



Craig S. Mullins

[Return to Home Page](#)

April / May 2009



zData Perspectives

by Craig S. Mullins

DB2 System Configuration and Performance

There are multiple types of IT professionals who deal with, use, and manage DB2 for z/OS on a daily basis. The vast majority are end users, followed by application developers and programmers. Then there are the DBAs, of which there are many varieties: application-focused, system-focused, performance-focused, and even jacks-of-all-trades. And finally we have the system programmers, those who work with installation and system tuning.

Of all these constituents, usually only the system programmers and some of the DBAs get involved with system-level performance and configuration issues. But

this topic is getting more complex and difficult to manage all the time, so let's take a look at some of the issues.

Probably the most important aspect of system configuration and tuning involves understanding and managing DSNZPARMs, aka ZPARMs. These are the system parameters that control the overall behavior of a DB2 subsystem.

There are seven macros that, once assembled, produce the DSNZPARMs: DSN6ENV, DSN6ARVP, DSN6LOGP, DSN6FAC, DSN6GRP, DSN6SYSP, and DSN6SPRM. I don't want to get into an explanation of how to set ZPARMs or what they are all in this column, though.

So far, so good -- but keep in mind that there are literally hundreds of these parameters, with more being added all the time. And the documentation on them is limited.

The sheer volume of DSNZPARM parameters can be overwhelming. Oh sure, some of them are more visible than others. For example, many people know about the locking parameters (NUMLKUS – maximum locks per user; NUMLKTS – maximum locks per user per table space) and some of the others like:

- CTHREAD – maximum users
- CONDBAT, MAXDBAT – maximum remote connections, active
- IDFORE, IDBACK – maximum TSO, batch
- DSMAX – maximum number of open data sets

These parameters have been in DB2 for ages and many of us have had exposure to them. But every new DB2 release adds more parameters and options, as well as changes old parameters, so we have to tango with many, many more ZPARMs

than before. As such, sometimes we just go with the defaults and hope for the best.

But this is not really a viable option in this day and age when we need to squeeze every last drop of performance out of our systems. Setting even just one ZPARM inappropriately for your system can have a huge negative impact on performance; and if you just take the defaults, you may have more than one thing set sub-optimally.

Consider, for example, CHKFREQ, or checkpoint frequency. First some background: periodically DB2 must take a system checkpoint. The checkpoint contains information on the currently open unit of recoveries within DB2, all open page sets, a list of page sets with exception states, and a list of page sets updated by any currently open unit of recovery. Also, buffer pool pages that need to be externalized are written out at system checkpoint. So it is important to set CHKFREQ appropriately, as well as your buffer pool deferred write thresholds (DWQT). Taking appropriate checkpoints speeds up the DB2 restart process. But taking a checkpoint also causes some minimal contention within DB2 and will also increase CPU usage by DB2.

The default for this parameter is 500,000, which means DB2 will take a checkpoint after writing 500,000 log records. But is this ideal for your environment? How much time will it take to write that many log records? And won't that time vary greatly between peak and off-peak hours? Simply accepting the default and forgetting it is probably not the best idea here.

As of DB2 V7, CHKFREQ can be set to a number of minutes, instead of the number of log records written, if you so choose. If you have variable logging rates, you might want to set CHKFREQ as a time instead. Doing so can bring some consistency to your checkpoint processing.

And that is just one example... remember, there are hundreds of these parameters, all with different impacts, effects, and concerns.

Of course, today's column just looks at the tip of DB2 system configuration and performance management. Other considerations include:

- Which parameters can be changed online, while DB2 is up, and which cannot.
- If, when, and how to go about modifying hidden DSNZPARM values.
- How to perform change management on system parameters.
- And do you document what parameters gets changed when, and why?

Indeed, there is a lot to consider in terms of what is happening with DB2 for z/OS under the covers... So end users and application folks, aren't you glad there are people that worry about this in your shop so you don't have to? There are people worrying, right?

From [zJournal](#), Apr / May 2009

.

© 2009 Craig S. Mullins, All rights reserved.

[Home](#).