Selecting a Data Warehousing Technology in Azure

1/31/2018
Selecting a Data Warehousing Technology in Azure

Melissa Coates
Solution Architect
Presenter

Leo Furlong
Principal Architect
Responding to Q&A in the chat window

Angela DeYoung
Marketing Coordinator
Webinar facilitator
This webinar will be recorded. Slides will be available.

In the next few days, you will receive an email notification with a link to the recorded session + the slide deck.

Please ask your questions using the GoToWebinar window.

If there are any questions that we don’t address before we conclude, we will follow up with you after the session. Depending on the extent of questions, we will also publish a blog post.
Selecting a Data Warehousing Technology in Azure

Webinar Objectives

- Present a series of data warehousing scenarios in Azure:
  - Relational → Nonrelational → Composite Use Cases

- Review the most common Azure services used for data warehousing

- Offer suggestions and things to consider when choosing technologies
Data Warehousing Objectives

DW objectives, regardless of the technical implementation:

✓ Integrated view from disparate data sources
✓ Historical analysis not available in source systems
✓ Minimize silos and multiple versions of the truth
✓ Centralize analytical data for many users to access
✓ Provide user-friendly data structure which can evolve & adapt
✓ Remove reporting query demand from source systems

And above all... Realize business value from the data
Data Warehousing Objectives

A DW implementation must support:

- Interactive analytical & tactical queries
- Acceptable query response time
- Diverse base of users & BI tools
- Near real-time data freshness

✓ High volumes of data
✓ Secure connectivity
✓ Many concurrent users
Numerous Options to Store & Process Data

**GOAL: BI & REPORTING**
- Office
- Other BI/Reporting tools

**ANALYZE**
- Reporting
- Data Lake Analytics Analysis Service

**PREDICT**
- SQL R, Python
- ML

**STORE & PROCESS**
- SQL Server
- CosmosDB
- SQL
- SQL DW
- HDI

**STAGE**
- Storage
- Data Lake

**PLATFORM**
- IaaS
- PaaS

**INTEGRATE**
- OMS
- Batch
- Data Factory
- Automation
- Run Book
- Function
- Load Balancer
- Gateway

**MANAGE**
- Key Vault
- Active Dir
- VNET
- ExpressRoute
- Load Balancer

**ENTERPRISE REQUIREMENTS**
- Security
- Compliance
- Geo-Availability
- Scale
- Performance
- Time-to-Market
- Cost
- Skills

Prime on Cloud Terminology & Service Offerings

- **Public Cloud**
  - **Infrastructure as a Service (IaaS)**
    - SQL Server in a Virtual Machine
  - **Platform as a Service (PaaS)**
    - Azure SQLDB, Azure SQLDW, CosmosDB
  - **Serverless Apps**
    - Azure Functions
  - **Software as a Service (SaaS)**
    - Power BI, Office 365

- **On-Premises**
  - **Virtual Server**
  - **Physical Server**

- **Azure Stack (Private Cloud)**

**Shared Infrastructure (Lower Cost)** vs. **Dedicated Infrastructure (Higher Cost of Ownership)**

**Easier to Scale** vs. **More Difficult to Scale**

**More Control (Higher Administration Effort)** vs. **Less Control (Lower Administration Effort)**
Relational Database Options for a Data Warehouse in Azure
Relational Data Warehouse Options in Azure

IaaS
Infrastructure as a Service

PaaS
Platform as a Service

SMP
Symmetric Multi-Processing

MPP
Massively Parallel Processing

Relational database of your choice in a virtual machine

Azure SQL Database

Managed Instance (in preview)

Standard

Azure SQL Data Warehouse

Azure Database for MySQL (in preview)

Azure Database for PostgreSQL (in preview)
The SQL Server Family

IaaS
Infrastructure as a Service

PaaS
Platform as a Service

SMP
Symmetric Multi-Processing

MPP
Massively Parallel Processing

SQL Server in a virtual machine

Unsupported in standard Azure SQL DB:
• PolyBase
• R Services
• Change data capture
• Some T-SQL syntax
• Among others...

Unsupported in Azure SQL DW:
• Primary, foreign, unique keys
• Computed columns
• Merge statements
• Some data types
• Among others...

Azure SQL Database

Managed Instance (in preview)

Azure SQL Data Warehouse
What Makes Azure SQL Data Warehouse So Different?

### MPP Architecture

**Control Node**
Interacts with apps & connections; coordinates activities of the compute nodes.

**Compute Nodes**
Provide the computational engines to process data.

**Distributions**
Every row of data is stored in a distribution. The method of distributing data is critical to achieving good performance.
The SQL Server Family

IaaS
Infrastructure as a Service

PaaS
Platform as a Service

SMP
Symmetric Multi-Processing

MPP
Massively Parallel Processing

SQL Server in a virtual machine

Azure SQL Database

Standard
Managed Instance (in preview)

Azure SQL Data Warehouse
When to Consider a Virtual Machine?

Consider when you want to:

✓ Closely resemble a traditional DW implementation
✓ Run an SMP DB larger than Azure SQL DB supports
✓ Quickly migrate an existing solution to the cloud
✓ Run the software or DB platform of your choice with full feature parity
✓ Run all aspects of SQL Server (SSIS, SSAS MD, MDS)
✓ Have full control & administer all aspects
When to Consider Azure SQL Database?

Consider when you want to:

✓ Create a new DW solution

✓ Run a small to medium-sized DW workload (up to 4TB currently)

✓ Take advantage of PaaS & reduced administration effort

✓ Optionally utilize automatic tuning features
When to Consider Azure SQL Data Warehouse?

Consider when you want to:

- Run a large-size DW solution (1-4TB+)
- Scale up/down, or pause, based on demand
- Integrate with multi-structured data
When to Consider Azure SQL Data Warehouse?

See the full decision tree on the BlueGranite blog:

https://www.blue-granite.com/blog/is-azure-sql-data-warehouse-a-good-fit
Non-Relational Database Options for a Data Warehouse in Azure
Non-Relational Data Warehouse Options in Azure

IaaS
Infrastructure as a Service

- HDInsight

PaaS
Platform as a Service

- Hive LLAP
- Spark

- HDInsight Interactive Query Cluster
- HDInsight Spark Cluster
- Azure Databricks (Currently In Public Preview)
When to Consider Hive LLAP on HDInsight?

Consider when you want to utilize open source technologies to:

✓ Project a data warehouse structure on top of data stored in Hadoop for the purpose of querying the data without relocating it (“SQL on Hadoop”)

✓ Interactive querying via HiveQL directly on multi-structured data in Hadoop (via LLAP)

✓ Do batch data processing with a SQL-like language
When to Consider Spark on HDInsight?

Consider when you want to utilize open source technologies to run a general purpose cluster computing system for:

✓ Interactive querying via SparkSQL directly on multi-structured data in Hadoop (“SQL on Hadoop”), and/or

✓ Batch data processing with a SQL-like language, and/or

✓ Processing of streaming data, and/or

✓ Libraries for machine learning and graph operations, and/or

✓ APIs for R, Python, Scala, etc.
When to Consider Azure Databricks?

Consider when you want to:

✓ Use a proprietary Spark framework which is optimized beyond open source Spark, and receives new releases much quicker than HDInsight Spark

✓ Utilize an environment which is closer to a true PaaS with less configuration; easier for non-IT personnel to create, manage, deploy

✓ Collaborate on data science projects with integrated workspaces
Composite Scenarios in Azure
The Modern Big Data Warehouse

Multi-Platform Architecture

Hive metastore

Azure SQL DB

HDInsight Spark Cluster

Batch processing

Azure Data Lake Store

Data Lake Engineers

Data Analysts

Data Scientists

Advanced Analytics & Data Science

Machine Learning

R, Python, APIs

Analytics

Data Exploration

Corporate Reporting

Self-Service BI

Streaming/Real-Time/Applications Integration

Business Users

Corporate Reporting

Self-Service BI
Considerations & Key Decisions

Data

What **types of data ingestion pipelines** do you have, at what frequency?
- Batch
- Micro-batch
- Streaming

What are the current + anticipated **data size volumes**, and in what **format**?
- Structured data (ex: corporate relational data, CSV)
- Semi-structured data (ex: JSON, logs)
- Unstructured data (ex: images)
- Geospatial data
Considerations & Key Decisions

Data Movement & Storage

What level of data integration (ETL or ELT) vs. data virtualization provides optimal data access?

- Data movement can be expensive
- Data might be too large to practically move
- Time window for data processing may be small
- Latency (freshness) of data varies

Which do you value more?

- Polyglot persistence strategy ("best fit engineering" based on the data itself)
- Architectural simplicity

Is this a brand new solution, or are you augmenting an existing solution?
Considerations & Key Decisions

Information Delivery

What type of data consumption do you support?
  • Centralized reporting & analytics
  • Decentralized self-service models
  • Departmental or subject-specific data marts
  • Application integration

What are the expectations + needs of your user population?
  • Casual users
  • Data analysts
  • Data scientists
  • IT, BI specialists, big data engineers
Considerations & Key Decisions

Cloud Objectives & Goals

Your goals for going to the cloud will affect tradeoffs & decisions you make.

<table>
<thead>
<tr>
<th>Common Pros</th>
<th>Common Concerns</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Eliminate/reduce data center management</td>
<td>✓ Uptime guarantees</td>
</tr>
<tr>
<td>✓ Elasticity (scaling up/down)</td>
<td>✓ Performance</td>
</tr>
<tr>
<td>✓ Self-service provisioning of services</td>
<td>✓ Security</td>
</tr>
<tr>
<td>✓ Ease of experimentation (agility)</td>
<td>✓ Compliance, regulations, legal</td>
</tr>
<tr>
<td>✓ Faster time-to-market</td>
<td>✓ Sharing of resources (multi-tenancy; noisy neighbors)</td>
</tr>
<tr>
<td>✓ Easier high availability and disaster recovery</td>
<td>✓ Data and intellectual property privacy</td>
</tr>
<tr>
<td>✓ Subscription-based operating expenses (rather than capital expenses with large up-front investment)</td>
<td>✓ Vendor lock-in/dependency</td>
</tr>
<tr>
<td></td>
<td>✓ Connecting legacy systems (hybrid/on-prem)</td>
</tr>
<tr>
<td></td>
<td>✓ Sprawl of self-provisioned services</td>
</tr>
<tr>
<td></td>
<td>✓ Lack of cloud expertise</td>
</tr>
<tr>
<td></td>
<td>✓ Complexity</td>
</tr>
<tr>
<td>The cloud is not <em>always</em> easier</td>
<td>✓ Cost</td>
</tr>
<tr>
<td>The cloud is not <em>always</em> cheaper</td>
<td>Difficult to estimate cost up-front</td>
</tr>
</tbody>
</table>
Final Suggestions

✓ Give serious consideration to if Azure SQL Data Warehouse is the best choice. Don’t choose it ‘by default’ because of the product name. Learn its design patterns & differences from SQL Server if you do utilize Azure SQL DW.

✓ A relational data warehouse can be an excellent complement to a data lake and/or Hadoop implementation in a multi-platform architecture.

✓ Do a small proof of concept before making a big commitment.

✓ Look at using PaaS first (over IaaS), and evaluate features support.

✓ Consider the experience level of your staff, and its ability to support a solution.

✓ Cloud offerings are constantly changing. Plan for how to keep up.
Q&A
Thank You for Attending this Webinar

https://www.blue-granite.com/