

# Using High Performance Change Accumulation at SBC

Neil Schechter

Jim Kelly

SBC Services, Inc.



# Overview

---

- IMS at SBC
- Change accumulation at SBC
- Structure and design of HPCA
- Implementation of HPCA at SBC
  - How it was deployed
  - Getting it to run
  - JCL consideration
- Performance test results
- Evaluation and Summary

# Introduction

---



# IMS at SBC

---

- Major IMS applications
  - Billing
  - Equipment Inventory
  - Work Orders

## IMS at SBC

---

- SBC has over 48,000 registered IMS data bases
- Over 150 IMS images exist at SBC
- Of those images, approximately 15% use HPCA
- On four or five systems, HPCA is considered critical

# The Need for High Performance Change Accumulation

---

- On some major applications there is little or no batch window time. Image copies can only be taken on weekends.
- With some applications, during sustained periods, a log tape will be written every three to five minutes, making it impractical to recover without change accum files.
- Ordinary change accum runs, using DFSUCUM0, simply can't keep up.

## Evolution of IMS Change Accumulation.

- IMS may often require use of logs to recover databases
- IMS utility, DFSUCUM0, consolidates logs, making a recovery faster

## Evolution of IMS Change Accumulation

---

- DFSUCUM0 may be inadequate for heavily used IMS systems
  - Log files must be re-read for each additional Change Accum group
  - Log files may be created faster than DFSUCUM0 can process them



# Evolution of Change Accumulation

---

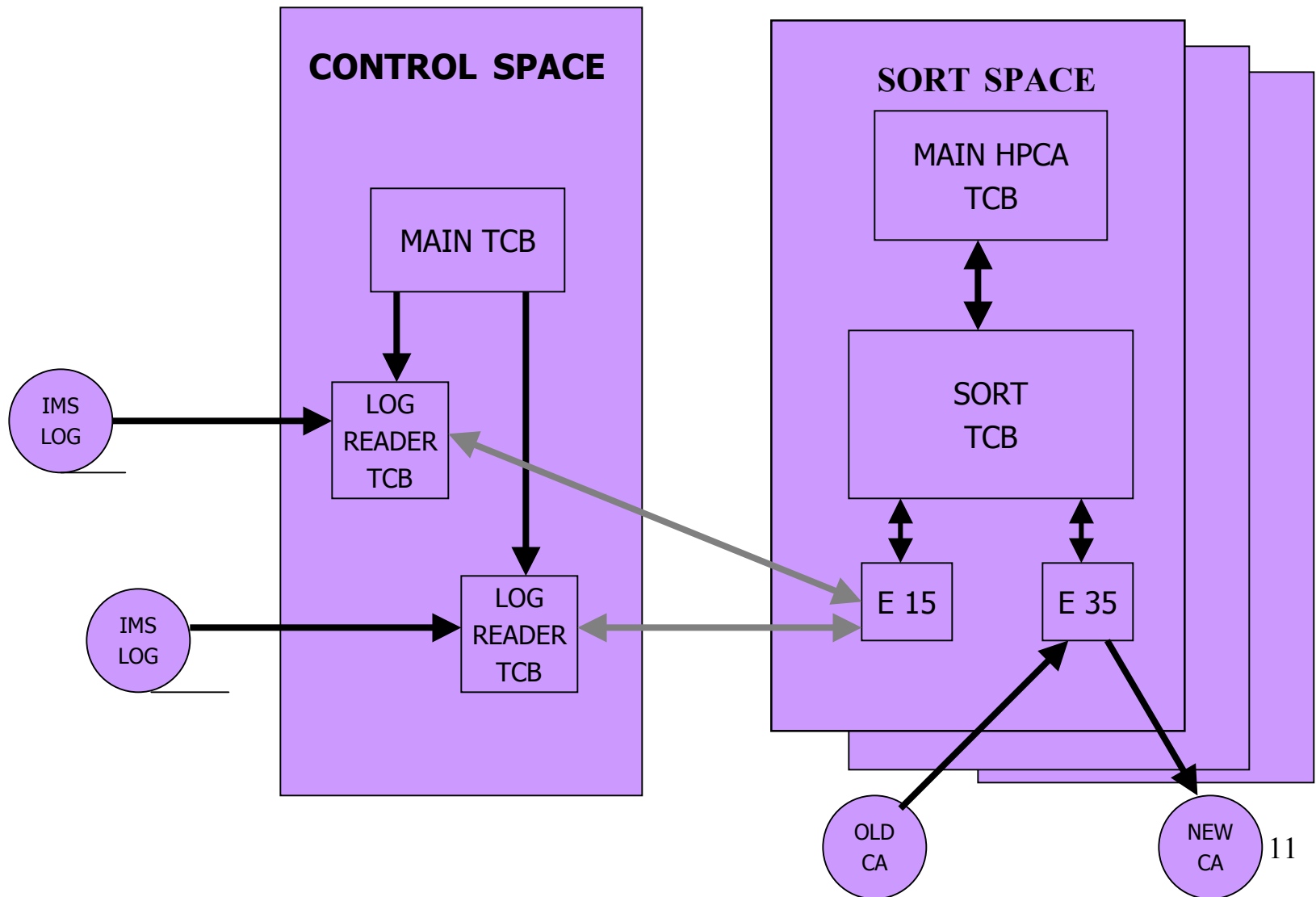
- To overcome this difficulty, product utilities were developed that would do change accums faster.
- SBC has had experience in using the following change accumulation products.
  - BMC'S Change Accum Plus (CHAMP)
  - IBM'S Parallel Change Accum (PCA)
  - IBM'S High Performance Change Accum (HPCA)

# Evolution of Change Accumulation

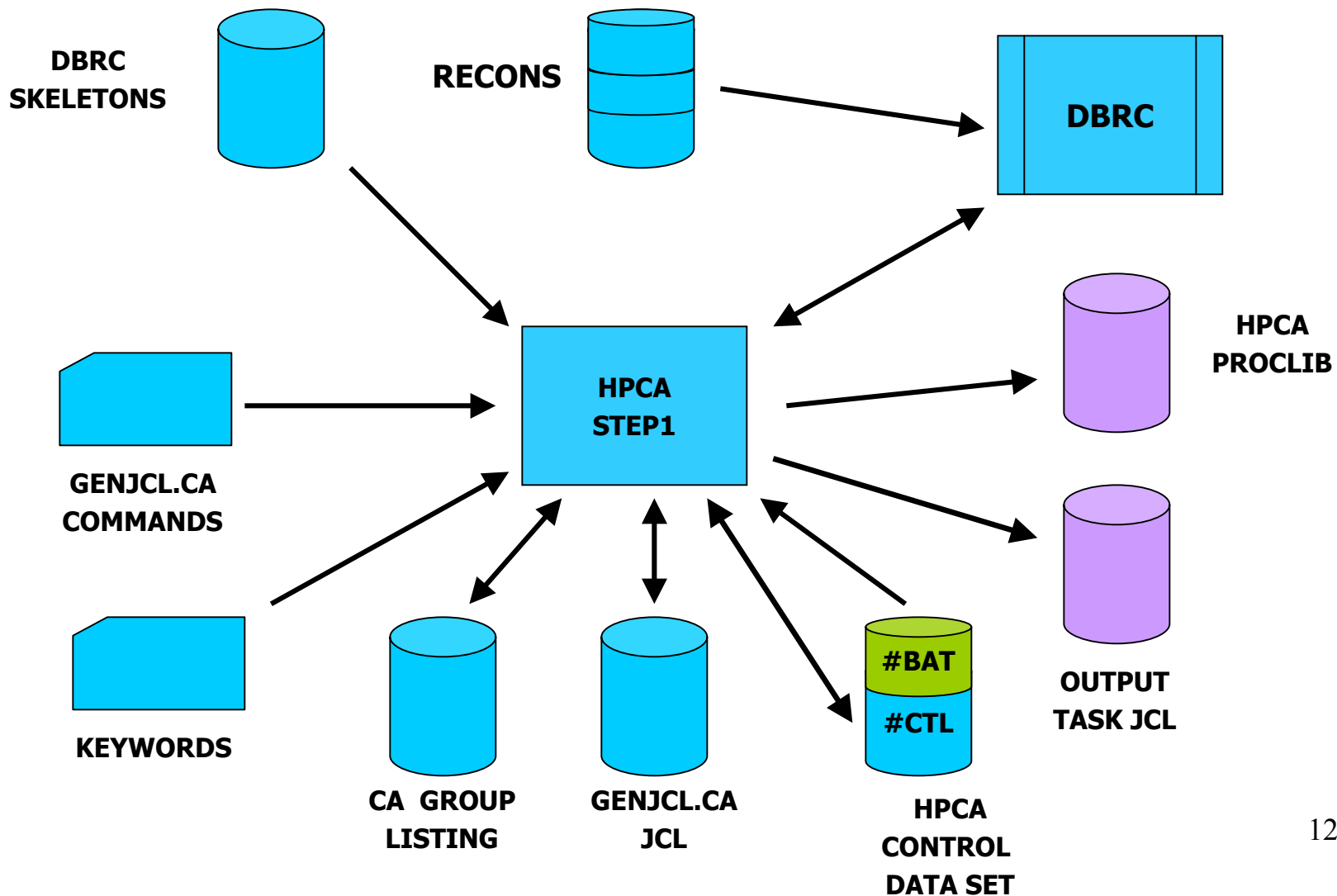
---

- These products improve change accum performance mostly by sorting and merging the log records of each change accum group in parallel.
- HPCA has the additional capability of reading log datasets in parallel as well.
- HPCA also has the ability to produce a second copy of the output CA file, which can be used for disaster recovery.

# Structure and Design of HPCA: Driver and Output Tasks

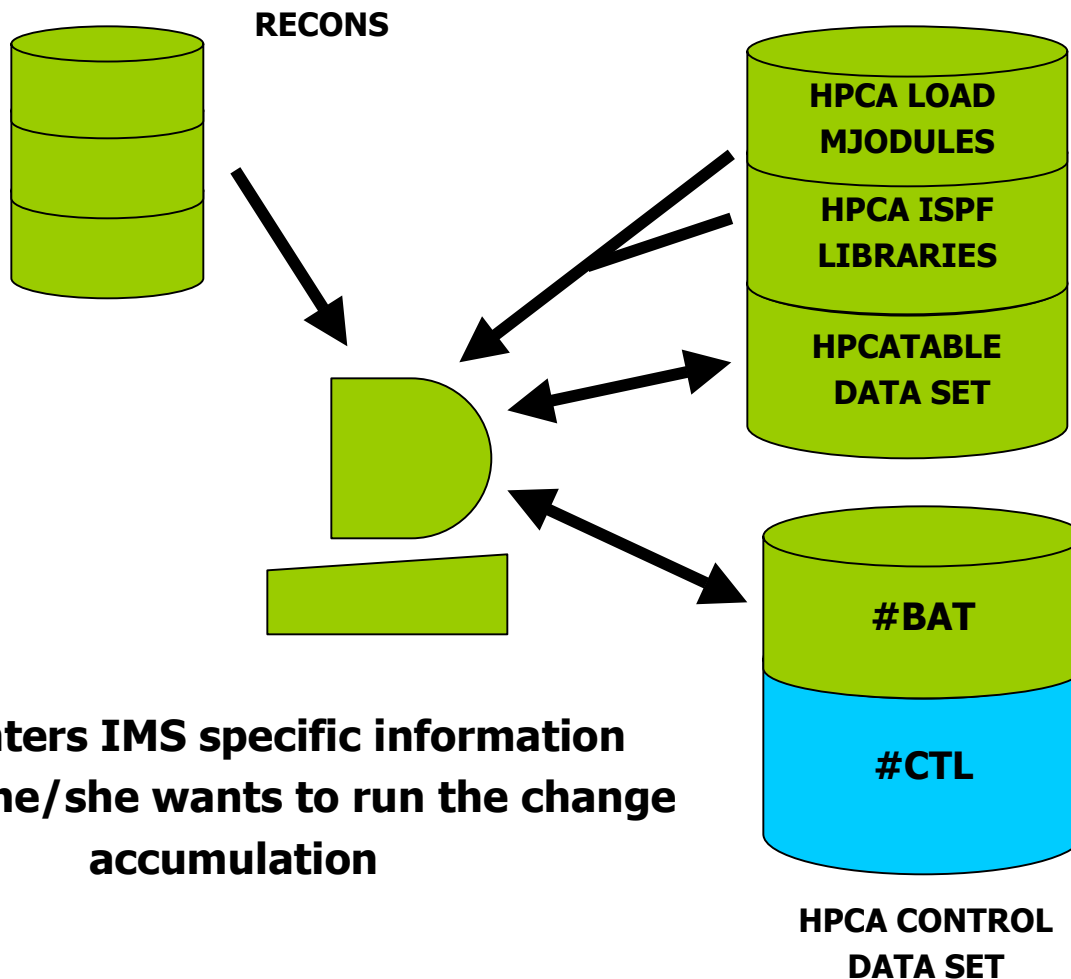


# Structure and Design of HPCA: HPCABAT0

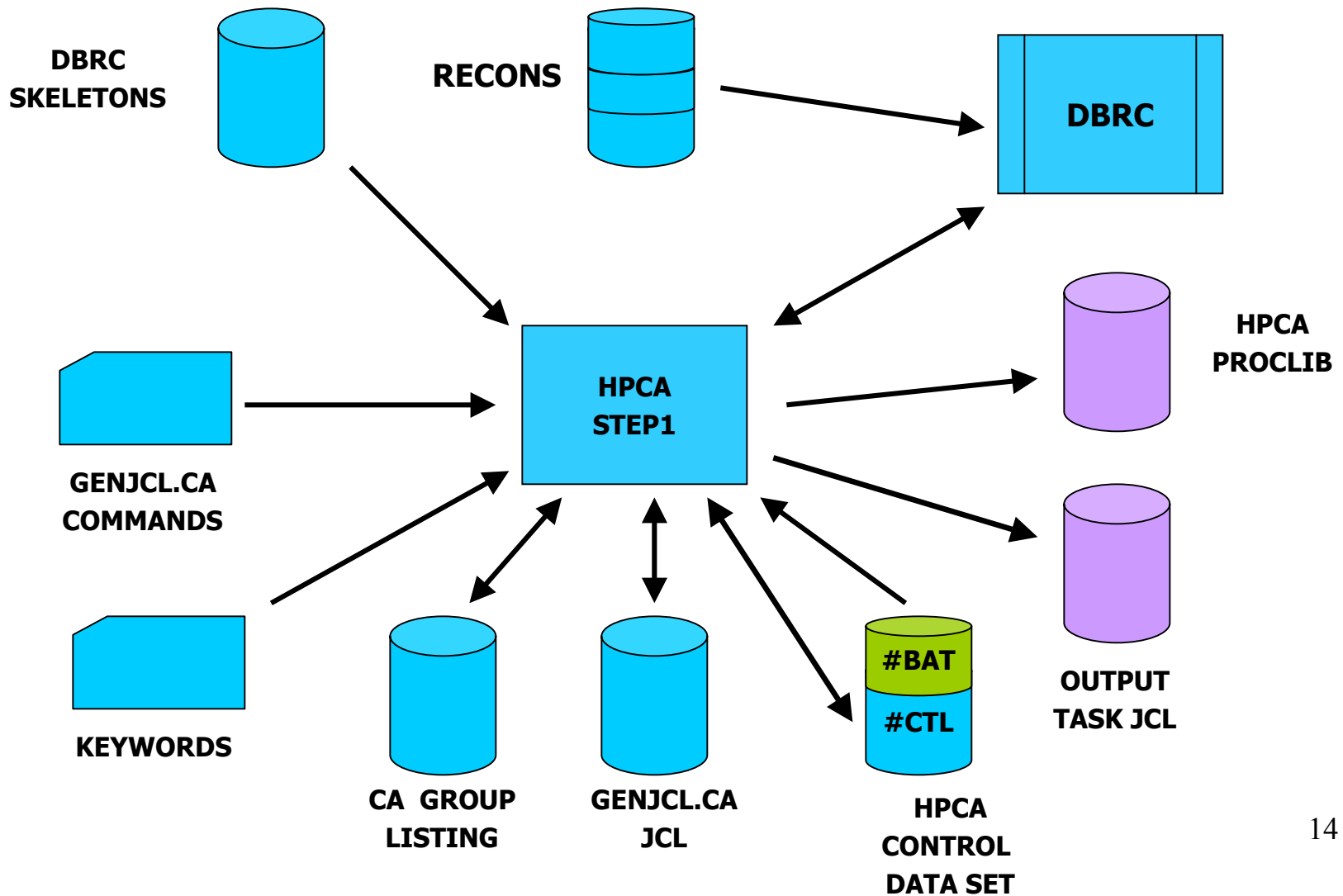


# Structure and Design of HPCA: ISPF User Interface

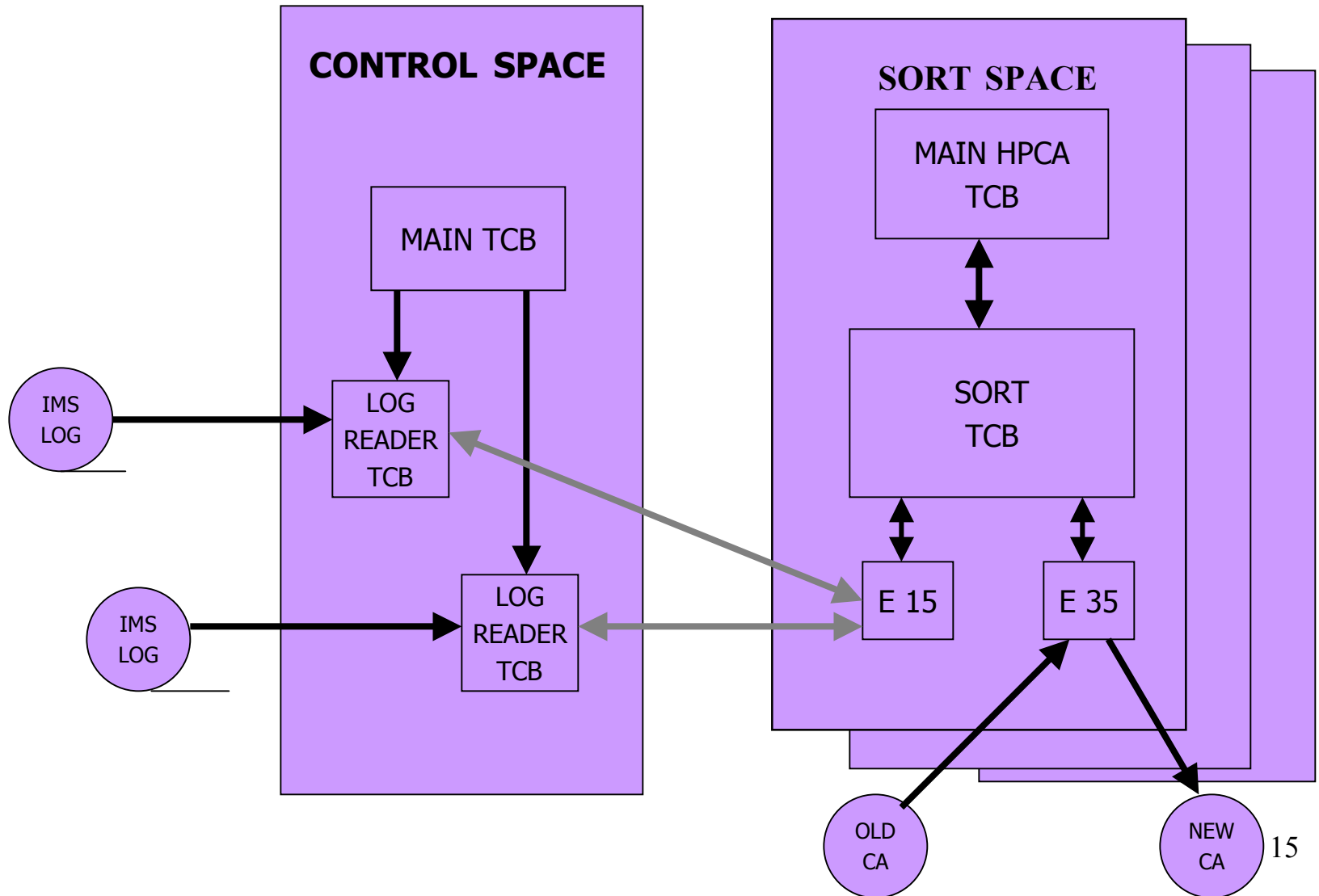
---



# Structure and Design of HPCA: HPCABAT0



# Structure and Design of HPCA: Driver and Output Tasks



## HPCA Installation and Deployment at SBC

- HPCA libraries were installed across a Sysplex, allowing several IMS systems to share the same product library set
- HPCA load modules and ISPF members were kept in their own separate libraries, apart from other products
- HPCA load library was APF authorized



## HPCA Installation and Deployment at SBC

- RACF profiles were defined for the output started tasks, which were allowed READ access to the IMS system logs
- The same output tasks were allowed ALTER authority by the IMS applications to create their change accum files

## HPCA Installation and Deployment at SBC

- The DBA would then tailor step one JCL for his/her application and run it
- After the step one process created the job members for the output started tasks, they needed to be copied into the MVS started task library

# HPCA Installation and Deployment at SBC

---

- An automatically scheduled job stream would then be set up to run.
- SBC uses more than one variation for these job streams.
  - One variation simply includes HPCABAT0 along with HPCADRIV . This allows complete tracking through an automatic scheduler.
  - Another variation submits HPCADRIV to an internal reader, allowing for last minute editing, providing diagnostics should trouble occur, and guarantees correct PROC resolution.

# Implementing HPCA

---

- Walkthrough some of the panels
- How do you specify the number of input tasks/readers?
- How do you specify the output/CA tasks?
- DBRC commands and skeletons
- JCL considerations

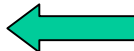
# Using the ISPF Interface

-----  
HPCA setup -

You are using system: IMSUPTS

Please provide a "user-specified HLQ" (high-level qualifier).  
This HLQ will be used as a 'prefix' to the datasets that HPCA  
creates/allocates for you during this session.

-----  
ENTER YOUR HLQ: SUPT.DBA



- This is our "front-end" panel to access the ISPF interface
- Entering the HLQ will cause HPCA to allocate some datasets if they don't exist.

# Selecting the HPCA-ID

HPCA-ID/IMS Profile

Row 1 to 1 of 1

COMMAND ==> \_\_\_\_\_ SCROLL ==> PAGE

Enter END (PF3) to exit.

To create a new HPCA-ID/IMS profile, enter an HPCA-ID and its description

HPCA-ID: \_\_\_\_\_ Profile Description: \_\_\_\_\_

Enter line command S to select an IMS System for this run

T to setup DBRC product datasets

D to delete an HPCA-ID profile

CMD HPCA-ID HPCA-ID/IMS Profile Description

-----

s SWCS SWITCH PRODUCTION SOUTH HPCA



Enter "s" to select your HPCA ID

# Using the ISPF Interface

HPCA CA Group Selection

Row 1 to 5 of 10

COMMAND ==> s \*



SCROLL ==> PAGE

Enter one of the Line or Primary commands. Enter END (PF3) to exit.

Line Commands:

Primary Commands:

S - Select the CA Group for processing.

S \* Select all cagroups

U - De-select the CA Group

Enter END (PF3) to exit. Enter CAN to exit without processing.

| CA-GROUP   | ACTION |
|------------|--------|
| -----      | -----  |
| — SWCSCAQ1 |        |
| — SWCSCAR1 |        |
| — SWCSCAS1 |        |
| — SWCSCAT1 |        |
| — SWCSCAU1 |        |

- Enter "s \*" on the command line to select all CA groups for processing

# Using the ISPF Interface

HPCA Selected CA Groups

Row 1 to 5 of 10

COMMAND ==> SET



SCROLL ==> PAGE

Enter one of the Line or Primary commands. Enter END (PF3) to exit.

## Line Commands:

- M - Display key defaults member list
- K - Keywords
  
- L - Allocate Log Spill Datasets
- S - Specify SORT Parameters

## Primary Commands:

- SET Setup run time environment
- GEN Generate CA-JCL
- ENV Show environment parms
- FAST Execute HPCA
- RUN Submit generated CA-JCL
- STAT Retrieve status of CA-job
- BAT Create Control parameters
- SORT Display by CAGROUP or PRTY

HPCA-ID: SWCS

| CMD | CAGROUP  | PRTY  | REUSE | SKEL.MBR | CATIME |
|-----|----------|-------|-------|----------|--------|
| --- | -----    | ----- | ----- | -----    | -----  |
| -   | SWCSCAQ1 | 01    | N     | HPCASKL2 |        |
| -   | SWCSCAR1 | 01    | N     | HPCASKL2 |        |
| -   | SWCSCAS1 | 01    | N     | HPCASKL2 |        |
| -   | SWCSCAT1 | 01    | N     | HPCASKL2 |        |
| -   | SWCSCAU1 | 01    | N     | HPCASKL2 |        |




From CAJCL parameter on INIT.CAGRP statement



# Using the ISPF Interface

SETUP

HPCA Setup Selection Menu

Command ==> ALL 

Enter a selection option. Enter END (PF3) to exit.

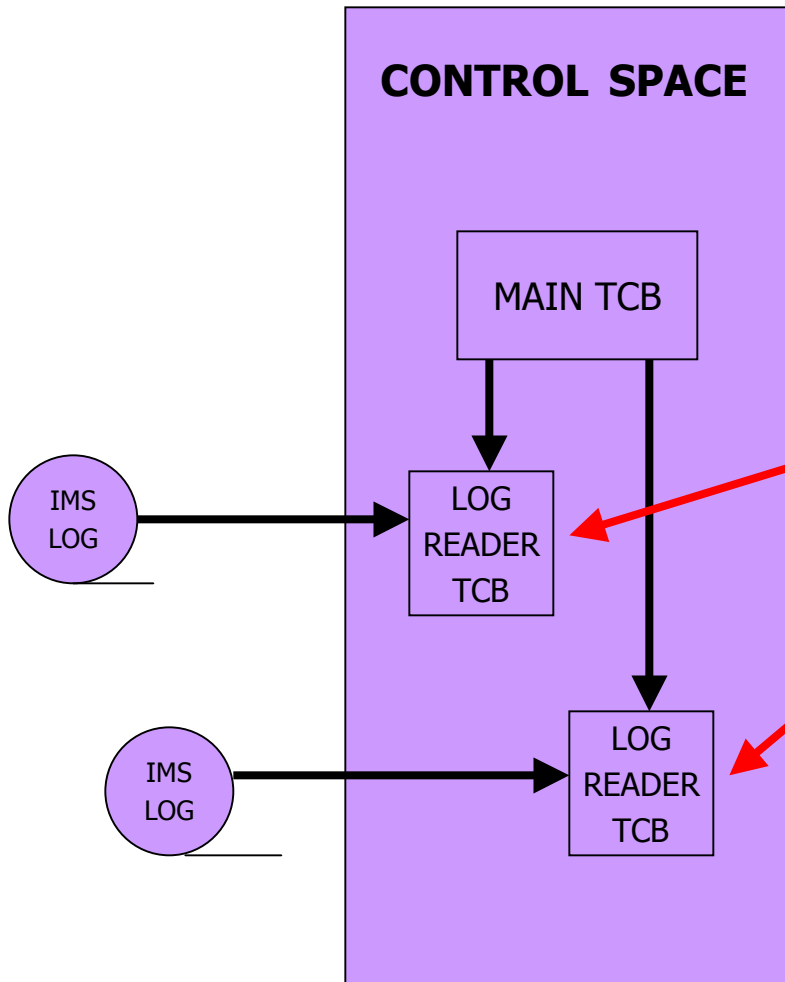
HPCA-ID: SWCS

Option. . . .

1. - JCL Job statements
2. - Global Control Parameters
3. - SORT Parameters
4. - Reserved
5. - LOG SPILL Parameters
6. - LOG SPILL Dataset Allocation Parameters
7. - Reserved
8. - Fastpath Work Dataset Allocation Parameters
9. - Fastpath SORT Parameters

A. - ALL of the above in succession

# Specifying Tasks

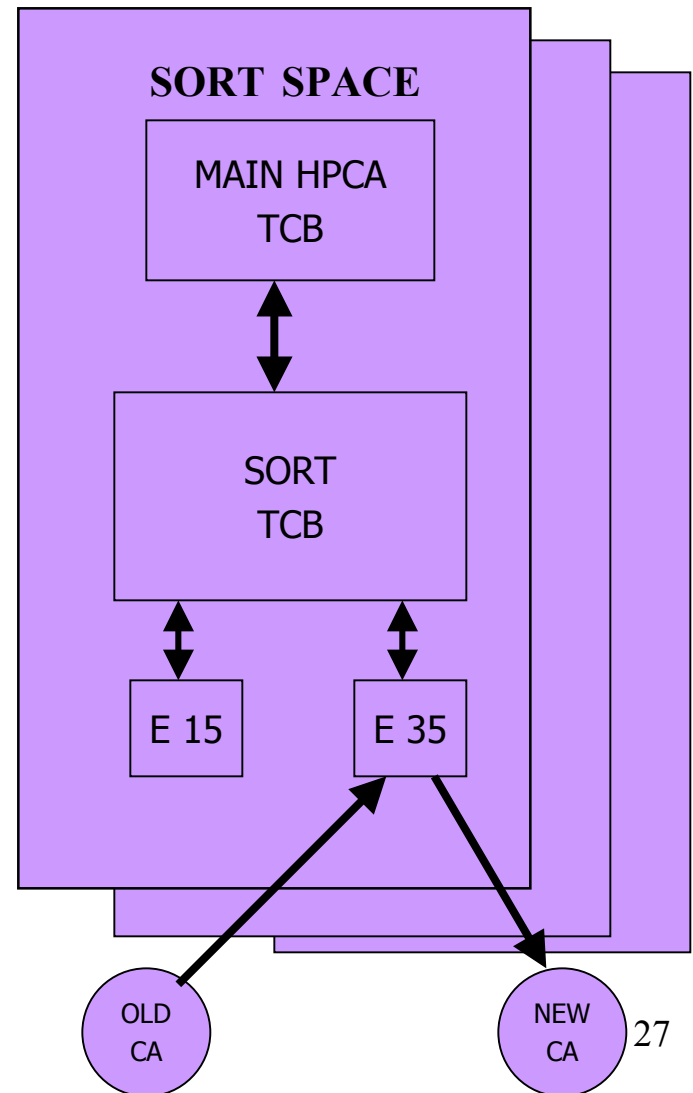


You control the number of concurrent log readers HPCA will use

# Specifying Tasks

---

You also control the number of output/sort tasks which execute concurrently.



# Specifying tasks

```
SETUP                HPCA Global Control Parameters

COMMAND ==>>> _____

Enter END (PF3) to exit. Enter CAN to exit without processing.

Global Control Parameters for CA-GROUP: ALL
  Max. number of input tasks: 6  ← (1-19. Default 9)
  Max. number of output tasks: 6 (1-19. Default 9)

      CA termination method: R      (A=ABEND, R=RC. Default R)
  I/O buffers for input datasets: 50 (1-255. Default 25)
  I/O buffers for output datasets: 50 (1-255. Default 25)

  HPCA Message dataset prefix: SYSOUT (Default SYSOUT)
  SORT Library specification if not in LINKLIB:
    SORT Load Library DSname: _____

      Write SMF records: N          (F=FULL, N=NO, S=SHORT. Default N)
```


- Maximum number of input tasks entry identifies how many parallel log readers are used
- Maximum number of output tasks entry identifies how many HPCA CA tasks will run in parallel

# Specifying spill files

```
SETUP          HPCA LOG SPILL Dataset Allocation Parameters      Row 1 to 2 of 2
COMMAND ==>> _____ SCROLL ==>> PAGE
```

Enter END (PF3) to exit. Enter CAN to exit without processing.

Log Spill dataset allocation parameters for CA-GROUP: ALL

```
Max number of SPILL volumes: 9  (1-9. Default 4)
Management class . . . . . : _____ (Blank for default management class
Storage class . . . . . : _____ (Blank for default storage class
Device type . . . . . : _____ (Generic unit or device address
Data class . . . . . : _____ (Blank for default data class
Space units . . . . . : CYLS (TRKS, CYLS)
Primary quantity . . . . . : 100 (In above units)
Secondary quantity . . . . . : 100 (In above units)
```

Volumes:

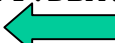
```
==> _____ ==> _____ ==> _____ ==> _____ ==> _____ ==> _____
==> _____ ==> _____ ==> _____ ==> _____ ==> _____ ==> _____
```


- If you need to split up the number of CA tasks running concurrently, a spill file will be created. Running out of space will cause HPCA to fail!

# Specifying Fast Path Sortout File

```
SETUP          FAST PATH Work Dataset Allocation Parameters      Row 1 to 2 of 2
COMMAND ==>> _____ SCROLL ==>> PAGE
```

Enter END (PF3) to exit. Enter CAN to exit without processing.

```
Work Dataset name prefix: SUPT.DBA.HPCA
Number of Output units: 9  (1-9. Default 2)
```

```
Management class . . . . . : _____ (Blank for default management class)
Storage class . . . . . : _____ (Blank for default storage class)
Device type . . . . . : _____ (Generic unit or device address)
Data class . . . . . : _____ (Blank for default data class)
Space units . . . . . : CYLS (TRKS, CYLS)
Primary quantity . . . . . : 2000  (In above units)
Secondary quantity . . . . . : 500 (In above units)
```

Volumes:

```
==> _____ ==> _____ ==> _____ ==> _____ ==> _____ ==> _____
==> _____ ==> _____ ==> _____ ==> _____ ==> _____ ==> _____
```

- If you have Fast Path, then all of the FP log records are sorted first. Running out of space will cause HPCA to fail!

## HPCA ISPF Panels

---

- Going through the panels should be a one time occurrence
- We rarely have gone back to the panels after initial setup

## GENJCL Considerations

---

- You must provide both DBRC commands and a GENJCL skeleton.
- GENJCL.CA commands
  - One statement is needed for each change accum group in the run
  - Read by the "generate" step (HPCABAT0)
- Change accum skeleton referred by GENJCL.CA statements
  - Various approaches - simple and complex
  - This presentation will focus on the simple
  - "Opportunities" where you may choose to improve this skeleton will be mentioned



# GENJCL Considerations

---

- **Sample GENJCL.CA commands:**

```
GENJCL.CA GRPNAME (SWCSCAQ1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
GENJCL.CA GRPNAME (SWCSCAX1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
GENJCL.CA GRPNAME (SWCSCAZ1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
GENJCL.CA GRPNAME (SWCSCAY1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
GENJCL.CA GRPNAME (SWCSCAU1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
GENJCL.CA GRPNAME (SWCSCAR1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
GENJCL.CA GRPNAME (SWCSCAV1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
GENJCL.CA GRPNAME (SWCSCAW1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
GENJCL.CA GRPNAME (SWCSCAT1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
GENJCL.CA GRPNAME (SWCSCAS1) NOJOB LIST MEMBER (HPCASKL2) NODEFLT
```

- **This is invoked by HPCABAT0**

# GENJCL Considerations

---

- GENJCL skeleton
  - You can choose any name you wish - this example uses HPCASKL2
  - This was modified from existing skeletons
  - This does use DD ICBUCUMN for off-premise (D/R) copy
    - This DD statement was added by the Parallel Change Accum (PCA) product and was found to be very useful for this purpose

# GENJCL Considerations

---

- Skeleton sample

```
//SWCSCA1S JOB (SWCH,31), 'DBA', CLASS=N, MSGCLASS=E, REGION=0M
//SWCSCA EXEC PGM=DFSUCUM0, PARM='CORE=1500000, DBRC=Y'
//STEPLIB DD DSN=IMSWCS.SDFSRESL, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//IMS DD DSN=SWCS.BCR.DBDLIB, DISP=SHR
//SYSOUT DD SYSOUT=*
%DELETE (%CAODSN EQ '')
//DFSUCUM0 DD DSN=%CAODSN,
// DISP=OLD, DCB=BUFNO=20
%ENDDDEL
%DELETE (%CAODSN NE '')
//DFSUCUM0 DD DUMMY, DCB=BLKSIZE=100
%ENDDDEL
```

# GENJCL Considerations

---

- Skeleton sample (cont.)

```
//DFSUCUMN DD DSN=SWCS.CA.%CAGRP.CA%DATE.CA%TIME,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=TAPE90,
//          RETPD=14,
//          DCB=BUFNO=20,
//          VOL=(,,,30)
//ICBUCUMN DD DSN=SWCS.OFFPEXPD.%CAGRP.CA%DATE.CA%TIME,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=BUFNO=30,
//          RETPD=14,
//          UNIT=TAPE90
//DFSUTAP  DD UNIT=(TAPE90,1,DEFER)
%SELECT  RLDS((%CAGRP),(FROM(%DSL LGTM)))
//DFSULOG DD DSN=%LOGDSN,
//          DISP=SHR,UNIT=AFF=DFSUTAP,DCB=BUFNO=20
%ENDSEL
%DELETE (%LOGSEL EQ 'YES')
//DFSULOG  DD DUMMY,DCB=BLKSIZE=100      NO INPUT LOG
%ENDDEL
//SYSIN    DD  *
%CADB0
```

# JCL Considerations

---

- Change accum process must take place in two separate steps
- You can add additional "housekeeping" steps
- We run it as a single job
- This execution JCL is very stable

# JCL Considerations

---

- Step 1 - Delete previous datasets
  - This is necessary for Fast Path applications in the event of a failure
  - We delete the SORTOUT dataset
  - We use IDCAMS to accomplish this

# JCL Considerations

---

- Step 2 - Compress datasets
  - Some library datasets are updated on a frequent basis so we compress them using IEBCOPY
    - CONTROL
    - PROCLIB
    - STCJOBS (the generated one, not the system one!)
  - As an option you can also define these datasets as PDS-E and eliminate this step.

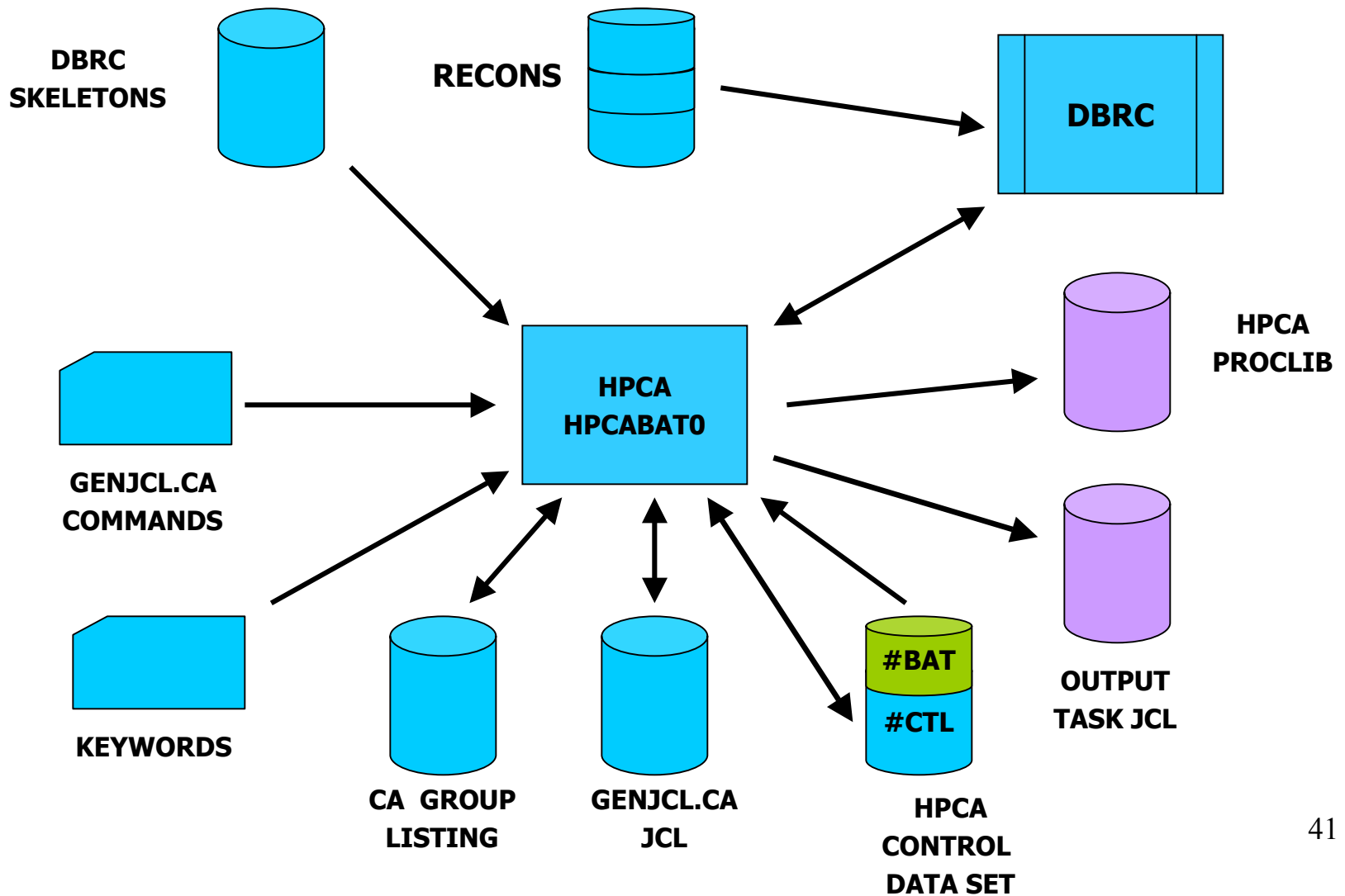
# JCL Considerations

---

- Step 3 - HPCABAT0
  - This is the generate step and uses the GENJCL.CA statements mentioned earlier
  - This step updates several of the controlling datasets used by HPCA
    - HPCABAT0 parses the generated jobs to update these datasets. Much of the information is stored in the control dataset (HPCACTL).
    - New procs are generated for each started task



# JCL Considerations: HPCABAT0



# JCL Considerations

---

## Sample HPCABAT0 JCL

```
//STEP03 EXEC PGM=HPCABAT0, PARM='SWCS'  
//STEPLIB DD DSN=IMSUPTS.HPCA.SHPCLMD0, DISP=SHR  
// DD DSN=IMSWCS.SDFSRESL, DISP=SHR  
//HPCACTL DD DSN=SUPT.DBA.HPCA.CONTROL, DISP=SHR  
//HPCAPROC DD DSN=SUPT.DBA.HPCA.PROCLIB, DISP=SHR  
//HPCAJLIB DD DSN=SUPT.DBA.HPCA.STCJOBS, DISP=SHR  
//** DSN=SYS1.STCJOBS ← Commented out  
//SYSUDUMP DD SYSOUT=*  
//SYSPRINT DD DSN=SUPT.DBA.HPCA.SYSPRINT, DISP=SHR  
//JCLOUT DD DSN=SUPT.DBA.HPCA.JCLOUT, DISP=SHR  
//JCLPDS DD DSN=DBAM.SWCS.SKELSPDS, DISP=SHR  
//GENJCLIN DD DSN=DBAM.SWCS.SKELSPDS(HPCASKEL), DISP=SHR  
//SYSOUT DD SYSOUT=*, DCB=BLKSIZE=133
```

GENJCLIN DD statement contains the GENJCL.CA statements

# JCL Considerations

---

- Step 4 - HPCADRIV
  - This step controls the execution of the change accumulation run
    - Starts the multiple readers
    - Starts the Fast Path sort task
    - Starts the individual CA tasks
  - HPCABAT0 (Step 3) must complete with  $cc = 0$ 
    - If all groups have no data to accumulate HPCABAT0 will complete with  $cc = 4$ . We found this on the Monday of a 3-day weekend.

# JCL Considerations

---

## HPCADRIV sample

```
// IF (STEP03.RC EQ 0) THEN
//IEFPROC EXEC PGM=HPCADRIV,REGION=3000K,
//          PARM='&PARM1'
//*
//STEPLIB DD DISP=SHR,
//          DSN=IMSUPTS.HPCA.SHPCLMDO
//          DD DISP=SHR,
//          DSN=IMSWCS.SDFSRESL
//HPCACTL DD DISP=SHR,
//          DSN=SUPT.DBA.HPCA.CONTROL(SWCS#CTL)
//HPCMSG00 DD SYSOUT=*
//SYSSNAP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
// ENDIF
```

## JCL Considerations

---

- HPCA uses started tasks invoked by HPCADRIV
- Task name is 4 character HPCA ID followed by 4 numeric characters assigned by HPCA.
- One task for each CA group.
- Optionally one task for Fast Path sort task

## JCL Considerations

---

- We put the simple execution JCL in SYS1.STCJOBS for each started task.
- HPCABAT0 will recreate this JCL but we write it to a separate library
- The task JCL points to a proclib member which changes for each run

# JCL Considerations

---

- If you have Fast Path the preliminary sort task is always numbered "0000".
- Sample Fast Path sort procedure:

```
//SWCS0000 PROC RGN=3000K,SOUT=*,
//          PARM1=
//IEFPROC EXEC PGM=HPCADRI2,REGION=&RGN,
// PARM='&PARM1'
//HPCMSG00 DD SYSOUT=&SOUT
//SYSSNAP DD SYSOUT=&SOUT
//SYSOUT DD SYSOUT=&SOUT
//SYSPRINT DD SYSOUT=&SOUT
//SYSUDUMP DD SYSOUT=&SOUT
//STEPLIB DD DISP=SHR,

//          DSN=IMSUPTS.HPCA.SHPCLMD0
```

- SORTIN and SORTOUT are dynamically allocated

## JCL Considerations

---

- Change accumulation tasks are numbered starting at "0001" and incremented by one.
- Each change accumulation group has its own started task assigned.
- The tasks are numbered in the order of the GENJCL.CA statements read by HPCABAT0.



# JCL Considerations

---

- Sample procedure for CA started task

```
//SWCS0010 PROC RGN=3000K,SOUT=*,
//          PARM1=
//IEFPROC  EXEC PGM=HPCADRI1,REGION=&RGN,
//  PARM='&PARM1'
//STEPLIB  DD DSN=IMSWCS.SDFSRESL,DISP=SHR
//          DD  DISP=SHR,
//          DSN=IMSUPTS.HPCA.SHPCLMDO
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//IMS      DD DSN=SWCS.BCR.DBDLIB,DISP=SHR
//SYSOUT   DD SYSOUT=*
//HPCMSG00 DD SYSOUT=&SOUT
//SYSSNAP  DD SYSOUT=&SOUT
//DFSUCUMO DD DSN=SWCS.CA.SWCSCAS1.CA02163.CA170021,

//          DISP=OLD,DCB=BUFNO=20
```

# JCL Considerations

---

- Sample procedure for CA started task (cont.)

```
//DFSUCUMN DD DSN=SWCS.CA.SWCSCAS1.CA02164.CA095011,  
//          DISP=(NEW,CATLG,DELETE),  
//          UNIT=TAPE90,  
//          RETPD=14,  
//          DCB=BUFNO=20,  
//          VOL=(,,,30)  
//ICBUCUMN DD DSN=SWCS.OFFPEXPD.SWCSCAS1.CA02164.CA095011,  
//          DISP=(NEW,CATLG,DELETE),  
//          DCB=BUFNO=30,  
//          RETPD=14,  
//          UNIT=TAPE90
```

# JCL Considerations

---

- Sample procedure for CA started task (cont.)

```
//DFSUTAP DD UNIT=(TAPE90,1,DEFER)
//DFSULOG DD DSN=SYS5.PRI.D02163.T1623276.IMSSWCS.SD.IMSLOGS,
//          DISP=SHR,
//          UNIT=AFF=DFSUTAP,
//          DCB=BUFNO=20
//          DD DSN=SYS5.PRI.D02163.T1704240.IMSSWCS.SD.IMSLOGS,
//          DISP=SHR,
//          UNIT=AFF=DFSUTAP,
//          DCB=BUFNO=20
//          DD DSN=SYS5.PRI.D02163.T1737318.IMSSWCS.SD.IMSLOGS,
//          DISP=SHR,
//          UNIT=AFF=DFSUTAP,

//          DCB=BUFNO=20
```

## JCL Considerations

---

- Log inputs are defined in all of the task procedures but are not read by them
- Logs are read by HPCADRIV
- The started tasks compare the logs read to the DFSULOG datasets to make sure they are bringing in the right logs.
- It's confusing...

# Performance Analysis

---

- CA runs compared 6 days before and after converting from PCA to HPCA
- Full function data bases, 4 CA groups, 2 HPCA input readers, no spill files
- STK Virtual Tape Subsystem used
- Change accum files moved from tape to disk for HPCA
- Run time decreased by 37%, CPU utilization rose by 18%
- Parallel input readers caused most of the improvement in elapsed time

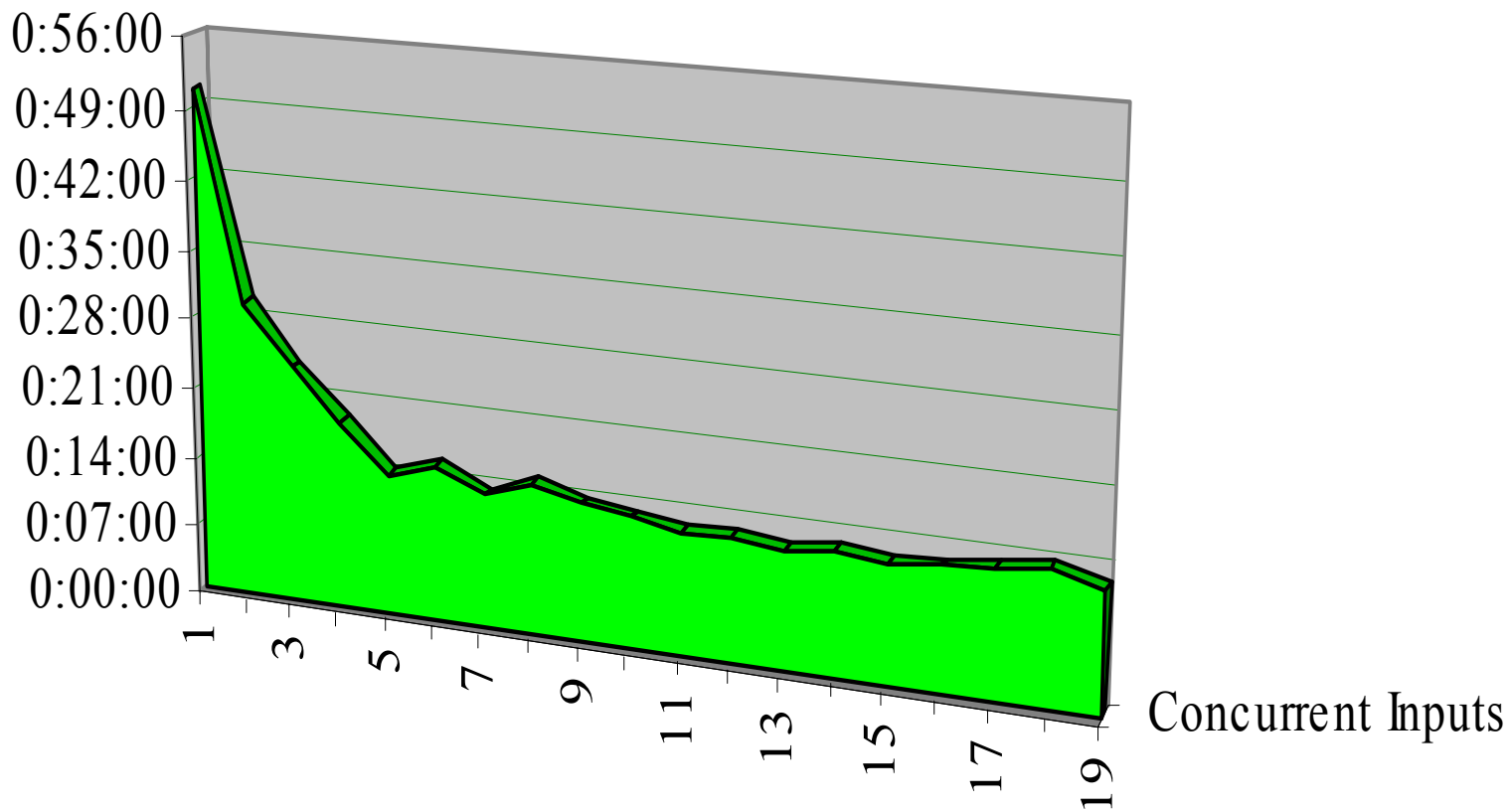
# HPCA Input Performance

---

- Tests made varying number of reader tasks from 1 up to 19
- Change Accum runs repeated using same 19 log files
- Tape subsystem used was STK Virtual Tape Subsystem
- Log files were staged and no spill files were used
- Hardware pushed to limit when reader tasks greater than 5

## HPCA "Inputs" Performance Data

Run Time



# HPCASingle Tape Read Times

Run Times

0:12:00

0:10:00

0:08:00

0:06:00

0:04:00

0:02:00

0:00:00

1

3

5

7

9

11

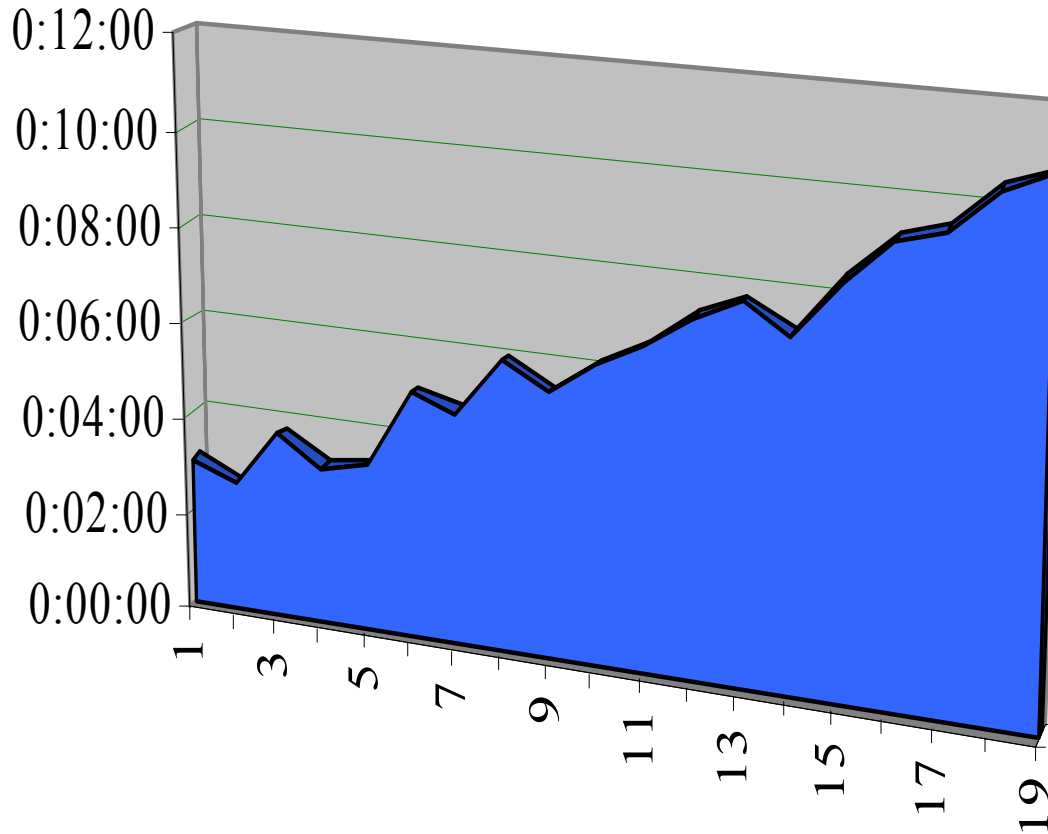
13

15

17

19

Concurrent Inputs





## Other Considerations

---

- RACF must be defined properly
- Started tasks need to be defined
- If using system started task library (SYS1.STCJOBS) you must have authority to update the library
- Started tasks must have authority to update output CA datasets and read input files

## Other Considerations

---

- If you have Fast Path, you may consider running HPCA more frequently.
- You will get dataset contention if you are accessing the ISPF panels while HPCABAT0 is running.

# Our Wish List

---

- We want to be able to limit the number of input logs brought into an HPCA run
- Fast Path processing is single-threaded and is a bottleneck
- Results checking is difficult. We want the results of the started tasks to be incorporated into the output of HPCADRIV.
- GENJCL requirement removed. We want HPCA to be control card driven.
- ÷ We want to have the logs removed from the started task procs and have the log validation done some other way.

# Implementation Suggestions

---

- Put a front-end on the ISPF interface.
  - This will eliminate the need to teach everyone what the CLIST name is (and where it resides!)
  - This allowed our DBA's enough flexibility to set up their own change accum runs
- Run a parallel test prior to going live
  - This will validate whether all of your pieces are in place before you really need them
- Make sure you are current on maintenance
- Implementing HPCA for a Sysplex rather than for individual IMS systems reduced administration of datasets

# Implementation Suggestions

---

- Put your largest groups first
  - This will decrease the size of the spill file if you use one
  - The product does not do this for you automatically
- Use deferred tape mounts in your skeleton
- Do not edit the control dataset
  - Doing this is equivalent to playing with fire!
- Do not separate your HPCABAT0 and HPCADRIV steps
  - Always run them together
  - Running HPCADRIV with the wrong control dataset can cause serious consequences to your change accumulation process

# Conclusions

---

- HPCA runs and is doing a good job for us!

# Questions???

---

- Please feel free to contact us if you have questions:
  - Neil Schechter - [neil.schechter@sbc.com](mailto:neil.schechter@sbc.com)
  - Jim Kelly - [james.kelly@sbc.com](mailto:james.kelly@sbc.com)