Handling Limit Conditions in CICS TS

Eugene S Hudders
C\TREK Corporation
PO Box 560069
Montverde FL 34756-0069
eshudders@aol.com
787-397-4150

Agenda

- What are limit conditions?
  - – Who sets the limits?
  - – What do you do when you start hitting/exceeding limit conditions?
- Review different limit conditions
  - – Determine cause
  - – Plan action
- Closing

What Are Limit Conditions?

- Limit conditions are specified target values
  - Represent resources available to CICS TS for execution
  - Sometimes limit conditions can be associated with some type of constraint
- Main purpose for limit conditions is to control the overall use of the assigned resources within CICS TS
  - Poor performance is usually associated with some type of congestion
- Exceeding limit conditions will increase system overhead and transaction wait times that can be reflected in longer response times
- Reaching a limit may affect other limits in the system

What Are Limit Conditions?

- Possible constraints:
  - Hardware
    - CPU cycles
    - Real/Virtual Storage
    - I/O
    - Network
  - Software
    - Data base design
    - Programming

What Are Limit Conditions?

- The limit values specified can be:
  - Inherited
  - A result of a study
  - Performance tuning
  - SWAG
- Limit conditions should be reviewed periodically
  - Stable systems – Once per quarter
  - Dynamic systems – Once a month
  - Exception reporting – On notification

What Are Limit Conditions?

- Some of the more commonly known limit conditions are:
  - MXT
  - MAXACTIVE (TCLASS)
  - SOS (Virtual Storage)
  - VSAM Strings
    - File
    - LSR Pool
  - LSR Buffers
  - Sessions
  - VTAM Receive Any RPLs

What Do You Do When You Reach a Limit Condition?

- Reaching a limit condition is not necessarily a problem that requires attention
  - However, not approaching or reaching limit conditions may indicate over allocation of resources
- Consider the possibility that reaching a particular limit may be an indication that attention is required

What Do You Do When You Reach a Limit Condition?

- Action to be taken depend on several factors:
  - Potential Exposure
  - Tolerance Level
    - Zero tolerance
    - “Rule of Thumb” (ROT) Tolerance
      - 1, 3 or 5% acceptance level
    - “Don’t Care” Tolerance
      - Usually installations that do not worry about performance

What Do You Do When You Reach a Limit Condition?

- Frequency of Limit Conditions
  - – Never – defined limits are never exceeded
    - Can identify over-allocated resources
    - Depending on the resource, this may or may not represent wasted resources that may be of use elsewhere
      - – Over-allocation of key length in LSR Pool – low cost
      - – Over-allocation of NSR strings – high cost

What Do You Do When You Reach a Limit Condition?

- Frequency of Limit Conditions
  - Infrequent – defined limits are occasionally exceeded
    - Can be interpreted as a “good sign” that the defined limit is using the allocated resources appropriately
    - Can also be a sign that you need to update the defined limit condition

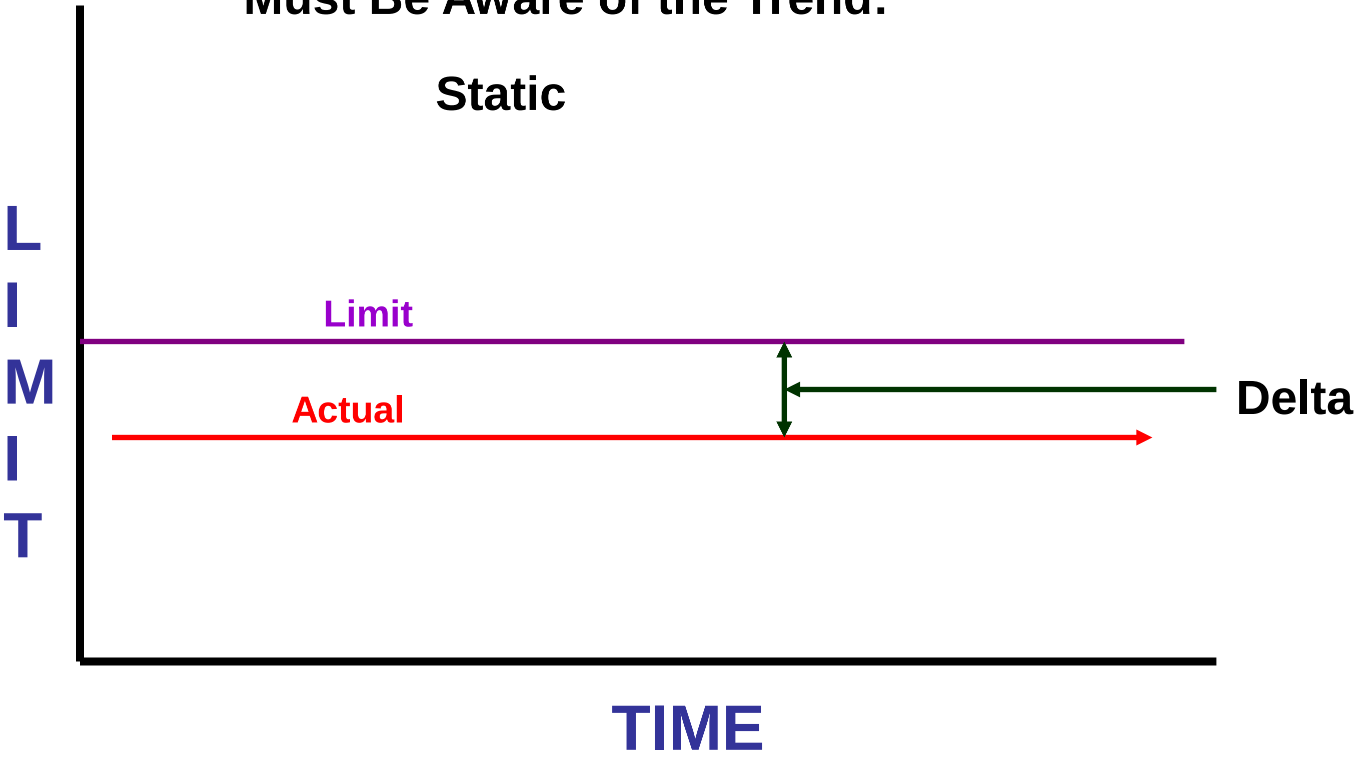
What Do You Do When You Reach a Limit Condition?

- Frequency of Limit Conditions
  - Frequent – a defined limit is consistently exceeded
    - Usually measured as a percentage (e.g., 3%)
    - If the percentage is higher (e.g., > 5%), you may want to take immediate action to correct
    - Frequently exceeding a particular limit condition may affect other limits in the system
      - Lack of virtual storage conditions may indirectly/directly cause additional program compressions that may result in additional program fetching activity and/or cancelation of tasks in the system (DTIMOUT and SPURGE) as in the case of system under stress

What Do You Do When You Reach a Limit Condition?

Must Be Aware of the Trend:

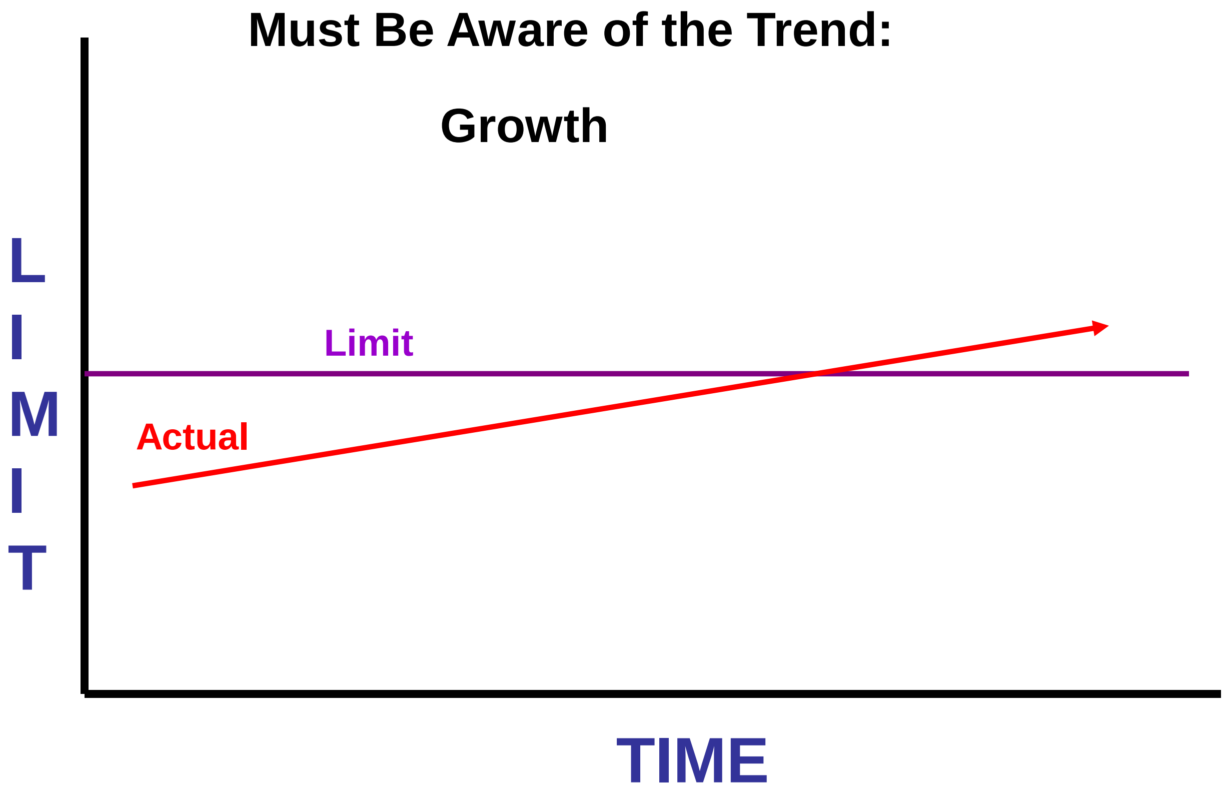
Static



What Do You Do When You Reach a Limit Condition?

- Static Trend
  - Usually represents a trend in a system that has little or no changes and/or a steady transaction volume
    - A “no/limited growth” system
  - The Delta represents the difference from the average peak use to the maximum limit defined
    - A large Delta may be an indication of wasted resources
  - Usually changes in this type of system are to lower the limit condition when resources can be better used elsewhere

What Do You Do When You Reach a Limit Condition?



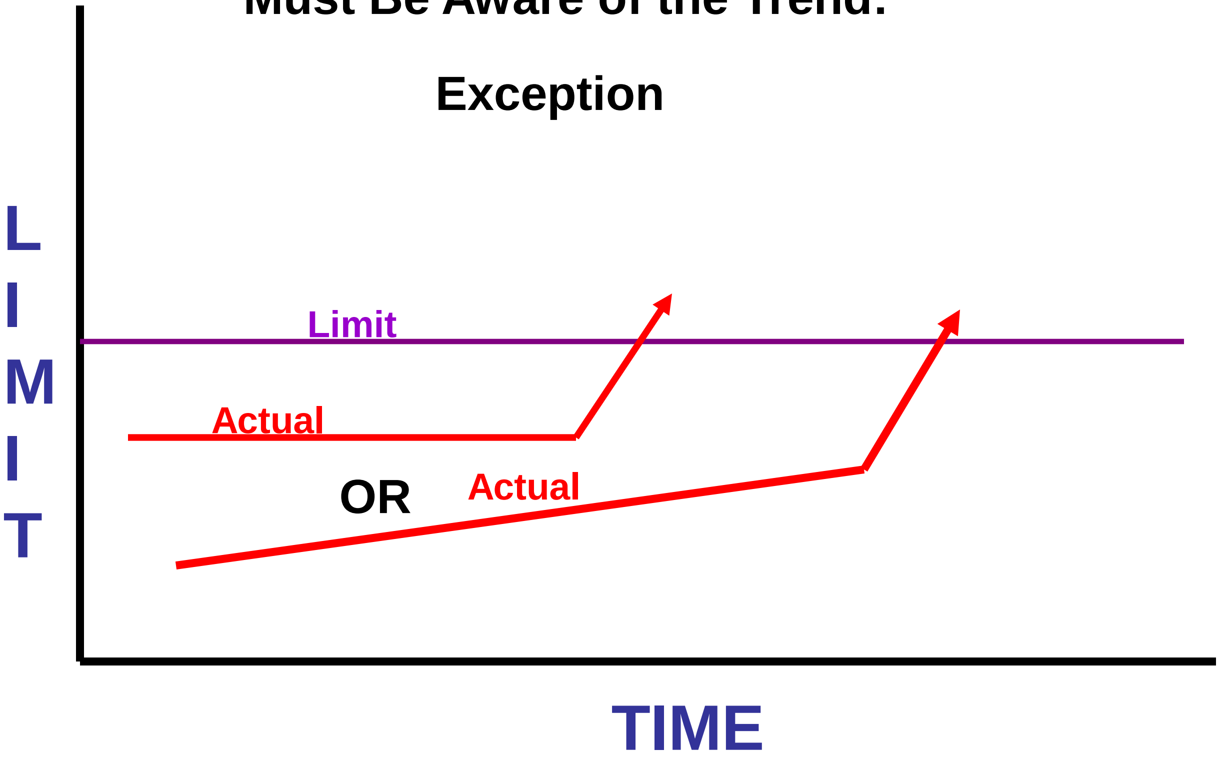
What Do You Do When You Reach a Limit Condition?

- Growth Trend
  - Represents a system that new applications, functions or users are being added
  - The volume can be used to measure a growth percentage that can be used to project approximately when the defined limit will be exceeded
  - Usually changes to this type of system are to adjust the limit definition to a higher value

What Do You Do When You Reach a Limit Condition?

Must Be Aware of the Trend:

Exception



What Do You Do When You Reach a Limit Condition?

- Exception Trend
  - Represents either a static or growth system that has been running well and suddenly a limit condition is exceeded
  - This type of occurrence requires further analysis to determine possible causes of this condition
    - System changes
    - Programming changes
    - External causes
  - If other factors involved, clear these first before increasing limit definition

What Do You Do When You Reach a Limit Condition?

- Summary
  - In all cases where you exceed the defined limit condition, you should analyze the cause
  - Solutions can vary:
    - Easiest solution is to raise the limit
      - But this may not be the answer
    - Make changes to other resources to improve length of time a resource is held
      - Residency time
    - Check for potential errors or recent system/programming changes

MXT

- MXT is a SIT parameter used to limit the number of user transactions that can be in the system at any one time
  - Used to control the overall use of resources in the system such as virtual/real storage
  - Can be set to a minimum of 1 to a maximum of 999 (Default = 5)
- Each MXT slot has a Performance Block (PB) associated with it used by WLM

MXT

- PBs are scanned by WLM
  - Every 10 cycles if region controls used (e.g., velocity goals)
  - Every 250 ms. if response time goals are used (e.g., 85% of transactions finish in less than ½ second)
- MXT allocations
  - Under allocation, MXT occurrences
    - Potential for an “artificial bottleneck”
  - Over allocation, additional CPU overhead

MXT

- Trend
  - Growth
    - increase MXT value
  - Exception
    - Check
      - For other bottlenecks or limit conditions in system
      - For system/application changes
        - » Fix these first
        - » Check to see if condition goes away when corrected
    - If all verified, then increase MXT

MXT

- Recommendation
  - Assign MXT to be around 50 to 70% of peak tasks
  - Example:
    - Peak 60 tasks
    - Recommended MXT = 85 to 120 tasks
    - MXT around 100
  - Beware of new applications where the transactions require additional MXT slots
    - IIOP
    - Web

MAXACTIVE

- TCLASS provide better granularity for controlling tasks for which MXT does not fit
  - Control high resource consumers
  - Single thread resource
  - Control transactions below the line
  - Control sessions (“Sympathy Sickness”)
  - Sometimes used as a “Safety Valve”

MAXACTIVE

- Using TCLASS places some overhead on the system
  - At transaction initiation to locate the appropriate TCLASS
    - Determine if available TCLASS slot
    - Update count (plus)
  - At transaction termination to locate the appropriate TCLASS
    - Update count (minus)

MAXACTIVE

- Major issues with TCLASS are:
  - Maybe inherited
  - Rarely reviewed unless problems occurs
  - Limit conditions may no longer be valid
      - Faster processors
      - More real storage
      - Faster I/O
      - Virtual storage constraint relief
  - PURGETHRESH not used due to some confusion on how to use it
    - MAXACTIVE = 50
    - PURGETHRESH = 10 (queue is n-1)
      - Once queue depth reaches 9, any transaction after that is purged
  - Potential for an “artificial bottleneck”

MAXACTIVE

- Trend
  - – Growth
    - increase MAXACTIVE value
  - – Exception
    - Check
      - – For other bottlenecks or limit conditions in system
      - – For system/application changes
        - » Fix these first
        - » Check to see if condition goes away when corrected
    - If all verified, then increase MAXACTIVE

MAXACTIVE

- Recommendation
  - Assign MAXACTIVE to be around 60 to 70% of peak tasks
  - Example:
    - Peak 60 tasks
    - Recommended MAXACTIVE = 86 to 100 tasks
    - MAXACTIVE around 90
  - Eliminate TCLASS, if not needed

SOS

- SOS is usually a reflection of insufficient virtual storage in the (E) DSAs or GDSA
  - DFHSM0133 – SOS above the line
  - DFHSM0131 – SOS below the line
  - DFHSM0606 – Insufficient storage above the bar
- Main throttles used to control SOS conditions are:
  - MXT
  - TCLASS

SOS

- General solutions to SOS conditions are:
  - Increase the (E) DSALIM (if possible)
    - Can be done via CEMT, if sufficient Region size available
      - Make sure you have a large REGION size (e.g., 0M)
    - Can probably be done above the line (EDSALIM)
    - May not be possible below the line (DSALIM)
      - “VS is free but sometimes you can’t buy any!”
  - Increase the MEMLIMIT
    - Requires a CICS recycling
    - May not be an issue today but may be in the future as more use of above the bar storage is made available

SOS

- Other solutions to resolve SOS conditions
  - Lower MXT
  - Use TCLASS to control storage “hogs”
  - Split CICS (more MRO)
  - Use DTIMOUT and SPURGE
  - Convert 24-bit programs to 31-bit
  - Eliminate unneeded resource definitions
  - Tune system by reducing the task residency time – Reduce Physical I/O and CPU utilization
    - LSR tuning
    - DB2 thread reuse/threadsafe
    - NSR buffering
    - DFHTEMP buffering

SOS

- Trend
  - Growth
    - Increase (E) DSALIM value(s) (if possible)
    - Consider using MRO to split workload
    - Reduce transaction residency time (tuning)
  - Exception
    - Check
      - For other bottlenecks or limit conditions in system
      - For system/application changes
        - » Fix these first
        - » Check to see if condition goes away when corrected
    - If all verified, then increase (E) DSALIM

SOS

- Recommendation
  - Verify that your virtual storage usage (above/below) is generally below 80%
    - If consistently above 80%, consider increasing (E) DSALIM
    - Consider tuning before SOS becomes an issue
  - Ensure safety valve capability (CEMT) by having a large REGION specified
  - Do not be afraid to allocate more (E) DSALIM than required
    - There is nothing wrong with (E) DSA use in the 30 to 60% range

VSAM Strings

- VSAM strings control the number of concurrent I/O requests you allow against a particular resource
  - NSR
    - Controls number of concurrent requests against the file
    - Cost = each string requires a BUFND and BUFNI (if applicable)

VSAM Strings

- VSAM strings control the number of concurrent I/O requests you allow against a particular resource
  - LSR
    - File level
      - Controls number of concurrent requests against the file
      - Cost = none at the file level
      - Maximum – 255 per file
    - Pool level
      - Controls the overall number of concurrent requests from all files assigned to the pool
      - Cost = minor
      - Maximum – 255 per pool

VSAM Strings

- String assignments tend to be over allocated
  - NSR files with low volume defined with many strings
  - LSR pools defined with maximum number of strings
- String assignment should consider
  - File activity
  - Transaction activity against the file
  - Duration of the normal requests
    - Browse versus read directs
    - Updates
- Objective is to be in the 50 to 70% range
  - Remember that 20% of strings are reserved for read-only requests

VSAM Strings

- String assignment for “add-only” ESDS files
  - Should be set to 1 to reduce system overhead caused by exclusive control overhead
    - String values are handled at the CICS level
      - So, if string is not available on the CICS request, the task is set to wait
      - When a string is available, then the waiting task is dispatched
    - Exclusive control is handled at the VSAM level
      - So, if string is available, CICS passes the request to VSAM
      - If VSAM finds that there is an exclusive control, the request is rejected to CICS to handle and re-schedule
      - The task is dispatched when the condition is resolved
- If ESDS file is both write and read (e.g., 80/20), then consider defining two separate FCT definitions for the file, one for writing and one for reading

VSAM Strings

- Trend
  - Growth
    - Increase number of strings
    - Consider using MRO to split workload
    - Reduce transaction residency time (tuning)
  - Exception
    - Check
      - For other bottlenecks or limit conditions in system
      - For system/application changes
        - Fix these first
        - Check to see if condition goes away when corrected
      - Ensure look-aside hit ratios are being met

VSAM Strings

- Recommendations
  - Verify that the affected file is achieving installation look-aside hit ratios
    - NSR
      - Ensure that there is proper buffering for the index
      - If not, fix this before increasing number of strings
      - If yes, increase strings
    - LSR – look-aside hit ratio percentages are applicable to the buffer size and not to a specific file (unless only file using buffer)
      - File
        - Ensure that there is proper buffering for the appropriate buffer size in the data and index
        - If not, fix this before increasing number of strings
        - If yes, increase strings but be careful with pool string usage

VSAM Strings

- Recommendations
  - Verify that the affected file is achieving installation look-aside hit ratios
    - LSR– look-aside hit ratio percentages are applicable to the buffer size and not to a specific file (unless only file using buffer)
      - Pool
        - » Ensure that there is proper buffering for all buffer sizes in the data and index pools
        - » If not, fix this before increasing number of strings
        - » If yes, increase strings

LSR Buffers

- Not usually considered a limit condition
- LSR Buffers are used to hold data and index information requested by different tasks
  - Can have a separate data and index pools
    - Reduces contention for like CI sizes
    - Allows for separate tuning of data and index
  - Uses an LRU algorithm to determine which buffers are to be overlaid with new records
    - This tends to be a self-tuning mechanism where older unused information is replaced with more current information
  - Uses a hashing search algorithm
    - Eliminates the concern of large buffer allocation

LSR Buffers

- LSR Buffers are used to hold data and index information requested by different tasks
  - Resources are “shared” by all files in pool
  - Maximum buffers – 32K (data and/or index)
    - Can use expanded storage buffer allocations if more than 32K needed
    - Note: There is a price to pay for using expanded storage buffers
  - Maximum 8 pools available

LSR Buffers

- Trend
  - Growth
    - Increase number of buffers
    - Consider using threadsafe VSAM
      - Use more pools to provide overlap
    - Reduce transaction residency time (tuning)
  - Exception
    - Check
      - For other bottlenecks or limit conditions in system
      - For system/application changes
        - Example, change a browse operation for an LSR file without issuing the ENDBR
        - Fix these first
        - Check to see if condition goes away when corrected
      - Ensure look-aside hit ratios are being met

LSR Buffers

- Recommendations
  - Verify that the buffers are achieving installation look-aside hit ratios
    - Suggested look-aside hit ratios
      - Data – 80%+
      - Index – 95% +
      - Overall – 93%+
      - Add buffers where appropriate
      - Make sure you have an ROI set to know when to stop adding buffers
    - Buffer monopolization
      - What metric did you use to determine monopolization?
        - » Add more buffers
        - » Segregate file to separate pool

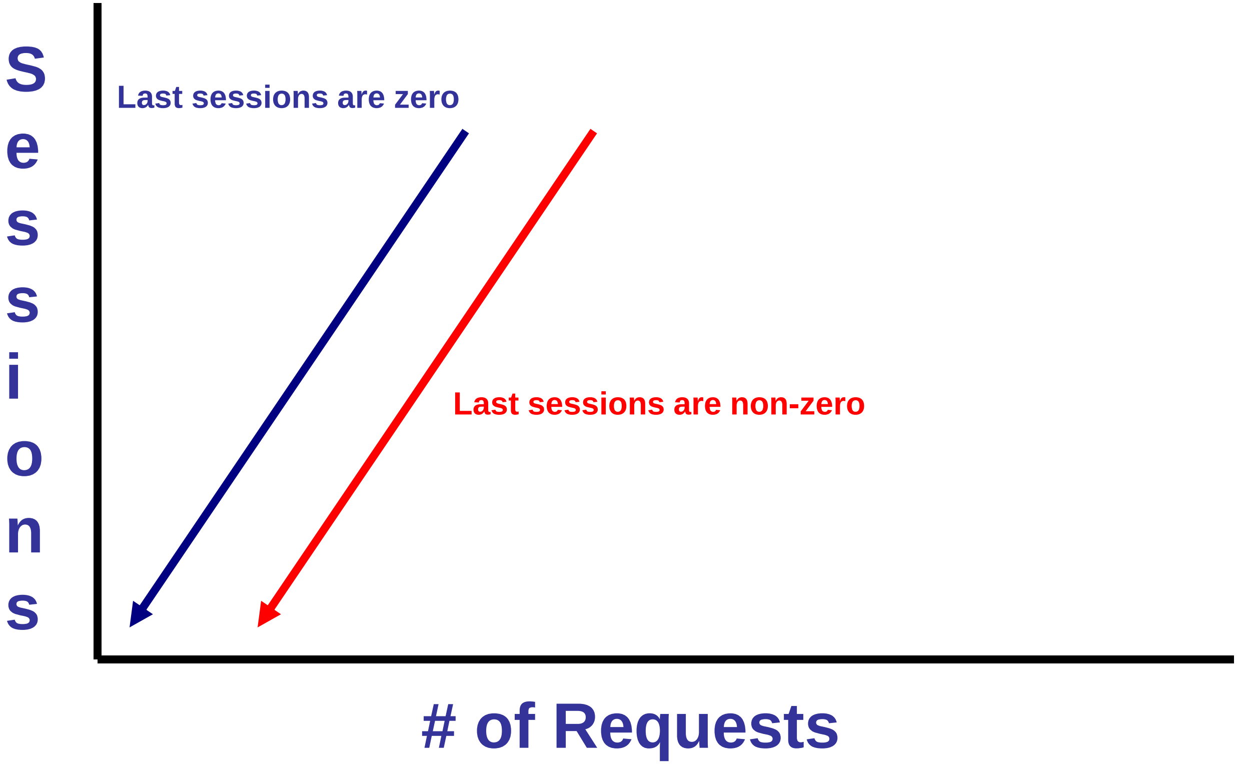
Sessions

- Sessions are used for communication with other systems
  - Usually define send and receive sessions
    - Send SYSA = Receive SYSB
    - Receive SYSA = Send SYSB
  - Problems with sessions are not necessarily seen as there is no message or condition flag raised

Sessions

- Sessions are used for communication with other systems
  - Some type of sessions are usually over-allocated
    - Usually in the second pair of transaction direction
    - Cost is mainly virtual storage
      - Approximately around 600 bytes per unused session
  - Session use is done via a top to bottom allocation algorithm

Sessions


Sessions
Last sessions are zero
Last sessions are non-zero
# of Requests

Sessions

- Trend
  - Growth
    - Increase number of sessions
    - Reduce transaction residency time (tuning)
  - Exception
    - Check
      - For other bottlenecks or limit conditions in this system and connected system
      - For any system/application changes
        - » Fix these first
        - » Check to see if condition goes away when corrected

Sessions

- Recommendations
  - Verify that there are some very low use or unused sessions at the end of the send or receive list
    - Add or delete sessions as appropriate
      - Trade off to control VS use

Receive Any RPLs

- Used for communications with VTAM
  - Used to control the number of concurrent sessions that may be active between CICS and VTAM
  - SIT parameter RAPOOL is used to define the number of Receive Any buffers are available for this purpose
    - 1st RAPOOL parameter is used for non-HPO systems
      - Usually 1.5 times the peak transaction rate (Default=50)
    - 2nd RAPOOL parameter is used for HPO systems
      - Usually small (e.g., LT 5) (Default=1)
    - Actual number of RPLs active depend on MXT setting and number of active tasks

Receive Any RPLs

- Used for communications with VTAM
  - RAPOOL(20,10) and MXT=50
    - Non-HPO system
      - If active tasks 45, then Receive Any=5
      - If active tasks 30, then Receive Any=20
    - HPO system
      - If active tasks 45, then Receive Any=5
      - If active tasks 30, then Receive Any=10
  - Measured by a High Water Mark (HWM)
    - May not be an effective measurement
      - What was the next tier?

Receive Any RPLs

- Used for communications with VTAM
  - RAPOOL importance has gone down because of more TCP/IP use
  - Excess use of RAPOOL is virtual storage
    - Need a RACE control block
    - Need RAIA for each entry
    - RAIA size is controlled by the RAMAX value
  - Not used for MRO sessions
  - If needed, use HPO

Closing

- When dealing with a limit condition
  - Analyze the type of system on which the situation occurred – the trend
    - Static
    - Growth
    - Exception
  - Increasing the set limit may not resolve the underlying situation
    - You may just be deferring the problem

Closing

QUESTIONS?

Thank you for your attention!