



RACF Performance Tuning

SHARE - 16810 - March 2015



Robert S. Hansel Lead RACF Consultant R.Hansel@rshconsulting.com 617-969-9050

Robert S. Hansel



Robert S. Hansel is Lead RACF Specialist and founder of RSH Consulting, Inc., an IT security professional services firm he established in 1992 and dedicated to helping clients strengthen their IBM z/OS mainframe access controls by fully exploiting all the capabilities and latest innovations in RACF. He has worked with IBM mainframes since 1976 and in information systems security since 1981. Mr. Hansel began working with RACF in 1986 and has been a RACF administrator, manager, auditor, instructor, developer, and consultant. He has reviewed, implemented, and enhanced RACF controls for major insurance firms, financial institutions, utilities, payment card processors, universities, hospitals, and international retailers. Mr. Hansel is especially skilled at redesigning and refining large-scale implementations of RACF using role-based access control concepts. He has also created elaborate automated tools to assist clients with RACF administration, database merging, identity management, and quality assurance.

Contact and background information:

- 617-969-8211
- R.Hansel@rshconsulting.com
- www.linkedin.com/in/roberthansel
- www.rshconsulting.com

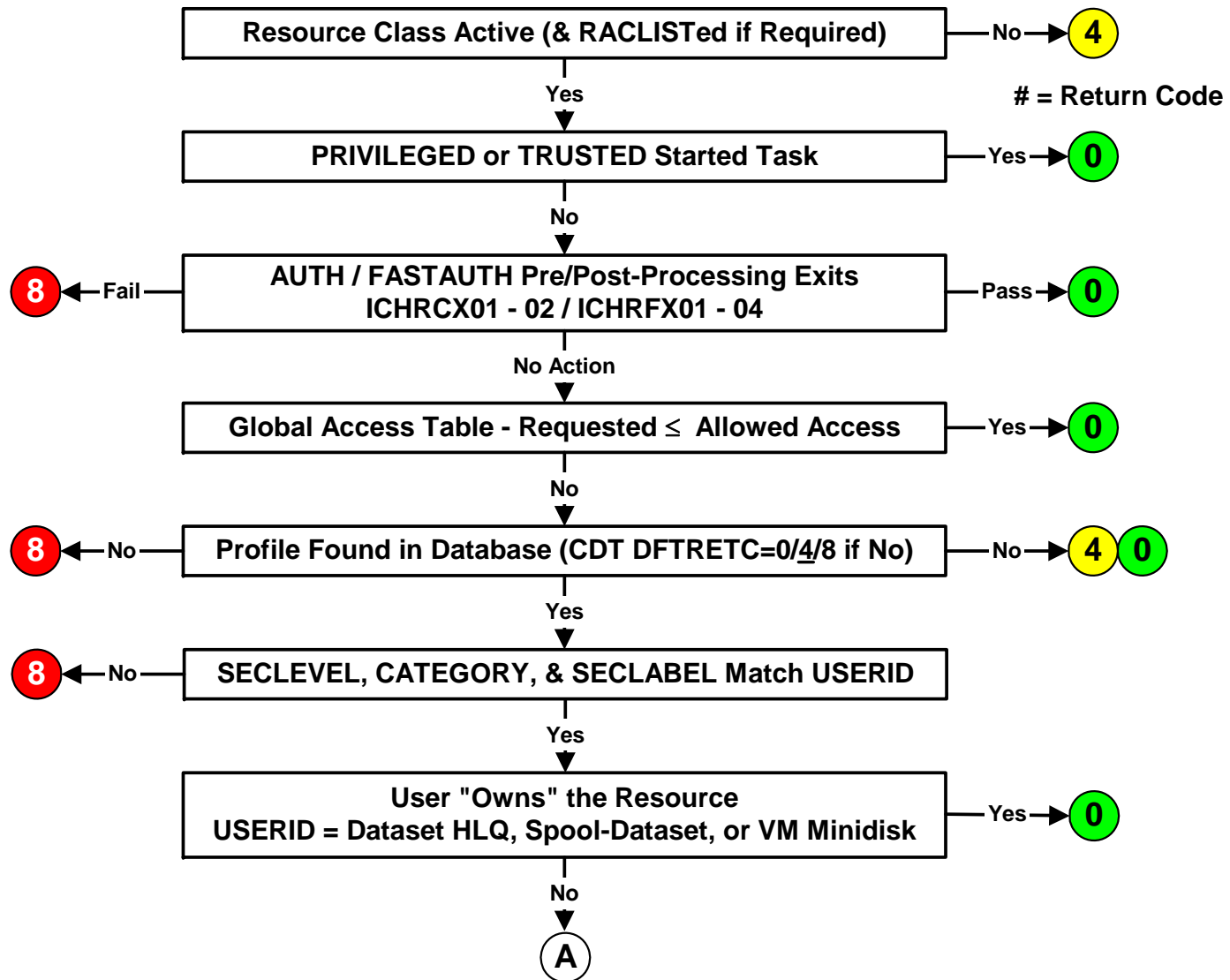
Performance Objectives



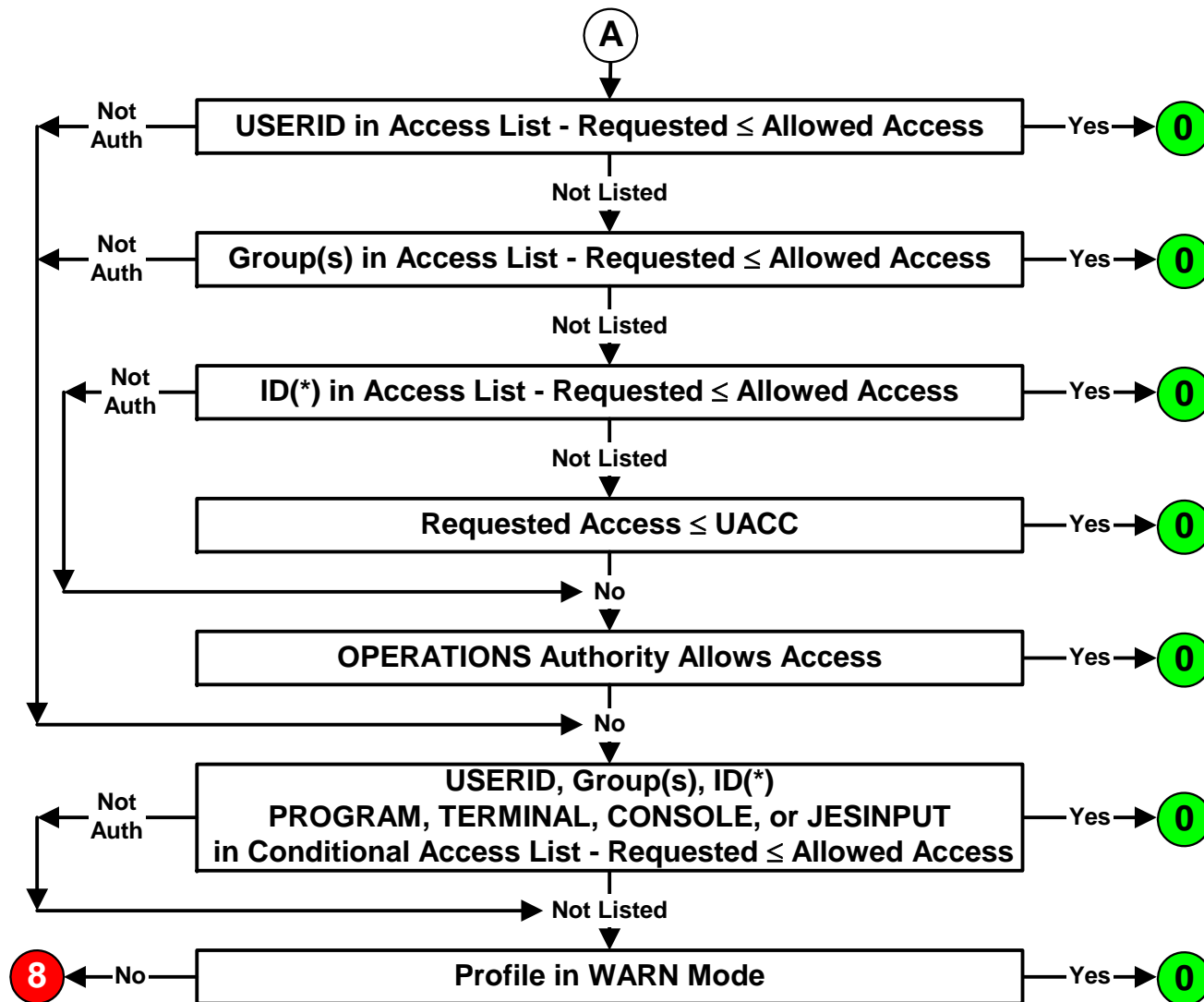
- Optimize Access Authorizations
- Expedite the Logon Process
- Minimize I/O Operations

RACF and z/OS are Trademarks of the International Business Machines Corporation

RACF Authorization Decision Logic



RACF Authorization Decision Logic



RACF Authorization Decision Logic



- Deactivate unused classes (be mindful of POSITs when deactivating)
 - Resource classes, including SECDATA & SECLABEL classes
 - Global Access Table classes
- Make access list processing efficient
 - Minimize the number of entries in access lists
 - ❖ Grant end-user access via groups instead of USERIDs
 - ❖ Remove obsolete residual entries - run IRRRID00
 - ❖ Remove redundant entries (e.g., access allowed equals UACC)
 - Minimize the number of group connects per user
- Reduce reliance on OPERATIONS authority by implementing Storage Administration authorities
- Write efficient exit code
- Implement the Global Access Table

Global Access Table



- Performance enhancement tool
 - Grants immediate access to a resource without referring to its profile and without logging
 - Used to grant access to common shared resources
- GLOBAL Class
 - Profile - Class name [RDEF GLOBAL DATASET]
 - Members - resource/access [ADDMEM('CTLG.USER'/UPDATE)]
 - ❖ Resource
 - Discrete or Generic - follows generic profile rules for General Resources
 - Need not match profile(s) protecting the resource(s)
 - For datasets, if not enclosed in quotes, appends user's USERID as the first qualifier
 - ❖ Access-levels - ALTER | CONTROL | UPDATE | READ | NONE (not EXECUTE)
- Special Variables - Used in resource names
 - &RACUID Substitute with requesting user's USERID
 - &RACGPID Substitute with requesting user's current connect group

Global Access Table



■ Sample entries

DATASET	&RACUID.*.**	ALTER	
DATASET	&RACGPID.*.**	UPDATE	(avoid - unintended access)
DATASET	CATALOG.MASTER	READ	
DATASET	CATALOG.USER	UPDATE	
DATASET	ISPF.LIBRARY	READ	
DATASET	SDSF.LIBRARY	READ	
DATASET	SYS1.BROADCAST	READ	
DATASET	SYS1.HELP	READ	
DATASET	SYS1.MACLIB	READ	
DATASET	SYS1.RACF	NONE	(precludes access)
DATASET	SYS%.***	READ	(avoid - too broad)
DATASET	*.PUBLIC.**	READ	(optionally allow TSO users to share data)
DATASET	*.**.#SMSTEST	ALTER	(optional catalog/SMS testing)
FACILITY	ERBDSB.*	READ	
FACILITY	IEC.TAPERING	READ	(probably obsolete)
FACILITY	STGADMIN.ARC.ENDUSER.**	READ	
JESJOBS	SUBMIT.*.&RACUID*.&RACUID	READ	
JESJOBS	CANCEL.*.&RACUID.*	ALTER	(not needed - post RTOKEN check)
JESSPOOL	*.&RACUID.**	ALTER	
JESSPOOL	*.*.\$JESNEWS.**	READ	
MQQUEUE	MQS*.ISF.USER.&RACUID.**	ALTER	(probable SDSF manual error)
OPERCMD5	MVS.CANCEL.TSU.&RACUID	UPDATE	
OPERCMD5	MVS.DISPLAY.*	READ	
OPERCMD5	MVS.MCSOPER.&RACUID	READ	
SDSF	ISFCMD.DSP. <i>option</i> .*	READ	(<i>option</i> : ACTIVE, HELD, OUTPUT)
TSOAUTH	JCL	READ	
TSOAUTH	RECOVER	READ	

Global Access Table



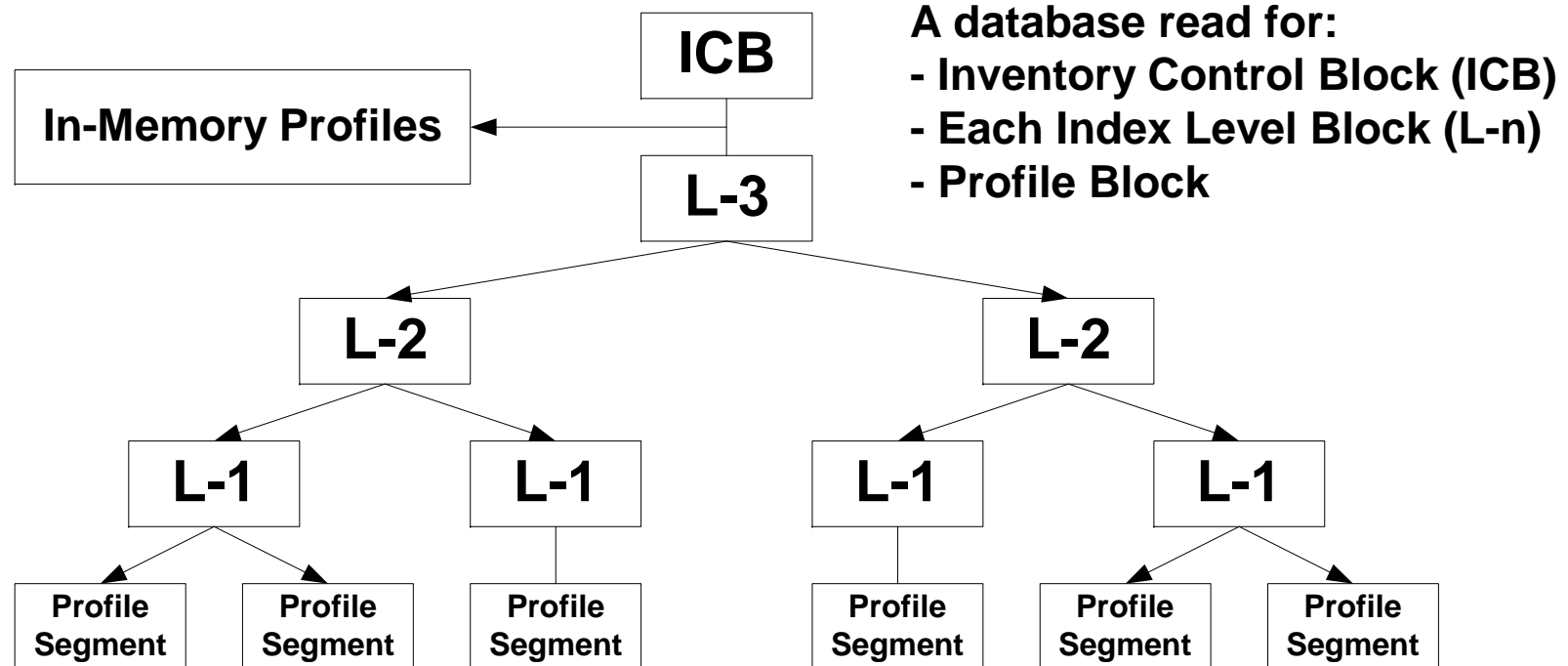
- Activated and managed via SETROPTS
 - SETROPTS GLOBAL(*class*) | NOGLOBAL(*class*) [REFRESH]
 - Must be refreshed if updated

- Can be used for most resource classes except ...
 - Not checked in RACROUTE REQUEST=FASTAUTH processing
 - Not checked in RACROUTE REQUEST=VERIFY processing for APPL, TERMINAL, JESINPUT, CONSOLE, APPCPORT, and SERVAUTH resources

- Keep list of entries short and efficient to minimize search

- Drawbacks
 - Precludes logging (except SETR AUDIT(*class*) resource defines)
 - Undermines protection if allows more access than profile UACCs

RACF Profile Retrieval



- Data is written and retrieved in 4K blocks
- Individual profiles and profile segments can be greater than 4K in size and span multiple contiguous blocks, each of which requires I/O to fetch - keep profiles as small as possible

Resident Data Blocks

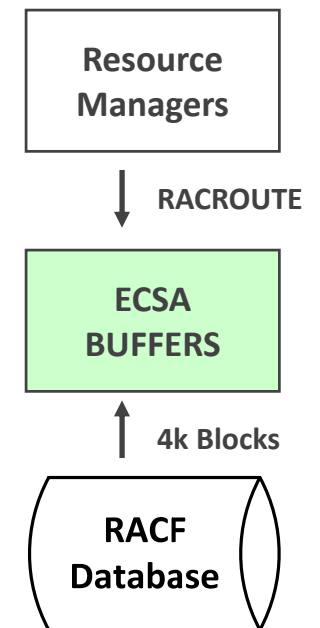


- RACF maintains buffers in ECSA to hold copies of most recently used blocks (index, BAM, and profiles)
- Frequently used blocks tend to stay in these buffers
- Desired number of resident blocks is specified in the Database Name Table - ICHRDSNT

AL1(1)	Number of databases
CL44'RACF.PRIMARY'	Primary DB name
CL44'RACF.BACKUP'	Backup DB name
AL1(100)	# of Resident Data Blocks
XL1'xx'	Flags

- Default/minimum number of blocks

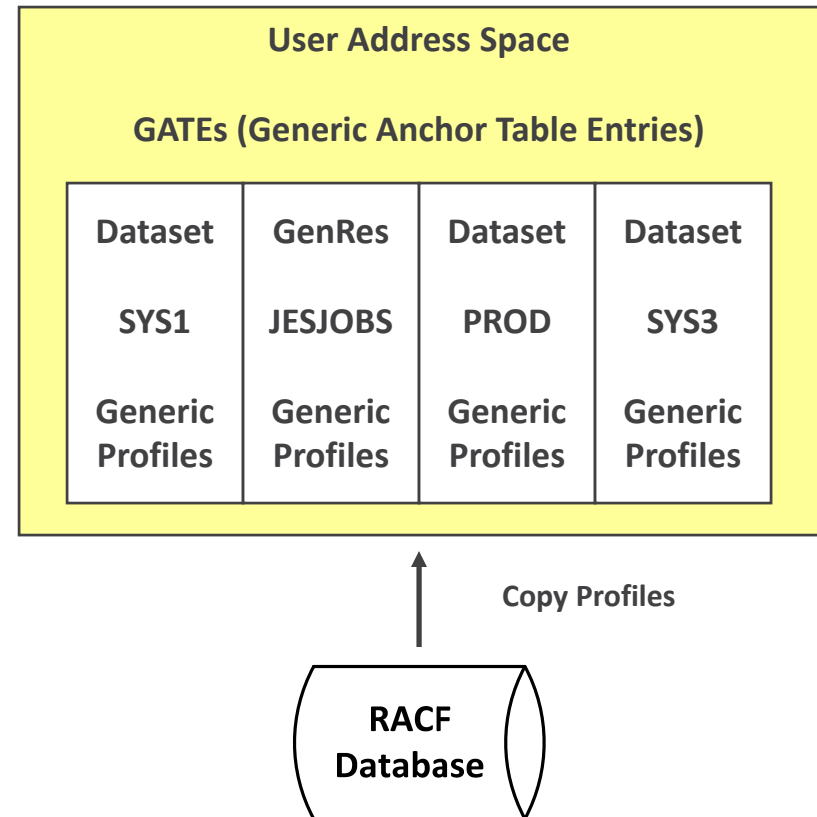
10 / 0	Non-RACF-Sysplex (<u>none</u> for backup database)
50 / 50	RACF-Sysplex (+ additional 20% for backup database)
- Maximum number - 255 (recommended)



Generic Profiles Stored In Memory



- Sets of generic profiles are cached in each individual user's address space memory
- Each set is comprised of generic profiles for either:
 - Dataset HLQ
 - General Resource class
- Upon first access to a resource class or HLQ, a list of all the associated generic profiles is retrieved and loaded into memory
- Individual generic profiles are retrieved as needed for authorization checking and retained in memory thereafter
- Profiles in memory are used for authorization checking - not those in the RACF database



Generic Profiles Stored In Memory



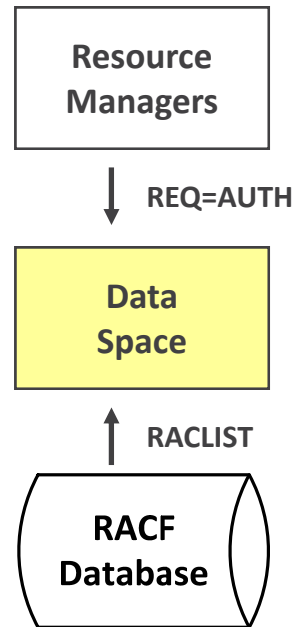
- Once all sets of generic profiles are filled, when the next new resource class or HLQ is accessed, the set with the least recently used profiles is dropped and replaced with the new one
 - Users accessing many different HLQs and/or general resources could experience thrashing (i.e. constant replacement) among the sets
- Dataset HLQs or general resources classes with many generic profiles take more I/O and CPU time to retrieve and load
- RACF can optionally keep up to 99 sets of profiles
 - Changed with the RACF operator command SET GENERICANCHOR(*option*)
 - Option can be configured for SYSTEM or JOBNAME(*jobname jobname* ...*)
 - Minimum/Default is 4

RACLIST



- All profiles for a specified class are cached in a shared dataspace
 - SETROPTS RACLIST(*class*), if RACLIST=ALLOWED in CDT
 - RACROUTE REQUEST=LIST,GLOBAL=YES by certain applications

CICS	IMS	VTAM	MQSeries	DB2
------	-----	------	----------	-----
 - Updated with SETROPTS RACLIST(*class*) REFRESH
 - Profile segments are not stored in memory (e.g., STDATA)
 - Required to exploit grouping classes (e.g., DASDVOL / GDASDVOL)



- CDT RACLREQ=YES - Required

APPCSERV	APPCTP	CRYPTOZ	CSFKEYS	CSFSERV
DEVICES	DIGTCIRT	DIGTNMAP	FIELD	IDIDMAP
NODES	OPERCMD5	PROPCNTL	PSFMPL	PTKTDATA
RACFHC	RACFVARS	RDATA LIB	SECLABEL	SERVAUTH
STARTED	SYSMVIEW	UNIXPRIV	VTAMAPPL	

- RACLIST recommendations:

APPL	CDT	DASDVOL	DIGT Classes	DSNR
FACILITY	JES classes	LDAPBIND	LOGSTRM	PRINTSRV
RRSFDATA	TSO classes	TERMINAL	SDSF	SURROGAT

RACF Database Sharing



- Sharing a database in non-Sysplex Data Sharing mode (no Coupling Facility)
 - RACF uses exclusive hardware RESERVEs to serialize the database for most updates
 - System holding exclusive RESERVE locks out other systems until it has processed all its update requests
 - Lock is on entire DASD volume
- Global Resource Serialization (GRS)
 - Can convert RESERVEs to global ENQs
 - Each system given exclusive control for one update request at a time
 - Only locks the RACF database - not the entire DASD volume
 - Avoids contention and monopolization
 - PARMLIB(GRSRNLxx) conversion entry
RNLDEF RNL(CON) TYPE(GENERIC) QNAME(SYSZRACF)
 - Restrictions
 - ❖ All z/OS systems must be part of the same GRS complex
 - ❖ Cannot be used when sharing a RACF database with a z/VM system
 - GRS required for Sysplex Data Sharing

RACF Database Caching



■ RACF Sysplex Data Sharing

- Uses Coupling Facility as large store-through cache for the Resident Data Blocks - caches ICB, index, & profile data blocks (can improve performance for single system)
- Enabled by ICHRDSNT flag on first database entry
 - XL1'x0' No Sysplex
 - XL1'x8' RACF-Sysplex data communication without data sharing
 - XL1'xC' RACF-Sysplex data communication with data sharing
- Coupling Facility Resource Manager (CFRM) sets cache policy
- To assist in calculating the coupling facility size for RACF, go to <http://www.ibm.com/systems/support/z/cfsizer/racf/>
- If feasible, specify size large enough to hold all index blocks plus all data blocks for non-RACLISTed resource classes

Logging



- Log judiciously
 - SETROPTS LOGOPTIONS(ALWAYS(*class*) | SUCCESSES(*class*))
 - SETROPTS OPERAUDIT
 - Resource AUDIT(ALL | SUCCESSES(*level*))
 - Resource GLOBALAUDIT(ALL | SUCCESSES(*level*))
 - User UAUDIT

Statistics



- Limit user logon statistics update to only once per day
 - Implemented via APPL class profiles for associated applications
 - Specify `APPLDATA('RACF-INITSTATS(DAILY)')` to activate

z/OS UNIX Identity Mapping



- Mapping required when corresponding identity must be determined (e.g., Unix 'ls' command - display RACF USERID and Group for Unix Owner uid and Group gid)
- Options to avoid searching all user and group OMVS segments for each look-up request
 - UNIXMAP Class
 - ❖ Contains profiles in the form *Unnn* and *Gnnn*, where '*nnn*' is a uid or gid
 - ❖ Users and groups are 'permitted' access to signify uid and gid assignment
 - ❖ Profiles are automatically maintained when OMVS segments are created or altered via RACF commands
 - ❖ Class must be activated to be used for mapping
 - Application Identity Mapping (AIM)
 - ❖ Restructured database with mapping index structure
 - ❖ Implemented using IRRIRA00 utility
 - ❖ Replaces UNIXMAP profiles
 - ❖ Enables use of *UID(nnn)* and *GID(nnn)* on SEARCH command
 - ❖ Required to use newest features to replace the Unix Default User
- Additionally, cache uid and gid mappings in VLF

Virtual Lookaside Facility (VLF)



- VLF can cache RACF information for reuse
 - Accessor Environment Elements (ACEEs)
 - Group tree
 - z/OS Unix mappings of uids and gids to USERIDs and Groups
 - z/OS Unix User Security Packets (USPs)

- MAXVIRT parameter - VLF Maximum Virtual Storage
 - Optionally specified in PARMLIB(COFVLFxx) for each VLF CLASS
 - MAXVIRT(*nnnnnn*) - 4K block increments
 - ❖ Default: 4096
 - ❖ Range: 256 - 524288
 - Monitor VLF use - SMF record type 41, subtype 3
 - Default normally sufficient

Virtual Lookaside Facility (VLF)



▪ Accessor Environment Elements (ACEEs)

- Created during logon process - contains user's attributes, lists of groups, and logon characteristics (e.g., Point-of-Entry (POE), application)
- Caching avoids repeated retrieval of user profile for subsequent logons
- PARMLIB(COFVLFxx) entry
CLASS NAME(IRRACEE)
EMAJ(ACEE)
- Altering a user profile causes purge of all cached ACEEs for that user
- Refresh of logon-related classes causes purge of all cached ACEEs

▪ Group tree

- Used to determine scope-of-groups for Group-level authorities
SPECIAL OPERATIONS AUDITOR
- Caching avoids repeated retrieval of group profiles and tree reconstruction
- Implement only if group authority is used extensively
- PARMLIB(COFVLFxx) entry
CLASS NAME(IRRGTS)
EMAJ(GTS)

Virtual Lookaside Facility (VLF)



- z/OS Unix mappings of uids and gids to USERIDs and Groups
 - Caching avoids repeated retrieval of mapping information
 - Needed even with AIM restructured database
 - PARMLIB(COFVLFxx) entry
 - CLASS NAME(IRRGMAP)
 - EMAJ(GMAP)
 - CLASS NAME(IRRUMAP)
 - EMAJ(UMAP)

- z/OS Unix User Security Packets (USPs)
 - Created when user dubs (invokes z/OS Unix function)
 - Caching avoids repeated rebuilding of USPs during subsequent dubbing
 - Especially helpful for applications using thread level security
 - PARMLIB(COFVLFxx) entry
 - CLASS NAME(IRRSMAP)
 - EMAJ(SMAP)

Enqueue Residency - ERV



- Contention issue - low priority TSO user or batch job gets swapped out while still holding an enqueue on SYSZRACF or a hardware RESERVE on the RACF database volume, and thereby holds up other address spaces and systems waiting on RACF

- Solution - grant more CPU Service Units to address spaces enqueued on system resources or holding hardware RESERVEs enabling them to complete work before being swapped out

- PARMLIB(IEAOPTxx) - ERV parameter
 - Range: 0 - 999999
 - Default: 500
 - Recommended: 40000 - 50000

RACF Commands & Utilities



- Avoid use of commands and utilities that are I/O or processing intensive during peak system activity periods

LD ID(), PREFIX(), or DSNS

SR NOMASK, AGE, USER, or WARNING

LU * LG * RL class *

ICHDSM00 IRRDBU00 BLKUPD

IRRUT100 IRRUT200 IRRUT400

SETROPTS GENERIC(*class*) REFRESH [especially DATASET]

SETROPTS RACLIST(*class*) REFRESH [classes with many profiles]

Large batches of commands - especially CONNECTs & REMOVEs

- Specify parameter NOYOURACC (or NOY) on RLIST commands to avoid retrieval and RACLIST processing of all grouping class profiles simply to determine your access

- Keep the RACF database clean of unnecessary profiles

