



www.redis.com

700 E El Camino Real Suite 250
Mountain View, CA 94040, USA
Tel: +1 415 930 9666
Email: info@redis.com

Redis Enterprise

The company

Redis is a leading in-memory, multi-model database platform. It is also the commercial provider of Redis Enterprise, an enhanced (and proprietary) version of Redis that provides additional functionality (primarily scale-out cluster support and high availability) designed to make it better suited for the enterprise. This is available both on-premises and in the cloud (Amazon, Microsoft, Google, Heroku and others) and can be deployed as a fully managed service. Redis itself is often rated as one of – if not the – most popular NoSQL databases, with more than 7,100 paying customers. The company also extends the native characteristics of Redis and Redis Enterprise to many popular industry use cases implemented through modules that the company has developed. By “module” the company means functionality embedded into the product as opposed to something tacked on top.

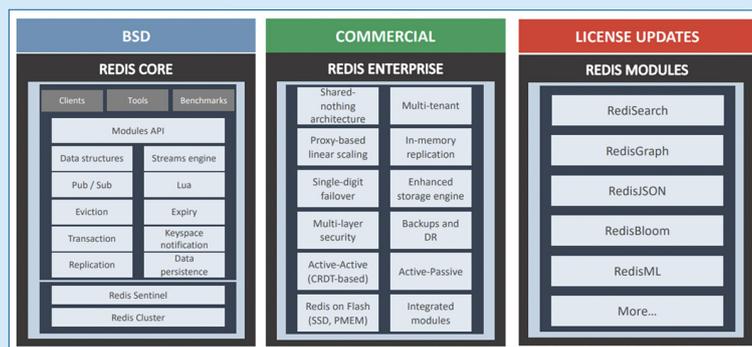
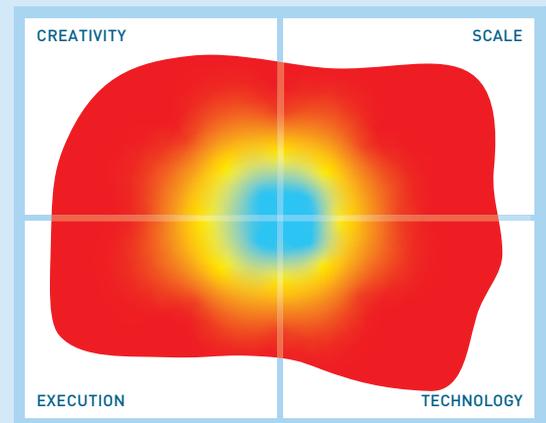


Figure 1 – The open source and commercial capabilities of Redis

Redis is a privately-held company backed by venture capital, as of February 2019 raising \$146 million in financing. It was founded in 2011 and is based in Mountain View, California. It has additional offices in London and Tel Aviv.

What is it?

Redis Enterprise was originally an in-memory, distributed (automated partitioning), NoSQL database with a key-value store as its underpinning. The core open source and commercial capabilities



The image in this Mutable Quadrant is derived from 13 high level metrics, the more the image covers a section the better. Execution metrics relate to the company, Technology to the product, Creativity to both technical and business innovation and Scale covers the potential business and market impact.

are shown in **Figure 1**. However, this description does Redis a disservice because the company’s approach to modules is such that, in reality, Redis is better thought of as a multi-model database that can be used to support document processing, graph traversals, stream processing (a core feature rather than a module though, rather surprisingly, there is no support for Kafka), machine learning, time-series, search and so forth. It is also possible to write your own modules.

From a transactional perspective, ACID transactions may be supported on data within a cluster, as long as the cluster is not geographically distributed. Both synchronous and asynchronous replication are supported and therefore both Active/Active and Active/Passive deployments, in the former case relying on a CRDT (conflict free replicated datatypes) approach and leveraging strong eventual and causal consistency. The choice of replication methods also allows tuneable durability (disk-based or replication-based) and consistency. Both strong and weak consistency (note: this is not the same consistency as ACID consistency) are supported within a local cluster but if you have a geographically dispersed environment (either Active/Active or using read replicas) then these would be eventually consistent.

Architecture	★★★★★
Concurrent analytics	★★★★★
Criticality	★★★★★
Ease of use	★★★★★

Flexibility	★★★★★
In-process analytics	★★★★★
Performance	★★★★★
Scale	★★★★★

“Redis Enterprise helped us deliver applications faster and with greater reliability than ever before. It enabled us in scaling by executing more than 2 million transactions – all within a short window of 6-8 hours.”
India's National Informatics Centre

How does it work?

The architecture of Redis Enterprise is illustrated in **Figure 2** where the Cluster Watchdog does what its name suggests; the Node Watchdog supports a secure (multi-tenant) user interface, a call level interface and a REST API; and the proxies handle the complexities of, for example, shared memory.

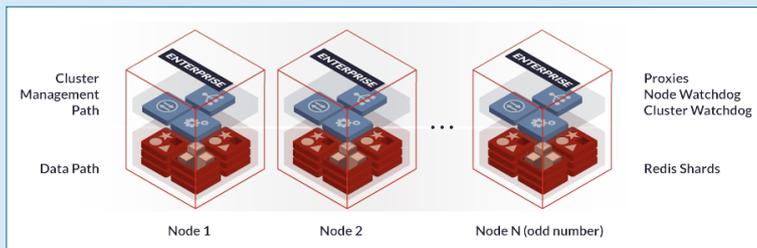


Figure 2 – Redis' shared nothing symmetric architecture with node and cluster watchdogs

Perhaps more interesting from the point of view of combining analytic processing with operational and transactional processing is the way that Redis leverages different data

structures. This is illustrated in **Figure 3**, along with Redis' own suggestions about where these particular different data structures might be useful. Alongside these data structures, Redis provides a variety of inline analytics through its modules, which range from specific functions such as Topk, which tracks the most frequent elements in a data set, to RedisAI, serving deep

learning models with built-in integrations into popular AI frameworks, supporting machine learning models and model serving. Further, there are also operational analytics capabilities such as RedisSearch, RedisGraph, RedisJSON and RedisTimeSeries, amongst others.

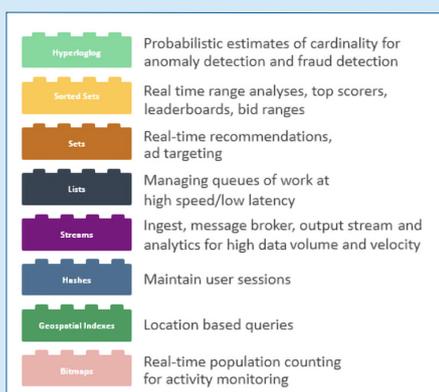


Figure 3 – Combining analytic processing with operational and transactional processing

More generally, Redis supports a wide range of programming languages, including Python, R and Scala. RedisAI supports TensorFlow and PyTorch and in the future, ONNX.

Finally, Redis is well-known for its performance, not least because it has historically run everything in memory. However, as the company moves away from focusing on caching use cases into wider environments, it can no longer assume that all of its clients can afford the amount of memory that may be required. For this reason, warm (as opposed to hot) data may be stored on SSDs and the company has worked closely with Intel on its Optane DC Persistent Memory technology, with general availability being announced in April 2019.

Why should you care?

Apart from its performance and scalability – which are obviously major factors – the most outstanding thing about Redis Enterprise is the flexibility that its support for different data structures and modules provides. Moreover, a significant number of these features explicitly support analytic as well as transactional capabilities. For example, prior to the introduction of modules, Redis did not have any of the core data structures to enable fast and efficient manipulation of JSON documents or the ability to use Redis as a Graph database. Many competitive solutions fall short as they impose restrictions on the data to fit their native data model. Modules like RedisJSON are designed to be intuitive regardless of whether someone is familiar with Redis or JSON. We should also say that we are especially impressed with the architecture behind RedisGraph, which uses sparse adjacency matrices. More broadly, we can say that Redis supports a broader range of pre-built analytic capabilities than most, if not all, other NoSQL databases.

The Bottom Line

It is interesting to observe how Redis has managed to leverage its initial success as a caching technology, into something more general-purpose. It is now a major contender in the provision of hybrid analytic and operational/transactional processing.

[FOR FURTHER INFORMATION AND RESEARCH CLICK HERE](#)