

S51

# DL/I Batch to BMP Conversion

Karen Tischer



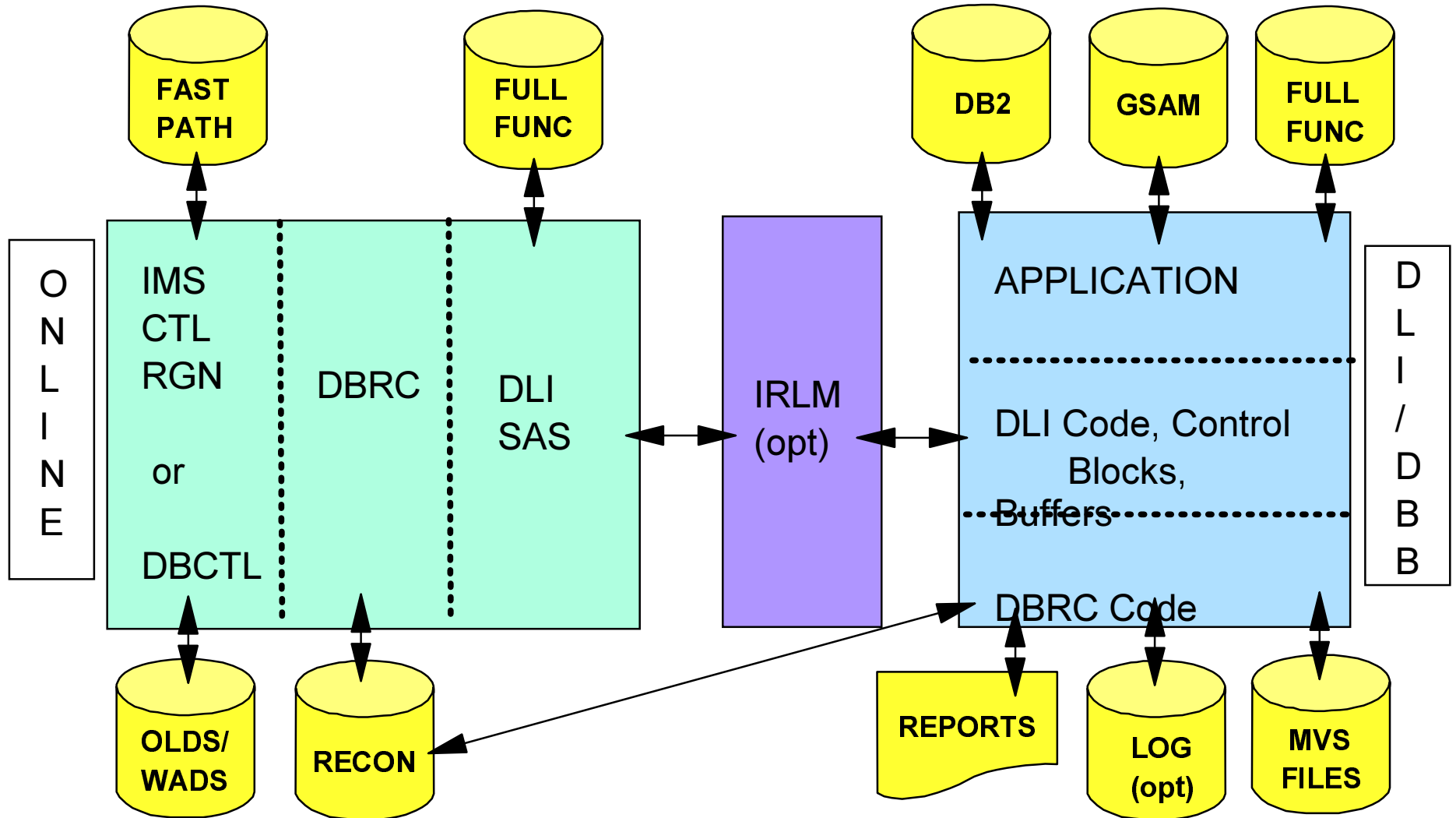
Miami Beach, FL

October 22-25, 2001

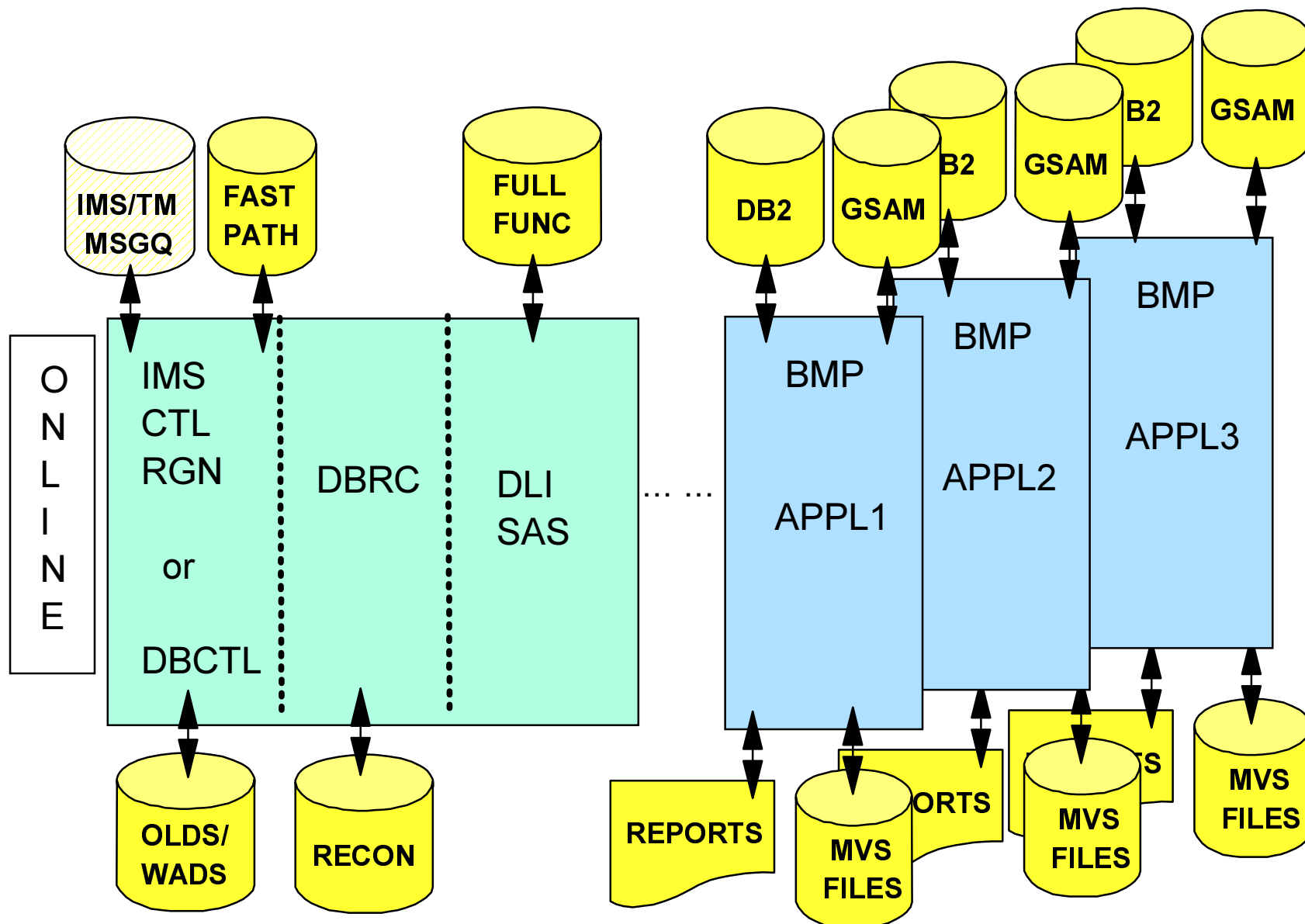
# Agenda

- DLI Batch versus BMP
- BMP Implementation
- Checkpoint/Restart
- Performance
- Summary
- Appendix: Sample Checkpoint Program Logic

# Topic 1: DLI Batch Address Space



# BMP Address Space



# DLI Batch Versus BMP

	BATCH	BMP
Online required for execution	N	Y
Requires sufficient batch window	Y	N
DLI and DBRC services in same address space as application	Y	N
Authorization/Unauthorization at each step	Y	N
Open/Close at each step	Y	N
Own database buffering	Y	N
Own log data set	Y	N
Can use HSSR	Y	N
Database resources locked until sync point	N	Y
Batch Backout required after application abend	Y	N
Access to IMS Message Queue	N	Y
Define in IMSGEN	N	Y

# DLI Batch Versus BMP . . .

	BATCH	BMP
Scheduled by MVS, not IMS	Y	Y
May experience deadlocks	N	Y
OSAM Sequential Buffering available	Y	Y
Block level data sharing required to concurrently update online databases	Y	N
CFNAMES,CFVSAM=..,CFOSAM=..,CFIRLM=.. required in DFSVSAMP for Block level data sharing	Y	N
Should include CHKP / XRST capability	N*	Y
Access to IMS Full Function Databases	Y	Y
Access to Fast Path Databases	N	Y
Access to MVS Files	Y	Y
Access to GSAM	Y	Y
Access to DB2	Y	Y

# TOPIC 2: BMP IMPLEMENTATION

- Adding an IOPCB
- Including Checkpoint/Restart logic
- Setting up the IMSBATCH procedure
- Using GSAM
- Including the BMP in the online system
- Allocating a JES Initiator
- Operating Considerations for the BMP

# The IOPCB

- Required by a BMP
- Acquired as 1st PCB in PSB at scheduling time
  - ▶ Need PCB Mask
  - ▶ Need linkage
- Used for 'CHKP'/'XRST' calls
- For testing in DLI Batch
  - ▶ PSBGEN . . . ,COMPAT=YES



# Checkpoint/Restart

- BMP concerned with concurrency of access as well as restartability
- All BMPs should have regular commit points
  - ▶ GU IOPCB
  - ▶ CHKP call
  - ▶ SYNC call
  - ▶ ROLL, ROLB, ROLS calls
- Resources Locked until commit
  - ▶ Span of data locked
    - Number of DB records (HDAM locks on RAP)
    - Control records
    - Twin chains
  - ▶ Lock enqueue space (PI or IRLM)
  - ▶ Deadlock possibility

# Checkpoint/Restart ... ..

- Affect on Operations
  - ▶ Cannot change DB status (/STO, /STA, /DBR, /DBD ... )
  - ▶ Cannot shut down online system
- Frequency depends on mode
  - ▶ MD-BMP ... MODE=SNGL recommended
  - ▶ 'GC' status code for DEDB with PROCOPT=P | H
  - ▶ User Interval
    - Elapsed time
    - Number of DB records read (not 'read' calls)
    - Number of DB records updated (not 'update' calls)
    - Use "master" for controlling interval
    - One DB or file read or updated once per iteration thru program

# Batch to BMP without CHKPs

- Add IOPCB
- Implement IMSBATCH procedure
- Add PSB to online IMS (APPLCTN)
- Use PROCOPT=E
  - ▶ Avoids locking overhead, unless data sharing
  - ▶ Prevents others in this IMS from scheduling
- Use PROCOPT=GON | GOT
  - ▶ Avoids locking overhead
  - ▶ Avoids data sharing requirements
  - ▶ Allows other online users to be scheduled in this and other IMSs

# IMSBATCH Procedure PARM='

- BMP (region type)
- MBR=
  - ▶ application program name
- PSB=
  - ▶ psbname if different for program name
- NBA=
  - ▶ fast path database buffers
- OBA=
  - ▶ fast path overflow buffers
- IN=
  - ▶ input transaction code
  - ▶ IMS/TM only - Message Driven (MD) BMP
  - ▶ OUT= ignored
    - Replies go to IOPCB
    - May use ALT-PCBs

# IMSBATCH Procedure PARM= ...

## ■ OUT=

- ▶ output transaction code or LTERM name
- ▶ IMS/TM only - non-Message Driven (NMD) BMP
- ▶ For sending output messages via the IOPCB
- ▶ Not for reading input messages from the input Qs

## ■ CKPTID=

- ▶ null: no restart
- ▶ 'LAST': restart from the last checkpoint issued
- ▶ 8 byte checkpoint id created by application
- ▶ 14 byte checkpoint id from DFS0540I or DFS681I messages
  - (IIIDDDHHMMSST) where IIII is region ID
- ▶ NOMSG681\* to suppress DFS681I messages
- ▶ NOMSG540\* to suppress DFS0540I messages
- ▶ NOMSG\* to suppress both DFS681I and DFS0540I messages

\*NOTE: not valid at restart

# IMSBATCH Procedure PARM= ...

## ■ LOCKMAX=

- ▶ value between 1 - 32767 (in units of 1000)
- ▶ When exceeded, BMP will 3301abend

## ■ CPUTIME=

- ▶ value between 1 - 1440 (minutes)
- ▶ When exceeded, BMP will U0240abend after DLI call completes
- ▶ Use instead of MVS TIME= parameter to avoid U113abend of IMS online

## ■ IMSID=

- ▶ 1-4 character ID of IMS online system where the BMP will run

# IMSBATCH Procedure PARM= ...

- PARDLI=0 | 1
  - ▶ 0: DLI processing under control of BMP TCB
    - Best for performance
    - Use for production
    - Advantages:
      - Page fault isolation to BMP's TCB
      - Multi-CPU exploitation
      - Priority dispatching
    - Disadvantage: Sx22 abend may cause IMS U113 abend

# IMSBATCH Procedure PARM= ...

- PARDLI=0 | 1 (continued)
  - ▶ 1: DLI processing under IMS Control Region TCB
    - Use for test
    - Eliminates IMS U113 abend which may occur when Sx22 abend occurs in the BMP region
      - S122 - operator cancel with dump
      - S222 - operator cancel
      - S322 - timeout
      - S522 - timeout due to 'wait'
      - S722 - output limit exceeded
    - Disadvantage: Bad for performance



# Using GSAM

- What is it?
  - ▶ OS files under control of DLI
  - ▶ BSAM or VSAM(ESDS)
  - ▶ F | FB | V | VB | U
- Used to ease restart - IMS automatically repositions
- Problems with GSAM
  - ▶ Backout does not remove updates
  - ▶ Out of space conditions
  - ▶ JES sysouts
- ACBLIB not used - IMSBATCH must contain DD statements for:
  - ▶ DBDLIB
  - ▶ PSBLIB

# Define BMP to IMS

**APPLCTN    FPATH=NO,PGMTYPE=BATCH,PSB=xxxxxxx,  
              SCHDTYP=SERIAL | PARALLEL**

▲ FPATH=YES is invalid

▲ PGMTYPE=BATCH: BMP [and CICS Transaction]

▲ SCHDTYP=

■ SERIAL:

- Single scheduling of PSB only
- Processing limited to one dependent region/thread

■ PARALLEL:

- Multiple scheduling of PSB for multiple transact codes
- Processing limited to MAXREGN parameter

# Define Transaction for MD-BMP to IMS/TM

- TRANSACT CODE=(aaaaaaaa,bbbbbbbb, . . .),DCLWA=YES, MODE=SNGL|MULT,SERIAL=YES|NO,[WFI]
  - ▶ DCLWA=YES:
    - Write input/output messages to log prior to enqueueing
  - ▶ MODE=SNGL:
    - Commit at each GU IOPCB
    - Performance option
    - Faster response to end-user
    - Forced for WFI
  - ▶ SERIAL=NO:
    - Input does not need to be processed in FIFO sequence
  - ▶ WFI:
    - Wait-for-Input - do not terminate BMP if there are no messages for it
  - ▶ PRTY: Normal and Limit priorities are set to 0
  - ▶ PROCLIM: ignored
  - ▶ PARLIM: not supported

# Starting the BMP

- JES INITIATORS:
  - ▶ Set up JES initiator(s) for BMP job classes
  - ▶ Use initiators to control when BMPs run
    - Start fewer initiators during peak transaction processing
    - Start more initiators during slow times
- BMP Started By:
  - ▶ JES job submission
  - ▶ JES START command
  - ▶ IMS command: /START REGION membername

# Stopping the BMP

- /STOP REGION | THREAD nn (normal case)
- /STO REG | THREAD nn ABDUMP
  - ▶ Software cancel - BMP issues own abend
  - ▶ Application looping or in wait state
- /STO REG | THREAD nn CANCEL
  - ▶ Only if /STO REG nn ABDUMP fails to work
  - ▶ Abends active TCB of BMP
  - ▶ May cause U113 abend of IMS if PARDLI=0
- Cannot use:
  - ▶ MVS or JES STOP | CANCEL (IMS traps and prevents)
  - ▶ MVS MODIFY

# Restarting the BMP

- If BMP does not issue CHKPs/XRST
  - ▶ Resubmit entire job
- If BMP issues CHKP/XRST
  - ▶ Specify CKPTID='LAST' and resubmit
    - Do NOT change jobname, psbname or program name
    - Last CHKP log record (X'18') must be on OLDS
  - ▶ Last CHKP (X'18') not on OLDS
    - Include //IMSLOGR DD
    - Supply checkpoint id from DFS0540I or DFS681I msgs in JOBLOG
  - ▶ Checkpoint ID not known
    - Resubmit job with CKPTID='LAST' ... not on OLDS then
      - U102 abend results with DFS0540I msg
    - Scan console log (or JOBLOG) for most recent DFS681I msg

# Topic 3: Checkpoint / Restart

- Checkpoint Call
- Restart Call
- Synchronization Point Call
- ROLL, ROLB, ROLS Calls

# Checkpoint Call

- Applies to Batch DLI and BMP
  - ▶ Commits all changes made
  - ▶ Establishes a restart point
  - ▶ Used for recovery purposes
- Basic Checkpoint - restart dependent on application logic
  - ▶ CALL 'xxxTDLI' USING CHKP, IOPCB | AIB, IOAREA
  - ▶ EXEC DLI CHKP ID('literal') | ID(areaname)
- Symbolic Checkpoint - requires use of Restart (XRST)
  - ▶ CALL 'xxxTDLI' USING CHKP, IOPCB | AIB, IOAREALN,  
IOAREA,  
  
AREA1LN, AREA1, . . .  
AREA7LN, AREA7
  - EXEC DLI SYMCHKP ID('literal') | ID(areaname)  
AREA1(area1) LENGTH1(expression1) . . .  
AREA7(area7) LENGTH7(expression7)



# Checkpoint Events

- Database updates committed
  - ▶ Before/After images written to system log
  - ▶ Modified segments written to database
  - ▶ Locks on modified segments released
- Checkpoint information written to log (X'18')
  - ▶ Checkpoint ID
  - ▶ All IMS database positions, including GSAM
  
  - ▶ Up to 7 user data areas
- Checkpoint ID sent to IMS master & MVS console
  - ▶ (DFS0540I & DFS681I)
- Database position lost except:
  - ▶ GSAM,
  - ▶ DEDB PROCOPT=P | H if 'GC'
- [Output messages enqueued for sending after logging]
- [Input messages dequeued - next input message returned]

# Checkpoint Program Flow

- Database driven program ('GN' processing)
  - ▶ Save database position
  - ▶ Issue CHKP call
  - ▶ Re-establish database position
  - ▶ Resume processing
- File driven program ('GU' processing)
  - ▶ Issue CHKP call
  - ▶ Read file
  - ▶ GU to re-establish database position
  - ▶ Resume processing

# Restart Call

- Restart a BMP that
  - ▶ abended
  - ▶ was terminated due to operator command
    - /CHE FREEZE
    - /STO REG | THREAD xx [ABDUMP | CANCEL]
  - ▶ abended due to an IMS abend
- Restart should be first program call (after GU IOPCB if MD-BMP)
- Restart must precede first checkpoint call
  - ▶ CALL 'xxxTDLI' USING XRST, IOPCB | AIB, IOAREALN, IOAREA,  
AREA1LN, AREA1, . . .  
AREA7LN, AREA7
  - ▶ EXEC DLI XRST MAXLENGTH(expression) ID('literal') | ID(areaname)  
AREA1(area1) LENGTH1(expression1) . . .  
AREA7(area7) LENGTH7(expression7)

# Restart Events

- GSAM repositioned by IMS
  - ▶ do not change blocksize
  - ▶ DISP=MOD positions to end with PROCOPT=L
- IMS Full Function databases repositioned, if possible, by IMS
  - ▶ identical position not guaranteed if
    - segments added or deleted
    - non-unique keys
    - no keys
  - ▶ check status code of each database PCB
    - if not blanks, reposition if necessary
- Fast Path databases not repositioned, user responsibility if necessary
- User areas restored

# Synchronization Point (SYNC) Call

- Usable only by NMD-BMPs
- Application must be SELF RESTARTING if restart required
- Not used in conjunction with CHKP
- No WTO
- No log data
- Releases resources that IMS has locked for the application
- CALL 'xxxTDLI' USING SYNC IOPCB | AIB
- No EXEC DLI equivalent

# ROLL, ROLB, ROLS Calls

- ROLL: Backout full function (FF) to last commit
  - ▶ CALL 'xxxTDLI' USING ROLL
  - ▶ EXEC DLI ROLL
  - ▶ Program abends with U778
- ROLB: Backout FF to last commit
  - ▶ CALL 'xxxTDLI' USING ROLB, IOPCB | AIB [,IOAREA]
  - ▶ EXEC DLI ROLB
  - ▶ Returns control to program
  - ▶ [returns first message segment into IOAREA]
- ROLS: Backout FF to earlier processing set point (SETS | SETU)
  - ▶ CALL 'xxxTDLI USING ROLS, IOPCB | AIB | DB-PCB, [IOAREA, TOKEN]
  - ▶ EXEC DLI ROLS TOKEN(token1) AREA(data-area)
  - ▶ Returns control to program or
  - ▶ DB-PCB: Can result in U3303 abend if DB2 or DEDB | MSDB in PSB



# Topic 4: Performance

- Monitors:
  - ▶ BMP - IMS Monitor: BMP tuning more difficult
  - ▶ Batch - DB Monitor: Batch tuning easier
- DLI & DBB may be swappable in non-data sharing environment
  - ▶ SWAP=Y | N (default is Y)
- Parallel DLI
  - ▶ PARDLI=0 : Best for Performance
  - ▶ PARDLI=1 : Best for testing where U113s are a problem
- BMP Initiators
  - ▶ 4 - 10 reasonable
- Start when online processing volumes are low

# Performance . . . . .

- Page fixing OSAM and VSAM control blocks and buffer pools
- Buffer Isolation - separate subpools to
  - ▶ Minimize buffer steals
  - ▶ Minimize buffer contention
- VSAM
  - ▶ Optimize buffer hit ratio
  - ▶ Minimize buffer steals
  - ▶ Use Hiperspace for high read:reread ratio
- OSAM
  - ▶ Use OSAM Sequential Buffering when applicable
  - ▶ Minimize read requests
  - ▶ Minimize buffer steals



# Performance . . . . .

## ■ OSAM Sequential Buffering

### ▶ Optional

### ▶ One pool of sequential buffers for each per DB PCB/DSG

- 4 buffer sets by default
- 10 buffers per buffer set
- long-term page fixed
- no lookaside between dependent regions

### ▶ Activation

- SBONLINE control statement in DFSVSMxx requests SB for IMS DB/DC or DBCTL
- PCB ...,SB=COND requests SB for the BMP
- //DFSCCTL DD with control statement in IMSBATCH JCL
  - SBPARAM used to override SB and default number of buffer sets by PCB in PSB
- SB Initialization User Exit routine - optional
  - Request conditional activation
  - Change default number of buffer sets
  - Disallow usage of SB for this execution

# Performance . . . . .

- DEDB High Speed Sequential Processing (HSSP)
  - ▶ Optional
    - Reduces elapsed time
    - Optionally can concurrently image copy - requires DBRC registration
  - ▶ Three buffer sets equal to UOW size long-term page-fixed
    - Will be dynamically increased to six buffer sets if necessary
    - 4 buffer sets to 7 buffer sets if ASIC
  - ▶ Activation
    - PCB ...,PROCOPT=H to activate HSSP for the NMD-BMP only
      - Appl must 'CHKP' at 'GC' (UOW boundary crossed)
    - //DFSCCTL DD control statements in IMSBATCH JCL
      - SETO can turn off HSSP request in PCB
      - SETO can request Asynchronous Image Copy (ASIC)
      - SETO can request No Read Ahead (NORDAH)
      - SETR can specify which areas are to be processed and in what order

# Topic 5: Summary

- BMP Limitations
- XRF Considerations
- BMP Advantages

# BMP Limitations

- Can only backout to LAST checkpoint
  - ▶ Batch DLI can backout to any checkpoint if not block level data sharing
- Cannot recover to beginning of BMP and re-run
- HSSR won't work with BMPs
  - ▶ OSAM Sequential Buffering is a good alternative
- No IMS commands from BMP regions in DBCTL environment

# XRF Considerations

- When ACTIVE IMS fails, BMP fails:
  - ▶ dynamically backed out to last CHKP
  - ▶ manually restart BMP on alternate (new active)
- Use initiators / job classes to control execution CPU
  - ▶ Stop initiators on old active
  - ▶ Start initiators on new active
  - ▶ Don't use system affinity
    - JES2: /\*JOBPARM SYSAFF= . . .
    - JES3: //\*MAIN SYSTEM=...

# **BMP Advantages: Better Operational Environment**

- Uses IMS Online Logs:
  - ▶ Simplified database recovery
  - ▶ Simplified operations
  - ▶ Central log control
  - ▶ Dynamic backout for all failures
- Avoids data sharing within a single MVS image
- Protection from inadvertent operator cancels
- U828 (ISRT duplicate index entry) eliminated
- No 'batch window' constraints

# BMP Advantages: Appl. Architecture

- Access to Fast Path DEDBs (data entry data bases)
  - ▶ alternative to user partitioning (240 AREAs) of databases
  - ▶ alternative to DB Partitioning (only 32 partitions permitted - pre-V7)
  - ▶ operations at AREA level
  - ▶ utilities executed against AREA
  - ▶ utilities run online while AREA is in use
  - ▶ High Speed Sequential Processing (HSSP)
  - ▶ Asynchronous Image Copy (ASIC) concurrent with HSSP
- IMS/TM: Access to message queues

# BMP Advantages: Performance

- Databases already OPEN'd by online
- Databases already AUTHORIZED by online (less RECON access)
- WFI (wait-for-input) processing: IMS/TM only
- High performance LWA (log write ahead) to WADS



S51

# DL/I Batch to BMP Conversion

Thank you for your evaluation



Miami Beach, FL

October 22-25, 2001

# Appendix:

## Sample Checkpoint Program Logic

- Uses program specified UNIQUE checkpoint IDs
  - ▶ Must be unique, else from where to restart?
  - ▶ IMS generated not known to application, hard to use in automated process
  - ▶ Minimize JCL changes - just resubmit with necessary logs
  - ▶ Eliminate recompile for frequency changes

# Sample Checkpoint Program Logic . . .

- Components
  - ▶ Checkpoint database - HDAM root only
    - Program name is KEY
    - JES Job number
    - Counter
    - Good place to store CHKP frequency information
      - To alter frequency, change value in database
      - No need to recompile program
  - ▶ Generalized checkpoint code copied into program
  - ▶ PCB for Checkpoint database

# Sample Checkpoint Program Logic . . .

- At program START
  - ▶ GU CHKP-DB-PCB using KEY = PGMNAME
  - ▶ If input record blanks, then normal execution
  - ▶ If input record contains chkp-id, restart indicated
- Issue XRST call
  - ▶ If normal execution, use blanks in IOAREA
  - ▶ If restart, use CHKP-ID from CHKP-DB in IOAREA
    - Saved program areas restored
    - GSAM databases repositioned by IMS
    - IMS databases repositioned if possible by IMS

# Sample Checkpoint Program Logic . . .

- If restarting
  - ▶ Check status code of all database PCBs for blanks
    - If not blank, reposition database if necessary
  - ▶ Update CHKP-DB with new JES job number
  - ▶ Issue initial CHKP call
- Normal processing
  - ▶ Obtain checkpoint frequency from CHKP-DB
  - ▶ Increment and test CHKP counter
    - elapsed time
    - # DB records updated
    - # DB records read

# Sample Checkpoint Program Logic . . .

- When CHKP to be taken
  - ▶ Update user areas to be CHKP'd
  - ▶ Update CHKP-ID counter
  - ▶ REPL root in CHKP-DB
  - ▶ Issue CHKP with new CHKP-ID and up to 7 user areas
  - ▶ Reset CHKP-ID counter
  - ▶ Reposition databases if necessary (not needed for GSAM or DEDB with 'GC')
- At normal completion
  - ▶ REPL CHKP record with blanks in CHKP-ID field so next execution is normal start
  - ▶ Terminate program