when the size of VFA is hundreds of gigabytes or terabytes and VFA is being initialized.

Also, filing of VFA delay file pending buffers can take a long time to complete in restart when a large number of these buffers exist.
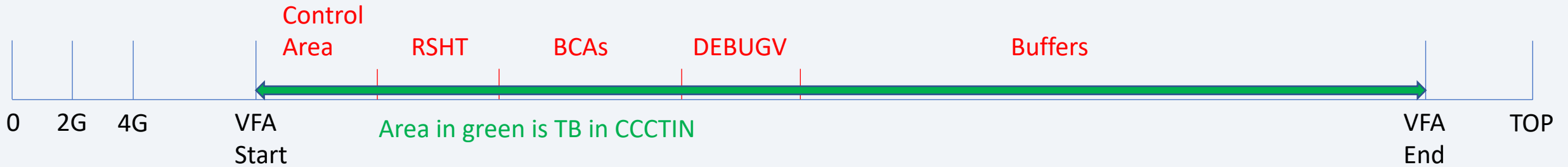
# PJ45401 – No impact to IPL time with large VFA

When VFA is hundreds of gigabytes or terabytes in size, IPL time grows significantly because of the following reasons.

- A test block (TB) is done on every 4 K in VFA to validate whether the memory is good when VFA is reinitialized.
  - TB and initialization are done in CCCTIN.
  - VFA that is 16 gigabytes has 4.2 million 4 K frames to TB.  Time needed to TB is minor.
  - VFA that is 2 terabytes has 537 million 4 K frames to TB.  Time needed to TB is significant.
- Filing of delayed file pending buffers.

**Technical Details**

# PJ45401 – No impact to IPL time with large VFA

## VFA memory area before PJ45401

**Technical Details**

# PJ45401 – No impact to IPL time with large VFA

## Test block (TB) and initialization of VFA area

- CCCTIN no longer does TB and initialization of the entire VFA area.
- TB and initialization are done when the following areas are first used:
  - Record Sharing Hash Table (RSHT)
  - Buffer Control Area (BCA)
  - DEBUGV Area
  - Buffers
- Use of BCAs, DEBUGV, and Buffers is sequential starting from the lowest address.
  - Pointers in the control area are used to determine the next blocks to be TB and initialized.
- New table is created to manage TB and initialization for the RSHT
  - RSHT usage is not sequential.
  - New table is called the RSHT Initialization Table (RSIT).

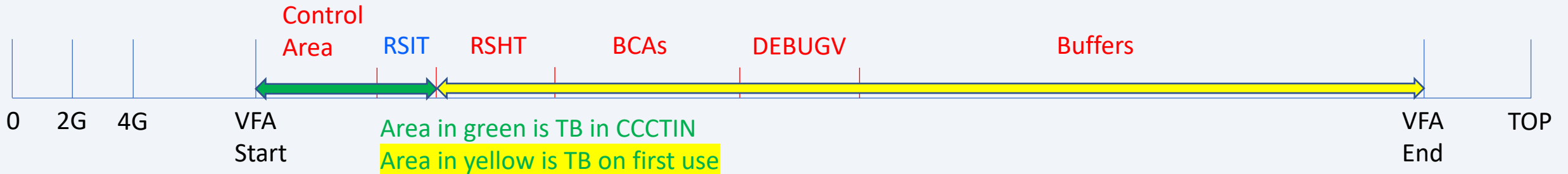# PJ45401 – No impact to IPL time with large VFA

## Test block (TB) and initialization of VFA area (continued)

- CCCTIN does TB and initialize enough BCAs, DEBUGV, and buffers to populate the reserve list.

- Reserve list size is the lesser of the following:
  - 100,000 items
  - Percentage of allocated buffers (default is 5%)
    - Set by ZVFAC DEFINE RES4095-percentage

# PJ45401 – No impact to IPL time with large VFA

## VFA memory area with PJ45401



CCCTIN does TB and initialization enough BCAs, DEBUGV, and buffers to populate the reserve list.

**Technical Details**

# PJ45401 – No impact to IPL time with large VFA

When all the BCAs and buffers have been TB and initialized, the following message is sent.

```
VFAC0089I 16.17.05 UNINITIALIZED VFA BLOCKS ARE EXHAUSTED.
                    ALL VFA BLOCKS ARE IN USE.
```

# PJ45401 – No impact to IPL time with large VFA

## Test results

- Using:
  - z14-701
  - Dedicated LPAR with 1 I-stream
  - VFA size was 1 terabyte
  - Hard IPL with clear
- Before PJ45401
  - 2 minutes 45 seconds for VFA processing in CCCTIN
- With PJ45401
  - 15 seconds for VFA processing in CCCTIN

# PJ45401 – No impact to IPL time with large VFA

## Delayed file pending buffers

- New capability is provided to file all delayed file pending buffers on a time initiated basis.
  - Purpose is to prevent the number of delayed file pending buffers from becoming too large.
    - For example, you might target to allow up to 250,000 VFA delayed file pending buffers.
    - Using ZVFAC displays, determine the growth rate of delayed file pending buffers.
    - Using the growth rate, set the time frequency so that the number of delayed file pending buffers does not become larger than 250,000.
    - Periodically monitor the number of delayed file pending buffers to validate that the 250,000 buffer target is not exceeded.
- Files delay file pending buffers and leaves the data in VFA.
  - Similar to command ZVFAC FILE.

**Technical Details**

# PJ45401 – No impact to IPL time with large VFA

## Delayed file pending buffers

- New capability is disabled when first installed.
- To enable do the following (note: this command is processor unique):

  ➔ ZVFAC DEFINE DFPTIME-*minutes*

- Changes take effect immediately.

# PJ45401 – No impact to IPL time with large VFA

How to determine DPFTIME (frequency to file out delay file pending buffers)?
- First, determine the maximum number of VFA delayed file pending buffers that you want to allow.
- Observe the amount of time required to file delay file pending buffers
  - On cycle from NORM state to 1052 state
  - While ZVFAC FILE is running in NORM state
  - Based on the observations calculate a rate per second for your configuration
- Determine a maximum time period that you are willing to allocate to filing of VFA delayed file pending buffers during an outage.
  - 10 seconds?
  - 30 seconds?
- Maximum target VFA delayed file buffers = observed rate * time period

# PJ45401 – No impact to IPL time with large VFA

How to determine DPFTIME (frequency to file out delay file pending buffers)?

- Determine the growth rate of delayed file pending buffers.
  - Do command ➔ ZVFAC DISP BUFF
  - Command should be done multiple times once a minute over a couple of hours.
  - Best to do the displays after an outage.
- Use the Maximum target VFA delayed file buffers from previous slide.
  - DPFTIME = Maximum target VFA delayed file buffers / growth rate per minute

# PJ45401 – No impact to IPL time with large VFA

## Time initiated message

```
CVF10001I 13.43.57 TIME-INITIATED DELAY FILE PROCESSING IS COMPLETE.
                    THE NUMBER OF RECORDS THAT WERE FILED IS 1000.+
```

# PJ45401 – No impact to IPL time with large VFA

Preserves TPF ability to have a small IPL time when the size of VFA becomes large.

Minimizes downtime for a scheduled and unscheduled outages.

➔ PJ45401 was available for download on June 13, 2019

# Agenda

PJ45401 – No impact to IPL time with large VFA

**→ PJ45598 – Recoverable system heap**

PJ45739 – System recovery boost

Improved handling of sick DASD

PJ45930 – CTL-571 due to RIOSC time out

PJ45934 – RIOSC time out enhancement

PJ46019 – VFA changes for Recoup chain chase

Other enhancements

PJ46026 - Manage recovery log as a resource

PJ45025 - Include name-value pairs in dump messages

PJ45959 - Display core enhancements

What's next

Multi-mod copy

**Problem Statement**

# PJ45598 – Recoverable system heap

Keeping large amounts of data in memory can reduce processing and I/O costs.

➢ Logical record cache is one example.

**Problem Statement**

# PJ45598 – Recoverable system heap

Cost savings are lost after an outage that includes an IPL.

➢ When a system IPLs, some data in memory is lost.

    ➢ Data in system heap is lost.

    ➢ Data in VFA may not be lost.

➢ After the system recovers and network traffic is restarted, additional processing and I/Os are needed because the data is not in memory. Throughput is reduced.

➢ A considerable amount of time is needed to repopulate large amounts of data in memory and to regain the efficiencies that existed before the IPL.

# PJ45598 – Recoverable system heap

- Time and processing power required to rebuild large core tables or caches following an IPL.
    - Rebuilding a 4 KB area is not expensive
    - Rebuilding a 1 MB area is not expensive (equivalent of 256 4 KBs)
    - Rebuilding a 10 MB area is borderline expensive (equivalent of 2,560 4 KBs)
    - Rebuilding a 100 MB area is expensive (equivalent of 25,600 4 KBs)
    - Rebuilding a 1 GB area is very expensive (equivalent of 262,144 4 KBs)
    - Rebuilding a 10 GB area is very expensive (equivalent of 2,621,440 4 KBs)

    Expensive means that it take time and processing power to rebuild the table or cache.  Also, it means that transactional work will require additional processing power to complete without the table or cache being fully populated.

# PJ45598 – Recoverable system heap

## System heap before PJ45598

- 31-bit system heap
    - Below 2 gig in SVM and EVM
    - Virtual memory that is backed by 1 MB frames when needed
    - Data lost when system IPLs
- 64-bit system heap
    - Above 4 gig in SVM and EVM
    - Preallocated area.  Used first.
    - Also, virtual memory that is backed by 1 MB frames when needed.
        - Used when unable to satisfy request in preallocated area.
    - Data lost when system IPLs

# PJ45598 – Recoverable system heap

## System heap memory area before PJ45598



31-bit
System heap

64-bit System heap

0      Virtual      2G      4G      Preallocated      Virtual      VFA      VFA      TOP
       Backed by                                      Backed by    Start    End
       1 MB frames                                    1 MB frames
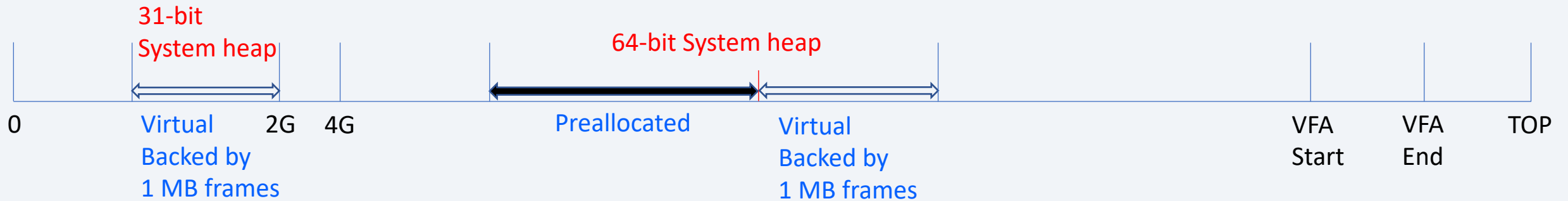
# PJ45598 – Recoverable system heap

## Addition of recoverable system heap

- Infrastructure where large memory tables or cache can reside and the data is not lost on most IPLs.

- Purpose is hold tables and caches that have significant cost to rebuild following an IPL.
    - Tables or caches that are megabytes or gigabytes in size

# PJ45598 – Recoverable system heap

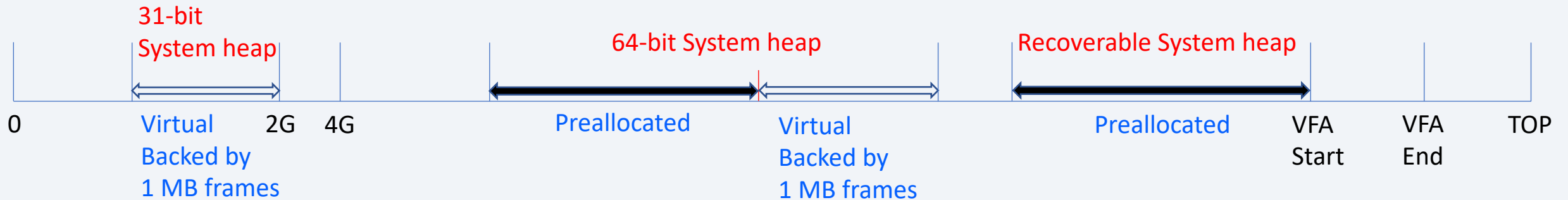## System heap with recoverable system heap

- 31-bit system heap
  - No change in characteristics
- 64-bit system heap
  - No change in characteristics
- Recoverable system heap
  - Above 4 gig in SVM and EVM
  - Preallocated area
  - Data is recovered on most IPLs

# PJ45598 – Recoverable system heap

System heap memory area with recoverable system heap

# PJ45598 – Recoverable system heap

When is data recovered in recoverable system heap?

• There cannot be a change in the memory configuration

• Software IPL (most unplanned outages)

• Hard IPL without clear

When is data not recovered in recoverable system heap?

• Hard IPL with clear

• IPL and the memory configuration has changed

• General rule: when VFA is not recovered, recoverable system heap will also not be recovered.

# PJ45598 – Recoverable system heap

## Size of recoverable system heap

- Size is in a memory configuration
- SIP
  - CORREQ MEMCONFIG=*name*,RSH=*size_in_MBs*
- On TPF
  - Command ➔ ZCTKA ALTER MEMORY *name* RSH-*size_in_MBs*
- Minimum size is 10 MB
- Dump processing (CPSE) internal trace table is allocated in recoverable system heap
  - Internal diagnostics that tracks information about most recent dumps
  - 1 MB in size
  - CMMSET used to access area

# PJ45598 – Recoverable system heap

## System heap APIs today

- GSYSC – obtain system heap
    - Both 31-bit system heap and 64-bit system heap.
    - 8 byte token is required.  Token can be unique.

- RSYSC – release system heap
    - Both 31-bit system heap and 64-bit system heap.

- FSYSC – find system heap that has a unique token.
    - Both 31-bit system heap and 64-bit system heap.

# PJ45598 – Recoverable system heap

## New API - SHEAPC or tpf_sheapc()

- Data is passed using a parameter block
- Supports options: Get, Release, Find, Extend, Swap
- Tokens
  - Must be unique
  - Must be a character string
  - Character string can be up to 64 bytes including the null terminator
- Recoverable system heap buffers must be managed using SHEAPC or tpf_sheapc()
- API can be used for
  - 31-bit system heap buffers
  - 64-bit system heap buffers
  - Recoverable system heap buffers

# PJ45598 – Recoverable system heap

SHEAPC or tpf_sheapc() using the EXTEND action.

- Extend – get another system heap buffer that is associated with the unique token
  - Only allowed for recoverable system heap buffers.
  - SHEAPC ACTION=GET allocates first buffer associated with the unique token.
  - SHEAPC ACTION=EXTEND allocates another buffer that is associated with the unique token.
    - New buffer is NOT contiguous with the original buffer.
    - User responsible for pointing to the new buffer out of the original buffer.
  - SHEAPC ACTION=FIND returns the address of the buffer that was obtained by the GET.
  - Following an IPL all buffers associated with a unique token are managed as one unit.
    - If one buffer is not recovered, all buffers associated with the unique token are not recovered.
    - Otherwise, all buffers associated with the unique token are recovered.

# PJ45598 – Recoverable system heap

SHEAPC or tpf_sheapc() using the EXTEND action.

SHEAPC ACTION=GET using TOKEN A

➔ Address of Buffer 1 is returned
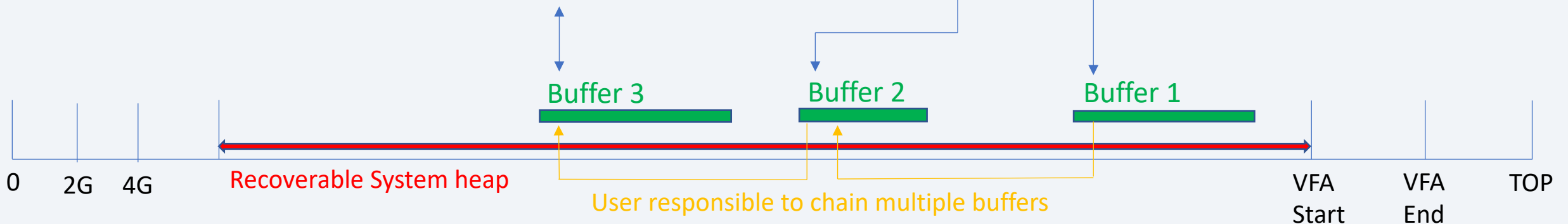
SHEAPC ACTION=EXTEND using TOKEN A

➔ Address of Buffer 2 is returned

SHEAPC ACTION=EXTEND using TOKEN A

➔ Address of Buffer 3 is returned

Buffer 3          Buffer 2          Buffer 1

0     2G     4G          Recoverable System heap          VFA     VFA     TOP
                                                          Start   End

User responsible to chain multiple buffers

# PJ45598 – Recoverable system heap

SHEAPC or tpf_sheapc() using the SWAP action.

- Swap – swap the starting address of two system heap buffers
    - Only allowed for recoverable system heap buffers.
    - Intent is for the user to do:
        - A buffer already exists using TOKEN A.
        - Get a new buffer with a different unique token called TOKEN B.
        - Populate the TOKEN B buffer by:
            - Copying data from the TOKEN A buffer to the TOKEN B buffer, or
            - Direct updates only to the TOKEN B buffer.  Reads first look at the TOKEN B buffer.  If the data is not in the TOKEN B buffer, look in the TOKEN A buffer.
        - Use SHEAPC ACTION=SWAP to swap the buffer addresses for TOKEN A and TOKEN B.
        - Release the original buffer, which is now pointed to by TOKEN B.
    - Purpose is to allow the size of a buffer that is pointed to by a unique token to change without requiring an outage.

# PJ45598 – Recoverable system heap

SHEAPC or tpf_sheapc() using the SWAP action.

SHEAPC ACTION=GET using TOKEN A

➔ Address of Buffer 1 is returned; TOKEN A is associated to Buffer 1.

SHEAPC ACTION=GET using TOKEN B

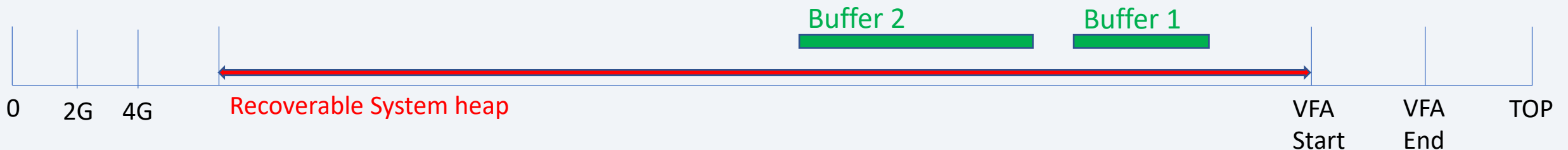➔ Address of Buffer 2 is returned; TOKEN B is associated to Buffer 2.

SHEAPC ACTION=SWAP TOKEN A and TOKEN B

➔ TOKEN A is now associated with Buffer 2.

➔ TOKEN B is now associated with Buffer 1.

Buffer 2          Buffer 1

0     2G     4G          Recoverable System heap                    VFA      VFA     TOP
                                                                     Start    End

# PJ45598 – Recoverable system heap

Allocating a recoverable system heap buffer (assume that the allocation is done following an IPL)

1) Call SHEAPC with the Find option to determine if the buffer is already allocated
   - Recoverable system heap must have a unique token.
   - It is possible that the buffer was recovered during the IPL and that the buffer already exists.
2) If not allocated, call SHEAPC with the Get option

# PJ45598 – Recoverable system heap

## Fragmentation in recoverable system heap

- Fragmentation happens when buffers of varying sizes are obtained and released over and over.

- For 31-bit and 64-bit system heap an IPL cleans up fragmentation.
    - After the IPL, all the space in 31-bit and 64-bit system heap is available.

- For recoverable system heap an IPL might not clean up fragmentation.
    - For example, enough space exists to satisfy a 10 GB request.
    - If the space is not in a continuous buffer, the request is rejected.
    - When the system IPLs and recoverable system heap is recovered, the same request is rejected again because nothing has changed in the recoverable system heap area and the available space is not in a continuous buffer.

# PJ45598 – Recoverable system heap

## How to manage fragmentation in recoverable system heap?

- Display fragmentation
  - Summary display ➜ ZMRSH DISPLAY FRAGMENTATION
  - Detailed display ➜ ZMRSH DISPLAY FRAGMENTATION MAP

- Request that a buffer of a specific size be available on an IPL
  - This might result in one or more buffers in the recoverable system heap area to be lost following an IPL
  - Command ➜ ZMRSH SET MINIMUM-*buffersize_in_MBs*

- Request to reinitialize recoverable system heap on an IPL without reinitializing other recoverable areas such as VFA.
  - Command ➜ZMRSH SET REINIT

# PJ45598 – Recoverable system heap

## Orphaned buffers in recoverable system heap

- A buffer can be obtained and never released.
  - Is the buffer in use?
- For 31-bit and 64-bit system heap, the buffer is no longer used following an IPL.
- For recoverable system heap, the buffer continues to exist following an IPL.
  - If an SHEAPC is called to find the buffer, it is assumed that the buffer is still being used.
  - If no finds are done to locate the address of the buffer, the buffer is orphaned. It is not being used.

# PJ45598 – Recoverable system heap

## How to manage orphaned buffers in recoverable system heap?

- Display orphaned buffers following an IPL
  - Command ➜ ZMRSH DISPLAY ORPHAN

- Release a specific orphaned buffer following an IPL
  - Command ➜ ZMRSH RELEASE ORPHAN TOKEN-*name_of_the_token*

- Release all orphaned buffers following an IPL
  - Best to wait until after the system reaches NORM state and some period of time has passed so that all restart activities have completed and finds have been done for all recoverable system heap buffers.
    - Command ➜ ZMRSH RELEASE ORPHAN
  - An option exists to wait a specified number of minutes before releasing orphans.  The intent is to add this to the ZSTIM table so that it is executed when the system reaches NORM state.
    - Command ➜ ZMRSH RELEASE ORPHAN WAIT-*minutes*

# PJ45598 – Recoverable system heap

Keep processing and I/O efficiencies of having large amounts of data in memory after most unplanned IPLs and some planned IPLs.

➔ PJ45598 was available for download on November 27, 2019

# Agenda

PJ45401 – No impact to IPL time with large VFA

PJ45598 – Recoverable system heap

➡ **PJ45739 – System recovery boost**

Improved handling of sick DASD

PJ45930 – CTL-571 due to RIOSC time out

PJ45934 – RIOSC time out enhancement

PJ46019 – VFA changes for Recoup chain chase

Other enhancements

PJ46026 - Manage recovery log as a resource

PJ45025 - Include name-value pairs in dump messages

PJ45959 - Display core enhancements

What's next

Multi-mod copy

# PJ45739 – System recovery boost (SRB)

Downtime is costly.

➢ IPLing a system takes time.

➢ After reaching NORM state following the outage, pent up demand in the network can result in higher transaction rates for a short period of time.

# PJ45739 – System recovery boost (SRB)

- After an outage when z/TPF is back in NORM state and transactions start flowing, a surge of traffic hits the system. Requests that have been waiting to be processed while the system was unavailable are now processed.
  - For a period of time transaction rates are higher than at steady state.
  - Caches might not be fully populated.
  - Additional capacity might be needed to handle the surge.

# PJ45739 – System recovery boost (SRB)

On an IBM z15 (or later processors) system recovery boost increases capacity when an LPAR IPLs

- Speed boost
  - Increases the LPAR on a subcapacity machine to full capacity.
  - During IPL, speed boost is activated in CCCTIN.
- I-stream boost
  - Uses I-streams above the I-stream cap (i.e. fenced I-streams).
  - During IPL, I-stream boost is activated in restart.  The I-stream cap is not enforced.

# PJ45739 – System recovery boost (SRB)

## How is system recovery boost turned on?

- Boost must be enabled.
- IPL z/TPF on an IBM z15 (or later) processor.
- To enable all boosts
  - Command ➔ ZBOST ENABLE ALL
- To enable only speed boost
  - Command ➔ ZBOST ENABLE SPEED
- To enable only I-stream boost
  - Command ➔ ZBOST ENABLE ISTREAM
- To display status
  - Command ➔ ZBOST DISPLAY

# PJ45739 – System recovery boost (SRB)

## How long is system recovery boost active following an IPL?

- Speed boost is turned off
    - 15 minutes if you IPL and reach NORM state or
    - 5 minutes if you IPL and stay in 1052 state
- I-stream boost starts to be turned off
    - 20 minutes if you IPL and reach NORM state or
    - 10 minutes if you IPL and stay in 1052 state
    - I-stream boost is turned off gradually
        - One I-stream is collapsed.
        - After one minute, a second I-stream is collapsed.
        - Every minute another I-stream is collapsed.
        - I-streams stop collapsing when the I-stream cap is reached.
    - Allows users the ability to change the I-stream cap if the capacity at the I-stream cap is not enough to support the current utilization.

**Technical Details**

# PJ45739 – System recovery boost (SRB)

## How to stop boost early?

- All boosts
  - Command ➜ ZBOST STOP ALL
- Speed boost
  - Command ➜ ZBOST STOP SPEED
- I-stream boost
  - Command ➜ ZBOST STOP ISTREAM

**Value Statement**

# PJ45739 – System recovery boost (SRB)

On an IBM z15 processor reduce the impact of outages.

- Extra capacity is provided for a specified time period at no additional charge.
- Minimize the business impact of an unplanned outage.
- Reduce the time needed for planned outages.

➔ PJ45739 was available for download on November 18, 2019

# Agenda

Reduce outage impact
  PJ45401 – No impact to IPL time with large VFA
  PJ45598 – Recoverable system heap
  PJ45739 – System recovery boost

➡ **PJ45930 – CTL-571 due to RIOSC time out**
➡ **PJ45934 – RIOSC time out enhancement**
  PJ46019 – VFA changes for Recoup chain chase
Other enhancements
  PJ46026 - Manage recovery log as a resource
  PJ45025 - Include name-value pairs in dump messages
  PJ45959 - Display core enhancements
What's next
  Multi-mod copy

**Problem Statement**

# PJ45930 – CTL-571 due to RIOSC time out
# PJ45934 – RIOSC time out enhancement

If a DASD control unit is not responding, a catastrophic CTL-571 LOCKC time out error might be received and cause an outage.

# PJ45930 – CTL-571 due to RIOSC time out
# PJ45934 – RIOSC time out enhancement

## Scenario

Lost interrupt time out = 7 seconds

DASD does not respond for 10 seconds

Multiple I-streams
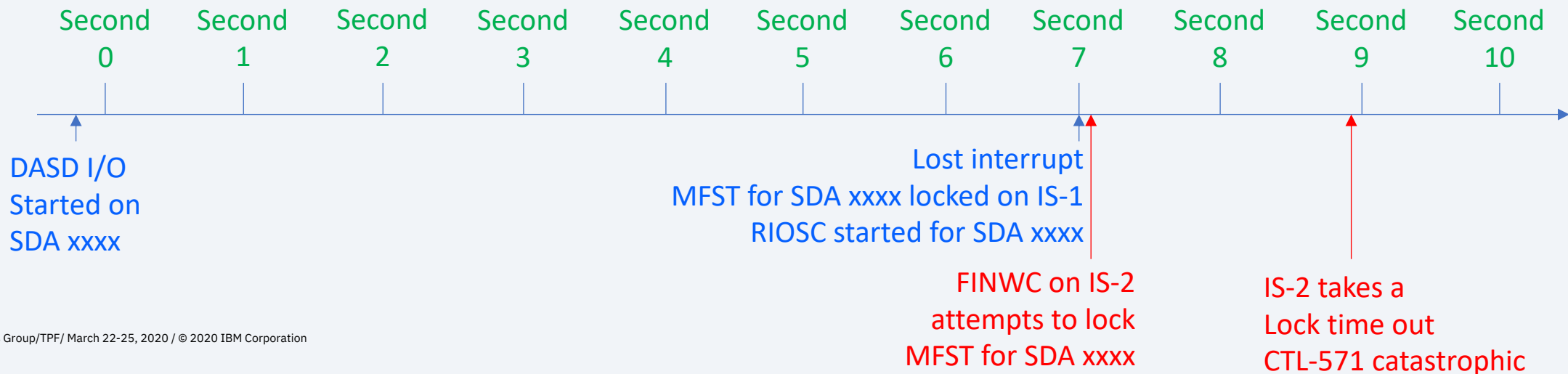
| Second 0 | Second 1 | Second 2 | Second 3 | Second 4 | Second 5 | Second 6 | Second 7 | Second 8 | Second 9 | Second 10 |

DASD I/O
Started on
SDA xxxx

Lost interrupt
MFST for SDA xxxx locked on IS-1
RIOSC started for SDA xxxx

FINWC on IS-2
attempts to lock
MFST for SDA xxxx

IS-2 takes a
Lock time out
CTL-571 catastrophic

# PJ45930 – CTL-571 due to RIOSC time out
# PJ45934 – RIOSC time out enhancement

## Reason for the error

- LOCKC time out based on amount of time required to execute a loop
  - Inner loop is executed 32,767 times
  - Outer loop is executed 131,072 times
  - On a z14 it takes about 1.9 seconds to time out

- RIOSC issues Clear subchannel (CSCH).
  - CSCH goes to the device.  An interrupt is provided when the CSCH is complete.
  - RIOSC executes a Test subchannel (TSCH) loop waiting for the interrupt.
  - If the interrupt has not been received in 2 seconds, the RIOSC returns to the caller.

- DASD lost interrupt logic is holding the MFST lock for the module when RIOSC is called.
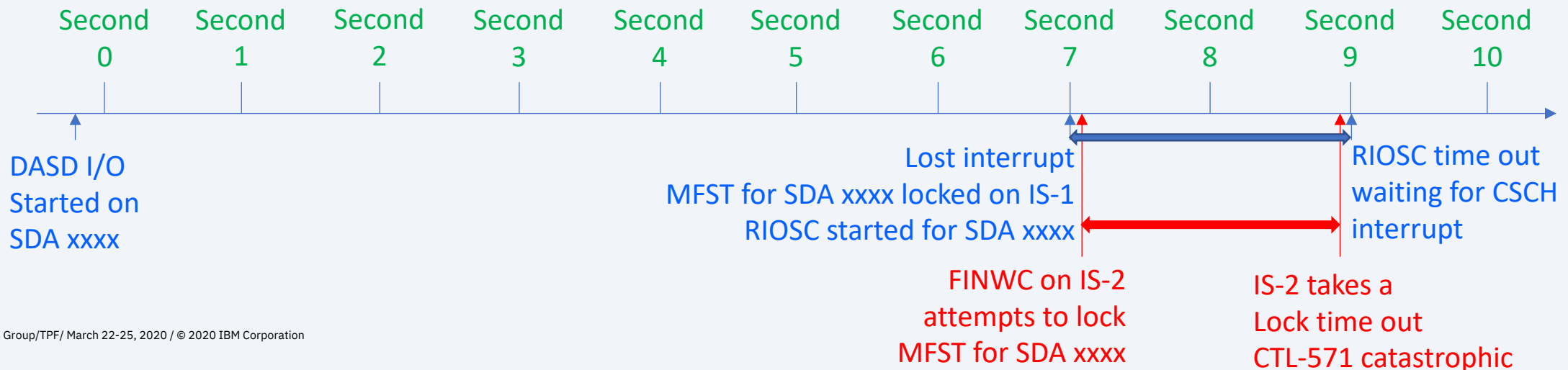
# PJ45930 – CTL-571 due to RIOSC time out
# PJ45934 – RIOSC time out enhancement

## Time outs within z/TPF

- LOCKC time out on a z14 is about 1.9 seconds

- RIOSC time out waiting for CSCH interrupt is 2 seconds
  - MFST lock is held in DASD lost interrupt processing when RIOSC is called.



Second 0   Second 1   Second 2   Second 3   Second 4   Second 5   Second 6   Second 7   Second 8   Second 9   Second 10

DASD I/O Started on SDA xxxx

Lost interrupt
MFST for SDA xxxx locked on IS-1
RIOSC started for SDA xxxx

RIOSC time out waiting for CSCH interrupt

FINWC on IS-2 attempts to lock MFST for SDA xxxx
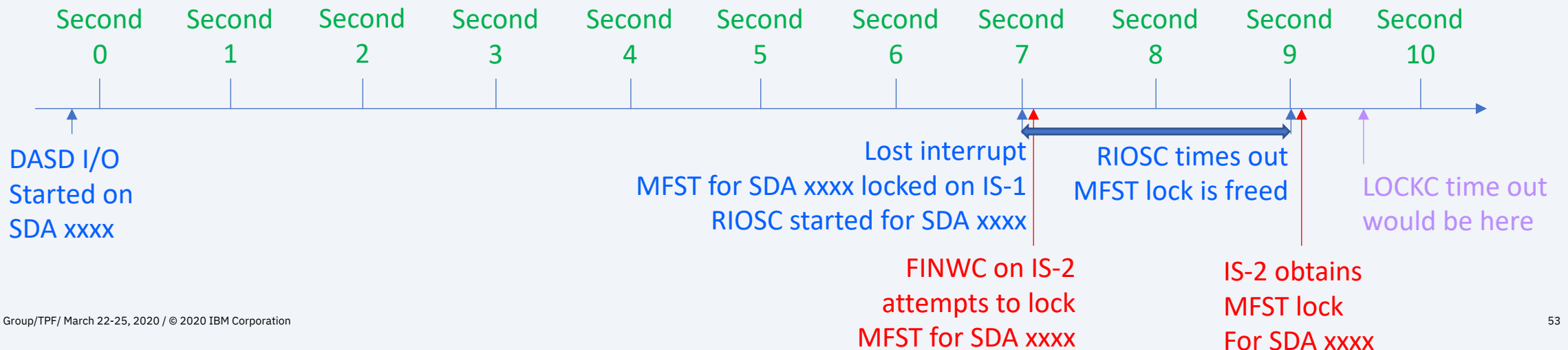
IS-2 takes a Lock time out CTL-571 catastrophic

# PJ45930 – CTL-571 due to RIOSC time out
# PJ45934 – RIOSC time out enhancement

## What does PJ45930 do?

- LOCKC time out has a minimum of 2.5 seconds before a CTL-571 is taken.
- LOCKC continues to execute a predetermined number of loops.
- Time out is the longer of 2.5 seconds or time required to execute the loops.

| Second 0 | Second 1 | Second 2 | Second 3 | Second 4 | Second 5 | Second 6 | Second 7 | Second 8 | Second 9 | Second 10 |
|---|---|---|---|---|---|---|---|---|---|---|

DASD I/O
Started on
SDA xxxx

Lost interrupt
MFST for SDA xxxx locked on IS-1
RIOSC started for SDA xxxx

RIOSC times out
MFST lock is freed

LOCKC time out
would be here

FINWC on IS-2
attempts to lock
MFST for SDA xxxx

IS-2 obtains
MFST lock
For SDA xxxx

53

**Technical Details**

# PJ45930 – CTL-571 due to RIOSC time out
# PJ45934 – RIOSC time out enhancement

## What does PJ45934 do?

- RIOSC time out is specified by the RIOSC caller
  - Default is 2 seconds

- DASD lost interrupt processing specifies a time out of 0 seconds.
  - RIOSC does not wait for CSCH interrupt.  Normal interrupt processing handles the interrupt

| Second 0 | Second 1 | Second 2 | Second 3 | Second 4 | Second 5 | Second 6 | Second 7 | Second 8 | Second 9 | Second 10 |

DASD I/O
Started on
SDA xxxx

Lost interrupt
MFST for SDA xxxx locked on IS-1
RIOSC started for SDA xxxx
MFST for SDA xxxx is unlocked

FINWC on IS-2
locks and then unlocks
the MFST for SDA xxxx

**Value Statement**

# PJ45930 – CTL-571 due to RIOSC time out
# PJ45934 – RIOSC time out enhancement

Prevent an outage with improved error handling when a DASD device is not responding.

➜ PJ45930 was available for download on December 9, 2019
➜ PJ45934 was available for download on January 24, 2020

# Agenda

Reduce outage impact

PJ45401 – No impact to IPL time with large VFA

PJ45598 – Recoverable system heap

PJ45739 – System recovery boost

PJ45930 – CTL-571 due to RIOSC time out

PJ45934 – RIOSC time out enhancement

**➡ PJ46019 – VFA changes for Recoup chain chase**

Other enhancements

PJ46026 - Manage recovery log as a resource

PJ45025 - Include name-value pairs in dump messages

PJ45959 - Display core enhancements

What's next

Multi-mod copy

**Problem Statement**

# PJ46019 – VFA changes for Recoup chain chase

Significant increases in DASD service time on a subset of the DASD modules while Recoup is running might result in an outage due to a catastrophic CTL-6DF no IOBs available error.

# PJ46019 – VFA changes for Recoup chain chase

## Scenario

- Most records are VFA candidates.

- Significant number of VFA delayed file pending buffers.
  - 20% of VFA buffers are delay file pending.

- DASD service time on one control unit increased.
  - Large queues on dupe modules and no queues on the primes.

- Recoup chain chase is active.
  - Uses LODIC before starting a new chain.
  - Has a limited number of ECBs
  - Finds records
    - If the record is retrieved from DASD, it will go to the DASD module with the least queue.

# PJ46019 – VFA changes for Recoup chain chase

What happened?

- Recoup chain chase finds the first record in a chain.

- Record is a VFA candidate.

- If the record is not in VFA, a VFA buffer needs to be obtained.

- VFA buffers are moved from the Aging list to the Reserved list.  If the current data in the buffer is delayed file pending, the current data is filed.
  - File goes to the prime and dupe module.
  - 20% of the buffers are delay file pending.
  - Files to the prime completes.  Files to the dupe takes longer and queues build.

- Find for Recoup first in chain completes and a find on the next in chain is done.

- With enough Recoup ECBs doing finds, queues for delay file writes build and the system might run out of IOBs (CTL-6DF catastrophic error).

# PJ46019 – VFA changes for Recoup chain chase

What does PJ46019 do?

- Recoup chain chase does not put records into VFA.

- A new option is added to the FINDC macro.
  - VFAFILL=YES | NO.
  - Default is VFAFILL=YES.

- When FINDC VFAFILL=NO is used, the following happens
  - If the record is in VFA, the record is obtained from VFA.
  - If the record is not in VFA, the record is retrieved from DASD and it is not put into VFA regardless of the record ID settings in the RIAT.

- When FINDC VFAFILL=YES is used and the record is retrieved from DASD, the record is put into VFA if the record ID attributes say that this is a VFA candidate.

- Recoup chain chase uses FINDC VFAFILL=NO.

# PJ46019 – VFA changes for Recoup chain chase

## PJ46019 technical benefits

- Significantly reduces filing of VFA delay file pending buffers because of Recoup chain chase.
  - Some filing of VFA delay file pending buffer might happen because of Recoup's use of pseudo directories.
- Maintains the efficiency of VFA during Recoup chain chase.
  - Significantly less turn over in the VFA cache.

**Value Statement**

# PJ46019 – VFA changes for Recoup chain chase

Prevent an outage when DASD service time increases significantly by reduced VFA interaction with Recoup chain chase.

Also, maintain VFA efficiency while Recoup chain chase is active.

➔ PJ46401 was available for download on March 11, 2020

# Agenda

Reduce outage impact
  PJ45401 – No impact to IPL time with large VFA
  PJ45598 – Recoverable system heap
  PJ45739 – System recovery boost
Improved handling of sick DASD
  PJ45930 – CTL-571 due to RIOSC time out
  PJ45934 – RIOSC time out enhancement
  PJ46019 – VFA changes for Recoup chain chase

  **PJ46026 - Manage recovery log as a resource**
  PJ45025 - Include name-value pairs in dump messages
  PJ45959 - Display core enhancements
What's next
  Multi-mod copy

**Problem Statement**
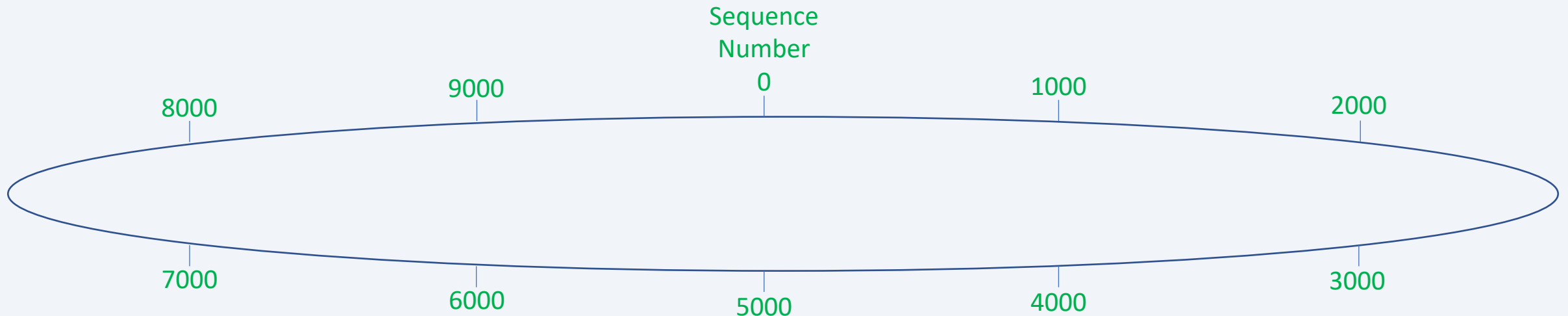
# PJ46026 - Manage recovery log as a resource

Significant use of commit processing might cause a catastrophic CTL-C117 Recovery log full error.  Currently, there is no method to control recovery log usage.

For example, TPFDF Cruise pack uses commit scopes when migrating to FARF6.  If a Cruise pack is done on a large subfile, a large number of commit scopes might be used in a short period of time.  Today, there is no way to control the amount of resource log used.  A CTL-C117 catastrophic error might happen.
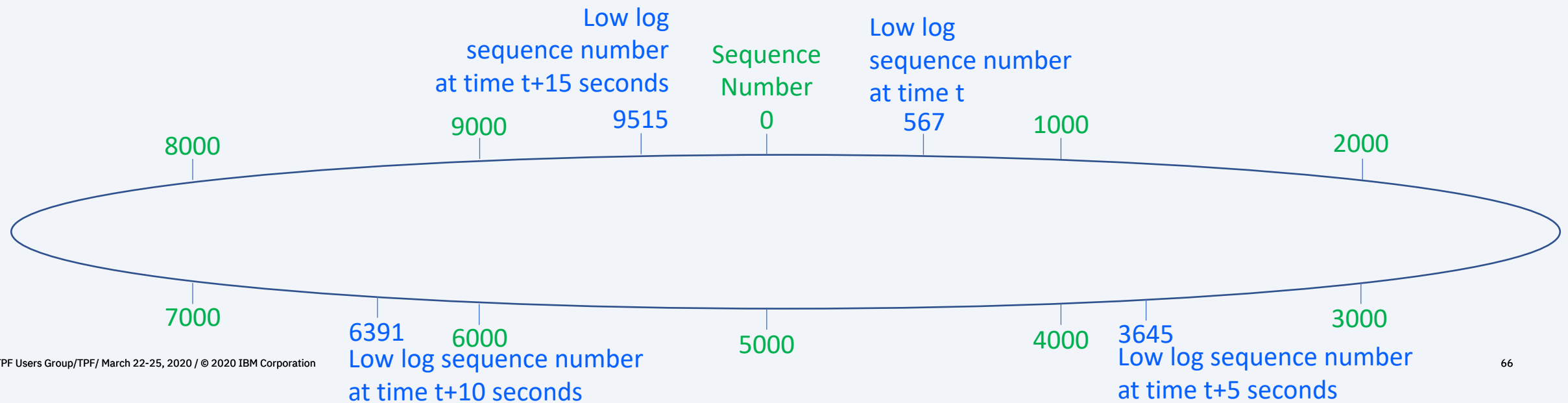
# PJ46026 - Manage recovery log as a resource

- Recovery log is circular.
    - Think of the recovery log as a series of sequence numbers.
    - Low log sequence number is the oldest recoverable data that is still needed.
    - Sequence numbers between the current sequence number and the low log sequence number are either in use or unusable.

- Example: assume there are 10,000 tracks in the recovery log.

Sequence
Number
0

9000        1000

8000              2000

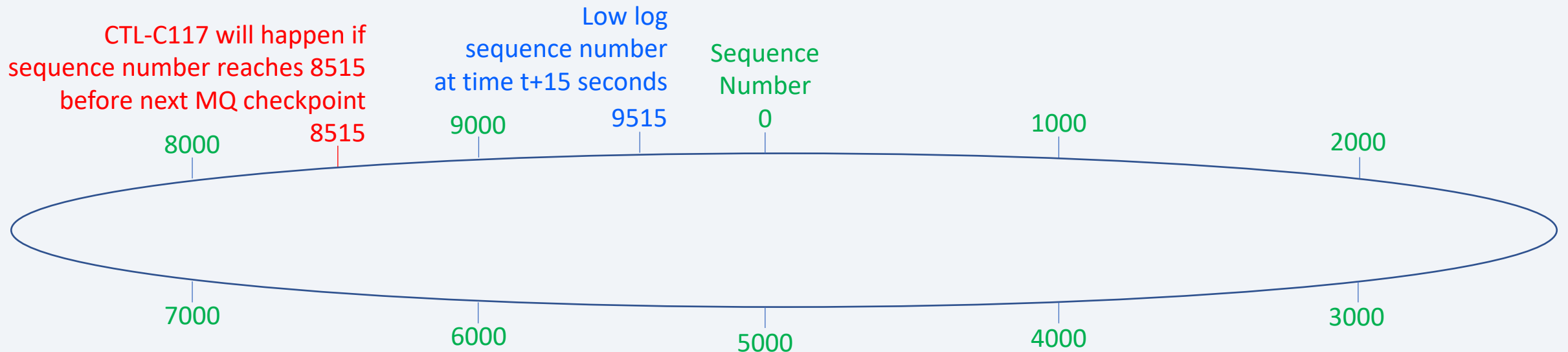7000              3000

6000    5000    4000

# PJ46026 - Manage recovery log as a resource

- MQ checkpointing sets a low log sequence number.
  - This is the current sequence number when MQ checkpointing starts.
- MQ checkpointing happens every 5 seconds
  - At this point a new low log sequence number is set.

Low log sequence number at time t+15 seconds 9515

Sequence Number 0

Low log sequence number at time t 567

8000

9000

1000

2000

7000

6391
Low log sequence number at time t+10 seconds

6000

5000

4000

3645
Low log sequence number at time t+5 seconds

3000

# PJ46026 - Manage recovery log as a resource

- Sequence numbers are used when items are added to the recovery log.
  - MQ PUT / MQ GET of persistent messages
  - A commit scope that has DASD records files and a commit is done.
- A CTL-C117 catastrophic error is taken when the current sequence number minus the low log sequence number represents 90% of the recovery log sequence numbers.
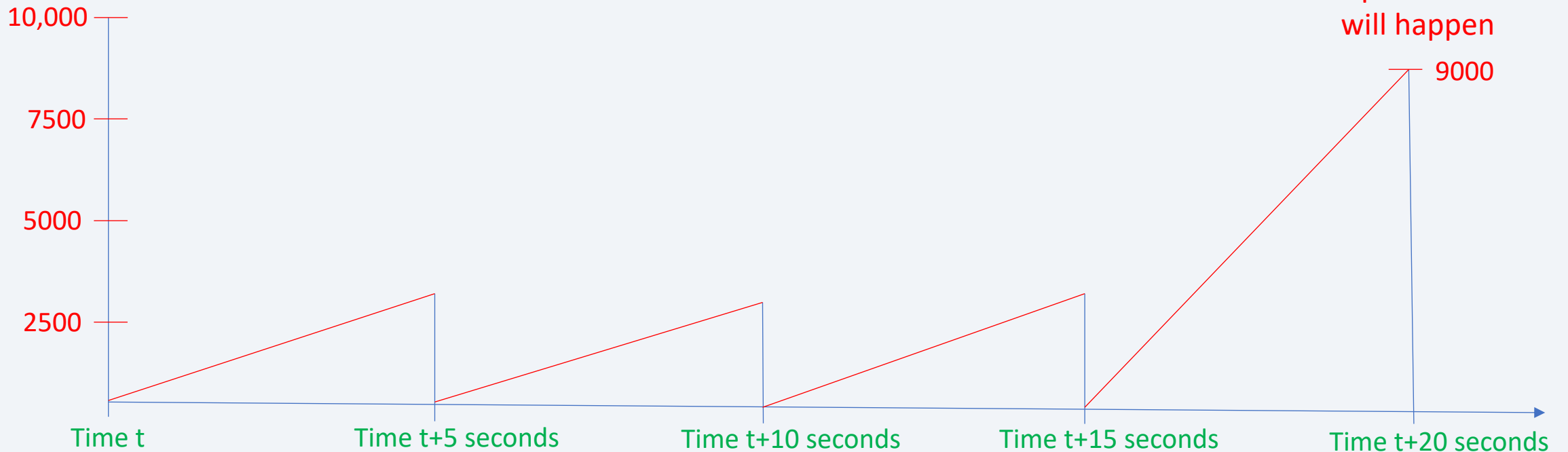


CTL-C117 will happen if sequence number reaches 8515 before next MQ checkpoint

8000    8515

7000    6000    5000    4000    3000

Low log sequence number at time t+15 seconds
9000    9515

Sequence Number
0    1000    2000

# PJ46026 - Manage recovery log as a resource

- Recovery log sequence number usage acts like a sawtooth wave.

# PJ46026 - Manage recovery log as a resource

- No controls exist to stop new work or slow down utilities to prevent unusable recovery log sequence numbers from reaching 90%.

- Because there are no controls, a CTL-C117 catastrophic error might happen if a large number of the following are following the last MQ checkpoint.

  - MQ PUT / GET of persistent messages
  - Commit scopes that have updates to DASD records

**Technical Details**

# PJ46026 - Manage recovery log as a resource

## What changes have been made?

- Before servicing the input list, check the amount of recovery log that is in use.
  - Check is based on the number of tracks in use.
  - Provide ability to set and display shutdown levels for recovery log.
  - Default behavior is to not shutdown the input list due to recovery log usage.
    - A shutdown level must be set (via ZACLV) in order for the input list controls to be effective.

- LODIC service routine checks the amount of recovery log that is in use.
  - Check is based on the number of tracks in use.
  - Provide ability to set and display shutdown levels for each LODIC resource class.
  - Default behavior is for LODIC to not check recovery log usage.
    - A shutdown level must be set (via ZSYSL) in order for LODIC to check recovery log usage.

# PJ46026 - Manage recovery log as a resource

## What changes have been made?

- Status displays contains recovery log information
    - Number of allocated and in use tracks.

- ECBs track recovery log usage
    - Number of bytes used.

```
==> ZSTAT
CSMP0097I 16.22.40 CPU-B SS-BSS  SSU-HPN  IS-01
STAT0041I 16.22.40 SYSTEM STATUS DISPLAY
                 RLOG      IOB    FRAME   COMMON      SWB      XWB      ECB   FRM1MB
ALLOCATED        7040     8192    18000      500    30720    30208     9000    20000
AVAILABLE        7018     8192    17974      494    28731    28671     8600    14228

ACTIVE ECBS               400
DLY/DFR ECBS                0
. . .
```

**Value Statement**

# PJ46026 - Manage recovery log as a resource

Prevent an outage by controlling recovery log usage.

➔ PJ46026 was available for download on July 17, 2020

# Agenda

Reduce outage impact
  PJ45401 – No impact to IPL time with large VFA
  PJ45598 – Recoverable system heap
  PJ45739 – System recovery boost
Improved handling of sick DASD
  PJ45930 – CTL-571 due to RIOSC time out
  PJ45934 – RIOSC time out enhancement
  PJ46019 – VFA changes for Recoup chain chase

  PJ46026 - Manage recovery log as a resource
➡ **PJ45025 - Include name-value pairs in dump messages**
  PJ45959 - Display core enhancements
What's next
  Multi-mod copy

# PJ45025 - Include name-value pairs in dump messages

Name-value pairs can be used to characterize transactions. For example, what user made this request? Where does the user reside?

If an error happens and a SERRC is taken, characteristics that are in name-value pairs can help understand why the problem happened. The only way to get the information is to open the dump; find the name-value pair area; and look in the area. This is cumbersome. Also, this information is lost when a SERRC is NODUMPed.

**Technical Details**

# PJ45025 - Include name-value pairs in dump messages

Name-value pairs can be used to identify such characteristics as

- Source = MomAndPopShop or MonsterShop
- Type = BalanceInquiry or Booking or Shopping
- Origin = Mobile or Web or Terminal

# PJ45025 - Include name-value pairs in dump messages

## What changes have been made?

- Allow up to 5 name-value pairs to be included in dump message.

- 2 names for REST are fixed.
  - REST OperationID (Name = ISrvcName)
  - REST Version (Name = ISrvcVersion)

- Up to 3 names can be defined by the user.
  - Manage the user defined names through the ZCDNV command.


- Include the ECB owner name in the dump message.

**Technical Details**

# PJ45025 - Include name-value pairs in dump messages

Example dump text

CSMP0097I 11.10.17 CPU-B SS-BSS  SSU-HPN  IS-01
CPSE0152E 11.10.17 IS-0001 SS-BSS  SSU-HPN  SE-001298 OPR-I911A21
010000B TRC-QTA2     OWNER-FARES   DATEVAL
QTA2    OBJ-qta2        00000860  LOADSET-BASE
ISrvcName-xmlSrvcGetQry
Source-MonsterTravel
Type-PriceInquiry
Origin-Web

# PJ45025 - Include name-value pairs in dump messages

## How to manage user defined name-value pairs to include in a dump message text?

- All settings are processor unique.
- Initialize the processor unique fixed file record where the names in the name-value pair are saved.
  - Command ➔ ZCDNV INITIALIZE
- Display the names that are included in dump message texts.
  - Command ➔ ZCDNV DISPLAY
- Add a name of a name-value pair that will be included in dump message texts.
  - Command ➔ ZCDNV ADD NAME-*nvpname*
- Stop including a name of a name-value pair in dump message texts.
  - Command ➔ ZCDNV DELETE NAME-*nvpname*

**Value Statement**

# PJ45025 - Include name-value pairs in dump messages

Triage of SERRCs can be improved by viewing key characteristics of the transaction without having to post process the dump.

➜ PJ45025 was available for download on June 17, 2020

# Agenda

Reduce outage impact
  PJ45401 – No impact to IPL time with large VFA
  PJ45598 – Recoverable system heap
  PJ45739 – System recovery boost
Improved handling of sick DASD
  PJ45930 – CTL-571 due to RIOSC time out
  PJ45934 – RIOSC time out enhancement
  PJ46019 – VFA changes for Recoup chain chase

  PJ46026 - Manage recovery log as a resource
  PJ45025 - Include name-value pairs in dump messages
  **PJ45959 - Display core enhancements**
What's next
  Multi-mod copy

# PJ45959 - Display core enhancements

When a command is input to display core, up to 4095 bytes can be displayed.  If data that is beyond 4095 from the start of the area is needed, multiple commands must be entered.  The data might have changed between commands and it might be inconsistent.

Also, when looking at the data, the user must have detailed knowledge of the structure that defines the area.

# PJ45959 - Display core enhancements

## What changes have been made?

- Allow up to 65,535 bytes on a display to the console.

- Provide ability to route the display to a file in the file system.
    - Up to 1,048,575 bytes can be displayed when the display goes to a file.

- Provide ability to use Data Format Description Language (DFDL) to parse a memory table and create a document.  Types of documents that could be created are:

    .csv, .json, .xml, .properties

- When data is written to a file in the file system, the intent is to move the data off of z/TPF.
    - DFDL created files are in UNICODE (not EBCDIC).

# PJ45959 - Display core enhancements

How to determine where to direct the output?

- Optional parameter is provided that specifies that the output should be directed to a file in the file system.
- The file extension determines whether to route the traditional output to a file or to use DFDL.
- File extension .txt specifies to route the traditional output to the file.
  - Example: ➔ ZDCOR 2000.200 FILE-'/tmp/DCOR2000.txt'
- File extension .csv, .json, .xml, or .properties uses DFDL to create the output.
  - Example: ➔ ZDDCA IST FILE-'/tmp/IST.properties'
  - Example: ➔ ZDCNF CMMTS1 FILE-'/tmp/CMMTS1.csv'
  - Example: ➔ ZDGBL IMRDCNTL OUTPUT-'/tmp/IMRDCNTL.xml' COPY-C
- Other file extensions are not allowed.

# PJ45959 - Display core enhancements

How to determine what DFDL schema file to use?

- A new configuration file exists that associates the name of memory area, the type (or how the memory was referenced), the DFDL schema file name, and the DFDL root name.
  - base/cntl/ibmMemorySchemaConfig.xml
  - base/cntl/usrMemorySchemaConfig.xml
- Internally, display core commands call function tpf_mschema_locate() to search both configuration files based on the call type and the name.  For example, command ZDDCA IST will request a search of the configuration file using a type of "DUMP TAG" and a name of "IST".  If a match is found, the DFDL schema file and root name is returned and used.

**Technical Details**

# PJ45959 - Display core enhancements

Several DFDL schema files are available for use.

- Business events control table (IBEVCNTL)
- I-stream status table (IST)
- Module file status table section 1 (FS1)
- Multiple release detection (IMRDCNTL)
- Processor ID table (PID)
- Program attribute table (PAT)
- Tape status table section 1, section 2, and section 3 (TS1, TS2, and TS3)

# PJ45959 - Display core enhancements

## What display commands use this new capability?

- ZDCOR – Display memory
  - Only the .txt file extension will be supported.
- ZDCNF – Display memory based on CINFC label
- ZDDCA – Display memory based on dump tag
- ZDSHP – Display system heap memory based on unique system heap token
- ZDGBL – Display the contents of a format 2 global

If the output is directed to a file and that file already exists, the existing file will be overwritten.

# PJ45959 - Display core enhancements

Example:

➔ZDDCA IST FILE-'/tmp/IST.properties'

CSMP0097I 13.15.14 CPU-B SS-BSS  SSU-HPN  IS-01

DDCA0100I 13.15.14 THE DISPLAY WAS REDIRECTED TO FILE /tmp/IST.properties.

Pieces from IST.properties file:

istuseis=2

istactis=3

...

istcmacutl=5

istcmgputl=4

istcmgpcap=1000

istcmgpeng=1

istcmcapis=2

istcmflooris=2

# PJ45959 - Display core enhancements

Reduce time needed to debug problems because the value of a specific field can be found quickly.

Improve productivity by having DFDL format the output based on the structure that defines the memory table.  Field names from the structure and their current value in the memory table are formatted. As a result, users do not need detailed knowledge of the structure to find the values of specific fields.

➔ PJ45959 was available for download on April 28, 2020

# Agenda

Reduce outage impact

PJ45401 – No impact to IPL time with large VFA

PJ45598 – Recoverable system heap

PJ45739 – System recovery boost

Improved handling of sick DASD

PJ45930 – CTL-571 due to RIOSC time out

PJ45934 – RIOSC time out enhancement

PJ46019 – VFA changes for Recoup chain chase

Other enhancements

PJ46026 - Manage recovery log as a resource

PJ45025 - Include name-value pairs in dump messages

PJ45959 - Display core enhancements

Multi-mod copy

# Disclaimer

Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

# Agenda

Reduce outage impact
  PJ45401 – No impact to IPL time with large VFA
  PJ45598 – Recoverable system heap
  PJ45739 – System recovery boost
Improved handling of sick DASD
  PJ45930 – CTL-571 due to RIOSC time out
  PJ45934 – RIOSC time out enhancement
  PJ46019 – VFA changes for Recoup chain chase
Other enhancements
  PJ46026 - Manage recovery log as a resource
  PJ45025 - Include name-value pairs in dump messages
  PJ45959 - Display core enhancements

➡ **Multi-mod copy**

# Multi-mod copy

Multi-mod copy allows up to 20 copies to run simultaneously. In addition there are controls that restrict the number of copies by complex, control unit, and processor. These controls were needed when this support was released 20 years ago. Because technology has advanced, these controls are too restrictive today.

After a DASD error, parts of the DASD subsystem might be running simplex. Current multi-mod controls extend risk of a complex outage when running simplex.

# Multi-mod copy

Current controls are:

- Maximum number of copies allowed in a z/TPF complex = 20.
- Maximum number of copies allowed on a processor = 10.
- Maximum number of copies allowed on a control unit = 5.


- ZMCPY SET manages the controls.

# Multi-mod copy

Assume:

- A module copy takes 20 minutes to complete.
- A DASD logical subsystem (LSS) has 64 modules on it.
- A DASD LSS takes an error and all the modules go offline.
    - This causes 64 modules to be simplex.
- Multiple mod copy controls are set to the maximum.
    - Copy up is going to a single control unit.
    - 5 copies are allowed to a single control unit.

Under these assumptions it will take 4 hours 20 minutes (260 minutes) complete the copy up.

- If another error happens on one of the modules that is simplex, a complex outage will happen.

# Multi-mod copy

## What changes might be made?

- Allow 100 copies to run simultaneously.

- Controls for complex, control unit, and processor will continue to exist.  However, the maximum allowed value will be 100.
  - Allow users to decide the proper control values.

- If high performance FICON is in use, transport-control-words (TCWs) will be used for copy.
  - Reduces cross talk on the channel when a request is initiated.
  - Intent is to increase copy speed.

# Multi-mod copy

Possible migration issues:

- Unable to do multi-mod copies during code install due to Keypoint V changes.

- Load Keypoint 6.

- Might need to increase lock space in MPLF partition 2.
    - MPLF partition 2 holds the copy locks.
    - Today, z/TPF allocates space for 5 locks.  If you want to do more than 5 copies on a control unit, more than 5 locks are needed in MPLF partition 2.
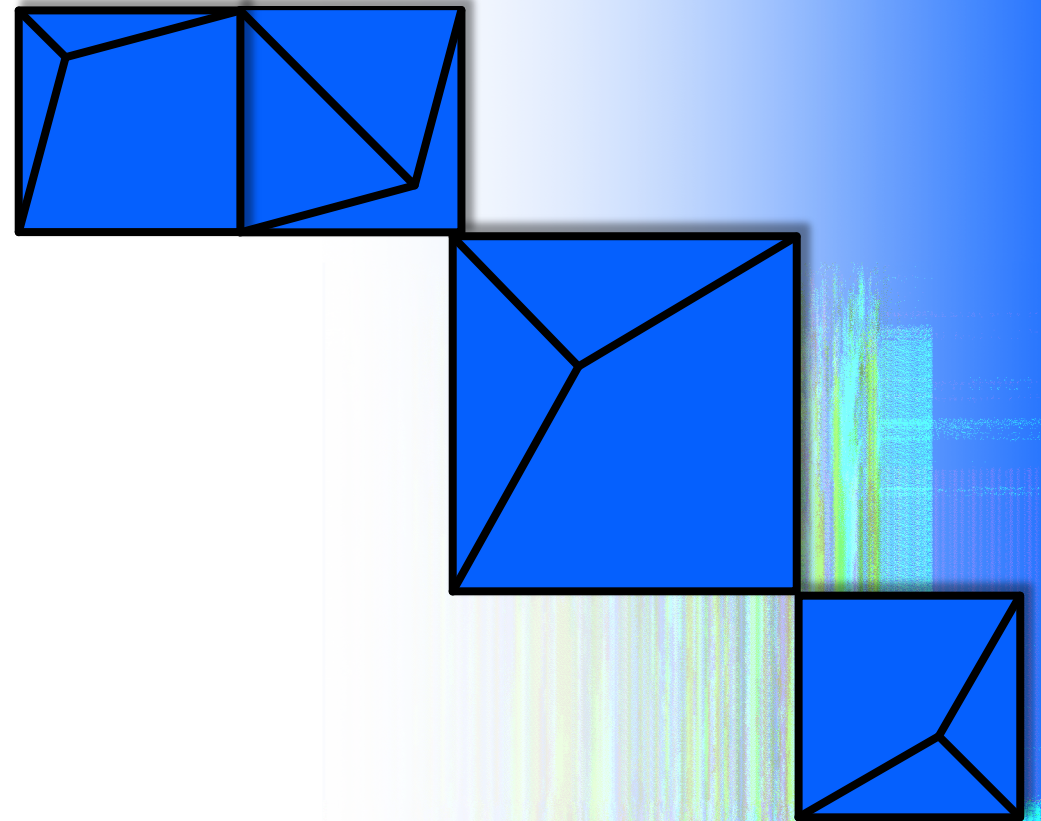
# Multi-mod copy

Reduce risk of a complex outage when recovering from a DASD hardware failure.

# Thank You

Questions? Comments?

# Virtual TPFUG Q&A

Summary of Q&A from the virtual TPFUG event:

| Question | Answer |
|---|---|
| Why not make SRB as default versus we turning it ON? I thought it was said Tuesday that this function was free. | System recovery boost is a new capability on the IBM z/15 processor.  You must purchase and install a z/15 processor in order to have access to system recovery boost.<br><br>A design decision was made to require users to enable System Recovery Boost.  If a problem might happen when Speed Boost or I-stream Boost is turned on, there needs to be a way to not use this capability.  For example, some customers have experienced timing issues when upgrading to a faster machine.  If a similar issue happens when Speed boost is turned off, there needs to be a mechanism to prevent its use.  Also, if a user is not ready for an I-stream to collapse, there needs to be a way to not use I-stream boost.  Therefore, requiring that System Recovery Boost be enabled before using it allows users to control when it is turned on. |

# Virtual TPFUG Q&A

Summary of Q&A from the virtual TPFUG event:

| Question | Answer |
|---|---|
| How is SRB reflected in the ISTREAM report? Will the report show just the cap or will it show using the full amount of engines? | New columns have been added to the I-stream cap report to indicate whether I-stream boost and speed boost are active when the entry was captured. The collected entry will still show the user-defined I-stream cap number.<br><br>Note that the I-stream cap report does not log changes to the number of in-use I-streams, which would change after speed boost has just been deactivated or the system is using HiperDispatch. |

# Virtual TPFUG Q&A

Summary of Q&A from the virtual TPFUG event:

| Question | Answer |
|---|---|
| If we IPL multiple times within e.g. a 30 minute, do we get the various time allowances per IPL?<br><br>We IPL multiple times during maintenance sometimes so would want the benefit on the final IPL before returning to normal operation. | If System Recovery Boost is enabled, Speed Boost and I-stream Boost will be turned on every time that TPF IPLs.  The number of IPLs does not matter.<br><br>When Speed Boost is turned on, it will remain active for 5 minutes if the system stays in 1052 state or 15 minutes if the system is in norm state.  I-stream Boost will start collapsing I-streams after 10 minutes if the system stays in 1052 state or after 20 minutes if the system is in norm state.<br><br>Here is an example of multiple IPLs.  A first IPL is done.  Speed Boost is turned on and the system is cycled to norm state.  After the system has been in norm for 7 minutes a second IPL is done.  The second IPL will turn on Speed Boost again.  If the system is cycled to norm state and stays in norm state, speed boost will be turned off 15 minutes after the second IPL was started. |

# Trademarks

IBM, the IBM logo, ibm.com and Rational are trademarks or registered trademarks of International Business Machines 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 on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

**Notes**

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment.  The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed.  Therefore, no assurance can  be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

All customer examples cited or described in this presentation are presented as illustrations of  the manner in which some customers have used IBM products and the results they may have achieved.  Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States.  IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice.  Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements.  IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products.  Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice.  Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law.  Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.