E71

IMS SYSPLEX Performance Considerations

Dave Viguers

St. Louis, MO **Sept. 30 - Oct. 3, 2002**

RETURN TO INDEX

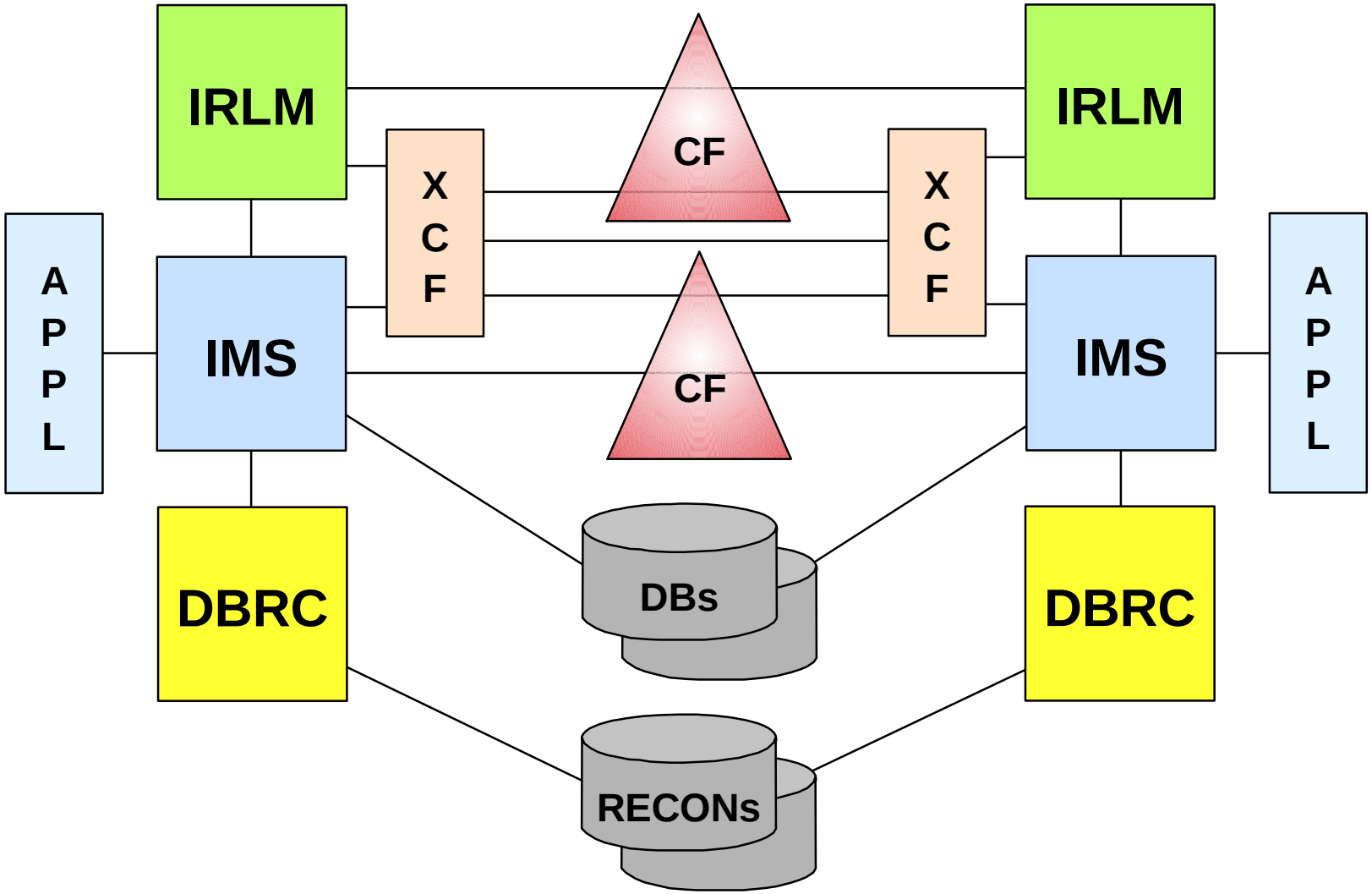
Session Description

- Now that many installations are sysplex enabled or are planning to implement IMS in a sysplex environment in the near future, we need to look beyond the standard IMS tuning considerations and take into consideration the effects of the sysplex components. This presentation looks at the IMS use of sysplex resources and what can be done to make your IMS perform its best in this environment.

Topics

- Sysplex Component Overview
- CF Structures
  - IRLM Lock Structure
  - OSAM / VSAM Cache Structures
  - SVSO Cache Structures
  - SMQ List Structures
  - MVS Logger List Structures
- Changes to IMS behavior
  - Scheduling
  - Local vs. Global processing
- Summary

Sysplex Components



Sysplex Component notes

- CF Structures
  - Use Fast engines
    - Normally the same as Operating Sys but...
  - Multiple links
    - For availability AND performance
  - Beware of dynamic dispatching for CF LPAR
  - If a particular IMS has a predominance of access then consider placing CF structure(s) on same physical machine
- XCF signaling paths
  - Multiple links - Multiple types (CTC & STR)
  - Let system determine which to use

IRLM

- IRLM
  - PC=NO
    - Less CPU but lock table in ECSA
  - DEADLOK=
    - Use low value to resolve quickly
    - Values less than 1 sec can be specified
  - F IRLM,SET,TIMEOUT=nnnn,dbms
    - Use to determine when long lock waits are occurring and why
  - Set dispatching priority high
  - Don't use trace unless necessary

IRLM Lock Structure

- Size
  - 32 or 64 MB good starting point
  - 1/2 is Lock Table Entries by default
  - Minimize False Contention
- IRLM parms
  - MAXUSRS=
    - # of IRLM's which will connect
    - use 2-7 to keep size of LTE to 2 bytes
      - maximizes number of entries in space
  - LTE=
    - Overrides default calculation above

VSAM Cache Structure

- Directory only
  - No data caching
  - Directory reclaims if not large enough
- Sizing
  - Total # VSAM buffers including hiperspace
  - times # shared IMS (assuming shared vspec)
  - times 200
  - plus some fudge factor
  - don't forget shared batch
- Alter supported if increased size required

OSAM Cache Structure

- Directory only - same guidelines as VSAM
- Data Caching (store through cache)
  - By OSAM subpool
    - Isolate DB's to subpools as appropriate
  - Cache All
    - Every block read is put in structure
    - Extra CPU to put in structure
    - Might be good for small or small locality of reference databases
    - Think hard before choosing this option

OSAM Cache Structure

- Data Caching (continued)
  - Cache Changed
    - Blocks only written to structure if updated
    - Might be good where same blocks updated / referenced constantly by multiple IMS's
    - Prevents re-reading from DASD for sharing systems referencing updated block
    - Again remember, it takes CPU to put block into the CF
    - And updated blocks still written to DASD

Shared VSO

- Store-in Data Cache
  - Data in CF may be different than on DASD
- Specified on Area by Area basis
- Use LKASID option for local buffer pools
  - Some exceptions can occur
    - Monitor % hits and % valid
    - If low then maybe LKASID is costing more than the benefit
- PRELOAD vs. NOPREL
  - Depends on size of area and available CF stor
  - Also locality of reference

Shared VSO

- CF access
  - Synchronous if CI size 4K or less
    - Generally better service time
  - Asynchronous if larger
- Castout processing
  - Initiated at checkpoint
  - Reads modified data from CF
  - Writes to DASD
  - Can cause delays in processing
    - Faster DASD is cure

Application Considerations

- PSB PROCOPTS
  - Review high volume trans
- Address deadlock problems
  - May be more prevalent with sharing
- Address contention problems
  - Will only get worse
  - Temporary bypass may be to 'route' trans or limit regions

Shared Queues

- Allows for considerable growth
- Potentially reduce individual IMS load
  - Spread out the work
  - Log data reduction
  - Terminal session reduction
    - Less time to checkpoint
- Will increase CPU consumption
  - Amount varies quite a lot
    - Appl/DB activity versus TM
    - CF access times
    - etc.

CF Structure Considerations

- SMQ uses LIST structures
- Establish proper LE to EL ratio
  - Need to determine average message sizes
  - Always err on side of too many LE's
  - Otherwise system might stop which is generally bad for performance
- Avoid overflow processing
  - Make primary structure size large enough
    - Use 'high used' DRRN as guide
  - Specify INIT and SIZE to allow for alter processing (manual or automatic) just in case
  - Queue activity quiesced during process

Structure Considerations

- Structure Checkpoint
  - Frequency determines amount of log data kept and time to restart in the event of a failure
  - Activity quiesced while in progress
  - CFRM couple data sets critical to performance
  - Of course so are the SRDS's
  - And then let's not forget about the CF access
  - If using both Full Function and EMH consider staggering the checkpoints to minimize any single delay

MVS Logger

- Used by CQS
  - Most of volume is input and output messages
- Do NOT use staging data sets
  - Can cause severe performance degradation
- Structure Offload occurs when threshold reached
  - % specified in LOGR policy
  - Frequency of offload determined by:
    - Data rate
    - Structure size
    - Threshold

Logger Structure

- Duplexed in data space
  - Assuming staging datasets not used
- The bigger the structure the more storage needed for the data space
  - could cause excessive storage demand
- The smaller the structure the more frequent offloads
  - should not be a problem with current OS/390 and Z/OS versions
- Logstream blksize should be large - 64K

Fast Path Processing Options

- 3 'SYSPLEX PROCESSING OPTIONS'
  - Determined by DBFHAGU0
  - Local first - default
    - Process locally if resources available
    - Else put on shared queue
  - Local only
    - Only process locally - never put on shared queue
  - Global only
    - Always put on shared queue even if possible to process locally

FF Scheduling

- Standard PARLIM algorithm can not be used
  - IMS doesn't know the depth of shared queues
- False schedules can occur
  - message may have already been processed
- Back to basics is best
  - The best schedule is the one that doesn't happen
  - High volume trans in specific classes
  - Use (P)WFI when possible
  - Specify MAXRGN to minimize 'thrashing'

Misc

- CF level changes can affect structure use
  - Can be hard failure or
  - Degraded performance
- Sometimes problem is outside of IMS
  - MVS functions
    - GRS
    - SMF
    - etc.
  - Other products
    - Security
    - DB2
  - Dispatching priority / WLM policy

Summary

- Fast CPU's
- Fast CF's
- Fast links
- Lots of memory
- Fast DASD
- Review high volume applications
- Review databases with contention
- Review class scheduling
- Monitor regularly