Moving to a cloud data warehouse:

HOW TO MODERNIZE AN ENTERPRISE DATA WAREHOUSE WITHOUT SACRIFICING EXISTING SOLUTIONS

Inside:

Why your organization should build a cloud data warehouse, the overall architecture for a modern cloud data warehouse, a selection of cloud-native technologies to use in building a data warehouse and the benefits of each.

PREPARED BY:

BAKER TILLY DIGITAL

bakertilly

INTRO

Implementing a data warehouse in the cloud using Microsoft® Azure technologies can help companies of all sizes move to a modern cloud data warehouse while leveraging existing on-premises data warehouses.

CONTENTS

INTRODUCTION

As organizations increasingly move their IT assets to the cloud[1], the continued presence of on-premises data warehouses represents an increasing burden. When most of an organization's assets are in the cloud, maintaining or even enhancing an on-premises data warehouse requires complex integrations comprising both code and infrastructure. However, these legacy solutions provide real business value that cannot be discarded.

How can an organization modernize and streamline its enterprise data warehouse without sacrificing the core benefits of an existing solution?

As businesses add value by storing data in a data lake for organizational access and research, the need for modelling the data and warehousing techniques for the data remains and can be implemented in a cloud platform to continue adding value.

Implementing a data warehouse in the cloud using Microsoft® Azure technologies can help companies of all sizes move to a modern cloud data warehouse while leveraging existing on-premises data warehouses.

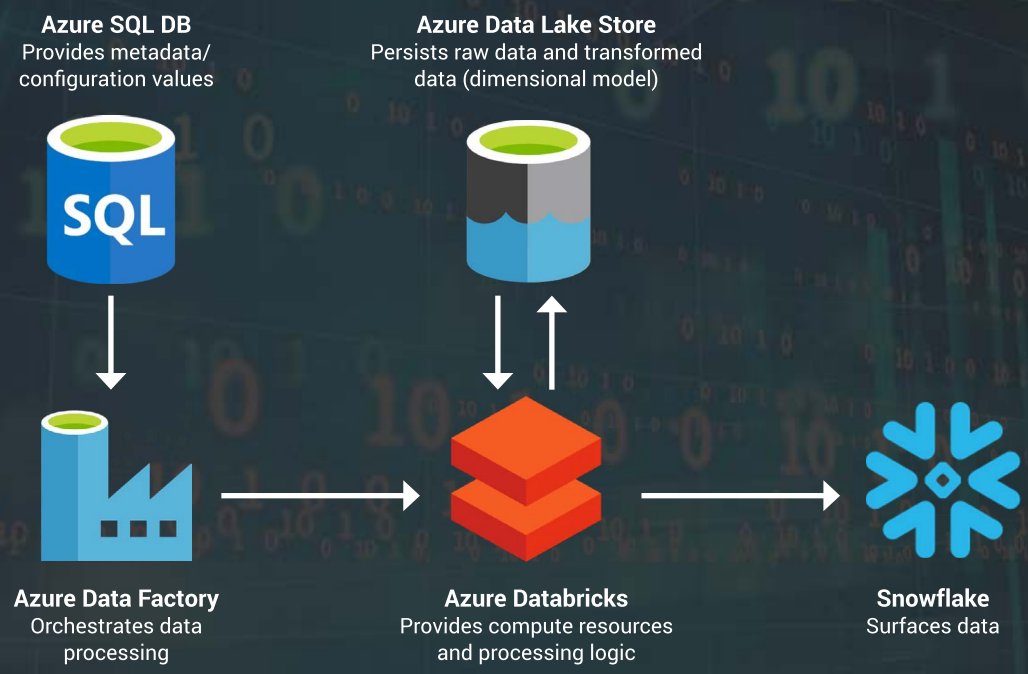
This paper discusses:

- Why an organization should build a cloud data warehouse
- The overall architecture for a modern cloud data warehouse
- A selection of cloud-native technologies to use in building a data warehouse
- Key benefits of each recommendation

RECOMMENDED ARCHITECTURE

Taking the benefits of dimensional modelling, the power of big data and distributed computing, and combining it with configuration-driven development, delivers a high-value and extensible cloud-based warehouse solution.

High-level model



AZURE DATA LAKE FOR SOURCE DATA ORGANIZATION

ASSUMPTIONS

In the rest of this white paper, we assume the existence of a data lake to house raw data from required sources. We do not cover how to organize, load, or secure a data lake in this paper.

BENEFITS OF A DATA LAKE

- Simple access and cost-effective[2] storage for scalable sourcing of data over time
- Standardized format and consistency across the organization
- Format offerings include open source options, such as Apache Parquet
- Partitioning of data in patterns can be performed in-line with best big data practices
- Access[3] can be distributed organization-wide
- Implementation early in the process allows for rapid and agile development and provides a way to reload the data model from a history of immutable data

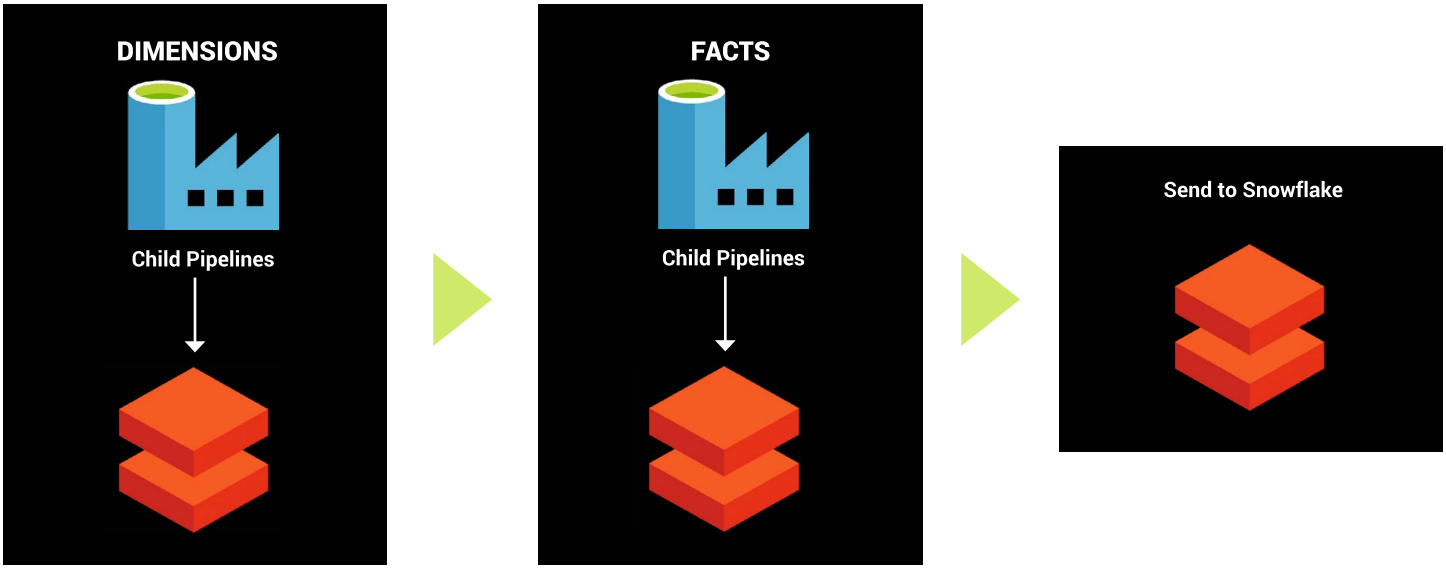
AZURE DATA LAKE GENERATION 1 AND 2

While Azure Data Lake Generation 1 provides an ample solution for storing and organizing source data, Generation 2[4] offers an improved product:

- Generation 2 can read and write faster[5] than Generation 1.
- Generation 2 also stores data the same way as other Azure Storage services, allowing users to access data using a variety of familiar methods.

USE AZURE DATA FACTORY FOR ORCHESTRATION

GENERAL ARCHITECTURE



WHAT IS AZURE DATA FACTORY?

Azure Data Factory[6] is a cloud tool for orchestrating events in a logical progression using "pipelines." It allows users to move data, call events, send notifications, look up data, and even call other pipelines. This can all be done in sequence, in loops, in parallel, or dependent on actions occurring within the pipeline. For this scenario, it is important to know that Azure Data Factory allows calls to Azure Databricks with parameters.

As a basic example as an ETL platform, a user can set up a pipeline to copy data to a staging area in a Data Lake, then run an Azure Databricks notebook to transform the data as needed, and finally, send the transformed data to a separate landing zone.

BENEFITS OF AZURE DATA FACTORY

There are several benefits to using Azure Data Factory for a cloud data warehouse.

- Integration with the Azure platform, providing a direct link to the Data Lake, Databricks, and SQL Server
- Pipeline triggers, either by schedule or a file being updated
- Parallel execution using Azure Data Factory's powerful for-each loop features
- Scalability and extensibility to data sets of all sizes
- Alerts and logging sent to Azure Log Analytics for operational management
- Deployments via Azure's DevOps platform

ORCHESTRATION ONLY

While Data Factory's Data Flow is available as a transformation tool, the practice for this type of project has been to write standardized and custom Databricks notebooks.

This keeps the areas of concern separate, contains business logic for building in one place, allows the use of Databricks' notebook version control for development releases, and allows the the organization to easily change where the final data sets are surfaced if necessary.

DRIVE PIPELINES WITH A METADATA FRAMEWORK

WHAT IS A METADATA FRAMEWORK?

Metadata is data that describes attributes of other data. A metadata framework is a repository – typically a relational database – for metadata related to the data model and data pipelines. For example, the framework might contain metadata describing dimensions and

their business keys; facts and their related dimensions; or target tables and their upstream data sources.

This metadata defines what work is done by the data pipeline; Azure Data Factory queries the metadata framework at runtime, and metadata is passed in to Databricks as parameters. This enables configuration-driven development and code encapsulation.

BENEFITS OF A METADATA FRAMEWORK

Providing a structure for parameterization of workflows – in this case, Azure Data Factory pipelines – allows for a plug-and-play, extensible solution. This enables reuse of standard data engineering patterns, such as updating a Type 2 slowly changing dimension. Encapsulating the code and driving it with metadata means that development of new data model objects does not require reengineering of the entire data pipeline. This reduces technical debt and development effort.

Additions to the data model are far simpler with a metadata framework. A new dimension, fact, or other object can be added with minimal of new code. Extract, transform, and load (ETL) operations can be configured via metadata instead of writing custom Python scripts or developing new Azure Data Factory pipelines.

TOOLS FOR METADATA

It is recommended to use Azure SQL Database[7] for use as a metadata storage tool. Not only will it interface well with the other Azure services used for this scenario, but it will be automatically updated with the latest SQL Server features by Microsoft to enhance its capabilities.

The structure of a SQL metadata framework can be retained in a version control system such as Git. Azure DevOps offers hosted Git repos as well as continuous deployment tools.

PROCESS DATA USING AZURE DATBRICKS

WHAT IS AZURE DATABRICKS?

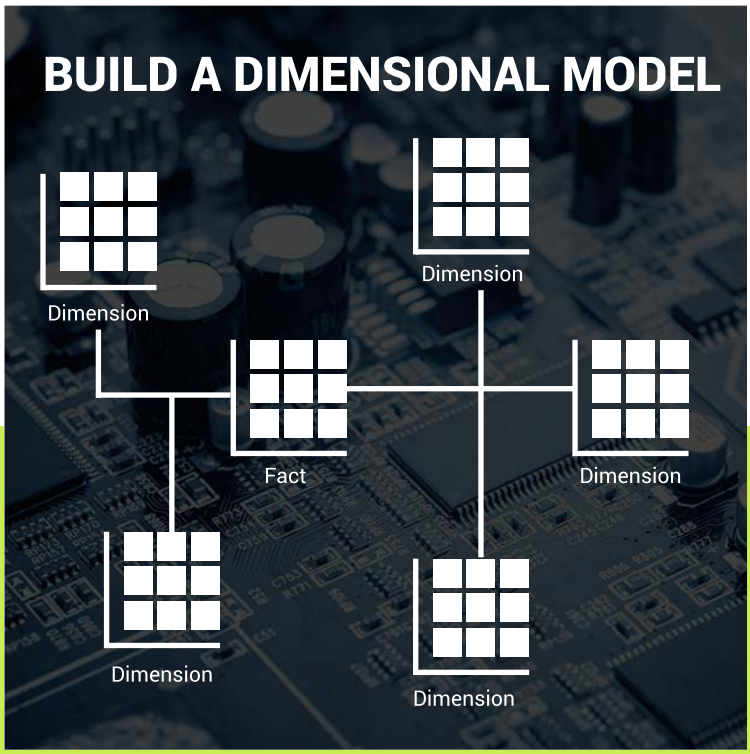
Databricks is a cloud platform built around Apache Spark. Spark is an open-source distributed processing framework that enables fast processing of big data with multiple improvements over the original MapReduce paradigm[8]. Databricks provides additional features for ease of use such as managed clusters, collaborative workspaces and a notebook-style interface. Azure Databricks[9] is the Databricks platform hosted on Azure.

WHY USE DATABRICKS?

Azure Databricks is currently the technology of choice for handling data engineering on Azure.

Key reasons to use Azure Databricks:
- Provides an array of cluster sizes and can automatically scale clusters up and down as needed
- Integrates with Azure Active Directory
- Supports multiple languages, including Python, R and SQL
- The runtime offers improved performance over Spark[10]
- Workspaces provide an interface for multiple users to collaboratively develop Spark applications

For a cloud-first data warehouse, Azure Databricks provides the tools and flexibility needed to integrate a variety of data types. It also provides the computational power needed to process big data without requiring data engineering teams to develop additional code for cluster management.



WHAT IS A DIMENSIONAL MODEL?

Dimensional modeling is a methodology commonly used to design data warehouses. Originally developed by the Kimball Group, dimensional modeling is the de facto standard for delivering data to the business. In the methodology, the data model is defined as a star schema, where key business processes are encoded as facts and business entities are encoded as dimensions. Dimensional models are designed to represent the business in data form.

For a full list of dimensional modeling techniques, see the Kimball Group glossary.

WHY USE DIMENSIONAL MODELING?

Accuracy

A key dimensional modeling technique is the establishment of keys and the definition of fact grain. This ensures that the data surfaced in user-facing tools accurately reflects business processes.

Comprehensibility

Dimensions naturally align with business entities; in a dimensional model, a business user never needs to ask, "Which data source should I use to find my top customers?" Instead, they simply use the customer dimension. Similarly, facts group together naturally related measures and make it obvious where to look.

Compatibility with BI Tools

Business Intelligence (BI) reporting and visualization tools tend to function best when the underlying data is a star schema. For example, implementing dimensional modeling techniques can help optimize performance in Microsoft Power BI[11] and other tools such as Tableau.

Extensibility

As new dimensions are added to the model, future facts can be built on top of the groundwork already laid. This means that the descriptive attributes added to existing dimensions can be reused for analysis of new facts. Reinventing the wheel with each new set of metrics is not necessary.

Leveraging existing solutions

Many organizations have a data warehouse in place, often conforming to dimensional modeling standards and hosted on-premises. These solutions typically utilize SQL to define datasets for extraction. An existing data warehouse comprises many assets that can be leveraged to build a cloud-first solution, such as:

- A logical data model (conceptual definitions of dimensions and facts)
- Naming conventions and code standards
- Business logic captured in code
- Common data transformations

Existing dimensions and facts can be harvested for a head-start on building a new data warehouse. Using Databricks' SQL API, it may even be possible to reuse code.

PERSIST MODELED DATA USING DELTA LAKE

WHAT IS DELTA LAKE?

Open Source

Delta Lake[12] is an open source storage technology under the Databricks platform that comes standard with a Databricks subscription. As an open source product, it is extensible to many solutions.

Parquet

Files that represent the stored data use the Parquet[13] file format, providing compressed and columnar storage. Parquet is also an open source technology provided by Apache, which works well with the Apache Spark ecosystem that drives Databricks clusters.

WHY USE DELTA LAKE?

Data Lake

Delta Lake provides storage of staged and transformed data for the model in Azure's Data Lake.

Malleable

It allows changes to the data in the form of deletes, updates and merges, which supports slowly changing dimensions, incremental data loading and ad-hoc data updates. The data is stored as snapshots so users can get data from a certain point in time in case an older version of the data set is needed. The retention period for snapshots is customizable[14]. Data files can be optimized[15] for even better performance based on key values and partitioning.

Transactional

Delta Lake data supports ACID transactions and data isolation. It conserves data quality, as model data and keys do not have to be rebuilt upon each run.

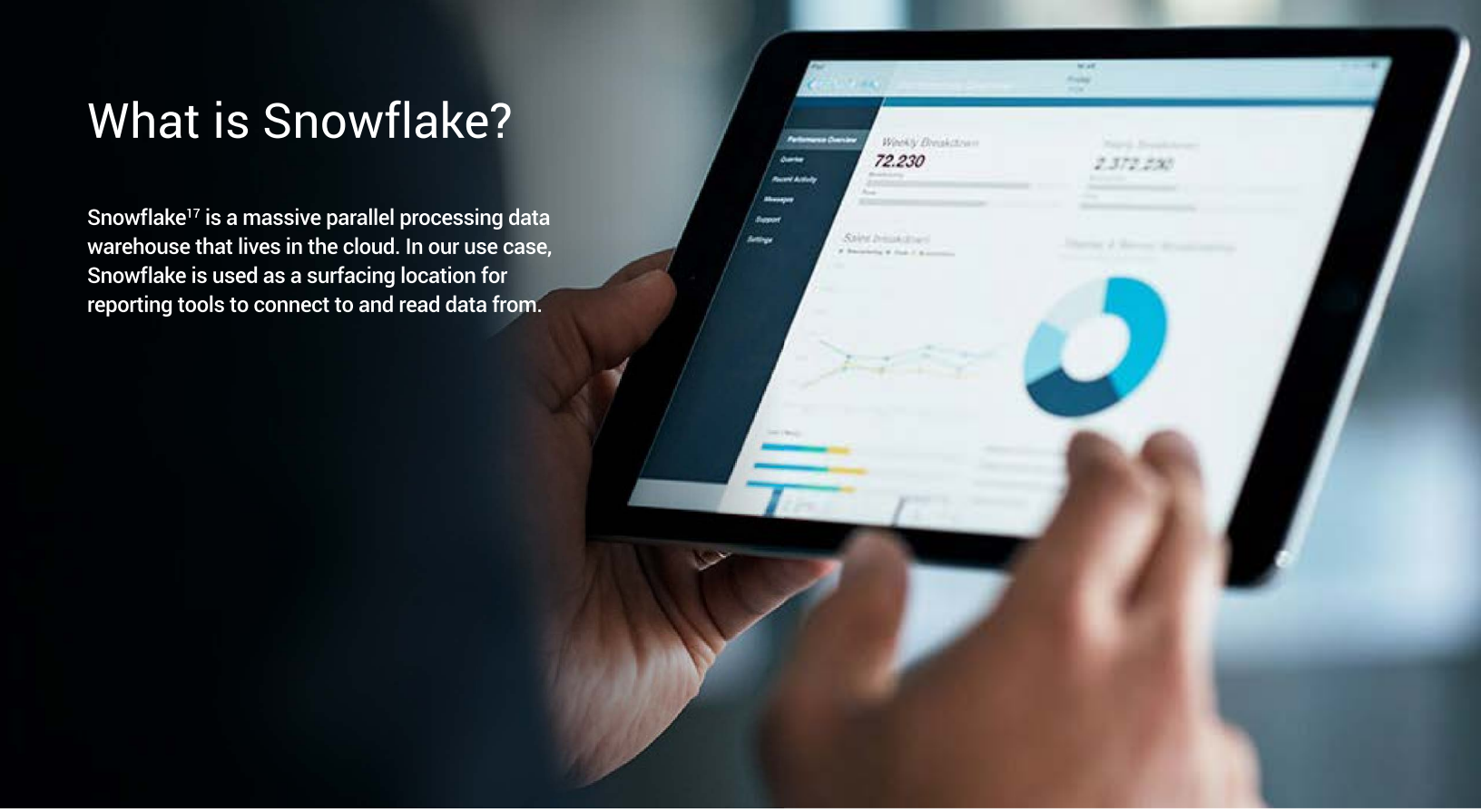
Efficient

It is an efficient way of storing data, using Parquet as a format in the lake. Parquet's storage footprint is small and can be read by Databricks very quickly compared to other formats.

Copy dimensional model to Snowflake

What is Snowflake?

Snowflake[17] is a massive parallel processing data warehouse that lives in the cloud. In our use case, Snowflake is used as a surfacing location for reporting tools to connect to and read data from.



BENEFITS OF SNOWFLAKE

Speed
Querying Snowflake from a reporting tool is quite fast, enabling end-users and power users to gain insights on data from the model quickly. This can be improved by establishing partitioning practices to allow Snowflake to prune queries for better performance.

Administration
Administration of Snowflake is easily manageable and accessible, allowing more time to distribute the data to the organization.

Connectivity
Snowflake can be connected through Databricks within an Azure tenant to perform data loads and push-down queries. Having a Snowflake account based in Azure also provides connectivity to other resources within the platform.

Security
Security is robust and implementable on many levels to expose the data only to the parts of an organization that specifically needs it. Snowflake can expose data via direct connection from a reporting service, allowing for row-level security. It also can provision read-only accounts for exposing specific databases for different parts of the organization.

Historical Data
Snowflake retains historical data to provide the ability to view snapshots of data from the past . This gives the organization the ability to look back in time or revert data to an older state.

CONCLUSION

A well-constructed data warehouse is a key asset to the data-driven organization. Building a data warehouse using cloud technologies enables organizations to reap the benefits of proven methodologies while scaling for continued growth.

Utilizing the technologies and techniques described in this white paper will allow an organization to support and enhance analytics as it shifts to the cloud.

REFERENCES

1. https://www.forbes.com/sites/louiscolumbus/2019/04/07/public-cloud-soaring-to-331b-by-2022-according-to-gartner/#67b7e8a95739
2. https://azure.microsoft.com/en-us/pricing/details/data-lake-storage-gen1/
3. https://docs.microsoft.com/en-us/azure/data-lake-store/data-lake-store-security-overview
4. https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction
5. https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-performance-tuning-guidance
6. https://azure.microsoft.com/en-us/services/data-factory/
7. https://azure.microsoft.com/en-us/services/sql-database/
8. https://www.infoworld.com/article/3236869/what-is-apache-spark-the-big-data-platform-that-crushed-hadoop.html
9. https://azure.microsoft.com/en-us/services/databricks/
10. https://databricks.com/blog/2017/07/12/benchmarking-big-data-sql-platforms-in-the-cloud.html
11. https://docs.microsoft.com/en-us/power-bi/guidance/star-schema#star-schema-relevance-to-power-bi-models
12. https://databricks.com/product/databricks-delta
13. https://parquet.apache.org/documentation/latest/
14. https://docs.databricks.com/spark/latest/spark-sql/language-manual/vacuum.html
15. https://docs.databricks.com/delta/optimizations.html
16. https://databricks.com/session/spark-parquet-in-depth
17. https://www.snowflake.com/
18. https://www.snowflake.com/news/snowflake-announces-availability-on-microsoft-azure/
19. https://docs.snowflake.net/manuals/user-guide/data-sharing-reader-create.html
20. https://docs.snowflake.net/manuals/user-guide/data-time-travel.html