WHITE PAPER

Why Your SQL Server Needs Redis

Roshan Kumar, Redis Labs

CONTENTS

SQL Server has been around for a few decades and is one of the most stable and widely used relational database management systems (RDBMS) in the market. Just like any other RDBMS, SQL Server encounters performance issues as the data it stores grows to larger volumes. SQL Server experts recommend performance tuning that includes setting up the right indices, correctly sizing TEMPDBs and ensuring maximum degrees of parallelism (MAXDOP), as well as properly configuring virtual machines (VMs), disk IO, main memory, etc.

Even with all of this fine tuning, SQL Server requires a heavy investment in the physical infrastructure in order to perform a few thousand operations per second. However, to stay ahead of the competition, most enterprises need to operate at the scale of a few million operations per second with sub millisecond latency. Today's competitive business environment also drives companies to innovate and bring new services to market faster than ever before. But they must do so without affecting any legacy applications.

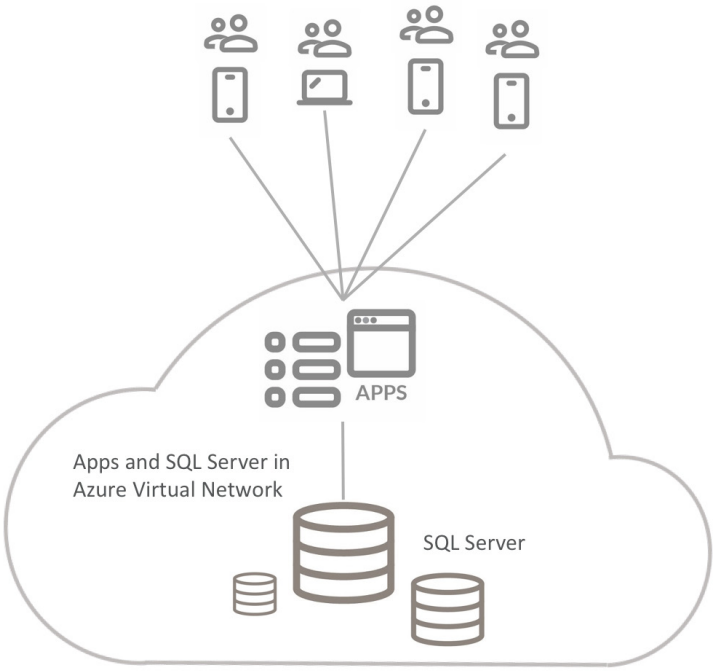

Figure 1. SQL Server has limited options to scale

How putting Redis in front of SQL Server can help

If you are innovating new solutions, the two most important questions you face are:

1. How can we scale our applications with limited effort and cost?
2. How can we introduce new features, applications and solutions faster?

Redis helps you meet both challenges. First, as an extremely lightweight, in-memory database, Redis performs many more operations per second (usually in the order of 100-1000 times more) with fewer computational resources. Second, Redis enables you to get your apps to market faster. Its built-in data structures allow you to store data in whichever format your program requires. In many cases you can avoid the burden of translating from a native data structure to a relational database. If you are using Redis as a scaling layer (such as caching), you can insert Redis between your data access layer and the database without disrupting your program stack.

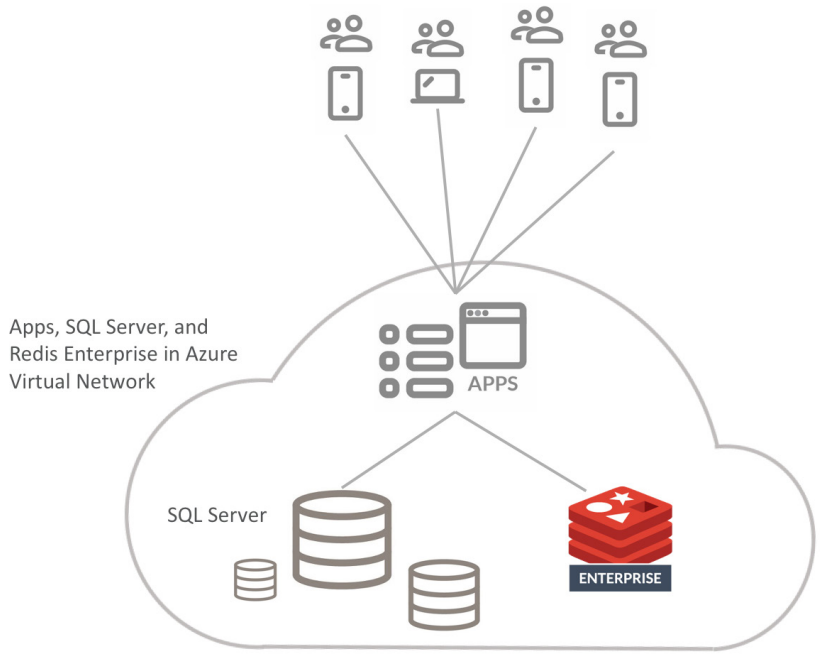

Figure 2. Redis Enterprise reduces the load on SQL Server and helps in linear scaling.

1. **Redis performance lets you do more with less**

Being an in-memory key-value data store, Redis is extremely fast and flexible. It's often called a "Swiss Army Knife" for data processing. It has built-in data structures such as Lists, Hashes, Sets, Sorted Sets, Bitmaps, Hyperloglog and Geospatial Indices that help you perform some data operations more efficiently and effectively when compared to a relational database. You can incorporate Redis into your architecture as a "system of engagement," (the system that stores the hot data users engage with) while designating your SQL Server as the "system of record" (the database that holds long-term user data). With this approach, you won't disrupt the compliance, regulatory and governance needs of your data.

2. **Redis is all pervasive, even more so with Windows and .NET**

Redis Enterprise from Redis Labs offers many options to develop applications for Redis on the Windows platform. If you are developing apps for Redis on Windows using C#, you will need a .NET Redis client. The complete list of .NET clients from redis.io can be found here. In particular, StackExchange.Redis is a popular choice among .NET developers.

Deployment Options

Redis Enterprise offers a variety of deployment options. By enabling you to deploy your Redis database closer to your application, this helps you achieve local data access latency.

Fully managed Redis Enterprise in your Azure Virtual Network (VNet)

Redis Enterprise is available as a fully managed service, which delivers cost effective, stable high performance Redis databases in dedicated clusters within your Azure VNet.

Redis Enterprise as software on your Azure server instance

Downloadable Redis Enterprise software allows you to deploy and run the Redis database in your environment of choice – an on-premises data center or your preferred public or private cloud platform – with full control of your data and configuration.

Redis use cases boost SQL Server performance

Redis is a perfect complement for SQL Server. As a system of engagement, it helps deliver the best user experience by keeping users engaged through low-latency responses. You can achieve this without your application, database or network layer creating a bottleneck. Popular use cases for adding Redis in front of SQL Server are caching, session stores, counting, metering, data streams and more.

1. Caching

Caching provides a tiered model for memory access. By storing common, repeatedly read objects in Redis, applications could retrieve data quickly and limit the load on your database server.

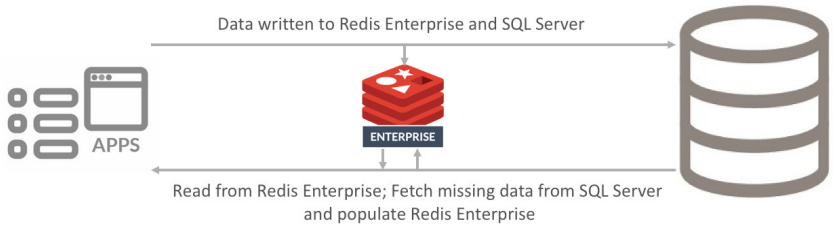

Figure 3. Redis Enterprise as a write-through cache.

When to use:

- If you have frequent reads and infrequent writes
- If your data is shared between user sessions
- Examples: images, documents, financial statements, reporting data, etc.

For more information about caching with Redis, download the white paper 15 Reasons Why Caching is Best Done With Redis.

2. Session stores

In all interactive apps, the server maintains a unique session for each active user and session objects are generally isolated from each other. Traditional designs relied on SQL Server-like relational databases to persist this data, but Redis takes session stores to the next level by enhancing the user experience with very low latency. A single cluster of Redis on decently sized servers with sufficient RAM can manage thousands, if not millions, of sessions. In addition to storing basic user information, Redis allows you to compute and serve many session metrics.

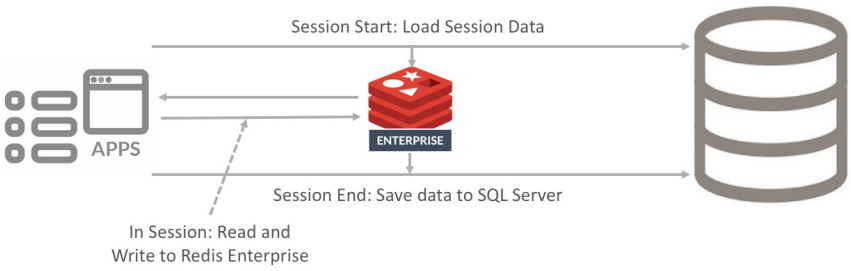

Figure 4. Redis Enterprise as a Session Store.

When to use:

• If you have frequent reads and writes

• If your data is isolated between sessions

• Examples: personalized applications, e-commerce, gaming, social applications

3. Counting and gamification

Gamification is a key component of many applications today, giving enterprises the opportunity to engage with their audiences. Leaderboards, dashboards, polls, messages, counters and other real-time aggregators require constant processing and communication with end-users. Relational databases such as SQL Server are designed to handle transactions well, but do a poor job collecting, processing and dissipating information to thousands of active users in real time. Redis' powerful and highly efficient data structures can easily support real-time gamification platforms.

When to use:

• For usage tracking, gamification, asynchronous communication

• If you need to count and track millions of simultaneous activities or objects

• Examples: social media, customer support, gaming, e-commerce, etc

4. Metering and rate-limiting calls to legacy servers

One major challenge for enterprises as they scale and add more applications is how to manage the load on their legacy servers during peak usage times. An expensive solution is to add more computational resources to legacy servers that act as a bottleneck. In cases where SQL Server is the bottleneck, you can scale out the database management system. However, this is also expensive, especially when expansion is not needed during off-peak hours. A more cost-effective way to handle the load is to rate limit the number of calls your applications make every few seconds. Redis is a popular tool for this kind of rate limiting solution. For more details, visit Redis Enterprise for Metering.

When to use:

• If your backend database cannot handle the peak load

• Examples: applications for retail, e-commerce, inventory management, etc.

Other user cases

In addition to the abovementioned use cases, Redis is popular for fast data ingest, high speed transactions, time-series data, job and queue management, messaging and real-time analytics. Redis Streams, a new data structure introduced in Redis 5.0, supports complex lifecycles for data streams with the help of consumer groups. Redis' traditional data structures, such as Lists, Sorted Sets and Pub/Sub, allow you to implement efficient, low-latency message queues in Redis. Redis Enterprise adds ACID properties to Redis with built-in consistency and durability features so you can use Redis as an ACID-compliant, in-memory database.

Embrace Redis today

Redis provides developers with all the tools they need to build highly engaging, data-centric applications. Given the data structures and highly optimized commands available in Redis, there is more than one way to leverage Redis for your applications. In addition to the examples described above, Redis is often used as a message broker, data structure store and temporary data store for a variety of use cases.

Essentially, Redis enables you to get your data closer and faster to your end user. If you flip this around, Redis also helps you collect data more quickly from your end users. With RAM's increasing affordability and the persistence features of Redis, more and more developers are using Redis as a primary database for both transactions and analytics.

Redis Enterprise takes your experience with Redis further. It offers critical enterprise capabilities, such as high availability, in-memory replication, automatic scaling, re-sharding, etc. It also delivers new features, such as CRDT-based active-active support for distributed databases and built-in Redis modules such as RediSearch, RedisJSON, Redis Bloom and Redis Graph. Redis Enterprise also offers flexible deployment options: you can deploy it on-premises, in your virtual private cloud as a fully managed service by Redis Labs, or as a hosted database-as-a-service in your public cloud environment. An investment in Redis Enterprise will not only meet immediate needs, but also make your applications future-proof.

700 E El Camino Real, Suite 250
Mountain View, CA 94040
(415) 930-9666
redislabs.com