Microsoft Cosmos DB: The new flagship internet database of Azure

Licensed Reprint

Publication Date: 12 Jun 2017 | Product code: IT0014-003285

Tony Baer

Ovum view

Summary

At its Build conference in May, Microsoft took the wraps off Cosmos DB, the new incarnation of its existing cloud-based Azure DocumentDB NoSQL database. With a nod to the dramatic, Microsoft terms Cosmos DB as its biggest database bet since SQL Server; it is positioning it as its flagship cloud database, suited for use cases ranging from security and fraud detection, to IoT (consumer and industrial), personalization, e-commerce, gaming, social networks, chats, messaging, bots, oil and gas recovery and refining, and smart utility grids. Cosmos DB is a good example of how cloud platform providers are rethinking databases for scalable, elastic environments and commodity infrastructure. The platform that is most comparable is Google Cloud Spanner, but each of these databases is engineered for different purposes: Cosmos DB as a globally distributed operational database and Spanner as a globally distributed SQL-supporting OLTP database.

The highlights of Cosmos DB include its flexibility in supporting multiple data models; an elastic scale-out architecture that supports globally distributed multiregion deployments with guaranteed low latency and four 9s availability; and a choice of multiple, defined consistency models. Cosmos DB is a flexible database that can be made to look and act like users want; for instance, it could be the globally distributed cloud storage engine of a MongoDB document or a graph database that supports the Gremlin language of popular Apache TinkerPop framework. While Cosmos DB is hardly unique in tapping into the cloud-native wave, it is the first to open up this architecture to data that is not restricted by any specific schema and it is among the most flexible when it comes to specifying consistency.

Beyond the checkboxes

As enterprises look to cloud, not just for tactical purposes such as DevTest or the running of standalone new apps, they expect cloud providers to offer a range of the "usual suspect" platforms that are a de facto checklist: an enterprise-grade relational transaction (OLTP) database; enterprise-grade relational data warehouse; a choice of key/value, document (JSON), and graph NoSQL data stores; and some form of big data/Hadoop/Spark service. When it was introduced two years ago, Microsoft Azure DocumentDB checked off the NoSQL JSON-style document database box.

For cloud providers, these checkboxes were about providing their own managed alternatives to on-premises databases/data platforms. More recently, cloud providers have started to add new services to extend their data platforms beyond the checklist requirements to use cases that leverage several unique capabilities of the cloud multitenant environment: inexpensive storage, elastic computing, and multiregion deployment. For instance, Azure SQL Database and Amazon Aurora exploit the economics of scale in the cloud to take new approaches to fault tolerance. Cloud providers are also offering the ability to collapse database architectures with direct query from cloud object storage; while Microsoft PolyBase exploited this capability early on, more recently, Amazon has come with its own answers such as Amazon Athena and Amazon Redshift Spectrum. And then came Google Cloud Spanner, which delivers a global OLTP database that supports SQL and scales to trillions of rows with a unique approach to ACID.

With Cosmos DB, now it's Microsoft's turn. Cosmos DB is Microsoft's multitenant, globally distributed database natively designed for the cloud. As one of the core services of Azure, Cosmos DB is live in all Azure regions, currently managing hundreds of petabytes of indexed data, and serving hundreds of trillions of requests every day from thousands of customers worldwide.

Cosmos DB is not a brand-new product per se, but a very significant expansion of the existing Azure DocumentDB NoSQL cloud database. When DocumentDB was introduced two years ago, it was the initial result of Microsoft's Project Florence, which began back in 2010 as an initiative to deliver an internet-scaled database. So it's not just a rebranding or example of "markitecture." Cosmos DB potentially provides the flexibility of ingesting just about any data model by virtue of being a schema-agnostic data store with a single logical namespace and the multiple options for database consistency. While some of these features are not necessarily unique to Cosmos DB, the combination and the opportunity to fill in the gaps provides Microsoft the opportunity of delivering a globally distributed database whose schema and performance can be tailored to the application.

Seeking the global elastic footprint

The biggest innovation that cloud brought to databases is the ability to spin up compute on demand (elastic compute); and its ability to scale across multiple data centers inside, and more importantly, outside a region and do so almost instantaneously. The economics of cloud are also very much driven by multitenant architectures and fine-grained resource governance that can optimize use of shared commodity infrastructure.

The prime beneficiaries of this have been data warehouses and analytics that do not have always-on loads. The same goes with global footprint, as it is far more straightforward to run analytics in massively distributed (shared-nothing) fashion because there is no need for maintaining consistency across database nodes.

The ability to span multiple regions is not unusual for cloud databases as they take advantage of the economies of scale that global cloud infrastructure provides; the same goes with horizontal sharding, which platforms like Amazon Aurora provide. However, the difference is that most existing cloud-native database platforms utilize the global footprint for automatic replication and failover purposes.

This is where Cosmos DB distinguishes itself. Through a combination of features, including automatic horizontal partitioning (sharding) and configurable consistency settings, Cosmos DB can operate as a single, global, logical *operational* database instance spread across multiple regions that also exploits the economics of elastic computing and multitenancy. Google Cloud Spanner has similar global aspirations, although it is not yet available for multiregion deployment. But as we noted above (and in our research), Cloud Spanner is optimized to run as a global transaction database with a specific ACID model, while Cosmos DB is a multimodel database that offers a choice of five consistency models, which we will discuss below.

Cosmos DB's secret sauce

As noted above, the key to Cosmos DB's horizontal scaling and elasticity as an operational database is attributable to its approach to resource management, partitioning, containerization, and consistency options, plus its schema-agnostic engine. Cosmos DB sits on automatically indexed SSD Flash storage, which we believe will become a de facto standard for cloud NoSQL databases (SSD is standard for Amazon DynamoDB, but still an option for Google Cloud Datastore). A single Cosmos DB table can scale from gigabytes to petabytes across multiple machines and regions.

Use of resources is managed through containers that encapsulate database functions such as stored procedures, triggers, and user-defined functions (UDFs). These containers act on atomic database records that can be represented through a variety of models; they are accessed using an innovative indexing scheme that originated with DocumentDB, and that has been extended through Cosmos DB's release of new APIs, making the multimodel capability reality. The index engine automatically tracks every path in the document tree. Cosmos DB supports policy-based tiering of colder data to HDFS-compatible Azure Data Lake or to Azure Blob storage; customers pay for throughput and storage.

Multimodel support comes through APIs that expose these atomic records in different forms, including JSON (through the existing DocumentDB API and the newly released MongoDB APIs); graph (via the Gremlin API); key/value (through Azure Table Storage API); and SQL (via the existing DocumentDB API). Note that while users can run a subset of SQL functions in Cosmos DB, it is not intended as a replacement for Azure SQL Database or Azure SQL Data Warehouse. As long as the data being migrated comes from a source database supporting these APIs, users should not have to recompile their data when moving to Cosmos DB.

Resource management and service levels are a function of the system's automated partitioning (which can be tweaked by the customer); data locality (e.g., directing data to the source that is physically closest and available); and configuration of consistency settings (more about that below). Each partition presents a single system image, with elasticity managed based on traffic patterns to partitions in different regions. Microsoft guarantees that Cosmos DB will deliver four 9s availability within a region.

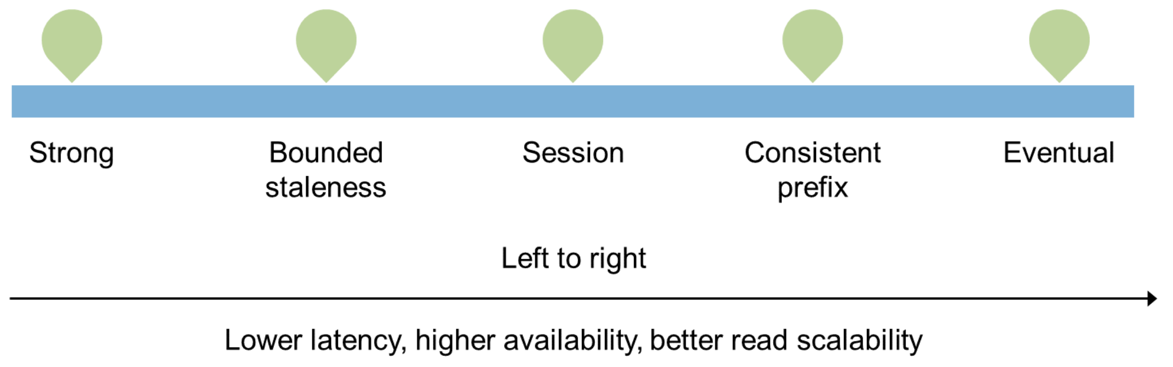
Cosmos DB's management of consistency is a key differentiator versus Amazon and Google Cloud. The others offer choices of strong consistency (which promises that all instances will have the same version of the data, but may have latency and availability issues with write-locks) and eventual consistency (which is better suited for large-scale deployments that require low latency and high availability). The only other globally distributed database, Cloud Spanner, supports an immediate consistency model based on its global time clock, for which Google promises 10msec resolution time. By contrast, Cosmos DB offers single-digit millisecond latency guarantees with five consistency models that can be imagined as a spectrum (see Figure 1):

- **Strong consistency,** the most stringent level of consistency that is typically associated with ACID databases. While providing a picture that, as the term states, is "consistent" across all nodes, strong consistency requires some form of locking when specific record(s) are updated. Update performance is the slowest compared to other consistency approaches.
- **Bounded staleness,** useful for PubSub applications, which allows reads of "stale" records only within a certain number of versions or time interval. It provides stronger consistency compared to session or eventual consistency.
- **Session,** where updates within a specific user session are updated instantly. This is useful for applications involving exchanges on social media.
- **Consistent prefix,** where updates are committed in the exact same sequence in which they were entered. This is useful for applications involving social media interactions, event-based

scenarios (tracking security intrusions), or some forms of IoT scenarios where it is essential to track the trajectory of device performance.

- **Eventual consistency,** which is the loosest form of consistency. Here, updates may be committed in any order, but this approach offers the lowest latency (highest performance) between reads and writes.

Figure 1: Cosmos DB consistency levels



Source: Microsoft

A shot across the bow

Coming out of the gate, Cosmos DB is in full release across all 40 Microsoft Azure regions worldwide, and has seen use through preview customers, such as the dozen or so whose logos appear on the Azure Cosmos DB home page. In actuality, there are thousands of customers if you count existing DocumentDB accounts that have been automatically upgraded to the new platform.

While we don't expect Cosmos DB to become a fully SQL ACID database, we expect that future releases will increase SQL functionality. And while it is not positioned as an analytic platform, integration with Spark (which has become a checkbox item for NoSQL databases) provides hints of how the platform could be harnessed for analytics. In the long run, NoSQL data stores with Spark connectivity will give Hadoop a run for its money. With its varied consistency options, there is also headroom for growth in accommodating real-time streaming data; for instance, the bounded staleness consistency setting sounds like a perfect match for integration with Apache Kafka. Azure Cosmos DB's change-feed capability (a form of change data capture) already supports lambda pipelines on Azure, where it can ingest data and feed changes to downstream processes. It could provide a similar capability in feeding Apache Kafka.

In its scale and mutability, Cosmos DB has upped the ante in cloud database options. The flexibility to represent data with almost any model and the ability to tune consistency will make it attractive to developers of applications requiring internet scale. Of course, there is always the question of whether a single database can be all things to all people. There continue to be uses for data platforms that are designed for specific functions such as data warehousing or OLTP. As noted, Azure customers will likely not use Cosmos DB to replace their Azure SQL database or Azure SQL data warehouse. But scalable internet applications are breaking down the boundaries of transactions and analytics; batch processing and real time; and structured and variably structured data. These are the new design targets that cloud-native databases are targeting. There remains plenty of white space here; the fact that Cosmos DB and Google Cloud Spanner are both globally distributed databases and yet so different indicates the potential variety of choices that will be coming from cloud providers. Now it's Amazon's turn to step up.

Appendix

Further reading

"Google Cloud Spanner differentiates the database portfolio," IT0014-003228 (February 2017)

Microsoft SQL Server 2016: An Initial Assessment, IT0014-003125 (June 2016)

"Microsoft Azure Data Lake takes big step in taming big data," IT0014-003078 (November 2015)

"Amazon's broader database footprint ratchets up the Oracle rivalry," IT0014-003195 (December 2016)

Author

Tony Baer, Principal Analyst, Information Management

tony.baer@ovum.com

Ovum Consulting

We hope that this analysis will help you make informed and imaginative business decisions. If you have further requirements, Ovum's consulting team may be able to help you. For more information about Ovum's consulting capabilities, please contact us directly at consulting@ovum.com.

Copyright notice and disclaimer

Whilst reasonable efforts have been made to ensure that the information and content of this product was correct as at the date of first publication, neither Informa Telecoms and Media Limited nor any person engaged or employed by Informa Telecoms and Media Limited accepts any liability for any errors, omissions or other inaccuracies. Readers should independently verify any facts and figures as no liability can be accepted in this regard – readers assume full responsibility and risk accordingly for their use of such information and content.

Any views and/or opinions expressed in this product by individual authors or contributors are their personal views and/or opinions and do not necessarily reflect the views and/or opinions of Informa Telecoms and Media Limited.

Ovum

TMT intelligence | informa

CONTACT US

www.ovum.com

analystsupport@ovum.com

INTERNATIONAL OFFICES

Beijing

Dubai

Hong Kong

Hyderabad

Johannesburg

London

Melbourne

New York

San Francisco

Sao Paulo

Tokyo