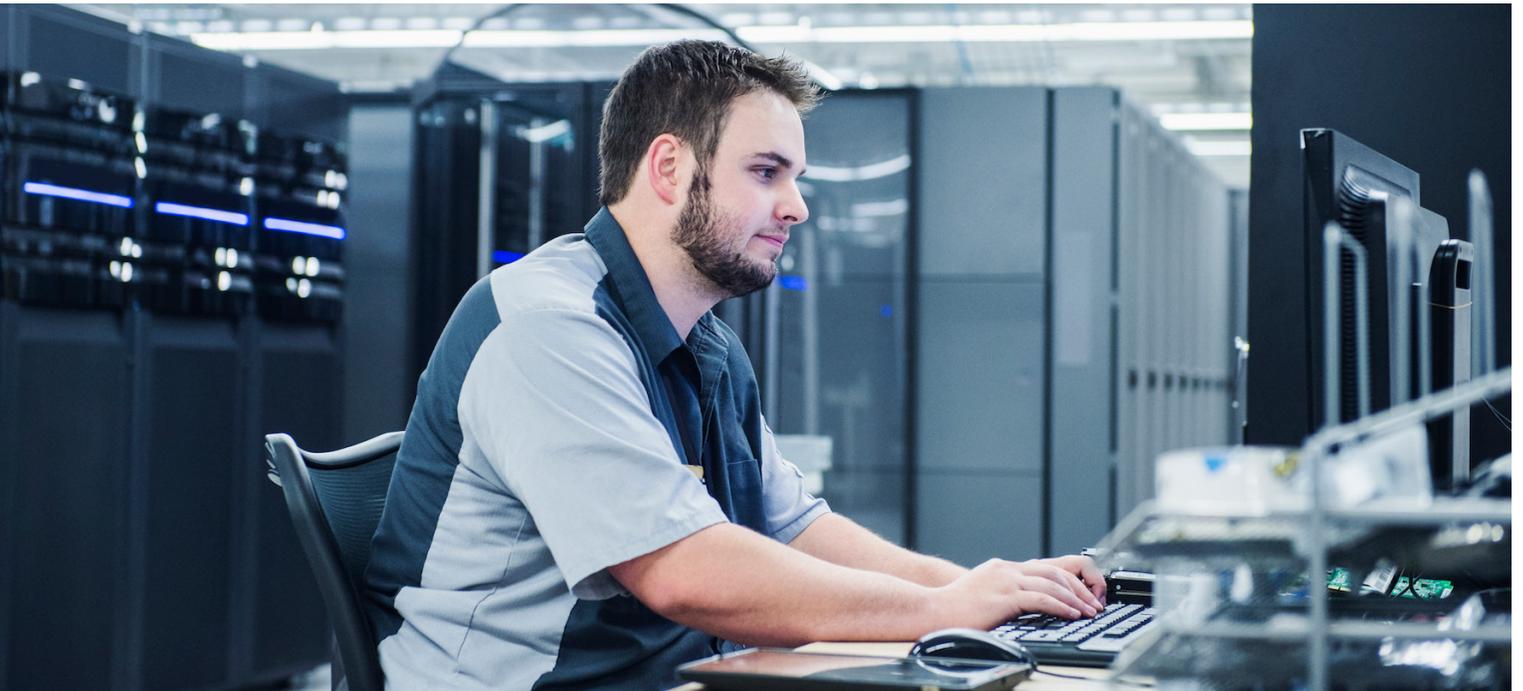# Quest™

Top 10 fallacies in benchmarking

Discover the truth about this important practice and learn to implement it with efficiency and effectiveness.

Most of us have great respect for the science of database benchmarking and the people who practice it. It's not a science for novices or haphazard attempts; rather, it requires proper preparation, adequate tools and reasonable expectations. In fact, when working with customers who have moderate to severe database benchmarking difficulties, we almost always discover that the problems stem from poor preparation, inexperience and impatience. For example, many customers who have never even read the TPC-C benchmark specification assume they can successfully perform that benchmark and know how to score or interpret it.

In this paper, we discuss the top 10 fallacies in benchmarking preparation and execution that we encounter on a regular basis. Understanding these misconceptions will help you be more efficient and effective in your benchmarking efforts.

1. I'M USING A TOOL SUCH AS QUEST® BENCHMARK FACTORY® FOR DATABASES, SO THAT'S ALL I NEED

While the right tools are definitely useful, you also need to read the specs for whatever industry-standard tests you are going to perform. The software you're using to automate these tests will ask questions or present options that you cannot really answer unless you understand their context.

For example, the highly popular "OLTP" test, known as the TPC-C Benchmark, defines "scale" factor as follows:

- Section 4.2.1: The WAREHOUSE table is used as the base unit of scaling. The cardinality of all other tables (except for ITEM) is a function of the number of configured warehouses (i.e., cardinality of the WAREHOUSE table). This number, in turn, determines the load applied to the system under test, which results in a reported throughput (see Clause 5.4).

- Section 4.2.2: For each active warehouse in the database, the SUT must accept requests for transactions from a population of 10 terminals.

So when a tool such as Benchmark Factory asks for the scale factor, it does NOT mean the number of concurrent users but rather the number of warehouses. For instance, a scaling factor of 300 means 300 warehouses and, therefore, up to 3,000 concurrent users. Failing to understand this definition of "scale factor," then, can skew your benchmark by an order of magnitude.

This requirement to read the spec is critical, and it will be an underlying issue for every remaining misconception discussed in this document.

> Failing to understand [the] definition of "scale factor," then, can skew your benchmark by an order of magnitude.

2. I HAVE AN EXPENSIVE SAN, SO I DON'T NEED TO CONFIGURE ANYTHING SPECIAL FOR IO.

Actually, the size, type and nature of the test may require radically different hardware settings, even all the way down to the deepest level of your SAN. For example, here are some of the things that might differ depending on whether you are performing a data warehousing test such as the TPC-H or an OLTP test such as the TPC-C:

- The TPC-H is best handled by a SAN with read-ahead and data-cache settings that are set more for read than write, while the OLTP TPC-C would benefit from just the opposite.

- The SAN hardware settings for stripe depth and stripe width should be set differently as well, and the file system and database IO sizes should be a multiple of the stripe depth. In fact, a common rule of thumb is:

  - Stripe Depth >= db_block_size X db_file_multiblock_read_count

- The optimal hardware RAID level is different: Although RAID-5 might be the best choice for OLTP, data warehousing might be better served by RAID-0+1.

- Finally, the number of disks can also be critical. For example, TPC-H tests start at around 300 GB in size, so anything less than 100 spindles at that size is generally a waste of time. And as you scale larger, 800 or more drives become common as the minimum recommended setup. The point is that no SAN cache is ever large enough for the monstrous workload of data warehousing queries.

In fact, we've seen up to 500 percent result differences when varying SAN settings and the number of disks. Relying on defaults can be a really big mistake.

3. I CAN JUST USE THE DEFAULT OPERATING SYSTEM CONFIGURATION RIGHT OUT OF THE BOX.

The truth is, most databases require some prerequisite operating system tweaks, and most benchmarks can benefit from a few additional adjustments. For example, we've seen 50 percent to 150 percent benchmark differences running TPC-C benchmarks for both Oracle and SQL Server by adjusting just one simple file system parameter. Yet that parameter is not part of either database's install or configuration recommendations.

Now you might argue that you can skip this step because it will be an "apples to apples" comparison since the machine setup will be the same across tests. OK, but why potentially wait three times as long for worse results? Because a 300 GB TPC-H test can take days just to load, efficiency is often critical to meet your deadlines.

Quest

4. I CAN JUST USE THE DEFAULT DATABASE SETUP AND CONFIGURATION RIGHT OUT OF THE BOX.

While some databases such as SQL Server might be "universally useful" as configured out of the box, other databases, including Oracle, are not. For example, the default number of concurrent sessions for Oracle is 50, so if you try to run a TPC-C test with more than 50 users, you're already in trouble.

Plus, as we've already seen, the nature of the benchmark dictates how numerous database configuration parameters should be set. A TPC-C test on Oracle would benefit from the following init.ora parameter settings:

```
CURSOR_SPACE_FOR_TIME = TRUE

CURSOR_SHARING = SIMILAR

OPTIMIZER_INDEX_CACHING = 80

OPTIMIZER_INDEX_COST_ADJ = 20

DB_FILE_MULTIBLOCK_READ_COUNT = 2
```

We've seen as much as a 533 percent performance improvement just from adjusting these five parameters alone

(see test runs 2–4) — imagine what a careful review of all the database configuration parameters for the test nature could provide.

5. TOOLS SUCH AS BENCHMARK FACTORY WILL CREATE OPTIMALLY DESIGNED DATABASE OBJECTS (I.E., TABLES, INDEXES, PARTITIONS, ETC.) FOR MY HARDWARE AND DATABASE SETUP.

The purpose of software such as Benchmark Factory is to automate the complex and tedious process necessary to execute a benchmark. While you can use the tool's default object sizing and placement, it's quite unadvisable to do so. Instead, overriding the default creation SQL to specify the tablespace, partitioning, sizing and other selections specific to your setup. With Benchmark Factory, you do this by clicking the "Custom Creation SQL" Tab, as shown below.

This manual effort can easily result in performance that differs by orders of magnitude, so it's almost always worth doing.

> The nature of the benchmark dictates how numerous database configuration parameters should be set.
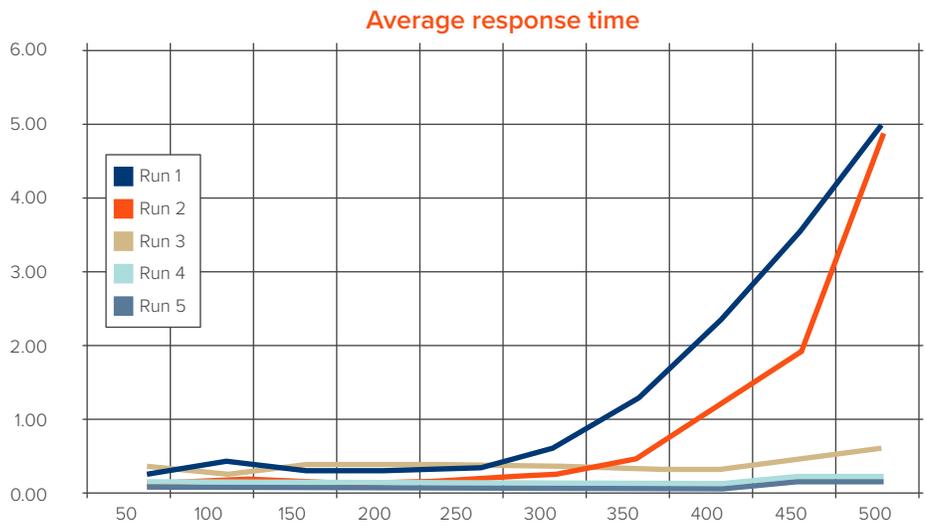

Figure 1. Changing init.ora parameters can have a dramatic effect on response time.
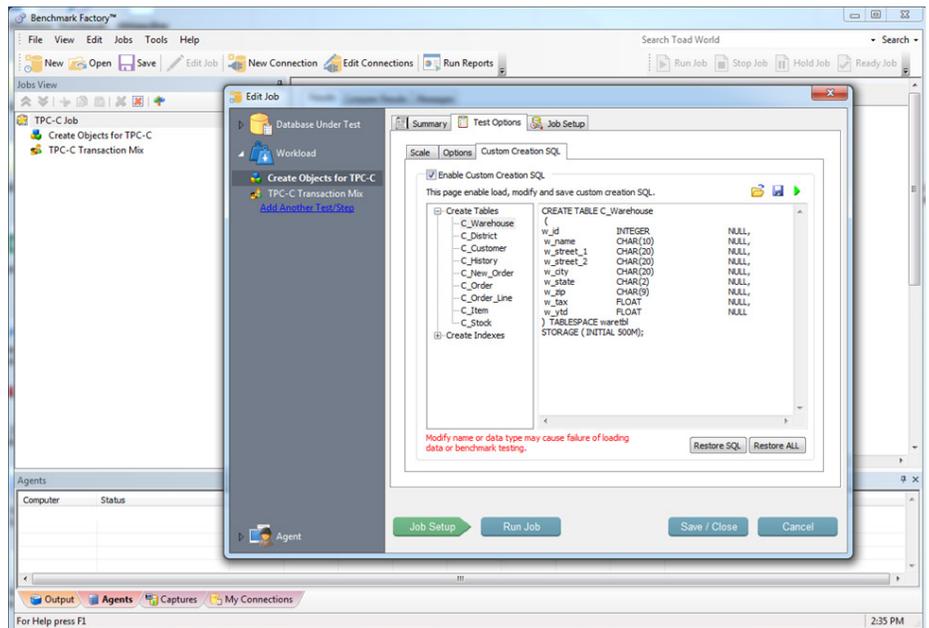
Quest

Figure 2. Specifying the object sizing and placement that's best for your setup can improve performance dramatically.

[TPC-H] tests the database optimizer's efficiency in handling complex statement explain plans and their executions.

6. TOOLS SUCH AS BENCHMARK FACTORY WILL AUTOMATICALLY MONITOR, TUNE AND OPTIMIZE ALL MY HARDWARE, OPERATING SYSTEM AND DATABASE CONFIGURATION PARAMETERS.

As we just saw, software such as Benchmark Factory is simply a tool to automate the complex and tedious process necessary to execute a benchmark. For example the TPC-H is a collection of 22 long and complex SELECT statements against a simple database design with some really big tables. It tests the database optimizer's efficiency in handling complex statement explain plans and their executions. If you want to monitor, diagnose or tune/optimize your database for such tests, you'll need different tools, such as Quest Spotlight® on Oracle, Foglight® Performance Analysis for Oracle and Toad® for Oracle with the DBA Module.

See Figure 3 for an example of some tuning available just within Toad. Spotlight on Oracle and Foglight Performance Analysis offer infinitely more complete and robust features to monitor, diagnose and tune/optimize your database. Remember, Benchmark Factory is simply a "load generator."

Quest

Toad DB tune and optimize checklist		
Static pre-check analysis and repair	Live instance monitor and correct	Dynamic app/session tune and optimize
Database health check utility	Database monitor screen	Session browser screen
Oracle parameters screen	Database probe screen	Indes monitoring screen
Redo log manager screen	ADDM/AWR admin/report utility	Code Xpert screen/utility
Tablespaces screen	Stats pack admin/report (still in beta)	Quest SQL optimizer interface
Tablespace map screen	Log switch frequency map screes	Explain plan screen
Segment advisor utility	SGA trace optimization screen	Analyze objects screen
Undo advisor utility	Top session finder screen	Pinned code screen
DBMS redefinition wizard	Server statistics screen	Oracle TKPROF utility

(Left side row labels: "Always" for top four rows, "Sometimes" for bottom four rows)

Legend:
- Toad Standard Edition
- Toad Xpert Edition
- Toad DBA Module

Figure 3. Tuning with Toad

7. I DON'T NEED A DBA TO PERFORM BENCHMARK TESTS — ANYONE TECHNICAL CAN DO IT.

Look at all the issues above again. Database developers or other technically savvy people may not be cognizant of these issues or authorized to make decisions about them. As we have seen, benchmarking requires more than just someone to run tests; it requires someone who knows benchmarking and can speak to all the issues above. Otherwise, you'll get results that are not really what your hardware can do, and they could be off by 500 percent or more. Those performance differences are more than just background noise — you could make bad strategic decisions with such misinformation.

Quest

Transactions / second
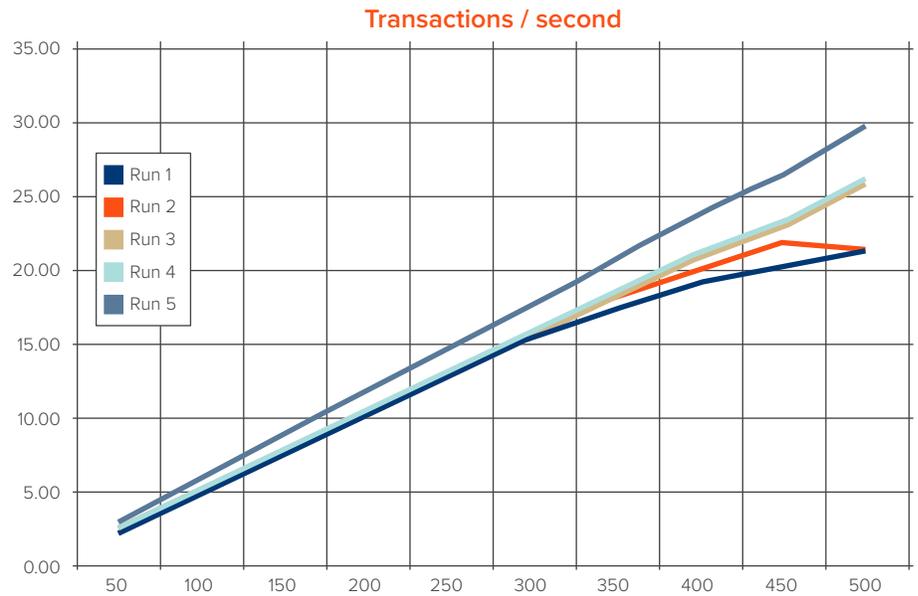
Figure 4. Transactions per second may not change as much as average response time.

8. I CAN VERY EASILY COMPARE DATABASE VENDORS ON THE SAME HARDWARE PLATFORM.

This is sometimes true but usually false. If you have one or more DBA who can address all of the above issues for each different database platform, then you can make an accurate comparison. Otherwise, you simply cannot compare the databases reliably by simply installing and running the same test for each. There are far too many dependencies and variables to trust such a simplistic approach.

9. TRANSACTIONS PER SECOND (TPS) IS WHAT MATTERS MOST IN BENCHMARKING.

This is rarely true. I think TPS is one of the most misleading values, but everyone seems to focus on that one first. Let me explain. As you increase the number of users or tasks being performed, by definition you're actually increasing the TPS. At some point it may plateau, but what does that tell you? Look at Figure 4.

This figure shows the TPS results for the same test runs shown in Figure 1, where we saw a 533 percent improvement in the average response time by adjusting the default configuration. The change in response time means something real in user or SLA terms, and it's also obvious and understandable.

The way the TPS chart should be analyzed, on the other hand, is via the first derivative of the various lines, by looking for the best growth in slope. In Figure 4, that's obviously the line for test #5, but it's hard to say what the differences in slope mean in terms of how many users we can support. On the other hand, using the chart in Figure 1, we can simply draw a line at the response time that we (or our users) deem acceptable and look where our line crosses that boundary — that's our maximum number of users.

If we had used the default database setup, the server would "run out of gas" at slightly more than 300 concurrent users. But the final tweaking efforts seem to indicate we might actually get almost double that number.

> The way the TPS chart should be analyzed... is via the first derivative of the various lines, by looking for the best growth in slope.

Quest

10. WE CAN SPEND JUST A FEW DAYS AND BENCHMARK EVERYTHING WE'RE INTERESTED IN.

Almost universally, the proper setup of all the dependent components can take a week or two — and when doing large-scale benchmarks, you can sometimes add up to a week for data loads and index creation. Make sure to carefully budget your time to complete these efforts. Remember, there will be surprises along the way, even if you do follow all the above advice.

CONCLUSION

Being aware of these common fallacies about database benchmarking will help you to be more successful in your efforts. Remember, the keys are proper preparation, adequate tools and reasonable expectations.

Quest

Quest