ibm.com

# Neat New V5R2
# Journal and Commit enhancements

BP09

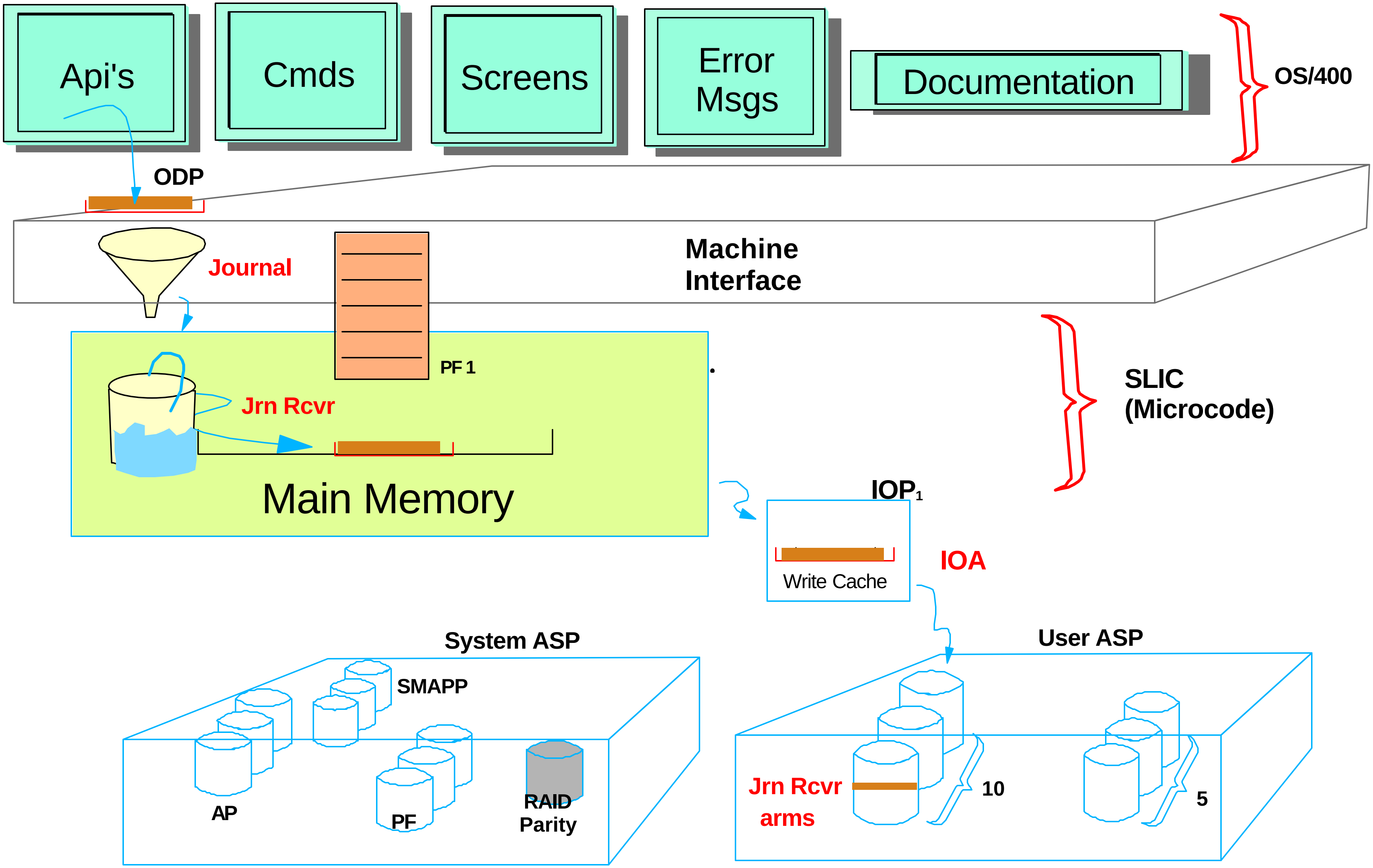
ITSO iSeries Technical Forum

Larry Youngren

Redbooks

International Technical Support Organization

# The Journal environment

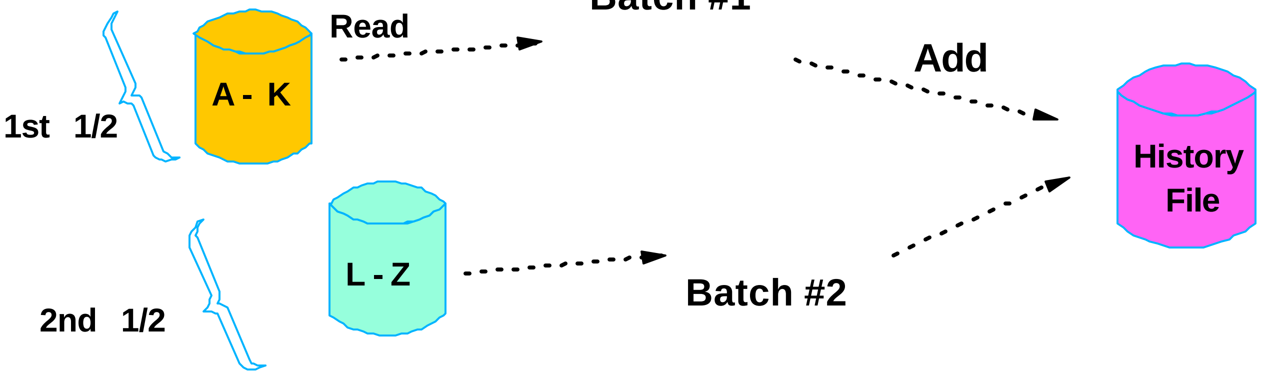


# Parallel jobs and Holey Inserts

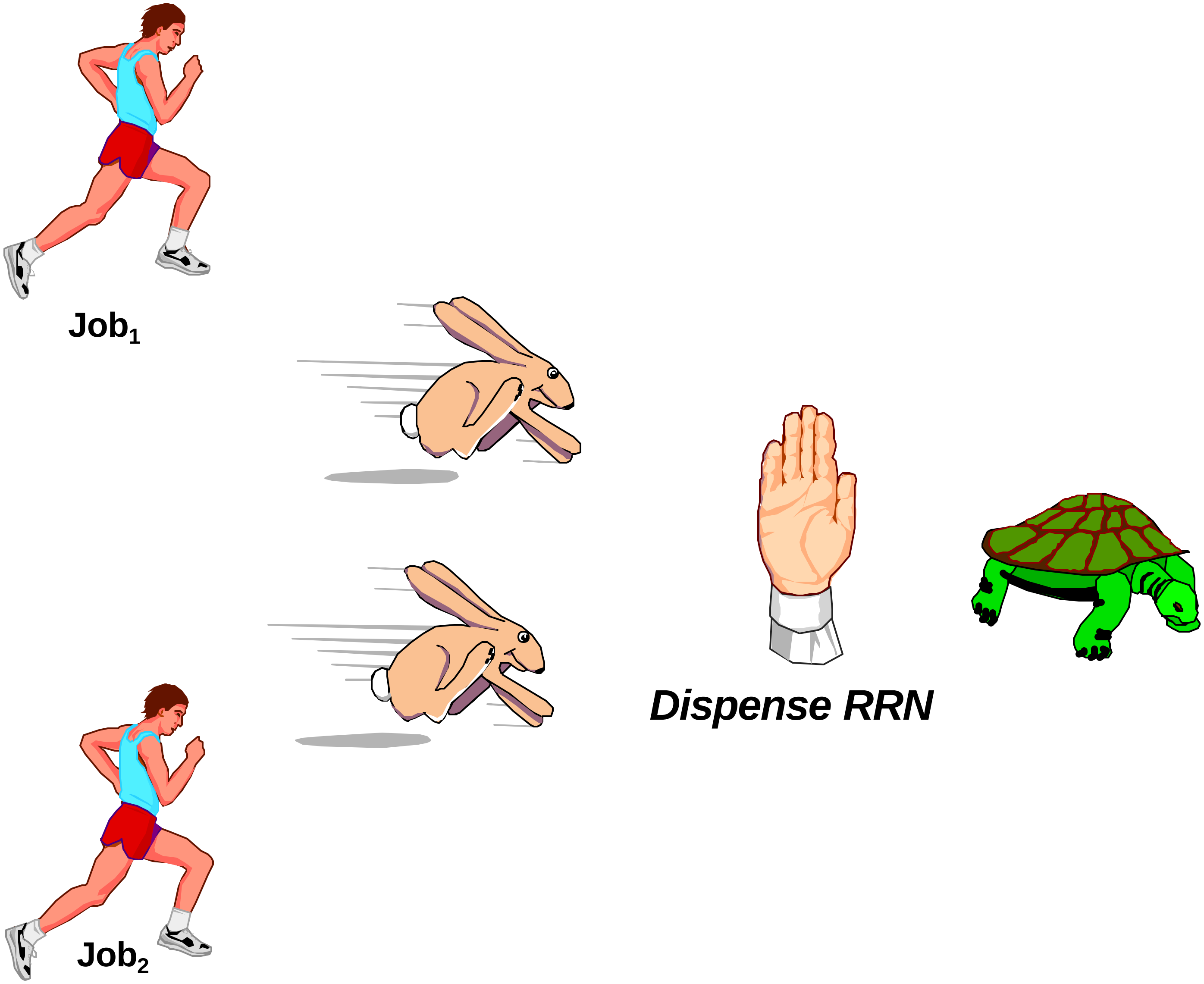
# The Problem:

**Many shops elect to reduce their batch windows by breaking up the original batch job into multiple threads**

- This is often a wise and effective move
- But... if the resulting threads are all still competing for the same Database file and if they're adding new records, they still must proceed **single-file**



# *The bottleneck:*



Job$_1$

Job$_2$

***Dispense RRN***

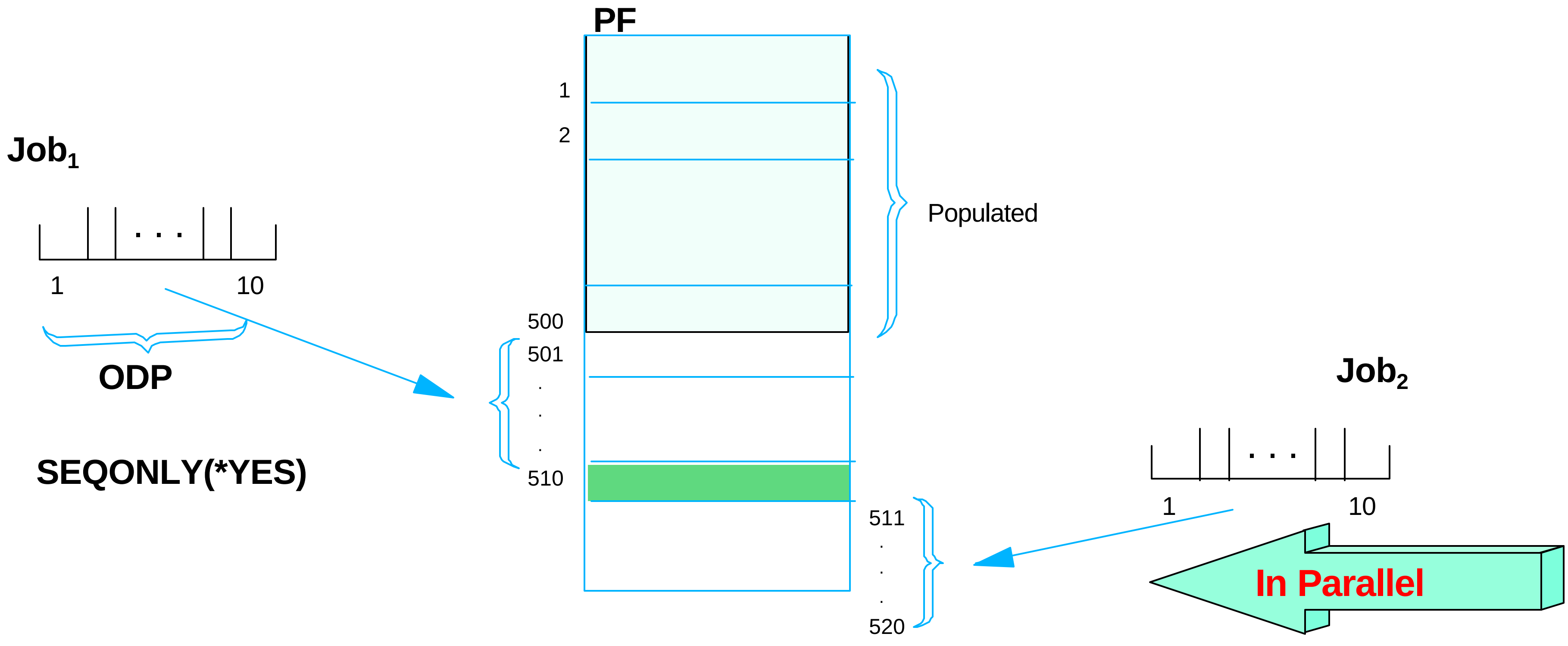
# The Solution:

- **Concurrent holey inserts**

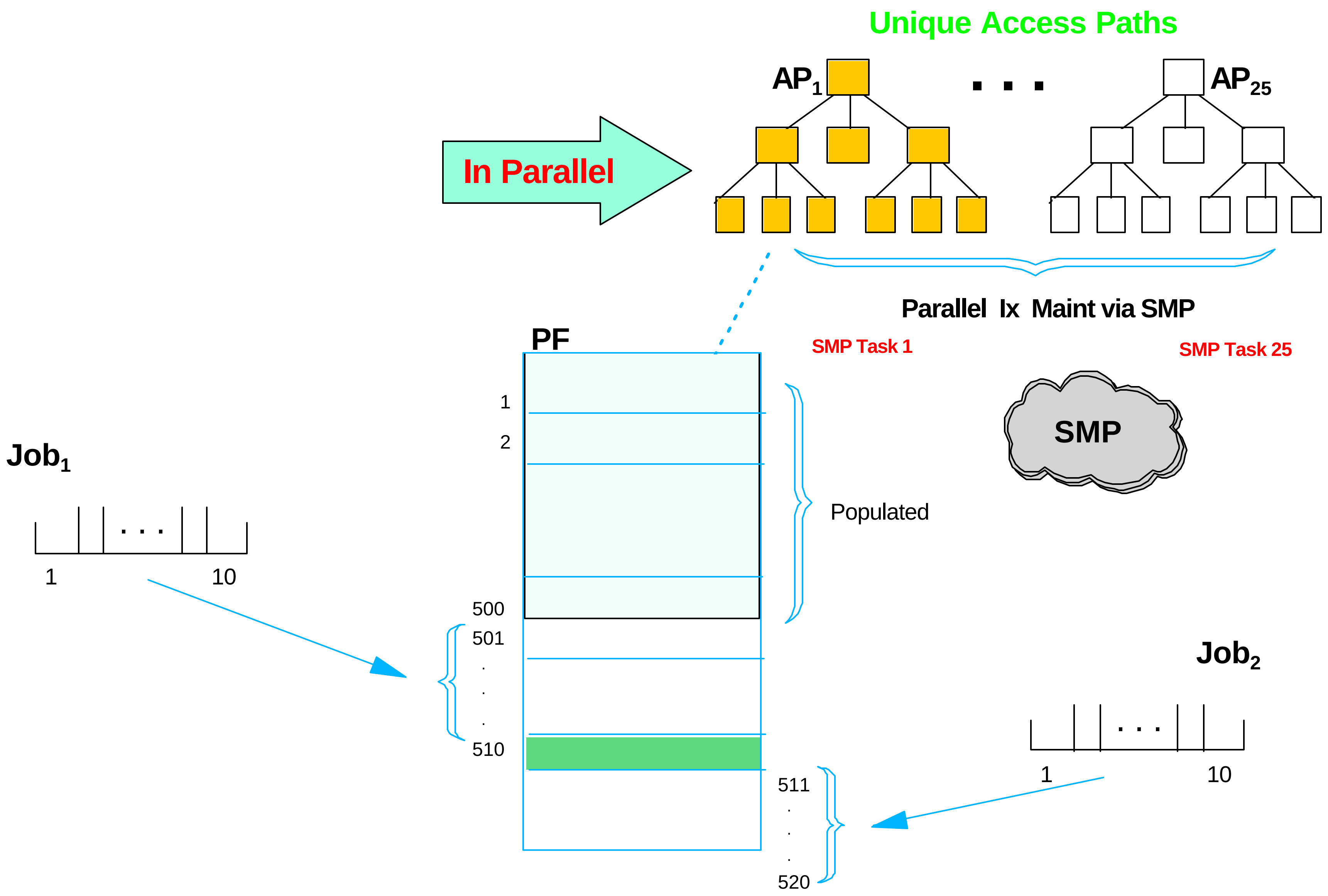
# Concurrent Holey Adds

CALL QDBENCWT '1'

This enables the support
(takes effect after the next IPL)



# Secondary SMP Parallelism for Index maintenance



Unique Access Paths

$AP_1$ . . . $AP_{25}$

In Parallel

Parallel Ix Maint via SMP

SMP Task 1

SMP Task 25

SMP

PF

Populated

$Job_1$

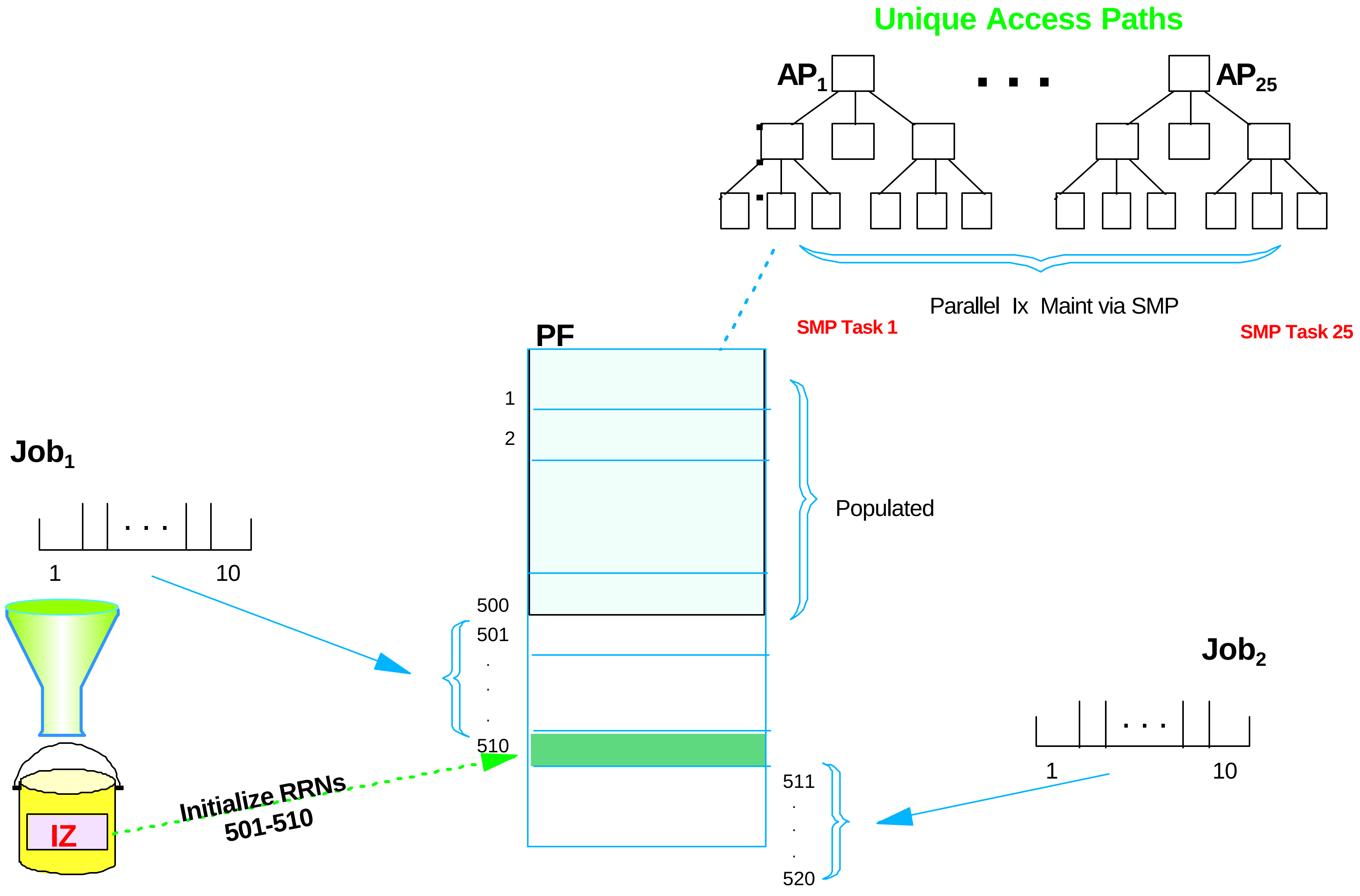
$Job_2$

# A Timing concern:

- **What happens if the concurrent jobs don't finish in strict starting order ?**
- How is our HA Vendor replay job going to cope with RRNs that seem to be out-of-order ?

**Please:   Not the dreaded  "Out_of_Sync" condition !**

# Using the Journal to keep timings straight



# New V5R2  Concurrent Holey Add support

- **Concurrent adds can make a substantial performance difference**
- **But...**
  - It has been difficult to replay them reliably without risking "Out_of_Sync"
- **For V5R2 we'll begin emitting  IZ  journal entries for these cases**
  - The **IZ** will reveal number of deleted entries to be inserted and RRNs
  - The IZ will be followed by proper number of **PX** entries

V5R2!

**Net:**

- **HA vendor software should cease seeing RRNs "out of order"**
- **Replay of IZ and PX should eliminate the former concerns**

# Summary

Speed up batch jobs by:

- ★ Break into parallel threads
- ★ Specify SEQONLY(*YES)
- ★ Install and Enable SMP
- ★ Use the API to enable Holey Inserts
- ★ Request a version of H Vendor software that's prepared for IZ and PX entries

# Customizing a System_Managed Journal

# Customizing your System-Managed wait/Retry values

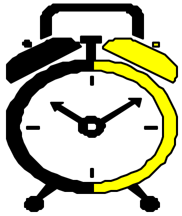
In V5R2 we are letting you tailor the **retry times** for CHGJRN/DLTJRNRCV  for **system managed** journals.

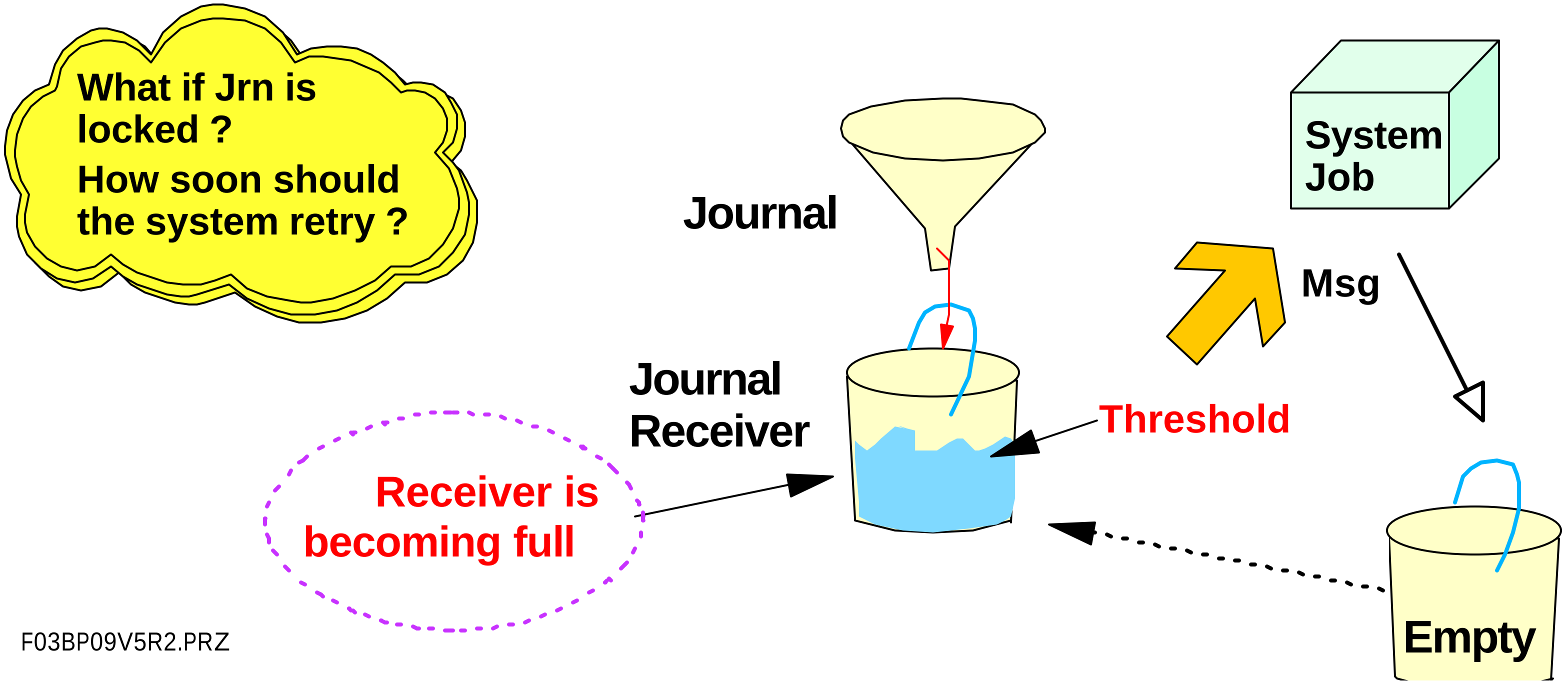
V5R2!

Two new parameters on CRTJRN/CHGJRN:

**MNGRCVDLY and DLTRCVDLY.**

Values can be 1 to 1440 minutes (a full day).  We had seen instances where 10 minutes was too long for some shops to wait to do the CHGJRN and the journal receiver hit **full,** also saw instances where re-checking every 10 minutes to do the delete when an exit program kept voting No was chewing up too much system resource.



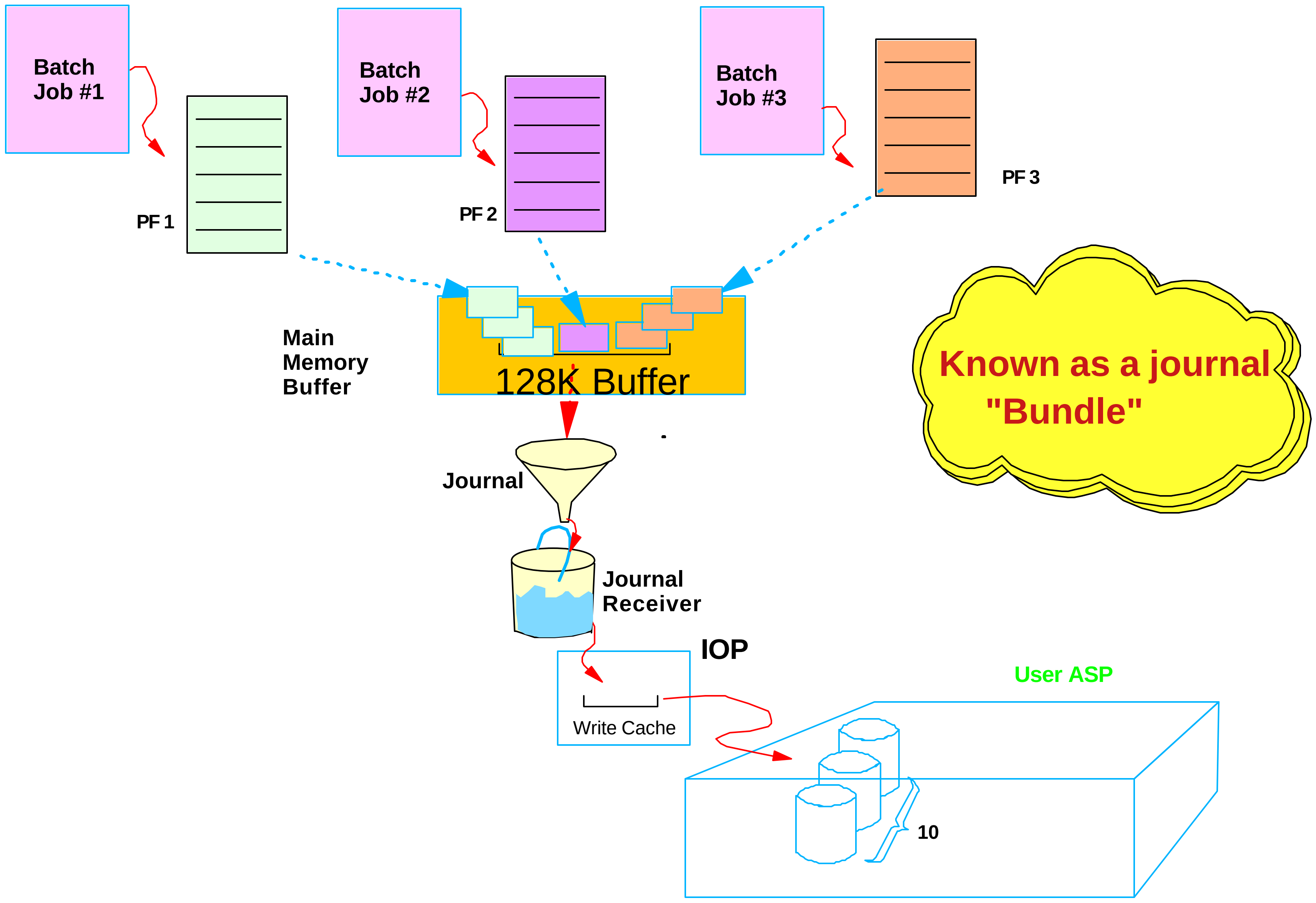


**What if Jrn is locked ?**

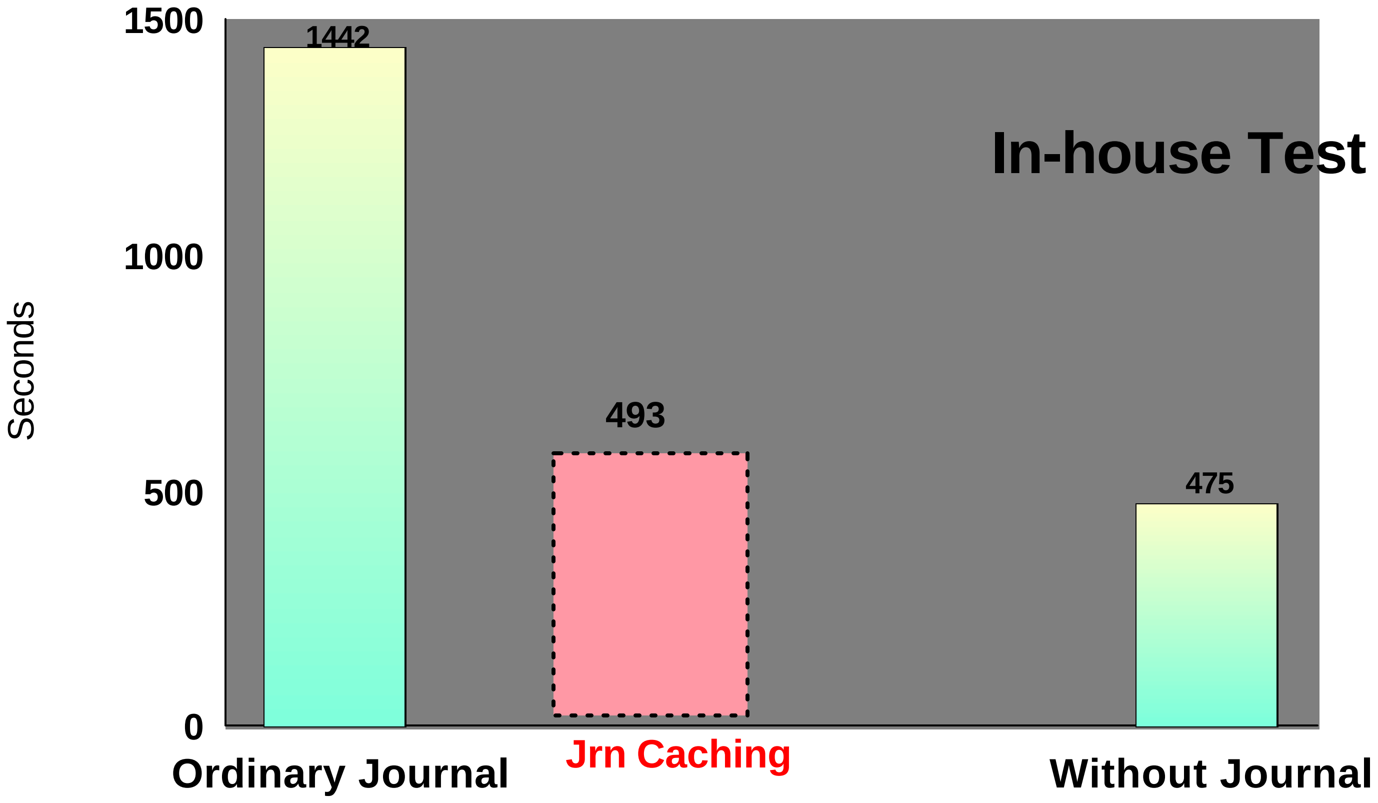
**How soon should the system retry ?**

# Aggressive Journal Caching

# Journal Bundling



# Impact of Jrn Caching



***DB Updates during Batch***

# For V5R2: The former PRPQ grows up !

- Becomes a priced feature of the Operating System

  Option 42 = **HA Journal Performance**

- ➢ Appears on **CHGJRN** command

  **CHGJRN JrnCache(*yes)**

- **Impact becomes broader**
  - ➢ **It's not just for DataBase any more**

    **Caching also for Data Areas, Data Q's, Stream Files**

# Don't limit your thinking to only Database objects

**Your applications *may* store critical data outside of the database:**

**Non-Database objects** can now be journaled/replicated

- New Objects eligible for Journaling:
  - **Data Areas**
  - **Data Queues**
  - **IFS objects**

**And...**

- **All of these can be "cached" as well !**

V5R2!

# Potential Performance Benefits

**Batch Job**

- 5 Million DB operations (10% Adds, 90% Updates)
- 9 Million resulting Journal entries

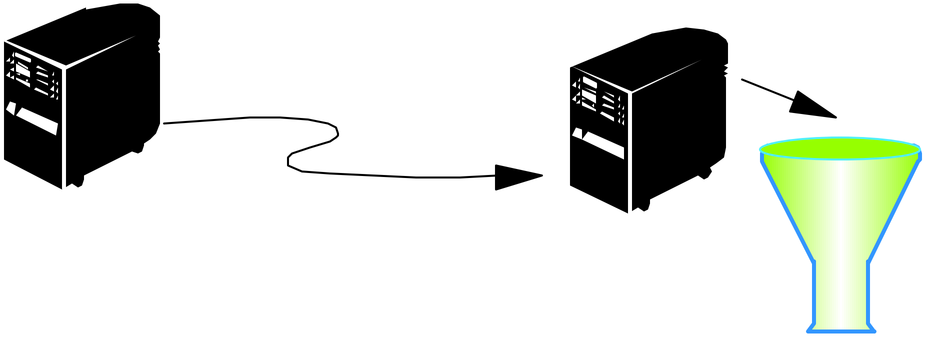
  (captured both before and after images)

| | **Elapsed Time** | |
|---|---|---|
| **Original Batch run, no Journaling** | **1118 Sec** | **Base Run** |
| **Ordinary Journaling enabled** | **9773 Sec** | |
| **Using the new Journal cache** | **1433 Sec** | |

# Potential Performance Benefits on Target side

(For Target Machine's Keep-up Mode)

| | Apply rate on Target machine |
|---|---|
| Before Caching | 600,000 transactions/Hr |
| With Caching on target | 2,400,000 transactions/Hr |



# Summary

Net:

- ★ Order & Install  Option 42
- ★ Enable  JrnCache(*Yes)
  - ➢ On both Source and Target systems

V5R2!

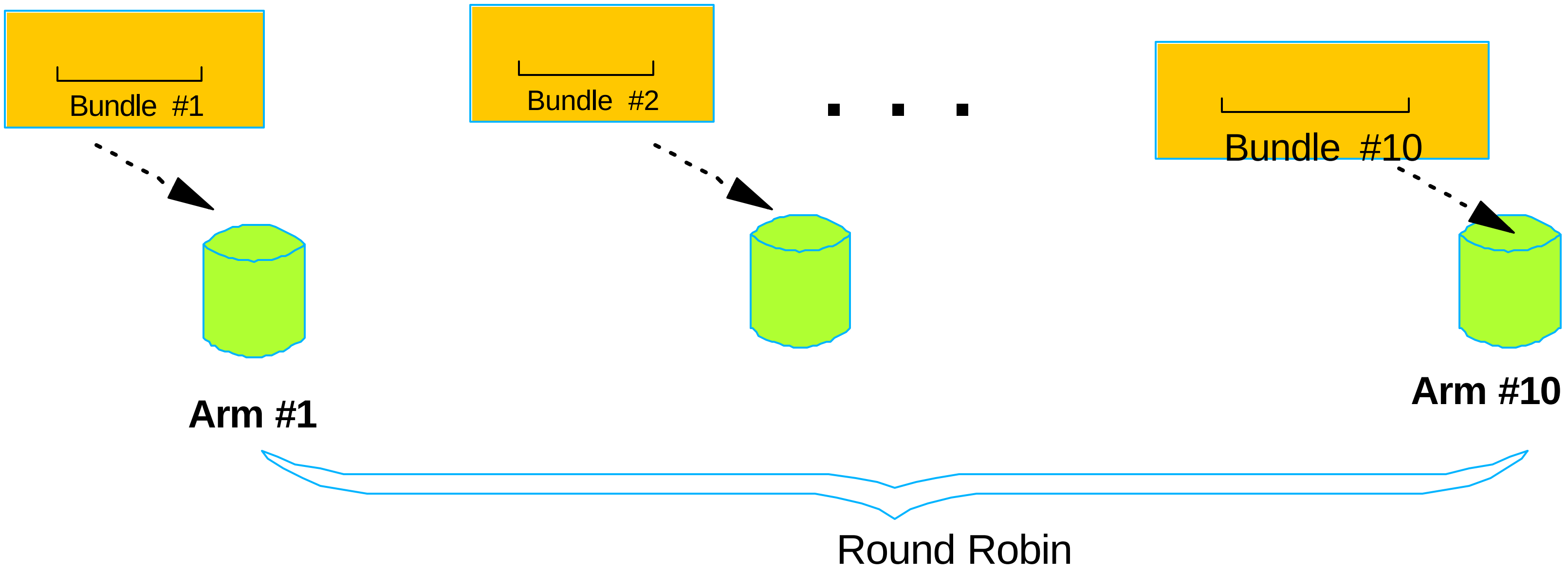
# Peeking behind the curtain

# Observing Journal "Bundling" characteristics

V5R2!

★ V5R2 has a new DSPJRN format (*TYPE5)

- You can "see" the **disk arm #** per Jrn Entry
- You can determine the average journal "**bundle**" width



# ***The Jan & Feb '03 issues of <u>iSeries NEWS</u>***

New V5R2 Journal features:

- ★ Eight "Tips" for Best Performance

Provides a copy of a sample program to calculate average bundle width

# Summary

Conduct a Caching investigation and analysis:

- ★ Run the sample cache-size analysis program
- ★ If you're not achieving optimal cache sizes...
  - i.e.  Seeing nearly average buffers 128k or wider
  - Find out WHY !

And now,

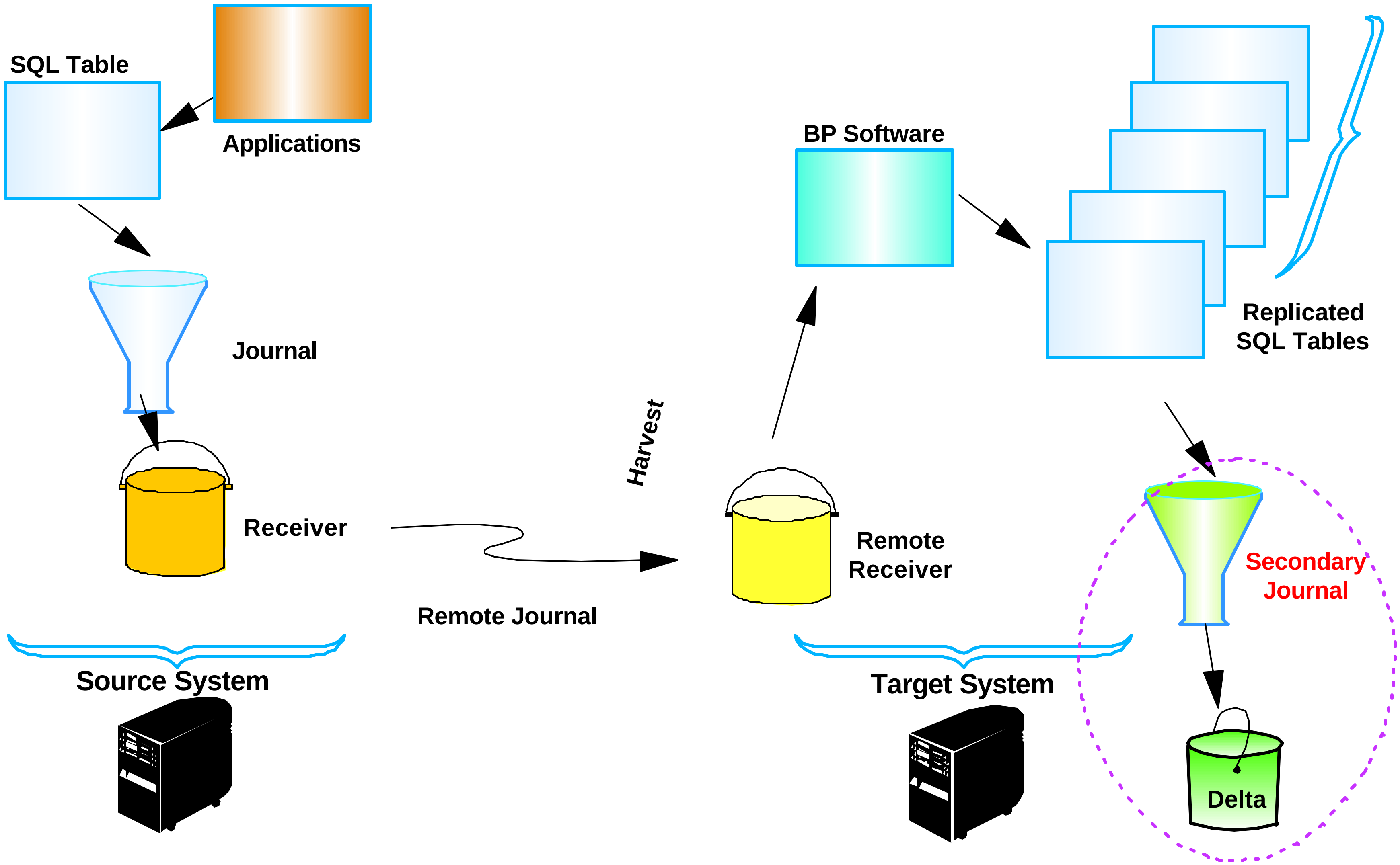
my personal favorite...

# Standby Journal

# The Problem:

- Applications need to fail over <u>rapidly</u> from production system to back-up system
  - You may want  to protect/replicate **thousands** of  objects
    - One ERP product replicates more than 64,000 database tables !
  - All of these objects are journaled on the <u>source</u> system
    - They also need to be journaled on the target system after the fail over
- It's expensive to keep them fully journaled on the target <u>before</u> it becomes the production machine
    - Being fully journaled slows down the HA BP "apply" jobs
  - It's similarly slow to start up journaling <u>after</u> the fail over
- The Solution:
  - **Journal Standby mode**

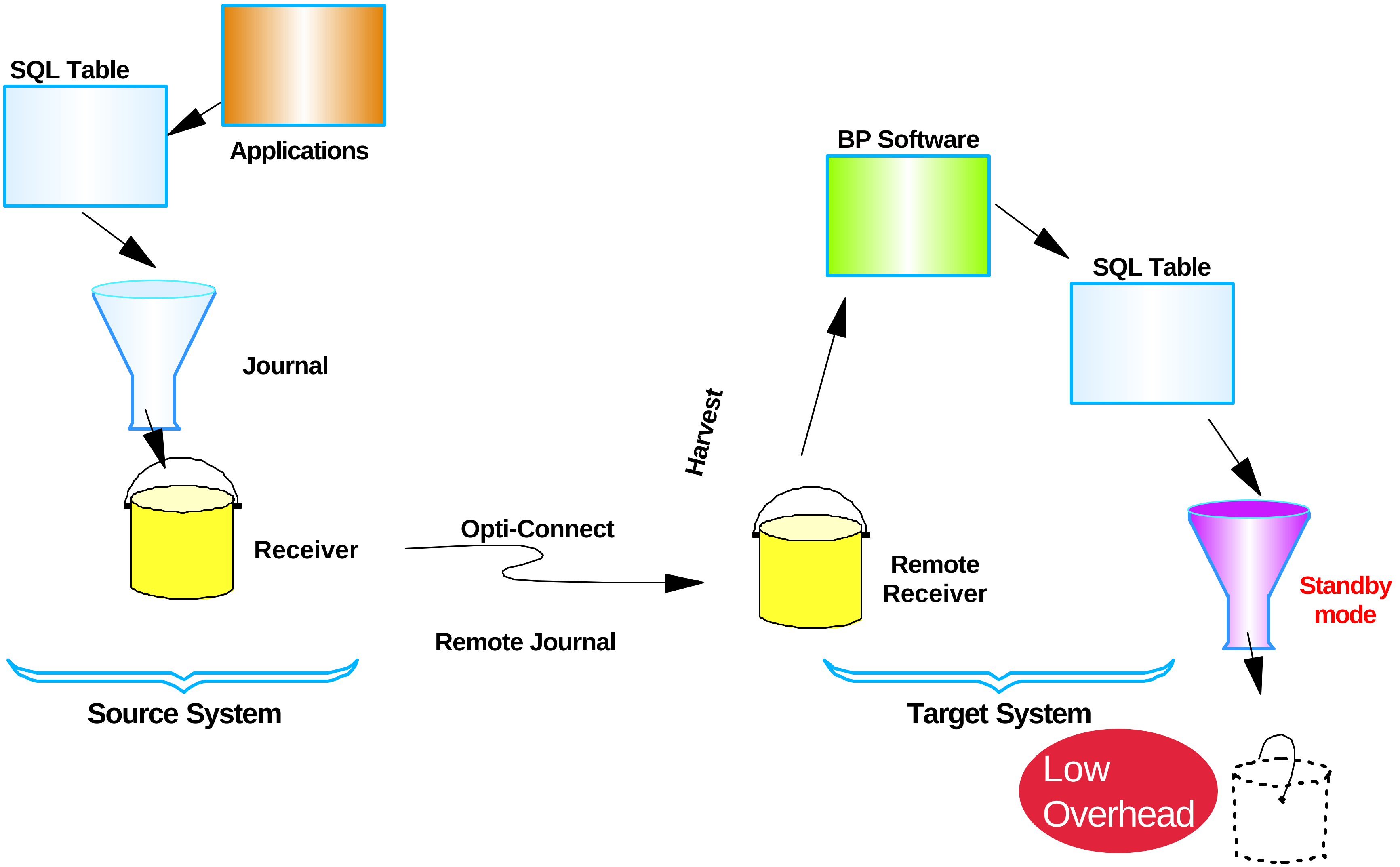
# *Cost of Enabling Secondary Jrn support on Target Sys*



# Standby mode

V5R2!

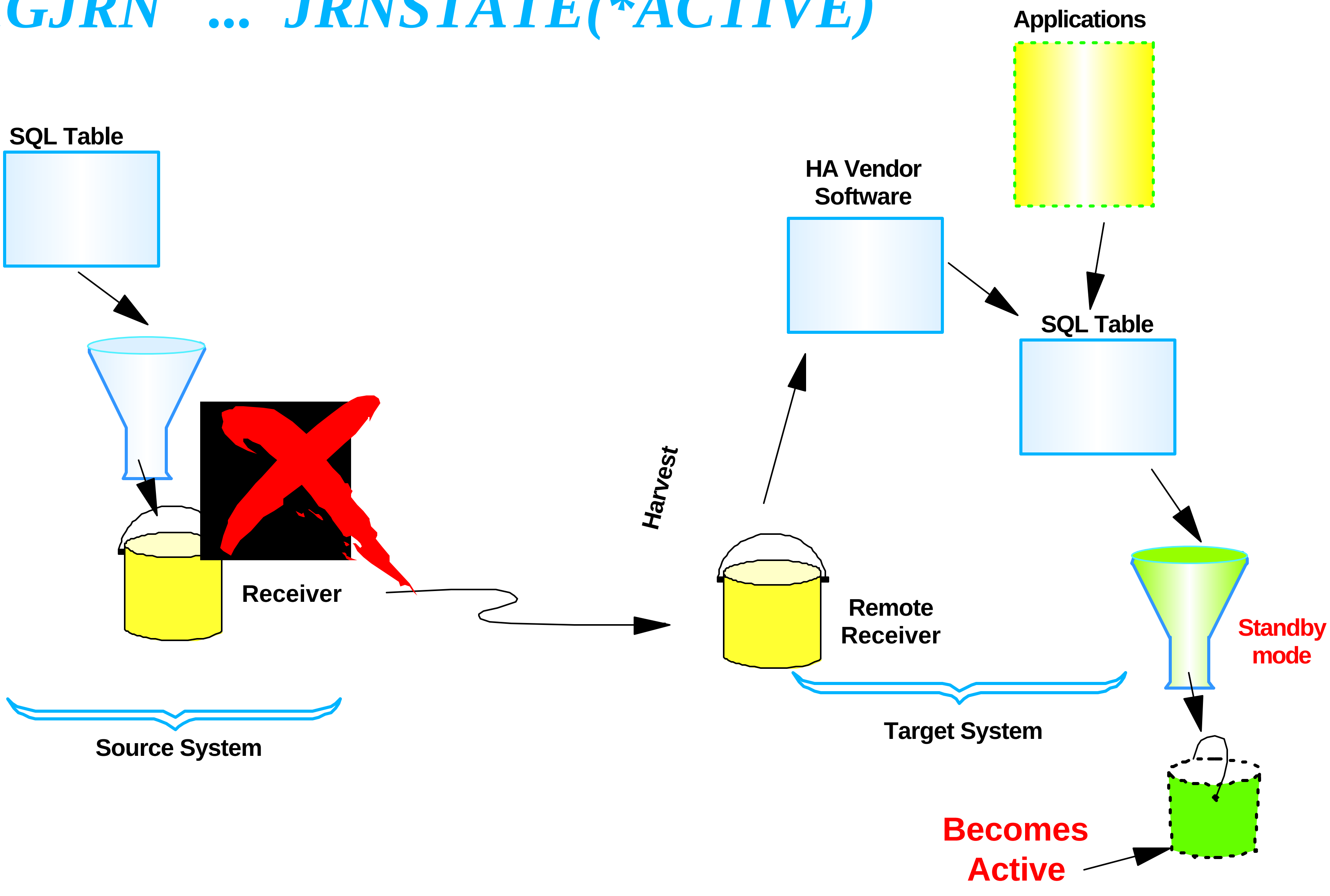
## CHGJRN ... JRNSTATE(*STANDBY)



Low Overhead

# Role Swap: *Switching out of Standby mode*

## *CHGJRN ... JRNSTATE(*ACTIVE)*



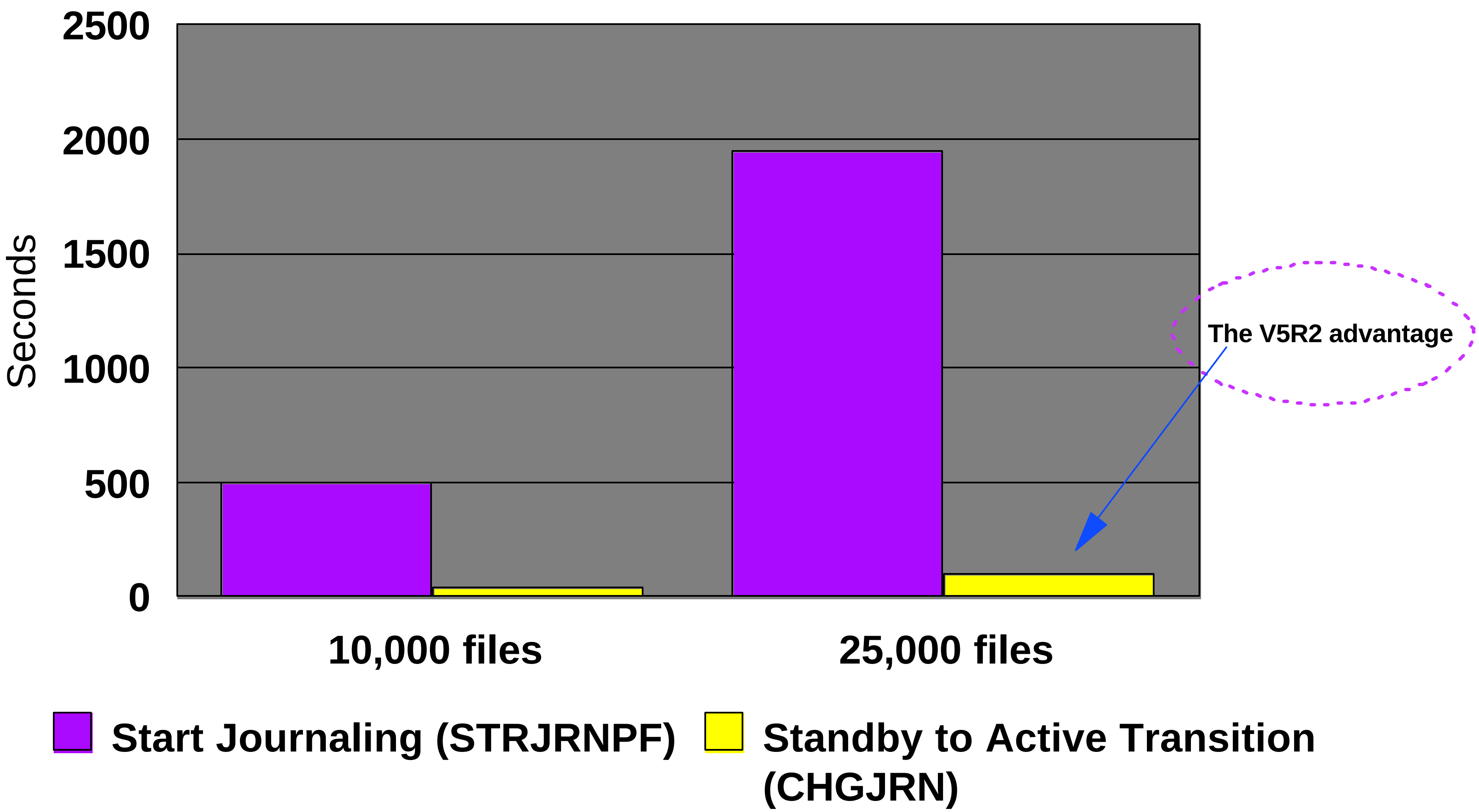
# Potential Benefit:

- Very little performance overhead on the target machine while in standby mode

- Much more rapid failover in the event of a failure
  - OS/400 merely flips **one** flag in the Journal rather than writing to disk thousands of Database file headers

- Mode transition rates during Role-swap as fast as...

# Start Journal vs. Standby to Active Transition



# Summary

Available via OS/400 Option 42:

- ★ Both:
  - Standby Mode
  - JrnCache(*YES)

Order, Install, and enable

V5R2!

# For some hands on experience consider the Journal Lab:

➡ V5R2 Newest Journal and commit features

- 9am Tues: BL09

➡ You can also try some "bonus" Remote Journal exercises !

## *Lab will give you hands-on experience with:*

9:00 AM
Tuesday

## V5R2 Newest Journal and Commit Features:

- New TYPE5 Journal Data
- Journal Standby Mode
- SMAPP Visibility and Tuning
- Savepoints
- Journal Performance Counters and Tracepoints

## Bonus HA Lab exercises:

- Remote Journal
- Simulated hot site recovery

# Selecting Optimal Journal attributes

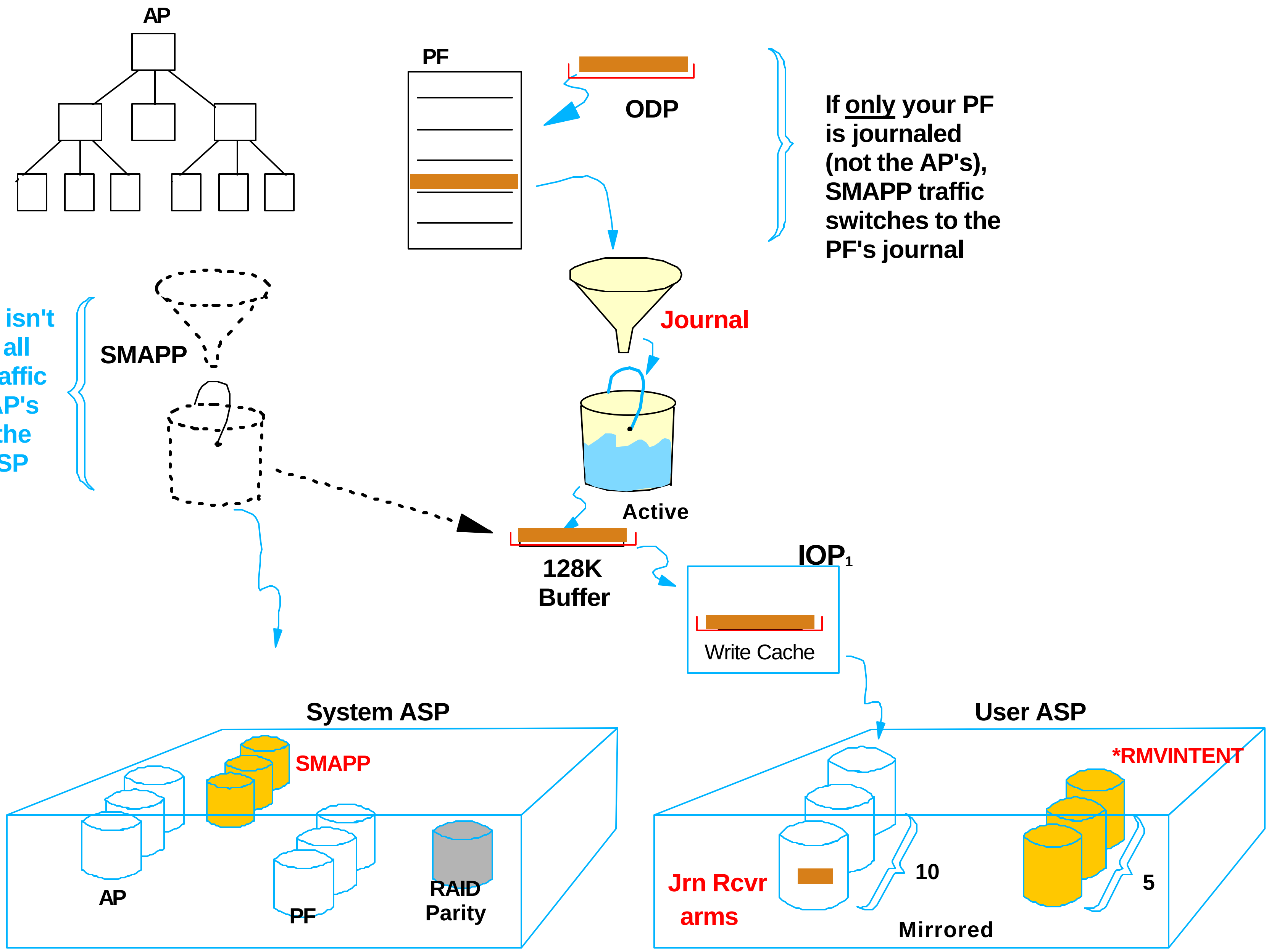
# Selecting an optimal CHGJRN... RCVSIZOPT

## How tall and wide do you want your journal Receivers ?

- ★ *NONE
  - Default, but slowest and most limiting
- ★ *MAXOPT1 (V4R5)
- ★ *MAXOPT2 (V5R1)
  - **Newest and best performing**
  - Especially if you have lots of Journaled objects
  - Especially if your applications create lots of new tables
  - Especially if you do lots of MOVEs or RENAMEs

# SMAPP performance and tuning

# SMAPP  traffic - - lurking the background



If only your PF is journaled (not the AP's), SMAPP traffic switches to the PF's journal

If your PF isn't journaled all SMAPP traffic for your AP's heads to the system ASP

# The Problem:

- SMAPP (System managed access path protection) is both a blessing and a curse
  - **Under-the-covers it's an implicit variety of journaling for Database files & Access Paths**
  - **The blessing:**
    - It shortens abnormal IPL time for the great unwashed
    - They need more of this castor-oil each release
  - **The curse:**
    - There ain't no free lunch!
    - All this background activity costs you something
    - SMAPP is very single-minded, he works diligently to achieve his objective no matter how much pain he causes
    - The pain stems from the fact that SMAPP can be a bully -- if he finds he can't pick on a big guy, he beats up the little guys (small access paths) to make up for it
  - **Standby mode will cause SMAPP to inflict even more pain unless we let you have more say so in who SMAPP picks on**

- The Solution:
  - **Let you see what SMAPP is doing, and throttle his bullying behavior**

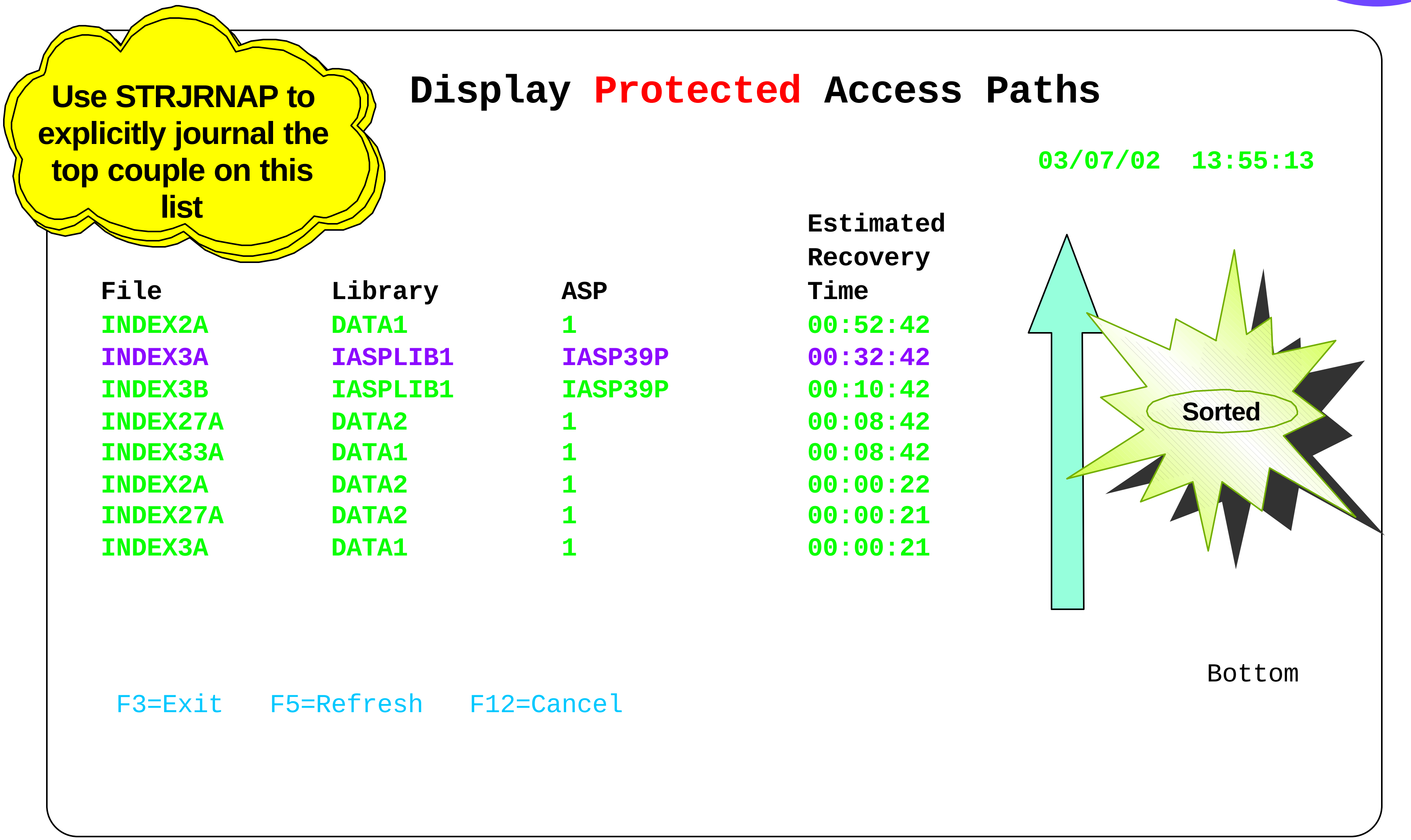
# SMAPP default values for protection

- **Originally default SMAPP protection was 150 minutes**
  - Meant IPL recovery time for Access Paths in the DB was capped at 2.5 Hrs
  - This original default cap was selected when disks and CPU's were much slower than those shipped today and the performance overhead of SMAPP was unknown
- **More recent default SMAPP settings have been drifting steadily downward:**
  - 120 Minutes for V4R5
  - 90 Minutes for V5R1
  - **70** Minutes for V5R2 } **New**

What should this drive you to do ?

# DSPRCYAP followed by F14

V5R2!



Use STRJRNAP to explicitly journal the top couple on this list

```
                 Display Protected Access Paths

                                                      03/07/02  13:55:13

                                          Estimated
                                          Recovery
File           Library        ASP         Time
INDEX2A        DATA1          1           00:52:42
INDEX3A        IASPLIB1       IASP39P     00:32:42
INDEX3B        IASPLIB1       IASP39P     00:10:42
INDEX27A       DATA2          1           00:08:42
INDEX33A       DATA1          1           00:08:42
INDEX2A        DATA2          1           00:00:22
INDEX27A       DATA2          1           00:00:21
INDEX3A        DATA1          1           00:00:21

                                                                Bottom
 F3=Exit    F5=Refresh    F12=Cancel
```

List of Indexes protected by SMAPP

(This screen is new for V5R2)

# For best performance...
# Bid Farewell to FRCACCPTH

# *Identify any uses of this ancient option*

## Not all Access Paths are eligible for SMAPP protection:

## How do you know if you have ineligible indexes?

- Starting in V5R2....
  - The EDTRCYAP/DSPRCYAP screens shows the "total not eligible recovery time"
  - **F13** from the EDTRCYAP/DSPRCYAP screens shows which indexes are **ineligible**
  - It's a snapshot in time

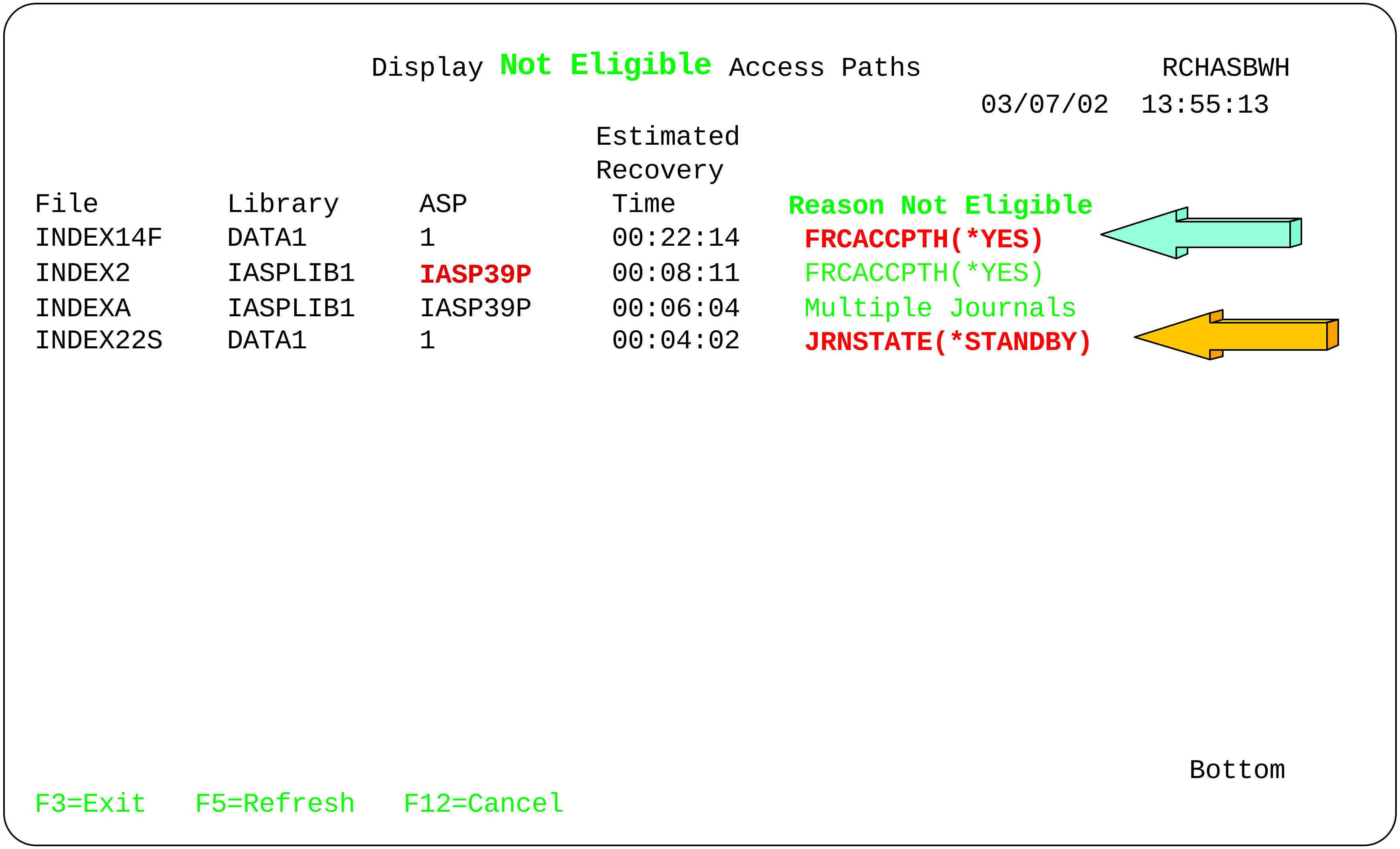
- Only indexes you can do something about are shown
  - Shows reasons for being ineligible
  - Prime offender...

**Replace with: *NO**

**CHGLF FRCACCPTH(*YES)**

# DSPRCYAP followed by F13

V5R2!



```
                     Display Not Eligible Access Paths                RCHASBWH
                                                           03/07/02  13:55:13
                                   Estimated
                                   Recovery
File        Library     ASP         Time       Reason Not Eligible
INDEX14F    DATA1       1           00:22:14    FRCACCPTH(*YES)
INDEX2      IASPLIB1    IASP39P     00:08:11    FRCACCPTH(*YES)
INDEXA      IASPLIB1    IASP39P     00:06:04    Multiple Journals
INDEX22S    DATA1       1           00:04:02    JRNSTATE(*STANDBY)




                                                                        Bottom
F3=Exit   F5=Refresh   F12=Cancel
```

List of Not Eligible Indexes protected

**(This screen is new for V5R2)**

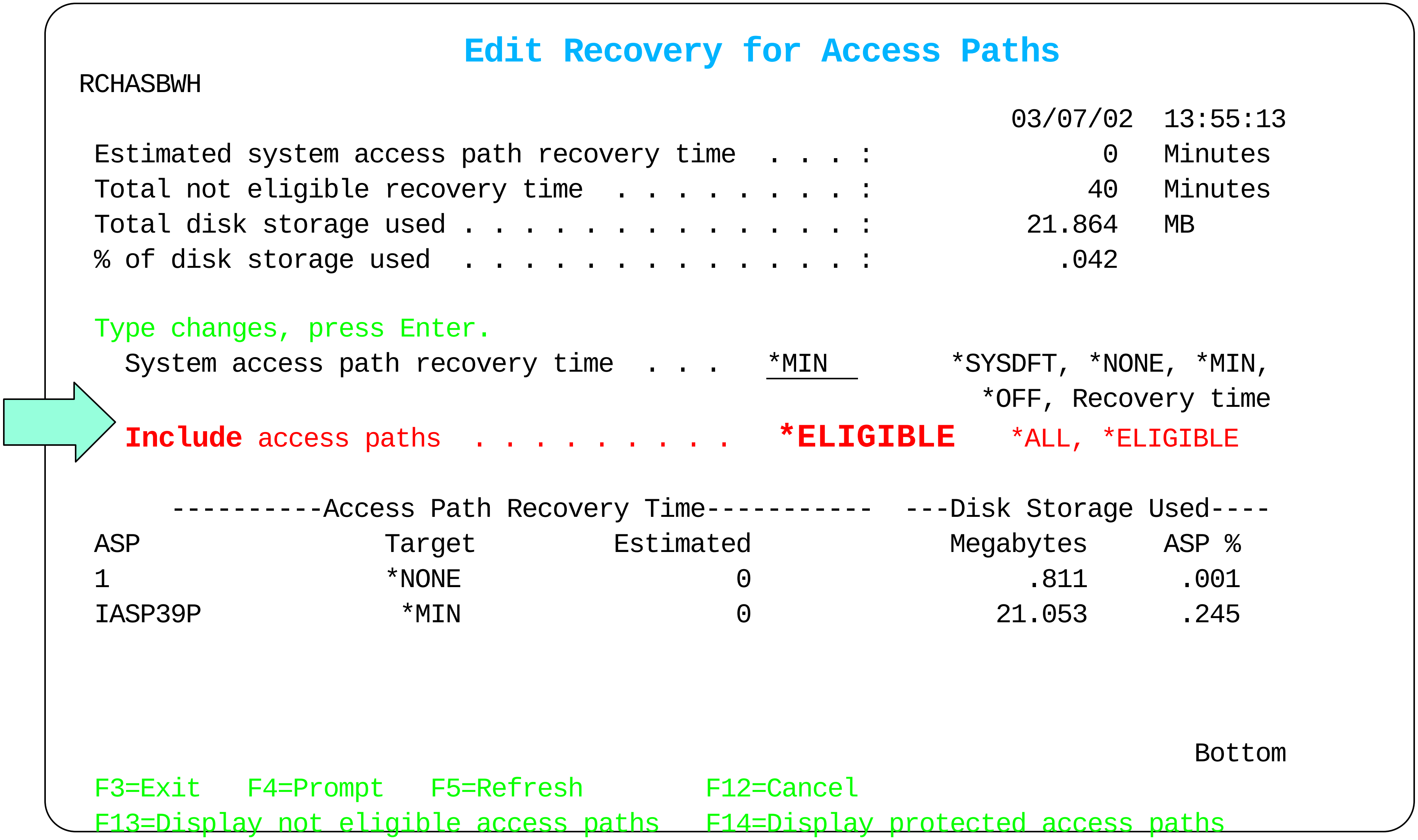
# SMAPP adjustments for Standby mode

- **Access paths defined over physical files which are attached to standby journals will not be protected via SMAPP**
  - These Access paths are considered "**ineligible**" for protection
- **A list of the top 500 ineligible Access paths can be displayed per ASP**
  - Their reason for being ineligible is noted
- **A similar list of the largest "at-risk yet protected" Access paths can also be displayed**
- **For the first time you can advise the machine whether to compensate for the ineligible access paths by picking on "other" access paths (or not)**
  - The best performance typically ensues if you override the default and tell the machine NOT to pick on the "others" to compensate (which might make your IPL longer)

# *Making the wisest SMAPP Choice*

- **What's the wisest choice on a Target Machine whose Journals are employing "Standby" mode ?**
  - Ans:  Instruct the machine to Include ONLY the *ELIGIBLE  Acc Pths
  - That way we don't waste cycles "overprotecting"

# EDTRCYAP



```
                        Edit Recovery for Access Paths
RCHASBWH
                                                             03/07/02  13:55:13
 Estimated system access path recovery time  . . . :                0   Minutes
 Total not eligible recovery time  . . . . . . . . :               40   Minutes
 Total disk storage used . . . . . . . . . . . . . :           21.864   MB
 % of disk storage used  . . . . . . . . . . . . . :             .042

 Type changes, press Enter.
   System access path recovery time  . . .   *MIN         *SYSDFT, *NONE, *MIN,
                                                             *OFF, Recovery time
   Include access paths  . . . . . . . . .   *ELIGIBLE    *ALL, *ELIGIBLE

      ----------Access Path Recovery Time-----------  ---Disk Storage Used----
 ASP                  Target         Estimated              Megabytes     ASP %
 1                    *NONE                  0                   .811      .001
 IASP39P               *MIN                  0                 21.053      .245



                                                                         Bottom
 F3=Exit    F4=Prompt    F5=Refresh          F12=Cancel
 F13=Display not eligible access paths   F14=Display protected access paths
```

Telling us to ignore the Ineligible Indexes

# More insight into Performance:

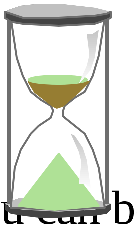
## Better counters and Traces

# Deep dark secrets...

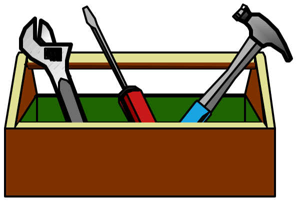
**Lurking in the background are Journal tasks**

**They perform housekeeping chores**

- Starting in V5R2....
  - You can examine:
    - Journal performance counters
    - Journal Trace points

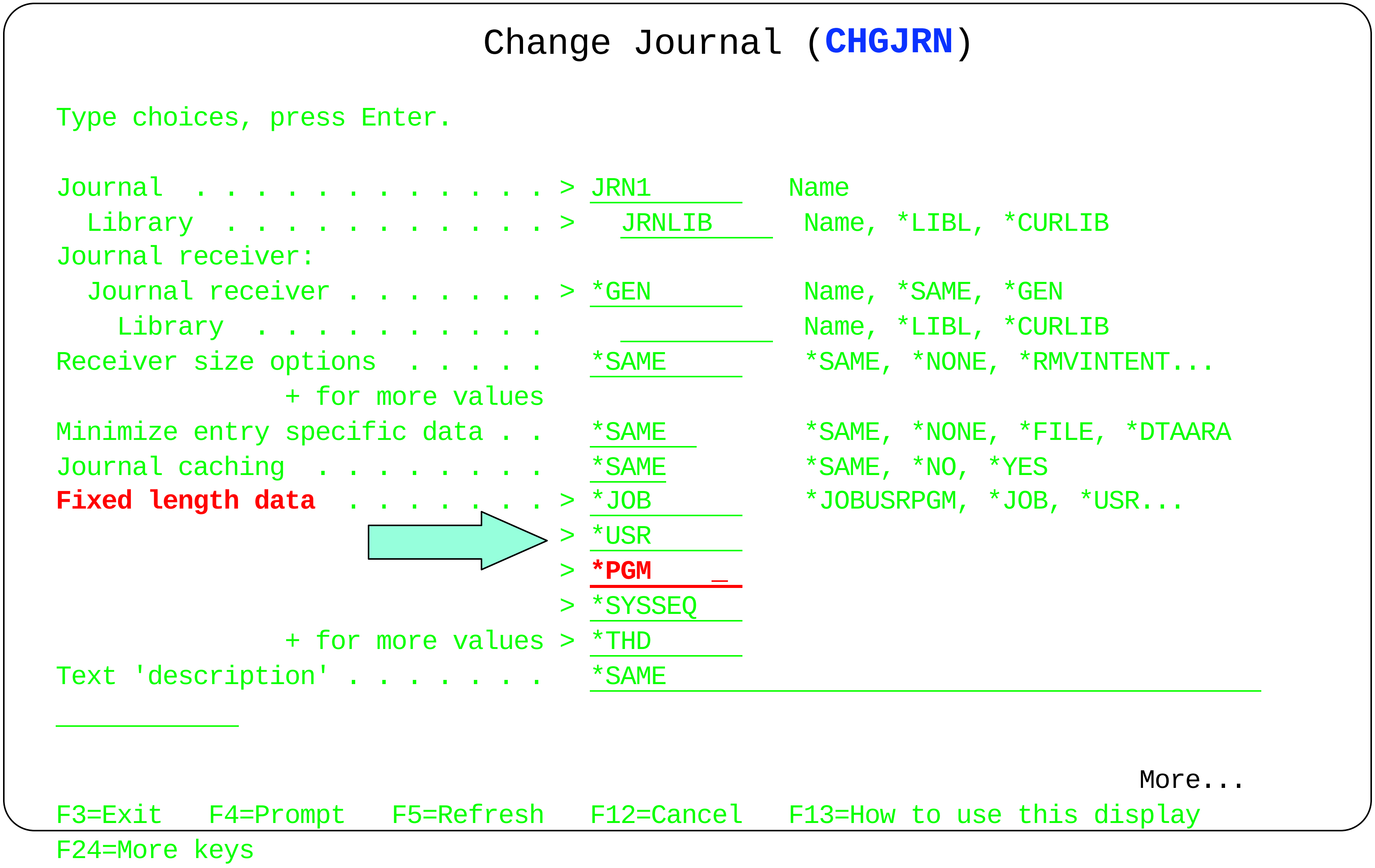


  - Armed with this new information you can better analyze what's really happening
  - Want to see this in action ?
  - Come to the V5R2 Journal lab...



# Customizing the Chaff

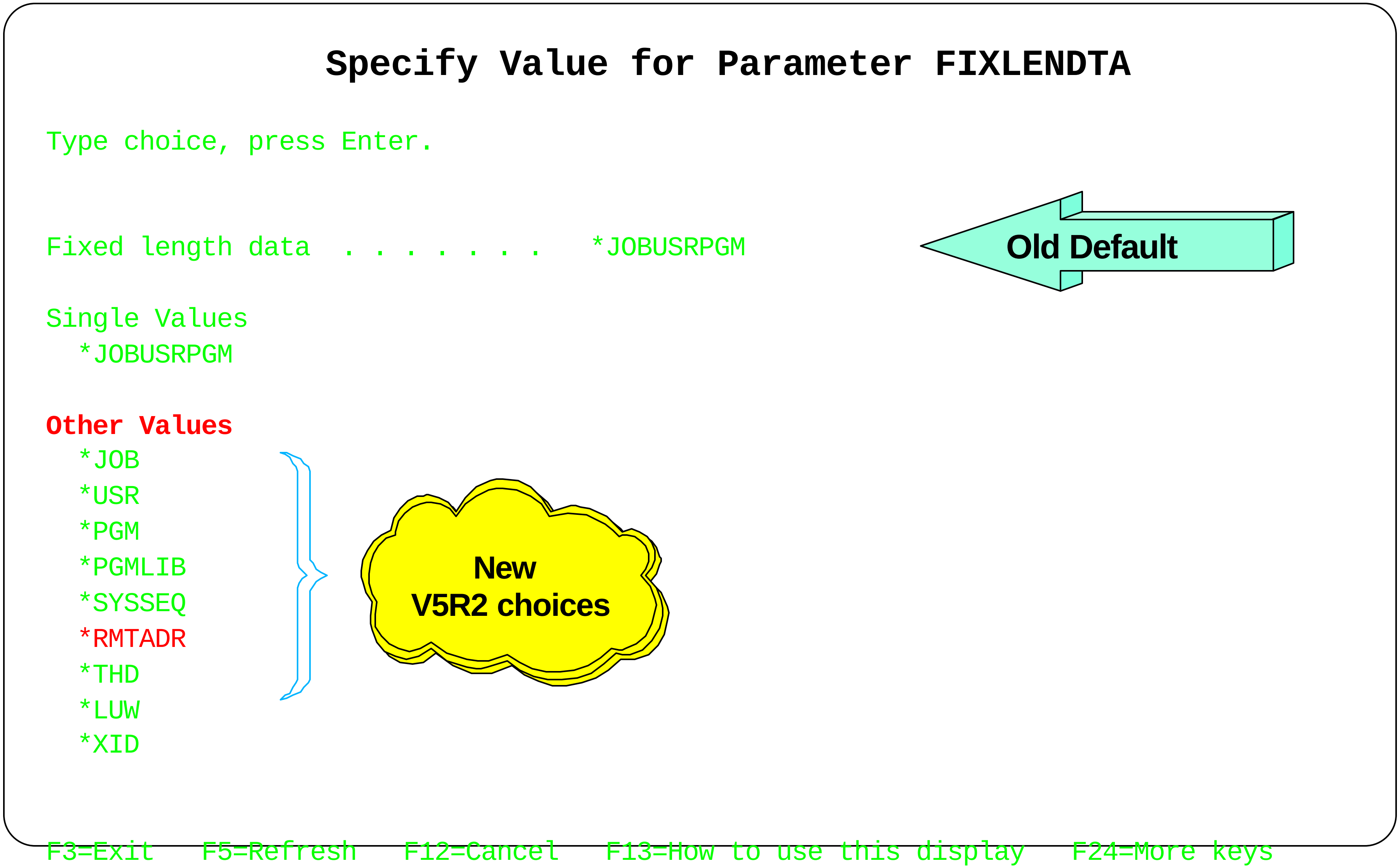
# Customizing what we collect



```
                          Change Journal (CHGJRN)

 Type choices, press Enter.

 Journal  . . . . . . . . . . . . . > JRN1          Name
   Library  . . . . . . . . . . . . >   JRNLIB        Name, *LIBL, *CURLIB
 Journal receiver:
   Journal receiver . . . . . . . . > *GEN            Name, *SAME, *GEN
     Library  . . . . . . . . . . .                   Name, *LIBL, *CURLIB
 Receiver size options  . . . . .    *SAME            *SAME, *NONE, *RMVINTENT...
                + for more values
 Minimize entry specific data . .    *SAME            *SAME, *NONE, *FILE, *DTAARA
 Journal caching  . . . . . . . .    *SAME            *SAME, *NO, *YES
 Fixed length data  . . . . . . . > *JOB              *JOBUSRPGM, *JOB, *USR...
                                  > *USR
                                  > *PGM     _
                                  > *SYSSEQ
                + for more values > *THD
 Text 'description' . . . . . . .    *SAME


                                                                        More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

CHGJRN command with new FIXLENDTA parm

# CHGJRN . . . FIXLENDTA(*RMTADR)



```
              Specify Value for Parameter FIXLENDTA

Type choice, press Enter.


Fixed length data  . . . . . . .   *JOBUSRPGM

Single Values
  *JOBUSRPGM

Other Values
  *JOB
  *USR
  *PGM
  *PGMLIB
  *SYSSEQ
  *RMTADR
  *THD
  *LUW
  *XID

F3=Exit   F5=Refresh   F12=Cancel   F13=How to use this display   F24=More keys
```

Old Default

New
V5R2 choices

## All options displayed for FIXLENDTA Parameter

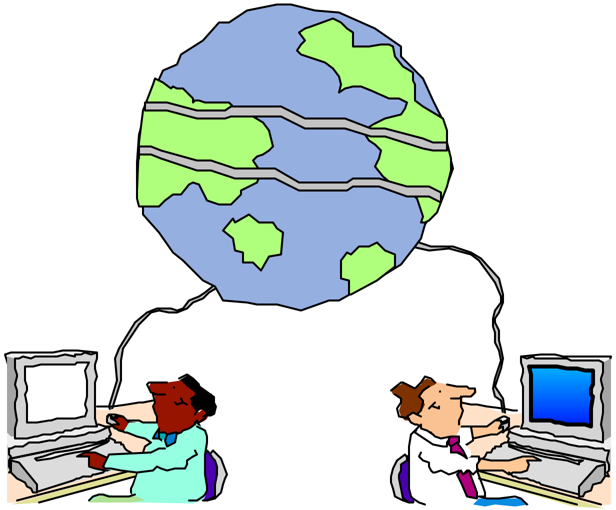
# Summary

Be Selective:

★ Advise us to collect only what you truly need,

no more and no less

# Intrusion Detection
## via Journaling

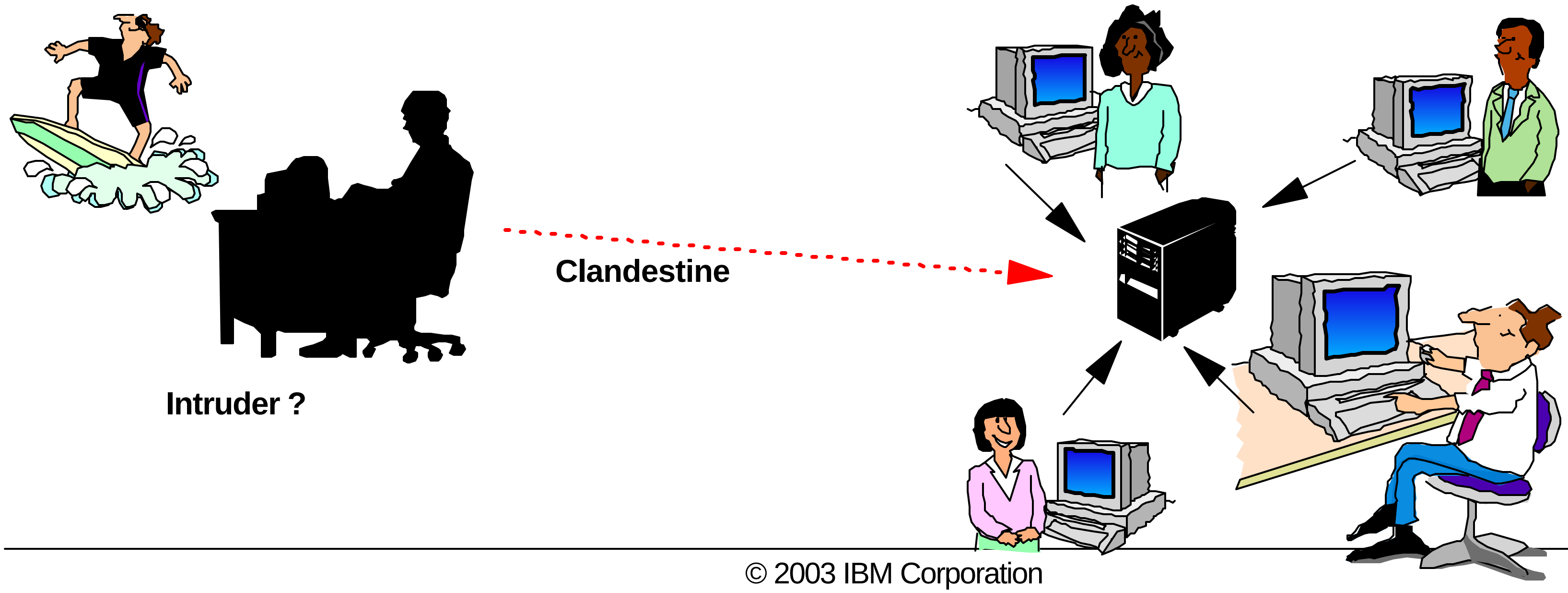


# Intrusion Detection via Journaling for V5R2

## You can instruct the Journal to record IP addresses

V5R2!

- Each DB modification will be tracked to the initiating address
- **This may assist in tracking the origin of changes originating "outside"**



# V5R2 Intrusion Detection Enablement

- New audit points are being added to help with the aspect of knowing when a system is being attacked.

- Storing additional identity information like the IP and Thread ID helps to trace the intruder's activity.

## This function is important because:

A system attached to a network can be attacked without the need for physical access or direct attached terminals.  System administrators need the ability to learn which IP address are initiating work or activity on the system.

(Knowing you are being attacked)

# V5R2: Collecting Additional Identity Information

## New data stored for each Audit record:

**(and optionally for other journals)**

- ❑ **Remote Address:**
  - ★ **Identifies the IP address**
  - ★ **Identifies the Port**
  - ★ **Identifies the Address Family**
- ❑ Thread ID - Identifies which thread within the process
- ❑ Program Library - Identifies the library in which the program resides
- ❑ Program Library ASP - Identifies the ASP, (both device name and number) housing the program library
- ❑ LUW ID - Identifies the Logical Unit of Work for the transaction
- ❑ XID - Identifies the specific transaction

# Better Intrusion Detection

- **Prior to V5R2 when system auditing is used, the information in the audit record can identify what happened and when, but when in a TCP/IP environment, it can be impossible to determine who caused it to happen. This is because audit records contain the job name, the job user profile, and the current user profile, but in cases where rejected signons are occurring, for example, there is no information about where it may be coming from.**

New

  - Some applications are developed such that all work is done under **one** iSeries **user profile**, and therefore it can be impossible to determine the client in the network the job is actually doing work for.

V5R2!

- **To provide additional information in security audit records, the thread id and the remote address will be included in the audit records**

# What does the DSPJRN information look like ?

( Note that **CHGJRN ...FIXLENDTA(*RMTADR)** needs to be specified to collect the data below)

| SEQUENCE | CODE | TYPE | ADDRESS | REMOTE | REMOTE IP |
|---|---|---|---|---|---|
| | | | FAMILY | PORT | ADDRESS |
| 00000000000000000011 | J | PR | 4 | 4,052 | 5.5.5.123 |
| 00000000000000000012 | D | CT | 4 | 4,051 | 5.5.5.123 |
| 00000000000000000013 | D | JF | 4 | 4,053 | 5.5.5.125 |
| 00000000000000000014 | F | JM | 4 | 4,053 | 5.5.5.125 |
| 00000000000000000015 | F | MC | 4 | 4,051 | 5.5.5.123 |
| 00000000000000000017 | R | PT | 4 | 4,051 | 5.5.5.123 |

# Summary

If you think you *might* have intruders...

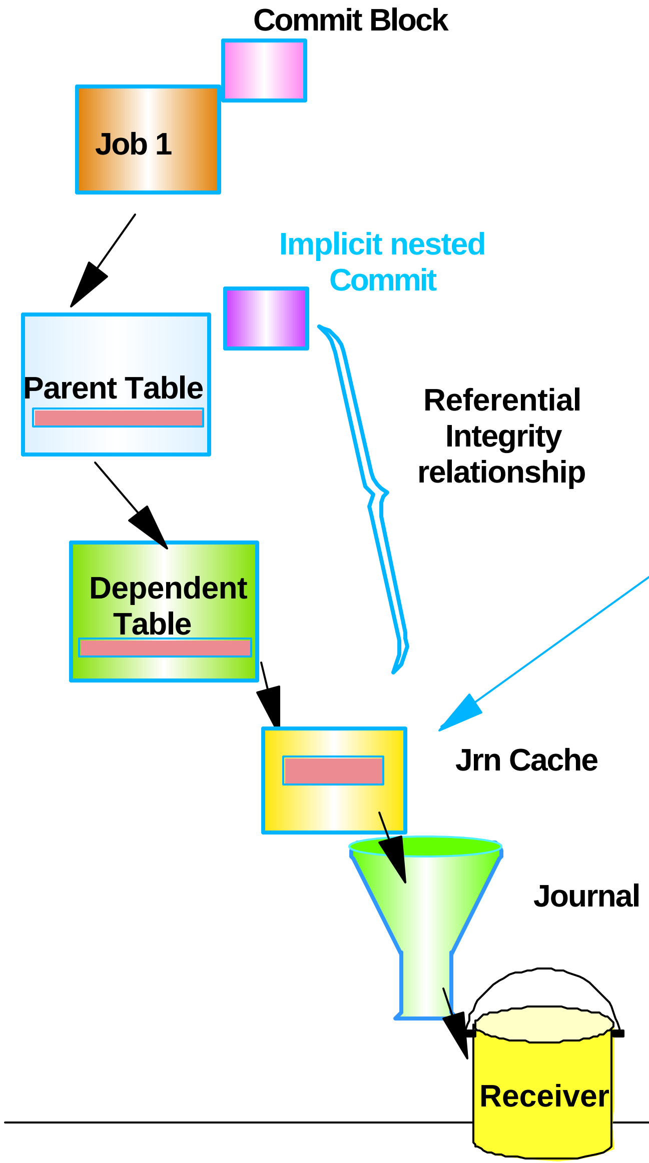
Perhaps you'll want to consider use of the new Journal featues to spot them !

# Soft Commit

# The Problem:

- The machine inserts subtle/hidden implicit commit cycles as its means of enforcing **Referential Integrity** constraints in the Database.
  - These *implicit* commit cycles hurt performance - Especially if you're caching !
  - **Standby mode** does not allow any commit cycles to be present
    - (not even implicit commit cycles)
    - Hence, any customer or application which employs **RI** can't use Standby mode
  - The alternative to Standby mode is **Caching mode**
    - Unfortunately, the performance benefits of caching is thwarted by any operation which flushes the caching buffer
    - All commit cycles (even implicit ones) flush the caching buffer
- The Solution:
  - **Soft commit** **(So novel, we're seeking a patent)**

# Soft implicit commit



Refrains from flushing this buffer

Primarily relevant for Cascade_Delete RI rule

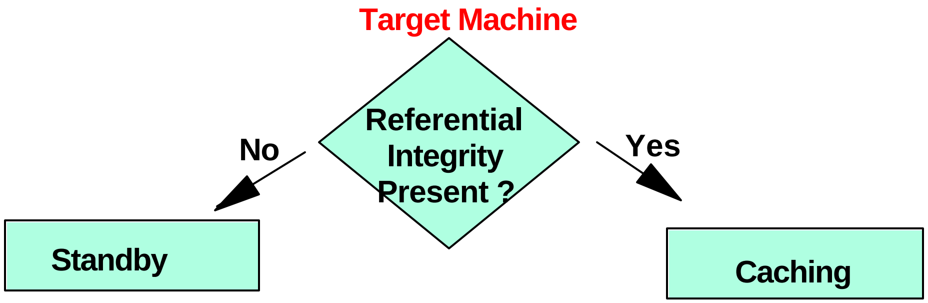
# Soft Commit

- Intended to help HA BP's achieve good performance on target machine when DB referential_Integrity rules insert an *implicit* commit cycle.

- The presence of soft commit assures that System initiated commit cycles will cease prematurely flushing the journal main memory buffer.

  - This makes use of Caching mode a good performer even when RI is present

**Net:**



- Use Caching mode (which capitalizes on "Soft" Commit) in place of Standby mode if RI is present

# Summary

Faster Referential Integrity processing arrives in V5R2 by enabling the Journal Caching option

(Soft Commit kicks in !)

V5R2!

# Nested transactions

# Nested Save points for transactions

## Partial nested sub-transactions can be rolled back

- Example:
  - Travel reservation w/ nested Hotel, car, airline

- Journal Entries:
  - SC - Start outer transaction
  - PT - New record for tentative Hotel in Rochester
  - <----- Save Pt #1 here (keep the Hotel Rm even if flight if full)
  - PT - New record for tentative Car rental from Rochester airport
  - --- attempt to lock in flight to RST shows plane is full
  - --- Application decides to fall back to Save_Pt #1
  - DR - Car reservation in RST is canceled (rolled back) but not Hotel
  - PT for new tentative Car rental in MSP
  - PT Flight to MSP (I'll drive down)
  - CM - Commit whole transaction (notice not committing RST car)

Nested

# SQL Save Point Syntax

V5R2!

**This example contains nested savepoints with the inner cycle being released, but the outer cycle being rolledback.**

**Only "Marriott" will survive after this rollback in this case.**

```
Insert into My_Lib/Hotel values ('Marriott');
            savepoint sp1 on rollback retain cursors;
Insert into My_Lib/Rental_Car values (‘Hertz’);
           savepoint sp2 on rollback retain cursors;
Insert into My_Lib/Airline values (‘Northwest’);
           release savepoint sp2;
Insert into My_Lib/Credit_Card values (‘Visa’);
           rollback to savepoint sp1;
commit;
```

Nested

# The nested rollback...

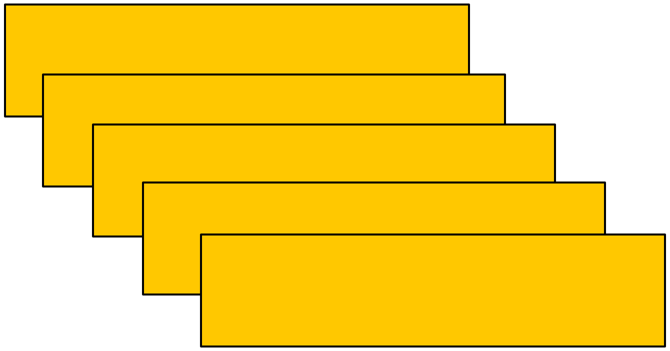
This example contains nested savepoints with the inner cycle being released, but the outer cycle being rolledback.
Only "Marriott" will survive after this rollback in this case.

```
Insert into My_Lib/Hotel values ('Marriott');
            savepoint sp1 on rollback retain cursors;
Insert into My_Lib/Rental_Car values (‘Hertz’);
           savepoint sp2 on rollback retain cursors;
Insert into My_Lib/Airline values (‘Northwest’);
           release savepoint sp2;
Insert into My_Lib/Credit_Card values (‘Visa’);
           rollback to savepoint sp1;
commit;
```

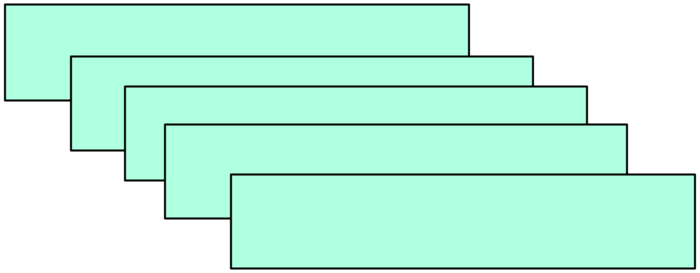
# More progress screens for long-running decommit operations

# If you've got run-away batch job with a large commit cycle..

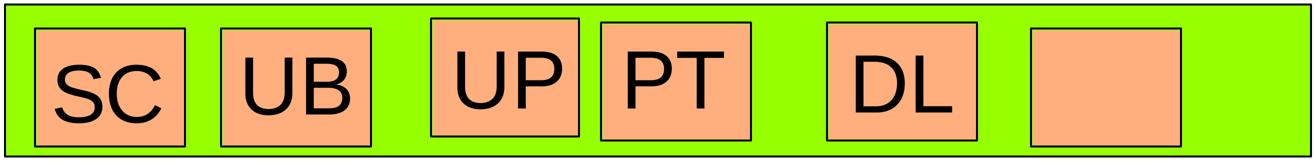
Modified Rows



Locks held



Journal Receiver



SC UB UP PT DL

# The Problem:

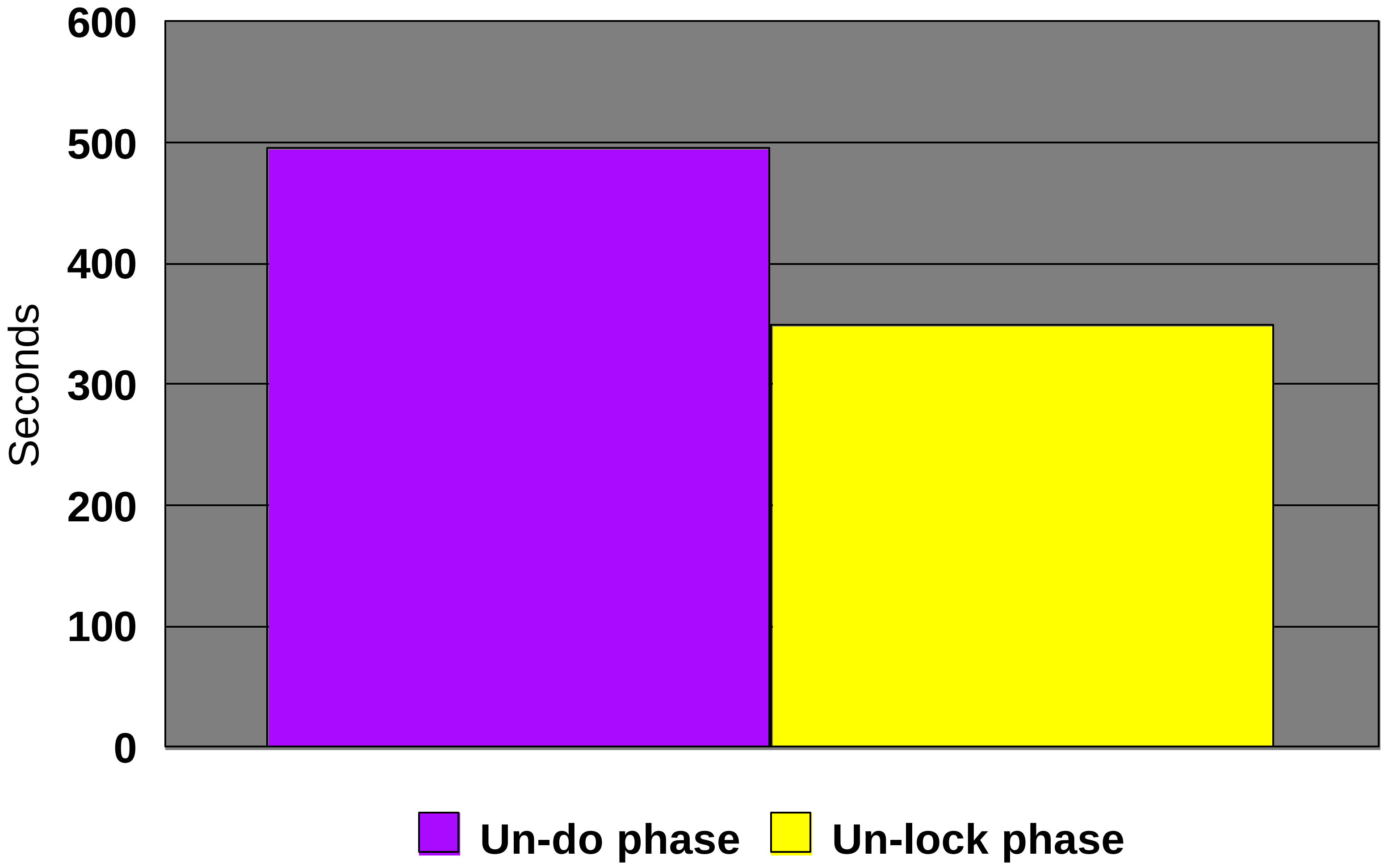
- Runaway jobs and sleepy operators often create enormous long-running jobs which tie up all the database and journal resources
  - If these jobs are running under commitment control as a single huge commit cycle, they can take your machine hostage
  - Customers often try to kill such jobs only to find that the ensuing decommit operation seems to take a month of Sundays
  - They want to know what's going on and how long it will be until they are whole again

- The Solution:

V5R2!

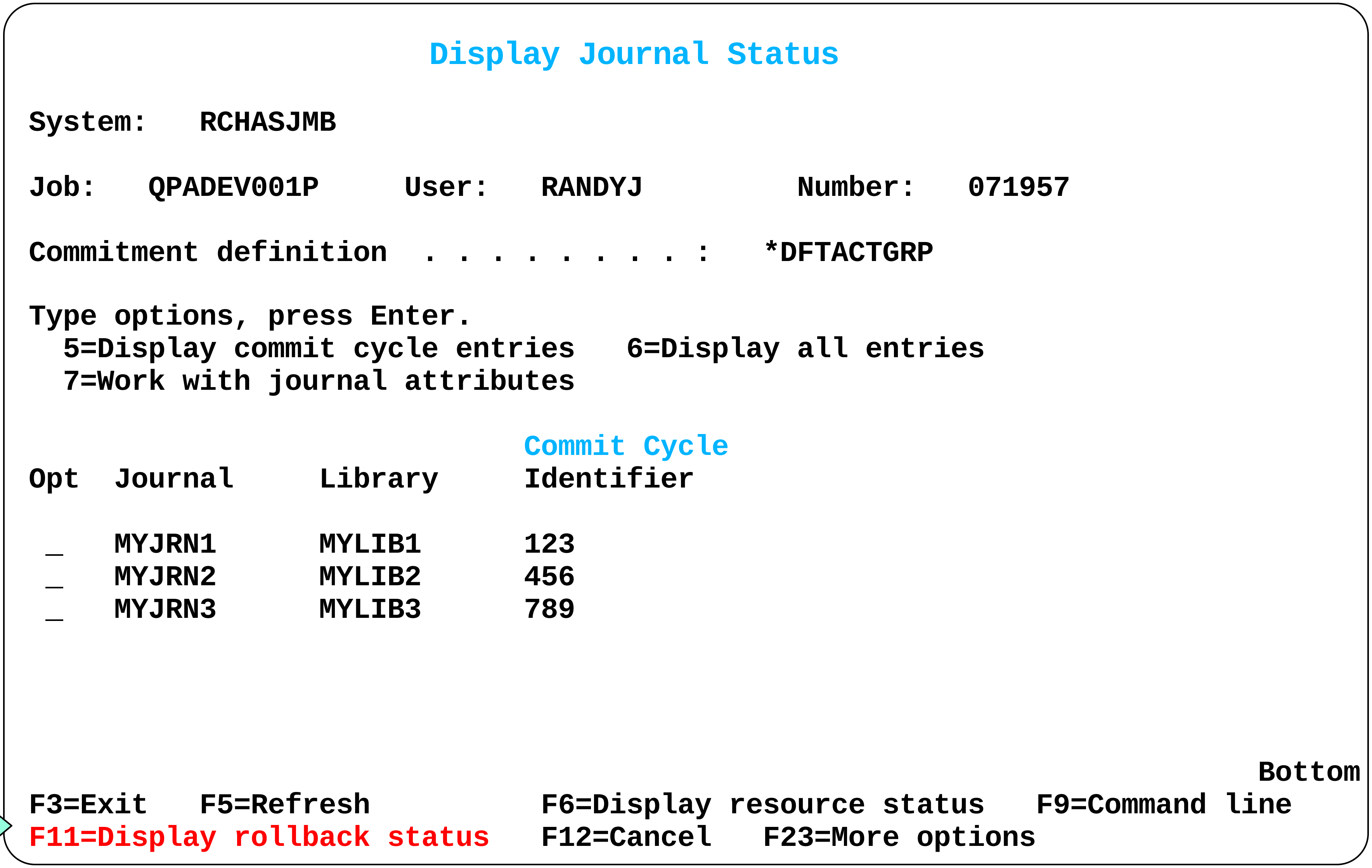
  - **New screen** to show progress during a long-running decommit operation
  - We'll now show both steps:
    - The actual **rollback** of database operations (so-called "un-dos")
    - The subsequent (and time-consuming) **unlock** of millions of records

## Rollback involves two phases:



# The Commit Status Screens:

## WRKCMTDFN

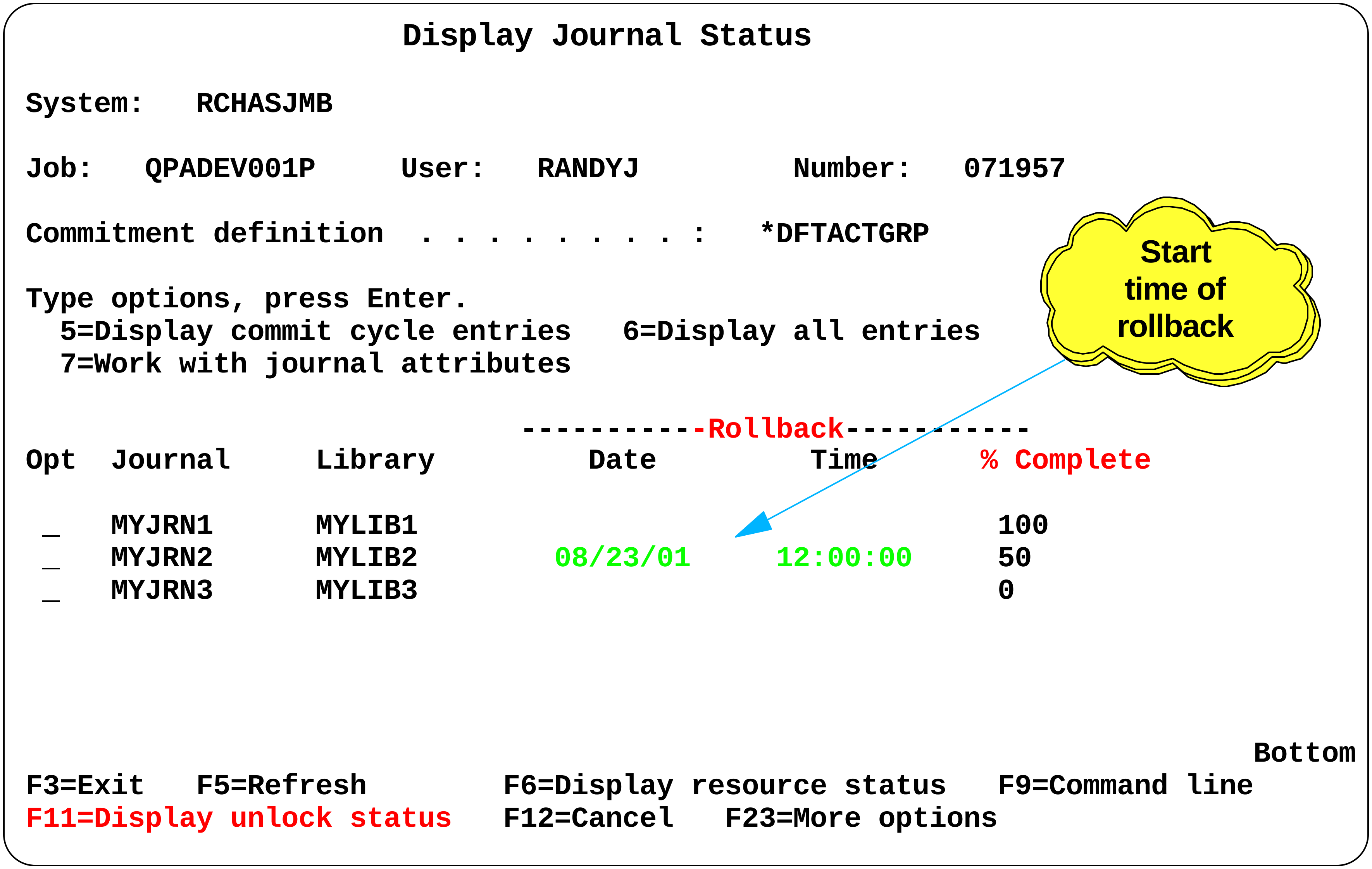


```
                     Display Journal Status

System:   RCHASJMB

Job:   QPADEV001P     User:   RANDYJ         Number:   071957

Commitment definition  . . . . . . . . :   *DFTACTGRP

Type options, press Enter.
  5=Display commit cycle entries   6=Display all entries
  7=Work with journal attributes

                           Commit Cycle
Opt  Journal     Library     Identifier

 _   MYJRN1      MYLIB1      123
 _   MYJRN2      MYLIB2      456
 _   MYJRN3      MYLIB3      789




                                                                    Bottom
F3=Exit   F5=Refresh         F6=Display resource status   F9=Command line
F11=Display rollback status  F12=Cancel   F23=More options
```

# The Screens: (The rollback phase)

V5R2!

F11 gives:

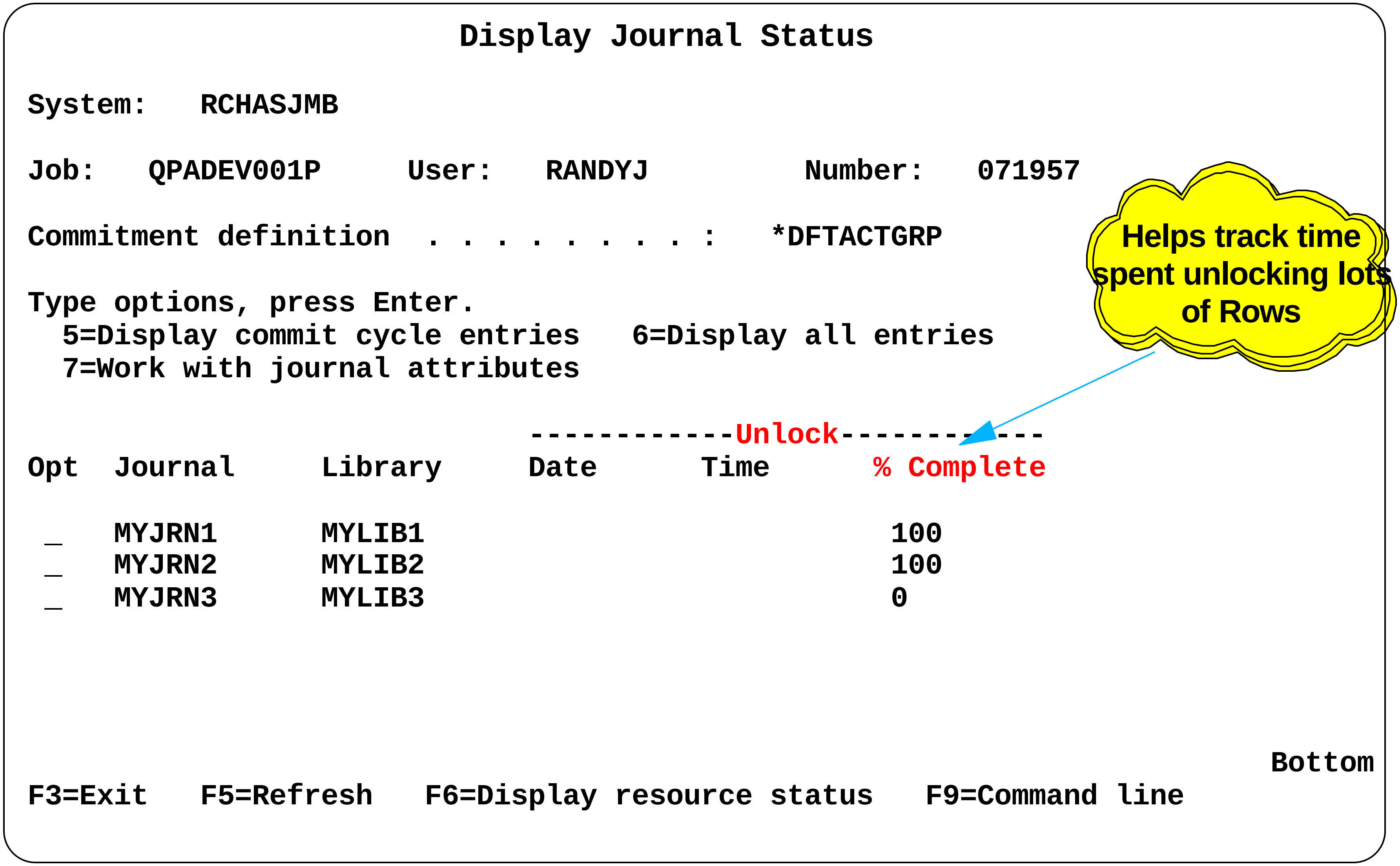


```
                         Display Journal Status

System:   RCHASJMB

Job:   QPADEV001P    User:   RANDYJ        Number:   071957

Commitment definition  . . . . . . . . . :   *DFTACTGRP

Type options, press Enter.
  5=Display commit cycle entries   6=Display all entries
  7=Work with journal attributes

                                      -----------Rollback-----------
Opt  Journal     Library          Date          Time       % Complete

 _   MYJRN1      MYLIB1                                      100
 _   MYJRN2      MYLIB2         08/23/01      12:00:00       50
 _   MYJRN3      MYLIB3                                      0




                                                                    Bottom
F3=Exit   F5=Refresh        F6=Display resource status   F9=Command line
F11=Display unlock status   F12=Cancel   F23=More options
```

Start time of rollback

# The Screens: (The unlock phase)

F11 again gives:



```
                         Display Journal Status

System:   RCHASJMB

Job:   QPADEV001P     User:   RANDYJ         Number:   071957

Commitment definition  . . . . . . . . :   *DFTACTGRP

Type options, press Enter.
  5=Display commit cycle entries   6=Display all entries
  7=Work with journal attributes

                             ------------Unlock------------
Opt  Journal     Library     Date      Time      % Complete

 _   MYJRN1      MYLIB1                           100
 _   MYJRN2      MYLIB2                           100
 _   MYJRN3      MYLIB3                           0



                                                                        Bottom
F3=Exit   F5=Refresh   F6=Display resource status   F9=Command line
```

Helps track time spent unlocking lots of Rows

## *Journal Lab*

# Come try out these new features for yourself !

- **9am Tues:   BL09**

➡ You can also try some "bonus" Remote Journal exercises !

# Trademarks and Disclaimers

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400
AS/400e
eServer

IBM
IBM (logo)
iSeries
OS/400

Other company, product or service names may be trademarks or service marks of others.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance 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 or performance improvements equivalent to the ratios stated here.