Fastest Informix DBA: How Did They Do It?

Here are the techniques that worked for the Fastest Informix DBA contest winners

Lester Knutsen
(lester@advancedatatools.com) is president of Advanced DataTools Corporation, an IBM Informix consulting and training partner specializing in data warehouse development, database design, performance tuning, and Informix training and support. He is president of the Washington, D.C. Area Informix User Group, a founding member of IIUG, an IBM Gold Consultant, and an IBM Data Champion.

At the IIUG Informix Conference in April, we ran a Fastest Informix DBA contest. I took a simple customer billing process and added some bad SQL and a default ONCONFIG file with some bad configuration options—recreating the sort of challenges we see every day. The unchanged benchmark took about 30 minutes to run, and I challenged participants to make it run faster. The fastest DBAs tuned it to run in fewer than 4 minutes.

In my last column, I talked about the challenge and listed the winners; this time, we'll look at what worked and how they did it.

First, study the problem

The DBAs who succeeded spent as much time as they could studying the problem. We had a document that described the benchmark, including all the code and the expected results. We also had a video where I described the problem they needed to solve, and I showed them how to run the benchmark. Those who spent more time studying all the material did better, because they were able to better focus their efforts. For example, I purposely designed the schema so that there was a very high buffer turnover rate, and the fastest DBAs looked at the data and the schema and figured this out. Also, the benchmark system had only one disk, so a lot of tuning to take advantage of Informix parallel disk reads and writes would be unlikely to help much. The first thing that all the successful DBAs did was study the problem, analyze the facts, and then come up with a plan.

Informix configuration: ONCONFIG file changes

One challenge involved deciding what changes to make to the Informix Dynamic Server (IDS) configuration file, the ONCONFIG file. These changes were specifically targeted for the benchmark environment and may not help in all situations, but they do give you a good idea of what to look for in your own ONCONFIG file.

BUFFERPOOL—All of the fastest DBAs increased the number of buffers that the server used. The server had 3 GB of RAM. The fastest DBA used almost half of the RAM for buffers, created a dbspace with a 16 KB page size (larger than the default 2 KB page size), and allocated one-third of the memory for buffers for this 16 KB page size. This solved the problem of the record length being too large to fit on a default 2 KB page size, kept the records together, and put the most number of records in memory. Overall, I suspect that BUFFERPOOL adjustments made the biggest difference in performance.

The fastest DBAs were also careful not to make the BUFFERPOOL too large, as this would have caused the OS to start swapping to disk and would have slowed down the entire system. When you add buffers, you also need to consider the number of Least Recently Used (LRU) queues that manage all these additional pages in the BUFFERPOOL. The fastest DBAs increased the LRU queues to more effectively handle the additional memory.

SHMVIRTSIZE—This is the amount of memory Informix will allocate to workspace and virtual memory. All of the faster DBAs increased it, and the fastest increased this the most. There were a couple of very large "group by" statements in the SQL, and increasing virtual memory along with some other changes to the Parallel Database Query (PDQ) process allowed more of this work to take place in memory.

DS_TOTAL_MEMORY—This is the amount of memory from the SHMVIRTSIZE memory that will be used for PDQ operations. The default is very small, and increasing this may have helped with sorts and index builds.

DS_NONPDQ_QUERY_MEM—This is the amount of memory allocated to sorting when a sort is performed that does not use PDQ. With only one disk drive on the benchmark system, not much could be done with PDQ. Increasing this parameter helped with sorts and index builds.

LOCKS—This is the number of LOCKS and memory available for these LOCKS. If this configuration parameter is not big enough, Informix will dynamically increase it, but increasing it on the fly is very inefficient. The fastest DBAs set the number of LOCKS so the server did not need to dynamically increase this parameter.

RESIDENT—Setting this keeps Informix in memory and tells the OS not to swap the database server out to disk. All of the faster DBAs set this to keep Informix in memory.

CPU VPs—The benchmark machine was a four-core machine and could support four Informix CPU virtual processors (VPs). All of the faster DBAs set the number of CPU VPs to between three and four to take advantage of all CPUs on the machine.

DBSPACETEMP—I created a temp dbspace in the base configuration but did not define it in the ONCONFIG file, so it was not used. Instead, the rootdbs was used for sorts and temp files. Again, all of the faster DBAs changed the ONCONFIG file to identify and define this parameter. Several even created two or three additional temporary dbspaces, so Informix could read and write to tempdbs in parallel.

Some of the fastest contest participants changed other ONCONFIG parameters, including PHYSBUFF and LOGBUFF, DIRECT_IO, VP_MEMORY_CACHE_KB, and CLEANERS. It's hard to know exactly how much these contributed, but these are the places where the very fastest DBAs found extra speed. I was also interested in which parameters did not get changed. It may have been because of time, but no one changed the read-ahead parameters RA_PAGES and RA_THRESHOLD, and no one changed the index-cleaning parameters or changed the BTSCANNER.

Informix schema changes

I purposely designed the database with two tables that had very large columns, a CHAR(2000) column in the customer table and a CHAR(1000) column in the bills table. However, most of this was wasted space. In the customer table, only about the first 100 characters were used, and in the bills table this field was never used. Not only did this waste space, but it caused the tables to overflow a 2 KB page and created most of the buffer thrashing and very high buffer turnover rates. There are a couple of solutions to this, one of which is to alter the table and turn these columns into LVARCHAR columns. This change reduced the number of buffers that were read and written during the benchmark and may have had one of the biggest impacts on overall performance.

Another schema change that a few DBAs made was to move the index creation on the bills table to after all the data was inserted, instead of before that data was inserted. This made for a faster load of the table without an index, and when the index was built, it was more compact and optimized. Also, in IDS 11.50, building an index performs an automatic UPDATE STATISTICS HIGH on the table, which would provide the Informix query optimizer with better information about the table. A few of the DBAs added additional indexes on the customer table, which may have helped the performance of the last query in the benchmark.

SQL optimization

The benchmark process comprised two INSERT statements into a bills table and three UPDATE statements. The UPDATE statements contained subqueries. At the end of the benchmark process, two SELECT statements with group by clauses were executed to check the numbers. The resulting numbers from the last two statements had to match the expected results; this is how we verified that the benchmark was completed and correct. I added redundant and unnecessary code to the SQL statements to make it challenging.

With some careful planning, I think the whole process could have been done as one INSERT statement and one or two UPDATE statements, but no one managed that. However, several of the faster DBAs did identify the unnecessary code in the SQL statements and removed that code from the benchmark process.

More to come at IOD 2009

The contest was a lot of fun to run and monitor, and it is exciting to see the ingenuity and creativity that all the Informix DBAs who participated put into it. Congratulations to the winners, who were announced in the last issue (**ibm.com**/developerworks/data/dmmag/archive.html) and are listed on our Web site at www.advancedatatools.com/Informix/index.html.

We sponsored another version of this contest, the Fastest Informix DBA Contest II, from June 18 to September 30, 2009. At the IBM Information On Demand 2009 Global Conference in October we will hold a Webcast and a Birds of a Feather session on the contest—visit the Advanced Data Tools Web site (above) for more details. Hope to see you there.