REPORT REPRINT

At RedisConf19, Redis delivers new modules and redefines being multi-model

JUNE 20 2019

James Curtis

The company made a number of key product announcements at RedisConf19, introducing several new modules that extend and refine its multi-model capabilities to provide inter-model communication, as well as the ability to programmatically operate on the modules.

451 Research®

Introduction

At its latest annual conference – RedisConf19 – Redis made a number of key product announcements. A majority of the announcements consisted of new modules the company plans to roll out. Specific modules include RedisTimeSeries, RedisAI and RedisGears – the latter functioning as a serverless engine to enable multi-model operations. The company also announced RedisEdge and the availability of Redis compatibility with Intel's Optane DC persistent memory.

451 TAKE

Providing a multi-model database is not necessarily new positioning for the company. What is new, however, is that with the new modules, Redis' multi-model positioning is much more refined, offering inter-model communication as well as the ability to programmatically operate on the modules. Given these updates, we would expect new and existing customers to take notice of the new capabilities as a means to extend Redis. The module strategy is proving to be quite beneficial for Redis because it allows for a good deal of flexibility, given that modules can be independently developed. The three new modules announced, as well as other modules comprising data models, are licensed under the Redis Source Available License (RSAL). While that could give some open source purists pause, the belief is that the benefits the modules provide for multi-model operations will likely outweigh any concerns on the licensing front.

Context

In early 2016, Redis introduced its modules strategy. Redis Modules are best thought of as customized, prebuilt functionality. Extending Redis' capabilities has always been a core approach for the company, such as with Lua scripting and with new data structures, so Redis Modules fit well within that strategy. When Redis introduced modules, the idea was that modules would open the door for the company to evolve as a multi-model NoSQL database, as well as enable other functionality.

That has certainly occurred. The company provides its Redis Modules Hub, which consists of a number of Redis enterprise modules and certified modules that work with specific versions of Redis Enterprise, and are available under the RSAL. A collection of other Redis modules, offered under various open source licenses, are likewise available, although not certified by Redis.

Also noteworthy is the company's securing of an additional $60m in a series E round. The round was headed by Francisco Partners and included participation from existing investors Goldman Sachs, Bain Capital Ventures, Viola Ventures and Dell Technologies Capital, bringing the company's total funding to $146m.

Product and announcements

Although Redis announced a number of modules at RedisConf19, the overriding message was around positioning Redis as a multi-model database. Redis' path to multi-model has been in the works for some time, beginning in 2016 when it rolled out its modules strategy. Since introducing modules, however, the company has developed and released a collection of modules that enable other data types, and now include probabilistic data, search, streaming data, document (JSON) and graph. Further, the company announced two data modules: time series and AI.

The first of the new data modules was RedisTimeSeries. Time series workloads are growing fast, so coming to market with a native data model in the form of a Redis module was the preferred strategy. Redis customers could do time series workloads previously by leveraging sorted sets and hashes and with Redis Streams, but these methods lacked certain functionality such as downsampling, aggregation, compaction, labeling, retention and compression. The TimeSeries module addresses these shortcomings by providing a streamlined way of working with time series data as a native data model inside Redis.

Another module that was introduced was RedisAI. Similar to the TimeSeries module, Redis is looking to not only simplify the task of training and serving AI/ML models, but to do so in a production environment. Redis could handle AI models before by leveraging existing modules that included Redis-ML, redis-tf and neural-redis, but these modules were either targeted at specific tasks or lacked some functionality. Redis-ML, for instance, is able to serve up ML models that were trained on other ML-based platforms.

Conversely, the new RedisAI module was jointly developed with Tensorworks. It is able to serve up ML and DL-type models within Redis, and to do so using known frameworks such as PyTorch and TensorFlow, with other frameworks coming soon such as ONNX. As part of this module, a new Redis data structure was developed called Tensor that effectively streamlines the effort in working with the ML and DL models within Redis.

With these new data models, Redis executives laid out their approach to a multi-model database, highlighting three techniques the company considers critical for its multi-model database. One technique consists of inter-model communication, which includes the ability for the modules to 'auto discover' other modules. As an example, Redis announced that RedisGraph now uses RedisSearch for graph traversal and queries.

Another technique promoted was the capability of Redis to leverage a single copy of the data between the cores and the Redis modules. The impact is that the data does not need to be copied between the modules, and can lead to new functionality such as active-active replication for RedisSearch, which was likewise announced as the conference.

The third technique is the ability to do in-database multi-model programming. To enable this, the company announced a third module called RedisGears. RedisGears serves a different purpose than RedisTimeSeries and RedisAI, which offer new data models. RedisGears is a serverless engine for carrying out multi-model operations within Redis. The whole premise behind RedisGears is to enable the building of operational pipes, which consist of data inputs and outputs to create operations within Redis.

A few examples include the ability to count messages that land in a time series database from Redis Streams. Another consists of taking existing data in Redis and indexing that data, then creating relationships between Redis objects using RedisGraph. For now, RedisGears supports Python, although support for other languages is forthcoming.

Other announcements included RedisEdge, a scaled-down but purpose-built Redis database for IoT environments. RedisEdge comes packaged with open source Redis, Redis Streams, RedisTimeSeries, RedisAI and RedisGears. The company also announced the availability of support for Intel Optane DC persistent memory (initially announced in 2018, which provides greater scalability and cost efficiencies for Redis deployments.

Competition

Redis has a number of competitors within the NoSQL market, as well as in the broader database market. The fact that Redis is positioning more deliberately as a multi-model database likely won't change the competitive landscape, although it could change how enterprises choose a database.

On the NoSQL front, particularly with multi-model databases, the company is likely to see MongoDB, DataStax, MarkLogic, OrientDB and ArangoDB, all of which offer multiple data models, including graph capabilities. Couchbase is another direct competitor with multiple data models, and, it should be noted, also gets leveraged as a caching layer.

Because Redis gets deployed as a caching layer, it is not uncommon to see it being paired with other databases, including relational databases in some circumstances, in which case it could be compared with in-memory data cache and database functionality from the likes of GridGain, GigaSpaces, Pivotal, Hazelcast and ScaleOut Software.

Cloud-only multi-model NoSQL databases are another competitive field for Redis, and include Microsoft Azure Cosmos DB and Amazon DynamoDB. There are, of course, a number of single model cloud NoSQL databases such as Azure Cache for Redis, Azure Table Storage, Amazon DocumentDB, Amazon ElasticCache for Redis, Amazon Neptune (graph, Google Cloud Bigtable, Google Cloud Datastore, IBM Cloudant and IBM Compose, just to name few.

There are also a handful of single-model NoSQL vendors (which provide cloud equivalents) that Redis might see, including Aerospike (key-value), Oracle NoSQL Database (key-value), Neo4j (graph) and TigerGraph (graph).

SWOT Analysis

STRENGTHS

Modules are a differentiator for Redis, and give the company a good deal of flexibility, particularly with respect to providing inter-model communication and programmability between the models, thus delivering a multi-model database.

WEAKNESSES

While the new modules bring newly added functionality, some are still maturing, such as forthcoming compression for RedisTimeSeries and additional language support for RedisGears, which currently only supports Python.

OPPORTUNITIES

Redis can build on a land-and-expand strategy. For new and existing Redis customers, however, the new modules provide additional use-case scenarios for which Redis can be applied, particularly around graph and search, where the company has focused a good deal of its development efforts recently.

THREATS

For the most part, providing multiple data models has become fairly commonplace among NoSQL vendors. The challenge, however, lies in how those models are fundamentally integrated and work together at the database level, and the benefits they provide, which aren't always apparent and can be hard to decipher for interested buyers.