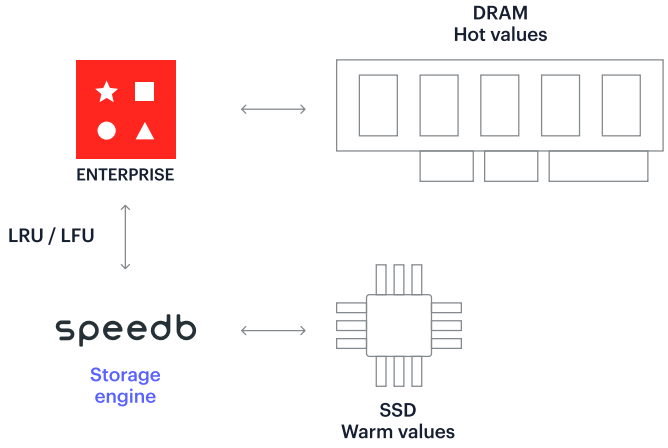
Redis Enterprise Auto Tiering

Cope with the challenges of large datasets

Real-time data platforms must deal with large volumes of data. However, DRAM is finite and expensive. As a palliative, operators want to incorporate solid state drives (SSDs) to extend their databases' available memory. But it's complicated to manage different kinds of memory in a cluster.

Auto tiering offers a better storage engine for Redis Enterprise. It extends databases beyond the limits of available DRAM using SSDs by keeping frequently used data in memory and storing less frequently used data in SSDs. As a result, Redis Enterprise can process large datasets, doubling the throughput and cutting latencies in half of previous solutions such as Redis on Flash.

Even better: Auto tiering uses the same resources and needs no code changes.



Redis Enterprise auto tiering

Redis Enterprise's auto tiering is based on a high-performance storage engine, Speedb. Speedb manages SSDs and DRAM complexity for storage management in a Redis Enterprise cluster. This implementation boosts performance up to 10K operations per second (ops/sec) per core of the database.

Workloads suitable for auto tiering

Two types of applications significantly benefit from auto tiering: applications that rely on large datasets and applications that need to turn dormant data into active data at a moment's notice.

Consider these two example scenarios.

Large volumes of data: banking industry
Banking applications retain historical data both because of regulatory requirements and also to provide features their customers demand. Banking application data regularly grows beyond available memory.

The cost of expanding DRAM significantly impacts infrastructure costs. Using SSDs with auto tiering can maintain throughput up to 10K ops/sec per core in the cluster, allowing Redis Enterprise's auto tiering to manage hot and warm data. And it does so without application changes and while controlling hardware costs.

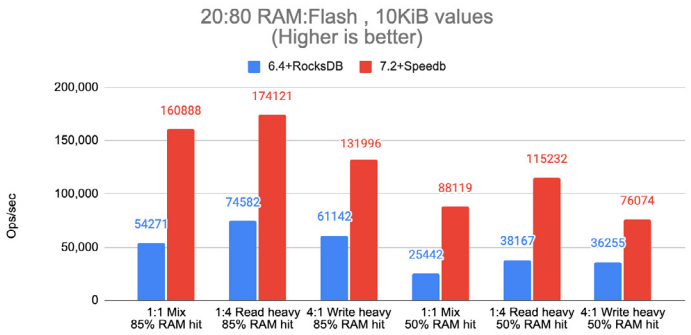
Dormant data: gaming sector
When gamers log in irregularly, their profiles consume memory that could be put to better use, such as speeding up gameplay for active users. Auto tiering can send warm data toSSDs while keeping the active game user profiles in DRAM. Data location is managed automatically, and warm data is promoted to DRAM when needed.

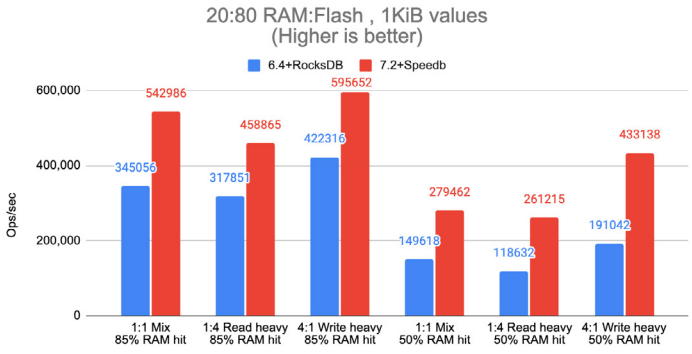
Performance comparison

The following benchmark represents three workload patterns and two different payloads. The patterns are 50% reads, and 50% writes (1:1), a read-heavy load (1:4), and a write-heavy load (4:1). In both cases, the DRAM hit rate is 50%, which means that half the time, the data was on DRAM and the other half on SSD.

1KiB objects on auto tiering			
	6.4	7.2	Differential
1:1 mix	149,618 ops/sec	279,462 ops/sec	1.9X
1:4 heavy read	118,632 ops/sec	261,215 ops/sec	2.2X
4:1 heavy write	191,042 ops/sec	433,138	2.3X

10KiB objects on auto tiering			
	6.4	7.2	Differential
1:1 mix	25,442 ops/sec	88,119 ops/sec	3.5X
1:4 heavy read	38,167 ops/sec	115,232 ops/sec	3.0X
4:1 heavy write	36,255 ops/sec	76,074	2.1X



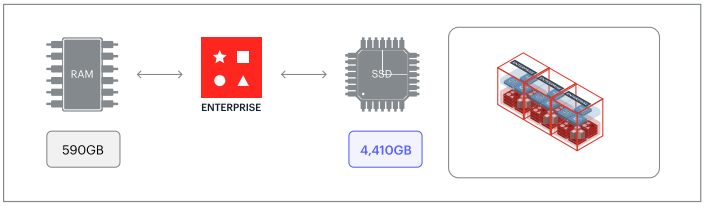


Auto tiering benefits

- Automatically access hot and warm data in a Redis Enterprise database
- Enable use cases that require large datasets
- Manage large volumes of data with up to 70% cost savings on infrastructure
- No application code changes required
- Works anywhere: cloud, on-premises, hybrid
- Simple deployment in Kubernetes with the Redis Enterprise Operator for Kubernetes
- High availability and geo-replication for large databases

Auto tiering key features

- Works on standard SSDs
- Automated tier data management
- Doubling the throughput and cutting latencies in half of previous solutions
- Throughput of up to 10K ops/sec per core
- Throughput of up to 3M ops/sec at sub-millisecond latency
- Tunable DRAM/SSD ratio for best performance or cost/benefit



You can tune the ratio of DRAM to SSD

Get Started

To learn more about auto tiering, visit the Redis Enterprise technology stack page.