Memory and performance

Redis Enterprise Software has multiple mechanisms in its architecture to help optimize storage and performance.

Memory limits

Database memory limits define the maximum size your database can reach across all database replicas and shards on the cluster. Your memory limit will also determine the number of shards you’ll need.

Besides your dataset, the memory limit must also account for replication, Active-Active overhead, and module overhead. These features can significantly increase your database size, sometimes increasing it by four times or more.

Factors to consider when sizing your database:

- **dataset size**: you want your limit to be above your dataset size to leave room for overhead.
- **database throughput**: high throughput needs more shards, leading to a higher memory limit.
- **modules**: using modules with your database consumes more memory.
- **database clustering**: spreading your data into shards across multiple nodes (scaling out) means you cannot disable clustering or reduce the number of shards later (scaling in).
- **database replication**: enabling replication doubles memory consumption
- **Active-Active replication**: enabling Active-Active replication requires double the memory of regular replication, which can be up to four times (4x) the original data size.

For more information on memory limits, see Memory management with Redis Enterprise Software or Database memory limits.

Eviction policies

When a database exceeds its memory limit, eviction policies determine which data is removed. The eviction policy removes keys based on frequency of use, how recently used, randomly, expiration date, or a combination of these factors. The policy can also be set to noeviction to return a memory limit error when trying to insert more data.

The default eviction policy for databases is **volatile-lru** which evicts the least recently used keys out of all keys with the 'expire' field set. The default for Active-Active databases is **noeviction**.

For more information, see eviction policies.

Database persistence

Both RAM memory and flash memory are at risk of data loss if a server or process fails. Persisting your data to disk helps protect it against loss in those situations. You can configure persistence at the time of database creation, or by editing the database’s configuration.

There are two main types of persistence strategies in Redis Enterprise Software: append-only files (AoF) and snapshots.

Append-only files (AoF) keep a record of data changes and writes each change to the end of a file, allowing you to recover the dataset by replaying the writes in the append-only log.

Snapshots capture all the data as it exists in one moment in time and writes it to disk, allowing you to recover the entire dataset as it existed at that moment in time.
Redis on Flash (RoF)

By default, Redis Enterprise Software stores your data entirely in RAM for improved performance. The Redis on Flash feature enables your data to span both RAM and SSD storage (flash memory). Keys are always stored in RAM, but Redis on Flash manages the location of their values. Frequently used (hot) values are stored on RAM, but infrequently used (warm) values are moved to flash memory. This saves on expensive RAM space, which give you comparable performance at a lower cost for large datasets.

For more info, see Redis on Flash.

Shard placement

The location of the primary and replica shards on the cluster nodes can impact your database performance. Primary shards and their corresponding replica shards are always placed on separate nodes for data resiliency and high availability. The shard placement policy helps to maintain optimal performance and resiliency.

Redis Enterprise Software has two shard placement policies available:

- **dense**: puts as many shards as possible on the smallest number of nodes
- **sparse**: spread the shards across as many nodes as possible

For more info about the shard placement policy, see Shard placement policy

Metrics

From the Redis Enterprise Software admin console, you can monitor the performance of your clusters, nodes, databases, and shards with real-time metrics. You can also enable alerts for node, cluster, or database events such as high memory usage or throughput.

With the Redis Enterprise Software API, you can also integrate Redis Enterprise metrics into other monitoring environments, such as Prometheus.

For more info about monitoring with Redis Enterprise Software, see Monitoring with metrics and alerts, and Memory statistics.

Updated: April 14, 2022