ONTOTEXT ANSWER:

The question of optimizing any database is often a critical one. Even with the rise of cloud computing and how easy it is to obtain hardware nowadays, computing power does not come cheap. So, what is most optimal for performance and how to save costs?

There are essentially three attributes that heavily impact GraphDB's performance. Those are memory, processor and storage.

Random access memory

Random access memory is probably the easiest to understand when it comes to optimizing GraphDB's performance. The more memory you can give to the database, the bigger part of your knowledge graph can be kept in it. Accessing data from RAM is much faster than reading it from disk, so a RAM increase would result in better performance. This increase is limited by two factors:

- GraphDB is not designed to load all of its data in memory. New triples always need to be stored on disk. Also, the sheer size of the database would be a limitation. Billions of triples can lead to data sizes over a terabyte and using so much memory would be prohibitively expensive.
- RAM is not used for computations. So, it would always be blocked, at least partially, by the CPU.

Beside those limitations, some operations are performed only in-memory. This means that, for each database, there is a minimum RAM size. This is noted in our documentation.

While RAM frequencies and other technical attributes also have an impact on performance, it is negligible.

Processing

As a multi-threaded application, GraphDB can benefit from more cores. The number of cores you can use is tied to your license. More cores lead to more concurrent queries and faster

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com

inference. This means that in general, CPU cores have a strong impact on your performance. However, there is always single-threaded computations. For example, writing into storage is always a single-threaded operation as we can't risk multiple threads overwriting the same data. The same applies for some queries, which also have a single-threaded part. Therefore, the individual clock speed of CPU cores also has a large impact on your overall performance.

CPU caches have a small effect on performance as well.

Storage

GraphDB is a database that persists the data to disk. Just like RAM, you need sufficient storage to actually store all your data. And, just like RAM, the actual speed of storage has a large impact. We always recommend using SSDs with GraphDB. In particular, more recent technologies like NVME are a huge boon to the database's performance.

On the other hand, technologies such as RAID or NAS, while good for redundancy, usually have a negative impact on performance. This is particularly evident with cloud providers. AWS, Azure, Google cloud all have quotas for IOPS and bandwidth on their disks. This can be a serious problem, as when you hit that quota, your requests are throttled. So, you need to keep a close eye on monitoring your disk usage.

To sum it up, the question of hardware optimization of GraphDB is a complex one. There's two barriers to entry, related to the database size – RAM and Storage size. Actual performance depends heavily on CPU speed, but it's important to watch out for throttling from IO operations. Concurrent operations, on the other hand, benefit greatly from an increased number of cores. In such an interconnected environment, very often, it's not enough to increase just one parameter. This is why we recommend keeping a close watch on GraphDB's resource utilization and boosting your hardware as needed.

You can check our benchmark page, which reflects some of those concerns. Get in touch with us for recommendations for cloud hardware to make the most out of your GraphDB.

Help us get better: review GraphDB on Gartner

Copyright: Ontotext New York | Sofia | Basel
Reach out to us: info@ontotext.com