RACLIST 101

aka "RACLIST boot camp"

New York RACF User's Group

10/30/2008

Russ Hardgrove
RACF Level 2
IBM – z/OS Software Support
Poughkeepsie, NY
hardgrov@us.ibm.com

Agenda

- RACLIST - a Brief History
- Why RACLIST?
- SETROPTS vs GLOBAL=YES RACLIST
- CDT Parms
- Differences
- AUTH vs FASTAUTH
- Some Mechanics
- SYSPLEX Comm - coordinator and peer
- RACGLIST
- Miscellaneous
- Review / Questions?
- Appendix / Reference Materials

RACLIST – A Brief History

Prior to RACLISTing, RACF needed to do I/O when performing it's authorization checks. This overhead made it impractical for fast application callers (CICS / IMS) to want to use RACF.

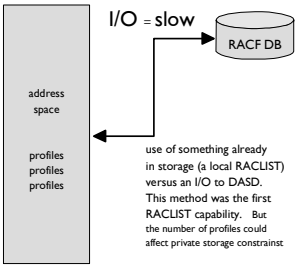
The initial form of RACLISTing was "LOCAL"; ie, profiles (for a given resource class) were placed in the user's / caller's address space.

This dramatically improved performance BUT at a cost some thought too high (many profiles meant MUCH region space was used - possibly below the 16 meg line).

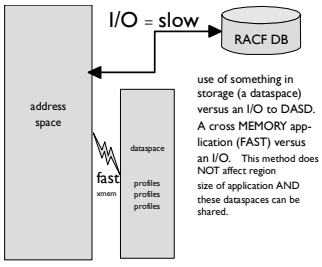
The answer ==> the use of MVS's dataspace technology. (common storage may have been used briefly, but was problematic at best)

The result was the speed of local RACLISTing w/o it's single biggest drawback (REGION use), and some other benefits as well... Multiple callers can use the SAME shared copy.

In-storage vs Database I/O



Dataspace vs Database I/O



Why even RACLIST ?

Mainly to avoid I/O. We recommend most general resource classes be RACLIST'd. (Dataset, Group and User profiles are ineligible.) Notible exceptions are classes that are volitile (ie, change frequently, like TAPEVOL).

The one downside (*if it can be called that*) is that profile changes (adds, deletes and alters - permits too) do not take effect immediately; the class must be REFRESHed. (more in a bit)

RACLIST – An Aside

Generic anchor tables (things anchored off one's ACEE for AUTH processing) is another way RACF attempts to limit I/O.

Up to four (4) of these tables are hung off the ACEE, each related to either a Dataset HLQ or an unRACLIST'd general resource. But, these tables only contain Generic profiles, and get recycled when a 5th HLQ/Class is needed.

So, better to use RACLIST. Also RACLIST avoids I/O & ENQs at (Fast)AUTH time.

SETROPTS vs GLOBAL=YES

Two different mechanisms that result in almost the very same thing. SETROPTS puts YOU in control; GLOBAL=YES gives programs (some) control.

Cmd: SETROPTS RACLIST(class)

Macro: RACROUTE REQUEST=LIST, GLOBAL=YES,CLASS=classname

SETROPTS LIST Output

```
...
GLOBAL CHECKING CLASSES = NONE
SETR RACLIST CLASSES = ACCTNUM CONSOLE FACILITY OPERCMDS
                       SDSF SERVAUTH STARTED SURROGAT
                       TSOAUTH TSOPROC XFACILIT
GLOBAL=YES RACLIST ONLY = NONE
LOGOPTIONS "ALWAYS" CLASSES = NONE
...
```

More SETROPTS

Once a class is RACLISTed (either way), any new profile changes will not take effect until a SETROPTS RACLIST(class) REFRESH is executed. In response to RALTER or RDEF:

ICH10006I RACLISTED PROFILES FOR clasname WILL NOT REFLECT THE ADDITION(S) UNTIL A SETROPTS REFRESH IS ISSUED. (* a caveat later)

And in response to an RLIST:

ICH13007I One or more requested profiles for _____ class are defined in the database, but are not listed. RACLIST REFRESH is required.

ICHERCDE (CDT) parms

RACLIST=ALLOWED|DISALLOWED
RACLREQ=YES|NO

ALLOWED means it can be RACLIST'd by any means.

DISALLOWED (default) means ONLY via RACROUTE. (i.e,. programs want to keep total control.) Results in

ICH14027I RACLIST OF CLASS class-name NOT ALLOWED BY THE CLASS DESCRIPTOR TABLE. OPERAND IGNORED.

RACLREQ means the using program(s) does not want I/O. It could want the performance boost from an AUTH (eg, OPERCMDS / CONSOLE) and/or will be using FASTAUTH without their own GLOBAL=YES (Health Checker). Results in:

ICH14040I WARNING! You must RACLIST class class-name before authorization checking can occur.

If using the CDT Class:

```
RDEFINE CDT dyn_class_profile
   CDTINFO( RACLIST(ALLOWED | DISALLOWED | REQUIRED)  )
```

More Differences

ICH10006I is only issued when the Class is under YOUR control; ie, RACLISTed via SETROPTS.

When issuing a SETR RACLIST (with or without REFRESH) for a class, all classes with that POSIT 'ride along'. This is not true for GLOBAL=YES (the **initial** RACLIST).

^^^^^

And then there is AUTH vs FASTAUTH…

AUTH -vs- FASTAUTH

An AUTH call does not expect / require (usually) a class to be RACLISTed (either way). But, to give YOU control, AUTH will first seek out a dataspace. **If found, it will be used (NO I/O). If not, off to do some I/O.**

On the other hand, FASTAUTH demands the class to be RACLISTed (either way) or the class will be deemed 'not found'. RCs 4/4/0:

The resource or class name is not defined to RACF or the class has not been RACLISTed.

With one exception, the class must be RACLISTed via GLOBAL=YES by the ASID in order to get addressability (via an ALET).

Still awake ???

Some mechanics

SETROPTS and GLOBAL=YES RACLIST can be an expensive process due to I/O's and ENQs. It is a function of size. So, the fewer done the better… and done offpeak if at all possible.

Never repeat classes (ala, shared POSITs); once is enough. Large numbers of profiles; large member / grouping classes; large access lists can all exacerbate the timing. (Minutes...)

RACGLIST (covered later) can mitigate this, especially in concert with SYSPLEX communications (next foil)...

LOCAL storage is used first (in caller's asid). Then the merged profiles are moved to a new dataspace. The old one (if REFRESH) is deleted. This is NOT serialized with AUTH/FASTAUTH. They get recoverable (and hopefully UNSEEN) 0E0 abends. New ALETs (MVS internal ptrs) are obtained by existing callers.

SYSPLEX Comm - coordinator & peer

Serialization is involved (for one SETROPTS RACLIST racing another).

The first occurs on the coordinator (where SETROPTS is issued). Once it ends there, the peers get their turn (at the same time). When all are done they report back to the coordinator who reports to the issuer when all are done:

ICH14063I SETROPTS command complete.

Try --NEVER-- to cancel a coordinator (or allow to time out) during this process. Be patient, especially for classes with large numbers of profiles:

IRRX017I NO RESPONSE RECEIVED FROM MEMBER memname WHILE PROCESSING function.

If during SETROPTS RACLIST activity, ENQS develop, the best advise is to let them continue and they should clear. These mean that OTHER SETR activities are colliding. Best to leave alone.

CPU times in coordinator asid and MASTER on peers can be high, especially if classes are large / complex (group and member).

When going to SYSPLEX communications for the FIRST time it is critical that serialization is correct. That SCOPE=SYSTEM is not increased and most importantly SCOPE=SYSTEMS is handled correctly. Also at that time, if prior procedures **"shotgun'd"** SETR jobs to all LPARS, **stop that.** Otherwise scary ENQs may be seen.

RACGLIST

The largest benefit comes when a class of profiles gets large. You MUST insure enough space exists on RACF DB to handle data because…

When SETROPTs RACLIST merging is done, the merged image is written BACK to the DB. Second and subsequent RACLISTs in the plex are much faster (ala peers). The same for the next IPL.

At the next IPL, the cast RACGLIST object is used; so in effect, a REFRESH is NOT done at IPL. (One IS done with non-RACGLIST'd classes.)

You can significantly improve IPL and region start-up times when employing this method. **And also for coordinated REFRESHes**.

Caveat: You need to realize that REFRESHes might be needed at / after IPLs.

Miscellaneous stuff

Local RACLISTing is still used. (VTAM)

With Local RACLIST, filtering is used (VTAM). It brings only a subset of profiles into it's storage. **Can use FILTER= too..**

GLOBAL=YES dataspaces are NOT deleted until a SETROPTS NORACLIST(clasname) is done. Using REQUEST=LIST,ENVIR=DELETE merely deregister's a program's interest in the class.

Lastly, data in a Segment (eg, STDATA) is NOT loaded into the dataspace; I/O is still required to get it. This does not affect AUTH. Only Extract….

Gotcha's

- CICS (or other) local class defs ==>>

  Do **NOT** use **RACLIST=REQUIRED**

  AUTH and FASTAUTH act differently …… triggered by AUDIT

- APAR OA26781 –

  There is an inconsistency in the Authorization processing between AUTH Check and FASTAUTH processing when the profile has

  - UACC(READ) & ID(*) ACCESS(NONE) -

Summary

- **RACLIST saves on I/O (&ENQs)**
- RACLIST can be done two ways:
  - SETROPTS RACLIST
  - RACROUTE REQUESTLIST,GLOBAL=YES
- CDT Parms determine which is allowed, or both
- There are differences !
  - AUTH vs FASTAUTH
  - whether shared POSIT is considered
- SYSPLEX Comm reduces the administration
- RACGLIST keeps data consistent when sharing

  and helps sysplex performance
- QUESTIONS??

Appendix / References / More

- z/OS V1R10 Security Server RACF Auditor's Guide
- z/OS V1R10 Security Server RACF Callable Services
- z/OS V1R10 Security Server RACF Command Language Reference
- z/OS V1R10 Security Server RACF Data Areas
- z/OS V1R10 Security Server RACF Diagnosis Guide
- z/OS V1R10 Security Server RACF Macros and Interfaces
- z/OS V1R10 Security Server RACF Messages and Codes
- z/OS V1R10 Security Server RACF Security Administrator's Guide
- z/OS V1R10 Security Server RACF System Programmer's Guide
- z/OS V1R10 Security Server RACROUTE Macro Reference

AT http://www-03.ibm.com/systems/z/os/zos/bkserv/hot_topics.html
See:

z/OS Hot Topics Newsletter #10 February 2008
"Rediscover the magic of RACLIST"
z/OS Hot Topics Newsletter #19 August 2008
"Demystifying the magic of RACGLIST / Conjuring data consistency across systems"

Ooo Rah !!!