

TWENTY YEARS OF PERFORMANCE TUNING

by Chris Lawson



TIP #7

Be Slow to Blame Insufficient Resources

If you don't know what to do, you can always say, "Add memory!" or "Add more CPU's!" It's easy to say, because it requires no analysis. And that's part of the fun--anybody, no matter how incompetent, can make that suggestion. The value of this suggestion matches the work. It is *rarely* the right answer. In fact, performance suggestions that rely on *generalities* are rarely the answer. Specific performance problems require *specific* performance solutions.

Considering all the performance problems I have analyzed, only a handful of performance problems were actually due to insufficient memory or CPUs. I've never actually seen even one performance problem that was due to insufficient database cache. (A friend of mine told me he saw one once.)

Normally, if some people are determined to add more memory to the database cache, I don't argue with them, but just continue on analyzing the real problem. After all, it won't *hurt* anything to have more memory. We did a test recently on a 100 TB database (which already had plenty of memory.) By drastically increasing the database buffers, we attained a 1% improvement in the cache "hit ratio."

Of course, if you are in the early stages of creating some application and setting up the database, you might indeed discover that you need to change some init.ora parameters. As part of the design process, you may discover you have grossly underestimated the CPU requirements. Usually, however, the siren call for more CPU or memory is based on guesswork, not actual analysis by experienced DBAs.



Chris Lawson is the author of *The Art & Science of Oracle Performance Tuning*, as well as *Snappy Interviews: 100 Questions to Ask Oracle DBAs*. When he's not solving performance problems, Chris is an avid hiker, runner, chorister, Amazon reviewer, and geocacher. Chris writes using the penname, "*Bassocantor*."

Twenty Years of Performance Tuning is a series of tips based on the author's experience solving performance problems over the last 20 years.