Department of Informatics

Technical University Munich

Bachelor's Seminar Paper

Assessment of Warm and Cold Storages for Azure Internet of Things

Minh Vu

Table of Contents

I. Introdution to Azure Internet of Things

The Azure Internet of Things (IoT) is a collection of Microsoft-managed cloud services that connect, monitor, and control billions of IoT assets. As described by Microsoft, by leveraging these services and connecting the intelligent devices, the users can transform businesses and enable new growth opportunities. Real-life IoT use cases include the manufacturing, retail, banking, governmental and healthcare sectors. Despite the massive potential, there have been significant worries about potential drawbacks, such as security loopholes, scaling issues and slow start-up time. Microsoft has been attempting to mitigate these issues by introducing a secured and robust system.

1. An Overview of Azure IoT

Things - Insights - Actions, these are the three pillars of IoT applications. IoT applications can be described as **Things** (or devices), sending data or events that are used to generate **Insights**, which are used to generate **Actions** to help improve a business or process. An example is a sensor (a thing), sending pressure and temperature data that is used to evaluate whether a machine is performing as expected (an insight), which is further used to tweak the parameters (an action).

An IoT application consists of the following subsystems:

1. Devices, or on-premises edge gateways, which are a specific kind of device that can securely register message sources (devices) with the cloud. The edge gateway may also transform messages from a native protocol to another format (such as JSON).
2. A cloud gateway service, or hub (such as Azure IoT Hub or Azure Event Hubs), to securely ingest data and provide device management capabilities.
3. Stream processors that consume streaming data. The processors may also integrate with business processes and place the data into storage.
4. A user interface, in the form of a dashboard, to visualize IoT data and facilitate device management.

These are the major steps of creating new insights:

1. Access the data and process it into a data stream.
2. Process and store the data.
3. Visualize or present the data.

The figure below is a diagram that shows the flow of data, from data source, to convert, to ingestion, to process and store, to presentation, and finally action.

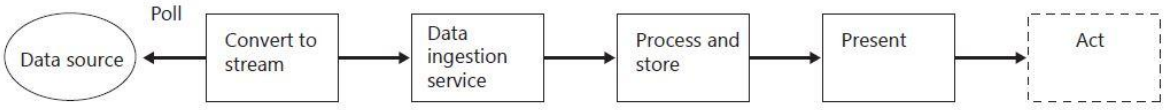

Figure 1: The flow of data from the data source to action.

Moreover, the IoT applications can inevitably introduce further challenges, as they are event-driven systems, that also need to keep and operate on historical data. The incoming data is an append type of data and can potentially grow large. There is a need to keep data for longer periods, primarily for these reasons: archive, batch analytics, and to build machine

learning models. On the other hand, the event stream is crucial for analyzing in near real-time to detect anomalies, recognize patterns over rolling time windows, or triggering alerts, if values go over or below a threshold.
Microsoft's Azure IoT Reference Architecture presents a recommended data flow for device to cloud messages and events in an IoT solution using Lambda architecture. The Lambda architecture enables the analysis of both near real-time, streaming data, as well as archived data, which makes it the best option for processing of the incoming data.

2. Azure IoT Lambda Architecture

The Lambda architecture addresses this problem by creating two paths for data flow. All data coming into the system goes through these two paths:

- A batch layer (cold path) stores all the incoming data in its raw form and performs batch processing on the data. The result of this processing is stored as a batch view. It is a slow processing pipeline executing complex analysis, for example combining data from multiple sources and over a longer period (hours, days, or longer), and generating new information such as reports, machine learning models or visualizations.
- A speed layer (warm path) analyzes data in real time. This layer is designed for low latency, at the expense of accuracy. It is a faster processing pipeline that archives and displays incoming messages, and analyzes these records generating short term critical information and actions such as alarms.
- The batch layer feeds into a "serving layer," which responds to queries. The batch layer indexes the batch view for efficient querying. The speed layer updates the serving layer with incremental updates based on the most recent data.

The warm path is where the stream processing occurs to discover patterns over time. However we also would like to calculate the utilization over a period of time in the past, with different pivots, and aggregations, such as production or efficiency. We want to merge those results with the warm path results to present a unified view to the user.
The cold path includes the batch layer and the serving layers. The combination provides a long-term view of the system. The cold path contains the long-term data store for the solution. It also contains the batch layer, which creates pre-calculated aggregate views to provide fast query responses over long periods of time. The technology options available for this layer on Azure platform is really diverse.

The following image shows five blocks that represent stages of transformation. The first block is the data stream, which feeds both the speed layer and batch layer in parallel. Both layers feed the serving layer, The speed layer and the serving layer both feed the analytics client.

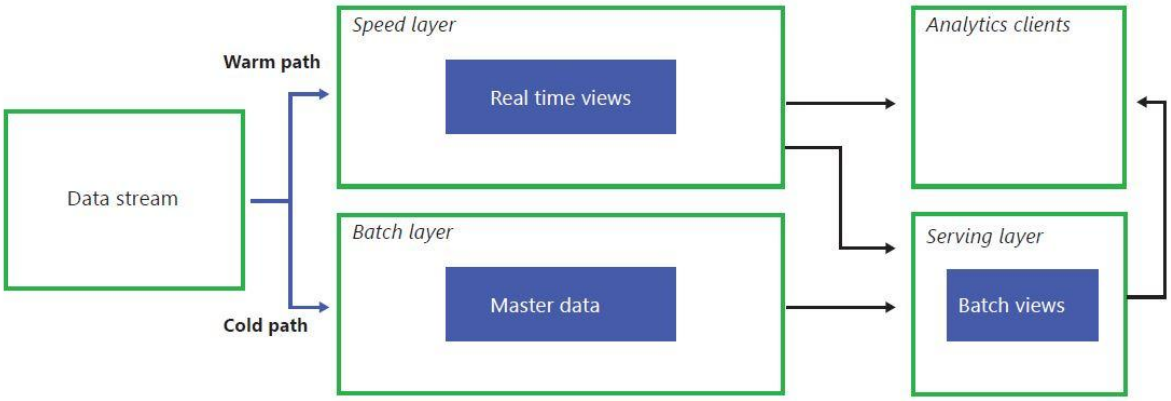

Figure 2: How the Lambda Architecture creates two seperate pathways for two different purposes

3. Storage Overview

Definition. IoT solutions can generate significant amounts of data depending on how many devices are in the solution, how often they send data, and the size of payload in the data records sent from devices. Data is often time series data and is required to be stored where it can be used in visualization and reporting as well as later accessed for additional processing. It is common to have data split into “warm” and "cold" data stores. The warm data store holds recent data that needs to be accessed with low latency. Data stored in cold storage is typically historical data. Most often the cold storage database solution chosen will be cheaper in cost but offer fewer query and reporting features than the warm database solution.

A common implementation for storage is to keep a recent range (e.g. the last day, week, or month) of telemetry data in warm storage and to store historical data in cold storage. With this implementation, the application has access to the most recent data and can quickly observe recent telemetry data and trends. Retrieving historical information for devices can be accomplished using cold storage, generally with higher latency than if the data were in warm storage. For general purpose scenarios Azure Cosmos DB is recommended for warm storage and Azure Blob Storage is recommended for cold storage. If the solution requires frequent queries that involve aggregation across many events and across many devices, Time Series Insights is recommended for warm storage. See below for a detailed description of warm and cold storage and a deep dive into the technology evaluation performed for storage technology in each category.

II. Technology Options for Warm Storage

A warm storage database stores device state for a pre-determined recent interval and may also store an easily accessible last known state per device. This data must be available in the database quickly (ideally within a matter of seconds from when the data is ingested into the cloud gateway from the device) and easily queried for simple scenarios such as visualizing current device sensor values or visualizing values over a recent timeframe. Common query patterns include: data for a device for a recent date and time range, aggregated data for one or many devices, and the last known value for a telemetry point for a specific device. The data stored in the warm database may be raw data, aggregated data, or both.

If a solution has a high rate of ingestion (on the order of many hundreds of thousands or millions of messages per second), a specialized high ingestion database may be required. The recommendations for high ingestion databases are forthcoming.

Evaluation Criteria

Warm storage solutions were evaluated based on the following criteria. These criteria will not apply for every IoT solution but were designed to be most generally applicable across IoT solutions.

- **Security**. The solution offers mature and robust features such as encryption at rest, authentication and authorization, and network security.
- **Simplicity**. The solution is well documented and has a well-defined architecture. Development tasks are documented, supported by software development kits ("SDKs"), and are testable in a local development environment. Deployment and operational tasks are supported by documentation, tools, and user interfaces.
- **Performance**. Reading and writing to the database is fast and scales to many concurrent reads and writes. Query performance is also fast.
- **Scalability**. The database supports storing gigabytes to terabytes of data. Scaling out does not require downtime. The ideal solution automatically adapts cost and computing power to the load provided.
- **Query Capability**. The database has the query capabilities necessary for the overall solution.
- **Price**. The database is affordable for both storage capacity and throughput needs.

1. Azure Cosmos Database

Azure Cosmos Database(DB) is recommended by Microsoft as a general purpose warm storage solution.

Azure Cosmos DB is a secure, highly scalable (no limits on data storage or throughput), low latency NoSQL database. It is best for datasets that can benefit from flexible, schema-agnostic, automatic indexing, and rich query interfaces. Azure Cosmos DB has 5 API types and data models—SQL, MongoDB, Graph, Table and Cassandra, which provide the flexibility to choose a data model based on the data needs of the solution. Cosmos DB allows multi-region read and write and supports manual failover in addition to automatic failover. In addition, Cosmos DB allows the user to set a time-to-live (TTL) on their data, which makes expiring old data automatic. Pricing is based on storage used and Request Units provisioned. Cosmos DB is best for scenarios that do not require queries involving aggregation over large sets of data across many devices, as those queries require more Request Units than a basic query such as the last event for a device.

Key Benefits

- **Turnkey global distribution:** Cosmos DB enables you to build highly responsive and highly available applications worldwide. Cosmos DB transparently replicates your data wherever your users are, so your users can interact with a replica of the data that is closest to them. Cosmos DB allows you to add or remove any of the Azure regions to your Cosmos account at any time, with a click of a button. Cosmos DB will seamlessly replicate your data to all the regions associated with your Cosmos account while your application continues to be highly available, thanks to the multi-homing capabilities of the service. For more information, see the global distribution article.
- **High Availability:** By virtue of deep integration with Azure infrastructure and transparent multi-master replication, Cosmos DB provides 99.999% high availability for both reads and writes. Cosmos DB also provides you with the ability to programmatically (or via Portal) invoke the regional failover of your Cosmos account.

This capability helps ensure that your application is designed to failover in the case of regional disaster.

- **Elastic scalability of throughput and storage, worldwide:** Designed with transparent horizontal partitioning and multi-master replication, Cosmos DB offers unprecedented elastic scalability for your writes and reads, all around the globe. You can elastically scale up from thousands to hundreds of millions of requests/sec around the globe, with a single API call and pay only for the throughput (and storage) you need. This capability helps you to deal with unexpected spikes in your workloads without having to over-provision for the peak. For more information, see partitioning in Cosmos DB, provisioned throughput on containers and databases, and scaling provisioned throughput globally.
- **Guaranteed low latency at 99th percentile, worldwide**: Using Cosmos DB, you can build highly responsive, planet scale applications. With its novel multi-master replication protocol and latch-free and write-optimized database engine, Cosmos DB guarantees less than 10-ms latencies for both, reads (indexed) and writes at the 99th percentile, all around the world. This capability enables sustained ingestion of data and blazing-fast queries for highly responsive apps.
- **Precisely defined, multiple consistency choices:** When building globally distributed applications in Cosmos DB, you no longer have to make extreme tradeoffs between consistency, availability, latency, and throughput. Cosmos DB's multi-master replication protocol is carefully designed to offer five well-defined consistency choices - strong, bounded staleness, session, consistent prefix, and eventual — for an intuitive programming model with low latency and high availability for your globally distributed application.
- **No schema or index management:** Keeping database schema and indexes in-sync with an application's schema is especially painful for globally distributed apps. With Cosmos DB, you do not need to deal with schema or index management. The database engine is fully schema-agnostic. Since no schema and index management is required, you also don't have to worry about application downtime while migrating schemas. Cosmos DB automatically indexes all data and serves queries fast
- **Ubiquitous regional presence:** Cosmos DB is available in all Azure regions worldwide, including 54+ regions in public cloud, Azure China 21Vianet, Azure Germany, Azure Government, and Azure Government for Department of Defense (DoD). See Cosmos DB's regional presence.
- **Secure by default and enterprise ready:** Cosmos DB is certified for a wide array of compliance standards. Additionally, all data in Cosmos DB is encrypted at rest and in motion. Cosmos DB provides row level authorization and adheres to strict security standards.
- **Significant cost savings:** Since Cosmos DB is a fully managed service, you no longer need to manage and operate complex multi datacenter deployments and upgrades of your database software, pay for the support, licensing, or operations or have to provision your database for the peak workload. For more information, see Optimize cost with Cosmos DB..
- **Industry leading comprehensive SLAs:** Cosmos DB is the first and only service to offer industry-leading comprehensive SLAs encompassing 99.999% high availability, read and write latency at the 99th percentile, guaranteed throughput, and consistency.

- **Globally distributed operational analytics and AI with natively built-in Apache Spark:** You can run Spark directly on data stored in Cosmos DB. This capability allows you to do low-latency, operational analytics at global scale without impacting transactional workloads operating directly against Cosmos DB. For more information, see Globally distributed operational analytics.

2. Azure SQL Database

Azure SQL Database(DB) is best for datasets that require relational storage and query capabilities. Azure SQL DB also provides advanced features for data management, protection and security, and business continuity. Pricing is based on a combination of storage provisioned and Database Transaction Units or elastic Database Transaction Units provisioned. Manual scaling to increase storage space has no downtime. SQL DB also has built-in replication and automatic region failover to ensure data is not lost in an outage. Nonetheless, the downsides of Azure SQL DB include the requirement of relational storage, the need to manually scale the database, and the limits on scale and throughput for write ingest.

Key Benefits:

- **Scalable Performance and Pools:** With single databases, each database is isolated from others and is portable. Each has its own guaranteed amount of compute, memory, and storage resources. The amount of the resources assigned to the database is dedicated to that database, and isn't shared with other databases in Azure. You can dynamically scale single database resources up and down. The single database option provides different compute, memory, and storage resources for different needs. For example, you can get 1 to 80 vCores, or 32 GB to 4 TB. The hyperscale service tier for single databases enables you to scale to 100 TB, with fast backup and restore capabilities. With elastic pools, you can assign resources that are shared by all databases in the pool. You can create a new database, or move the existing single databases into a resource pool to maximize the use of resources and save money. This option also gives you the ability to dynamically scale elastic pool resources up and down.
- **Elastic pools to maximize resource utilization:** For many businesses and applications, being able to create single databases and dial performance up or down on demand is enough, especially if usage patterns are relatively predictable. Unpredictable usage patterns can make it hard to manage costs and your business model. Elastic pools are designed to solve this problem. It is possible to allocate performance resources to a pool rather than an individual database. You pay for the collective performance resources of the pool rather than for single database performance.
- **Extensive monitoring and alerting capabilities:** Azure SQL Database provides advanced monitoring and troubleshooting features that help you get deeper insights into workload characteristics. These features and tools include:
    - The built-in monitoring capabilities provided by the latest version of the SQL Server database engine. They enable you to find real-time performance insights.
    - PaaS monitoring capabilities provided by Azure that enable you to monitor and troubleshoot a large number of database instances.
- **Availability capabilities:** Azure SQL Database enables your business to continue operating during disruptions. In a traditional SQL Server environment, you generally have at least two machines locally set up. These machines have exact, synchronously

maintained, copies of the data to protect against a failure of a single machine or component. This environment provides high availability, but it doesn't protect against a natural disaster destroying your datacenter.

- **Built-in intelligence:** With SQL Database, you get built-in intelligence that helps you dramatically reduce the costs of running and managing databases, and that maximizes both performance and security of your application. Running millions of customer workloads around the clock, SQL Database collects and processes a massive amount of telemetry data, while also fully respecting customer privacy. Various algorithms continuously evaluate the telemetry data so that the service can learn and adapt with your application.
- **Advanced security and compliance:** SQL Database provides a range of built-in security and compliance features to help your application meet various security and compliance requirements.

3. Azure Time Series Insights

Azure Time Series Insights(TSI) is an analytics, storage and visualization service for time series data, providing capabilities including SQL-like filtering and aggregation, alleviating the need for user-defined functions. All data in Azure TSI is stored in-memory and in SSDs, which ensures data is quickly ready for interactive analytics. Azure TSI also provides visualizations such as overlays of different time series, dashboard comparisons, accessible tabular views, and heat maps. Azure TSI provides a data explorer to visualize and query data as well as REST Query APIs. Further, it exposes a JavaScript controls library that enables embedding time series charts into custom applications. Azure TSI is suited for solutions that need visualization services built in and do not need to report on data immediately (TSI has an approximate latency for querying data records of 30-60 seconds). TSI is well suited for solutions that need to query aggregates over large sets of data, as TSI allows any number of users to conduct an unlimited number of queries for no extra cost. Currently, TSI has a maximum retention of 400 days and a maximum storage limit of 3 TB, so a solution using TSI will need to use a cold storage database (likely swapping data into TSI for querying as needed) as well if the customer needs a larger amount of storage or longer retention. TSI is the recommendation for time series data storage and analytics.

Primary scenarios

- **Store time series data in a scalable way:** At its core, Time Series Insights has a database designed with time series data in mind. Because it's scalable and fully managed, Time Series Insights handles the work of storing and managing events.
- **Explore data in near real time:** Time Series Insights provides an explorer that visualizes all data that streams into an environment. Shortly after you connect to an event source, you can view, explore, and query event data within Time Series Insights. The data helps you to validate whether a device emits data as expected and to monitor an IoT asset for health, productivity, and overall effectiveness.
- **Perform root-cause analysis and detect anomalies**: Time Series Insights has tools like patterns and perspective views to conduct and save multistep root-cause analysis: Time Series Insights also works with alerting services like Azure Stream Analytics so that you can view alerts and detected anomalies in near real time in the Time Series Insights explorer.
- **Gain a global view of time series data that streams from disparate locations for multi-asset or site comparison:** You can connect multiple event sources to a Time Series Insights environment. This way you can view data that streams in from

multiple, disparate locations together in near real time. Users can take advantage of this visibility to share data with business leaders. They can collaborate better with domain experts who can apply their expertise to help solve problems, apply best practices, and share learnings.

- **Build a customer application on top of Time Series Insights:** Time Series Insights exposes REST Query APIs that you can use to build applications that use time series data.

Key Benefits:

- **Short start-up time**: Azure Time Series Insights doesn't require upfront data preparation, so you can quickly connect to millions of events in your IoT hub or event hub. After you connect, you can visualize and interact with sensor data to quickly validate your IoT solutions. You can interact with your data without writing code, and you don't need to learn a new language. Time Series Insights provides a granular, free-text query surface for advanced users, and point-and-click exploration.
- **Near real-time insights:** Time Series Insights can ingest millions of sensor events per day, with one-minute latency. Time Series Insights helps you gain insights into your sensor data. Use it to spot trends and anomalies, conduct root-cause analyses, and avoid costly downtime. Cross-correlation between real-time and historical data helps you find hidden trends in the data.
- **Build custom solutions:** Embed Azure Time Series Insights data into your existing applications. You also can create new custom solutions with the Time Series Insights REST APIs. Create personalized views you can share for others to explore your insights.
- **Scalability:** Time Series Insights is designed to support IoT at scale. It can ingress from 1 million to 100 million events per day, with a default retention span of 31 days. You can visualize and analyze live data streams in near real time, alongside historical data.

III. Technology Options for Cold Storage

Instead of keeping all data in a warm data store with low latency, high throughput, and full query capabilities, data can be split into warm and cold storage paths. This can provide lower storage costs while still preserving historical data. A cold storage database holds data that is not needed as quickly and/or frequently as warm storage, but still may be necessary to access in the future for reporting, analysis, machine learning use, etc.

Evaluation Criteria

Cold storage solutions were evaluated based on the following criteria. These criteria will not apply for every solution but were designed to be the most generally applicable for an IoT solution.

- **Security**. The solution offers mature and robust features such as encryption at rest, authentication and authorization, and network security.
- **Simplicity**. The solution is well documented and has a well-defined architecture. Development tasks are documented, supported by software development kits (“SDKs”), and to some degree testable in a local development workstation. Deployment and operational tasks are supported by documentation, tools, and user interfaces.
- **Scalability**. The database supports storing a large amount of data. Scaling out does not require downtime. The database has very long (on the order of years) or unlimited retention. The ideal solution automatically adapts cost and computing power to the load provided.

- **Price**. The database is affordable for large amounts of data.

Technology Options

The best cold storage database for a solution is highly dependent on what purpose the database will serve. The two data storage solutions below are designed for high scale at a low price, but each has strengths for different scenarios. Azure Blob Storage is recommended by Microsoft for the general case, as it is cheaper than Azure Data Lake Storage Gen2, especially in terms of write requests. However, if the solution requires cold storage data analytics (with Hadoop, Azure Data Analytics, etc.), or requires querying with U-SQL, Azure Data Lake Storage Gen2 is designed with that scenario in mind and may be the better choice.

1. Azure Blob Storage

Azure Blob Storage is a simple, inexpensive file storage database. Blobs can be used to store raw device data. Using page blobs instead of block or append blobs should be considered depending on frequency of write operations. Azure blob storage has full security capabilities, local or geo-redundant storage options, and is available in all Azure regions. It is highly scalable—the maximum storage limit is 5 PiB and the maximum request rate per account is 20,000 requests per second.

Azure Blob storage is designed for:

- Serving images or documents directly to a browser.
- Storing files for distributed access.
- Streaming video and audio.
- Writing to log files.
- Storing data for backup and restore, disaster recovery, and archiving.
- Storing data for analysis by an on-premises or Azure-hosted service.

Azure Storage supports three types of blobs:

- Block blobs store text and binary data. Block blobs are made up of blocks of data that can be managed individually. Block blobs store up to about 4.75 TiB of data. Larger block blobs are available in preview, up to about 190.7 TiB
- Append blobs are made up of blocks like block blobs, but are optimized for append operations. Append blobs are ideal for scenarios such as logging data from virtual machines.
- Page blobs store random access files up to 8 TB in size. Page blobs store virtual hard drive (VHD) files and serve as disks for Azure virtual machines. For more information about page blobs, see Overview of Azure page blobs

Key Benefits:

- **Excellent scalability:** Azure Storage is scalable by design whether you access via Data Lake Storage Gen2 or Blob storage interfaces. It is able to store and serve many exabytes of data. This amount of storage is available with throughput measured in gigabits per second (Gbps) at high levels of input/output operations per second (IOPS). Beyond just persistence, processing is executed at near-constant per-request latencies that are measured at the service, account, and file levels.

2. Azure Data Lake Storage Gen2

Azure Data Lake Storage Gen2 is a distributed data store that can persist large amounts of relational and nonrelational data without transformation or schema definition. It is a good choice for a storage database if big data analytics and/or unlimited storage are required. It is slightly more expensive than Azure Blob Storage (specifically in terms of write operations), but it is optimized for big data analytics workloads. The database can be accessed from Hadoop via WebHDFS-compatible REST APIs or using the U-SQL language. It has locally redundant storage and is available in some US Azure regions as well as North Europe.

New Generation of Data Lake Storage

Azure Data Lake Storage Gen2 is a set of capabilities dedicated to big data analytics, built on Azure Blob storage. Data Lake Storage Gen2 is the result of converging the capabilities of our two existing storage services, Azure Blob storage and Azure Data Lake Storage Gen1. Features from Azure Data Lake Storage Gen1, such as file system semantics, directory, and file level security and scale are combined with low-cost, tiered storage, high availability/disaster recovery capabilities from Azure Blob storage.

Data Lake Storage Gen2 makes Azure Storage the foundation for building enterprise data lakes on Azure. Designed from the start to service multiple petabytes of information while sustaining hundreds of gigabits of throughput, Data Lake Storage Gen2 allows you to easily manage massive amounts of data.

A fundamental part of Data Lake Storage Gen2 is the addition of a hierarchical namespace to Blob storage. The hierarchical namespace organizes objects/files into a hierarchy of directories for efficient data access. A common object store naming convention uses slashes in the name to mimic a hierarchical directory structure. This structure becomes real with Data Lake Storage Gen2. Operations such as renaming or deleting a directory become single atomic metadata operations on the directory rather than enumerating and processing all objects that share the name prefix of the directory.

Data Lake Storage Gen2 builds on Blob storage and enhances performance, management, and security in the following ways:

- Performance is optimized because you do not need to copy or transform data as a prerequisite for analysis. Compared to the flat namespace on Blob storage, the hierarchical namespace greatly improves the performance of directory management operations, which improves overall job performance.
- Management is easier because you can organize and manipulate files through directories and subdirectories.
- Security is enforceable because you can define POSIX permissions on directories or individual files.

Key Benefits:

- **Excellent scalability:** Azure Storage is scalable by design whether you access via Data Lake Storage Gen2 or Blob storage interfaces. It is able to store and serve many exabytes of data. This amount of storage is available with throughput measured in gigabits per second (Gbps) at high levels of input/output operations per second (IOPS). Beyond just persistence, processing is executed at near-constant per-request latencies that are measured at the service, account, and file levels.
- **Hadoop compatible access:** Data Lake Storage Gen2 allows you to manage and access data just as you would with a Hadoop Distributed File System (HDFS). The new ABFS driver is available within all Apache Hadoop environments, including Azure

HDInsight, Azure Databricks, and Azure Synapse Analytics to access data stored in Data Lake Storage Gen2.

- **A superset of POSIX permissions:** The security model for Data Lake Gen2 supports ACL and POSIX permissions along with some extra granularity specific to Data Lake Storage Gen2. Settings may be configured through Storage Explorer or through frameworks like Hive and Spark.
- **Cost effective:** Data Lake Storage Gen2 offers low-cost storage capacity and transactions. As data transitions through its complete lifecycle, billing rates change keeping costs to a minimum via built-in features such as Azure Blob storage lifecycle.
- **Optimized driver:** The ABFS driver is optimized specifically for big data analytics. The corresponding REST APIs are surfaced through the endpoint dfs.core.windows.net.

IV. Conclusion

In this seminar paper, a suitable architecture for the IoT workload, that is Lambda Architecture, has been outlined. Multiple suitable warm and cold storage technologies have been proposed and assessed. Warm storage technologies are optimized for speed at the expense of accuracy and consistency, while cold storage technologies are optimized for storage space and retention time. The recommended option for warm storage is Azure Cosmos DB due to its universality, while Azure SQL DB is optimized for relational data and Azure Time Series Insights is the standard choice for time series. The recommended option for cold storage is Azure Blob Storage due to its price and robustness, with Azure Data Lake Storage Gen2 as the analytical option.

V. References

- Microsoft (January 2019), Azure IoT reference architecture
- Microsoft (December 2018), Solution guide: Extracting Insights from IoT
- Microsoft (October 2019), Azure Cosmos DB documentation
- Microsoft (April 2019), Azure SQL DB documentation
- Microsoft (April 2020), Azure Time Series Insights Documentation
- Microsoft (June 2020), Azure Blob Storage Documentation
- Microsoft (February 2020), Azure Data Lake Storage Gen2 Documentation