Azure Big Data Landscape
A high level overview of the big data services on the Azure cloud platform

Rolf Tesmer
Azure Data Solutions Architect, Microsoft
Why is data so important?
Because there's just so much of it!

- Internet Connected
- Digital
- Analog

Timeline:
- 30 years
- 10 years

Years:
- 1985
- 1990
- 1995
- 2000
- 2005
- 2007
- 2010
- 2015
- 2017
- 2020
On-Prem vs IaaS vs PaaS vs SaaS – Which One?
Agenda

2 Key Components of the Microsoft Azure Cloud Data Platform

**Big Data / Analytics PaaS**
(General Introduction & Overview)
Introduction: Data Size Over the Years...

- **Click stream**: Wikis/blogs
- **Sensors/RFID/devices**
- **Social**: Audio/video
  - **Web 2.0**
    - Web Logs
    - Digital Marketing
    - Search Marketing
    - Recommendations
  - **ERP/CRM**
    - Payables
    - Payroll
    - Inventory
  - **Collaboration**: eCommerce
  - **Advertising**: Mobile

- **Big Data**
  - Log files
  - Spatial & GPS coordinates
  - Data market feeds
  - eGov feeds
  - Weather
  - Text/image
Introduction: Big Data Definition - The Four V's

A Big Data "problem" exists when you must address more than one of the V’s. (Only one V indicates current technology is likely to satisfy your goals)

**Volume**
The data exceeds the physical limits of vertical scalability, implying a scale-out solution (vs. scaling up).

**Velocity**
The decision window is small compared with the data change rate.

**Variety**
Many different formats make integration difficult and expensive.

**Variability**
Many options or variable interpretations confound analysis.

To solve the "problem" you often need specialist technologies.
Business wish to solve the "problem" because it offers competitive advantage.
Big Data Business Applications & Use Cases

**Financial services**
- New account risk screens
- Fraud prevention
- Trading risk
- Maximize deposit spread
- Insurance underwriting
- Accelerate loan processing

**Retail**
- 360° view of the customer
- Analyze brand sentiment
- Localized, personalized promotions
- Website optimization
- Optimal store layout

**Telecom**
- Call detail records (CDRs)
- Infrastructure investment
- Next product to buy (NPTB)
- Real-time bandwidth allocation
- New product development

**Manufacturing**
- Supplier consolidation decisions
- Supply chain and logistics
- Assembly line quality assurance
- Proactive maintenance
- Crowd source quality assurance

**Healthcare**
- Genomic data for medical trials
- Monitor patient vitals
- Reduce re-admittance rates
- Store medical research data
- Recruit cohorts for pharmaceutical trials

**Utilities, oil, and gas**
- Smart meter stream analysis
- Slow oil well decline curves
- Optimize lease bidding
- Compliance reporting
- Proactive equipment repair
- Seismic image processing

**Public sector**
- Analyze public sentiment
- Protect critical networks
- Prevent fraud and waste
- Crowd source reporting for repairs to infrastructure
- Fulfill open records requests
Big Data and Data Warehousing Compared

Big Data does not negate the business drivers for a Data Warehouse. The technologies serve different business purposes. Big Data systems can be a feeder into the Data Warehouse.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Big Data (ADL, HDInsight, Hadoop, etc)</th>
<th>Data Warehousing (SQL DW, SQL in IaaS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution Type</td>
<td>Ecosystem, not a product</td>
<td>Product/Service</td>
</tr>
<tr>
<td>Typical Data Type</td>
<td>Structured, Semi-Structured, Unstructured</td>
<td>Structured (Operational)</td>
</tr>
<tr>
<td>Typical Data Size</td>
<td>TB – PB Linear Scale out = MPP</td>
<td>GB – TB Non-linear, Scale Up (SMP typically!)</td>
</tr>
<tr>
<td>Typical Data Artefacts</td>
<td>Files</td>
<td>Tables/Rows/Columns</td>
</tr>
<tr>
<td>Schema</td>
<td>Defined On Read</td>
<td>Defined On Write</td>
</tr>
<tr>
<td>Data Consistency, Quality and Accuracy</td>
<td>Low, loose structure, no ACID</td>
<td>High, complex structure, strong ACID</td>
</tr>
<tr>
<td>Azure Technologies</td>
<td>HDInsight, Data Lake Vendors (Cloudera, MapR, Hortonworks)</td>
<td>SQL DB, SQL DW SQL Relational Database in IaaS</td>
</tr>
</tbody>
</table>
Big Data as part of a Data Warehousing Solution

Relational

- Azure SQL Data Warehouse
- SQL Server in Azure VMs

Beyond relational

- Azure Data Lake
- Azure HDInsight
- Azure Marketplace
  - Hortonworks, Cloudera, MapR

Fastest insights
Real-time insights with breakthrough query performance

Analytics built-in
Real-time insights with analytics built in

Choice of deployment
Leading solutions—on-premises and in the cloud

Layers of security
Least vulnerable database 6 years in a row

Any data, any scale
A hybrid solution that grows in step with customer needs

More for the price
Customers do more with industry-leading TCO
Agenda

1. Lambda Architecture

Big Data / Analytics PaaS
What is the LAMBDA architecture?

“The Objective of Lambda Architecture is to leverage the combined power of both **batch** & **real-time** processing to address the business scenarios where it requires both **historic view of the data** as well as getting insight into the **data in real-time** as business happens.”

https://gallery.cortanaintelligence.com/Solution/Telemetry-Analytics
Big Data Pipeline and Data Flow in Azure

Ingestion

Bulk Ingestion
- Azure Data Factory
- Portal & Tools
- AddCopy Service
- Developer SDK
- Apache DistCp, Sqoop
- Import/Export Service

Event Ingestion
- Azure Event Hub
- STORM

Preparation, Analytics and Machine Learning
- Machine Learning
- Data Lake Analytics
- HDInsight (Hadoop and Spark)
- Stream Analytics

Azure Data Lake Store

Discovery
- Azure Data Catalog

Visualization
- Power BI
- SQL DW

DATA → INTELLIGENCE → ACTION
What exactly is Unstructured, Semi-Structured and Structured Data?
## Considering Data Types

Efficient **data compression** and **encoding** schemes with enhanced performance to handle complex data in bulk.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Storage Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unstructured</td>
<td>Store Natively (logs, pics, etc)</td>
</tr>
<tr>
<td>Semi-structured</td>
<td>JSON, XML, etc → Schema Evolution (Avro)</td>
</tr>
<tr>
<td>Structured</td>
<td>CSV → Columnar Storage (Parquet, ORC)</td>
</tr>
</tbody>
</table>

Columnar Formats: Why? ORC & PARQUET

All data for a column stored contiguously on disk. So you can read a column really fast. Just like SQL needs to do.

**Pro:**

Fast query

**Con:**

You have to convert data into it

Only do that if you need to query it many times
Columnar Formats: Options

ORCFile:
  - Best in Hive
  - Allows vectorized execution (Fast)
  - Allows ACID (Insert / Update / Delete)

Parquet:
  - Fully supported
  - No vectorization or ACID
  - Common for mixed Hive/Spark workloads
Query Times for Different Formats

Ran 4 queries (using Impala) over 4 Million rows (70GB raw), and 1000 columns (wide table)

Reference: Unknown
Data Size for Different Formats & Compression

Reference: Unknown
Agenda

2. What exactly is Hadoop?
What’s the deal with elephants?

“[Hadoop was] the name my kid gave a stuffed yellow elephant. Short, relatively easy to spell and pronounce, meaningless, and not used elsewhere: those are my naming criteria. Kids are good at generating such. Googol is a kid’s term“ – Doug Cutting, Hadoop creator
Introduction: What is Hadoop?

A platform with a portfolio of projects
 Governed by Apache Software Foundation (ASF) (Open Source)
 Comprises core services of MapReduce, HDFS, and YARN

<table>
<thead>
<tr>
<th>Governance &amp; Integration</th>
<th>Data access</th>
<th>Security</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Falcon, Sqoop, Flume, NFS, WebHDFS</td>
<td>HDFS (Hadoop Distributed File System) (3x replicas)</td>
<td>Storage: HDFS, Resources: YARN</td>
<td>Ambari, Zookeeper</td>
</tr>
<tr>
<td>Data management</td>
<td>YARN: data operating system</td>
<td>Access: Hive, ... Pipeline: Falcon Cluster: Knox</td>
<td>Scheduling</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Oozie</td>
</tr>
</tbody>
</table>
The various big data solutions

- **Azure MarketPlace**: HDP | CDH | MapR
  - Any Hadoop technology

- **Azure HDInsight**: Workload optimized, managed clusters

- **Azure Data Lake**: Specific apps in a multi-tenant form factor

- **Azure Data Lake Store**

- **Azure Storage**
## Context - Comparing Hadoop and SQL Server

<table>
<thead>
<tr>
<th>Hadoop</th>
<th>SQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDFS</td>
<td>Database</td>
</tr>
<tr>
<td>MapReduce</td>
<td>Windows Cluster</td>
</tr>
<tr>
<td>YARN</td>
<td>Relational Engine</td>
</tr>
<tr>
<td>Hadoop Common</td>
<td>SQL OS</td>
</tr>
<tr>
<td>Master Web Interface (HUE)</td>
<td>SQL Server Management Studio</td>
</tr>
<tr>
<td>Ambari</td>
<td></td>
</tr>
<tr>
<td>Sqoop</td>
<td>BCP</td>
</tr>
<tr>
<td>Hive / Impala</td>
<td>T-SQL (ie Create tables, etc.)</td>
</tr>
<tr>
<td>Pig</td>
<td>Powershell</td>
</tr>
<tr>
<td>Spark</td>
<td>In Memory SQL Stored Procedures</td>
</tr>
</tbody>
</table>

**Reference:** “Eating the Elephant” – PASS 2015 - Stuart R Ainsworth
Physical Structure

**Data Nodes**
- Blocks of data replicated 3 times across Data Nodes
- Store blocks on Local Storage
- Hadoop Distributed File System (HDFS)

**Name Node**
- Keeps the directory tree of all files in the file system
- Knows where to get each Block once per request
- Does not store File data itself
- Name node is critical – if down, cluster is down
Data Redundancy

Data Blocks are copied to three different nodes

File is split into 4 blocks (128MB x 3, 16MB x 1)
Agenda

2. Key Components of the Microsoft Azure Cloud Data Platform

Big Data / Analytics PaaS
HDInsight
Azure HDInsight

Hadoop as a Service on Azure (PaaS)

FULLY MANAGED AND SUPPORTED PaaS
Hadoop, Spark, Hbase, Storm, Kafka

Available on LINUX

100% OPEN SOURCE Apache Hadoop

Clusters up and RUNNING IN MINUTES (20-30)

Use familiar BI TOOLS FOR ANALYSIS like Excel
HDInsight: Azure PaaS Implementation of Hadoop

Hive
- HiveQL is a SQL-like language (subset of SQL) (Compiled into MapReduce jobs)

HBase
- Columnar, NoSQL database on data in HDFS

Spark
- In Memory Processing on Multiple Workloads

Storm
- Stream Analytics for Near-Real Time Processing (similar to Azure Stream Analytics)
HDInsight Supports Hive

SQL-like queries on Hadoop data in HDInsight

HDInsight provides easy-to-use graphical query interface for Hive

HiveQL is a SQL-like language (subset of SQL)

Hive structures include well-understood database concepts such as tables, rows, columns, partitions

Compiled into MapReduce jobs that are executed on Hadoop

Dramatic performance gains with Stinger/Tez

Stinger is a Microsoft, Hortonworks and OSS driven initiative to bring interactive queries with Hive

Brings query execution engine technology from Microsoft SQL Server to Hive

Performance gains up to 100x
HDInsight Supports Spark

In Memory Processing on Multiple Workloads
Single execution model for multiple tasks (SQL queries, Streaming, Machine Learning, and Graph)
Processing up to 100x faster performance
Developer friendly (Java, Python, Scala)
BI tool of choice (Power BI, Tableau, Qlik, SAP)
Notebook experience (Jupyter/iPython, Zeppelin)
HDInsight Supports HBase

NoSQL database on data in HDInsight
Columnar, NoSQL database
Runs on top of the Hadoop Distributed File System (HDFS)
Provides flexibility in that new columns can be added to column families at any time
HDInsight Supports Storm

Stream analytics for Near-Real Time Processing
Consumes millions of real-time events from a scalable event broker (ie. Apache Kafka, Azure Event Hub)
Performs time-sensitive computation
Output to persistent stores, dashboards or devices
Customizable with Java + .NET
Deeply integrated to Visual Studio

Event producers → Collection → Event Queuing System → Transformation → Long-term storage → Presentation and action

Event Hubs
Apache Storm on HDInsight
Kafka / RabbitMQ / ActiveMQ

Web/thick client dashboards
Search and query
Data analytics (Excel)
Devices to take action

HBase
HDFS
Azure DBs
Azure storage
Live Dashboards
Storage adapters
Agenda

2. Key Components of the Microsoft Azure Cloud Data Platform

Big Data / Analytics PaaS
Data Lake Store & Analytics
**Introduction:** What is Azure Data Lake Store & Analytics?

**Consists of 2 component parts:**
Data Lake Store & Data Lake Analytics

**Distributed PaaS** service

**Both Instantly scale** to meet performance needs

**Analytics** over **all data**
(unstructured, semi-structured, structured)

**U-SQL** to perform Analytics
(simple and familiar, easily extensible)
(Integrated into **Visual Studio** tools)

Built on open standards (**YARN**)

Can deploy other services on store (ie **HDInsight**)

**Microsoft Azure Data Lake**

- **ADL Analytics**
  - **U-SQL**
- **HDInsight**
  - Spark
  - Storm

**HDFS and ADL**

- **ADL Store**
  - Social Media
  - Video
  - Web
  - Logs
  - Files
Azure Data Lake Store

A hyper scale repository for big data analytics workloads

An enterprise wide repository of every type of data collected in a single place prior to any formal definition of requirements or schema.

**SCALE** No limits

**ANY DATA** Store in its native format

**HADOOP FILE SYSTEM (HDFS)** for the cloud

**NATIVELY** accessible via both HDFS and ADL

**ENTERPRISE READY** access control, encryption

**PERFORMANCE** Optimized for analytic workload

**PaaS** Service managed by Microsoft
Azure Data Lake Store – Technical Details

Durable & Highly Available
• Data is managed by Microsoft (PaaS)

Unlimited Storage
• Unlimited account sizes, no limits to scale
• Individual file sizes to PBs

Secure
• Secure files and folders, POSIX (ACL)
• Auditing and logging
• Encryption at rest

Optimised for Analytic Workloads
• Designed for large scale parallel processing
• Auto optimize to match active workloads
• Immediate read after write

Primary Use Cases
• Long term IoT storage
• Clickstream analysis
• Social analysis
• Web log analysis
• File based batch processing
• Staging files for DW loads
• Long term DW archive
• (+ similar use cases to Big Data)
Azure Data Lake Analytics

A elastic analytics service built on Apache YARN that processes all data, at any size

**AUTO SCALE** with no limits

**U-SQL** a language that unifies the benefits of SQL with the expressive power of C#

Optimized to work with **ADL STORE**

**FEDERATED QUERY** with Azure data sources

**ENTERPRISE READY**

Pay & Auto Scale **PER (U-SQL) ANALYTIC JOB**

**DEVELOP** jobs in **Visual Studio** or **Azure Portal**


Use Visual Studio DATA LAKE Tools

Create a new U-SQL Project

Develop U-SQL Analytic Job

Execute and Monitor Job

Dynamically select to Scale the Job
Apply schema on read

C# anywhere

Write to ADLS, & WASB in any format
```sql

@allRequests = EXTRACT UTCDate string,
                ActivityId string,
                Account string,
                operation string,
                HttpStatus string,
                Latency long

FROM @"/users/mwilke/output/telemetryExplore/jobRequests3.csv" USING Extractors.Csv();

@allRequests = SELECT *,
FROM @allRequests;

@summary = SELECT DISTINCT BusinessWeek,
                operation,
                PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY Latency) OVER (PARTITION BY BusinessWeek, operation) AS [25thPercentileLatency],
                PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY Latency) OVER (PARTITION BY BusinessWeek, operation) AS MedianLatency,
                PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY Latency) OVER (PARTITION BY BusinessWeek, operation) AS [75thPercentileLatency],
                PERCENTILE_CONT(0.90) WITHIN GROUP (ORDER BY Latency) OVER (PARTITION BY BusinessWeek, operation) AS [90thPercentileLatency],
                PERCENTILE_CONT(0.95) WITHIN GROUP (ORDER BY Latency) OVER (PARTITION BY BusinessWeek, operation) AS [95thPercentileLatency],
                PERCENTILE_CONT(0.99) WITHIN GROUP (ORDER BY Latency) OVER (PARTITION BY UTCDate, operation) AS [99thPercentileLatency]
FROM @allRequests;

@summaryCountMinMaxAvg = SELECT BusinessWeek,
                          operation,
                          COUNT(DISTINCT Account) AS Accounts,
                          COUNT(DISTINCT ActivityId) AS Operations,
                          MIN(Latency) AS MinLatency,
                          MAX(Latency) AS MaxLatency,
                          AVG(Latency) AS AvgLatency
FROM @allRequests GROUP BY BusinessWeek, operation;

@totalSummary = SELECT s.BusinessWeek, s.operation,
            cma.Accounts,
            cma.Operations,
            cma.MinLatency,
            cma.MaxLatency,
            cma.AvgLatency
```

Rich support for SQL operators
Reference: “Architect Robust Big Data Solutions with Azure Data Lake” – Matt Winter, Ignite Australia 2017
# Data Processing Technology Choices

<table>
<thead>
<tr>
<th></th>
<th>Azure SQL DW</th>
<th>Azure HDInsight with Spark</th>
<th>Cloudera Impala</th>
<th>Azure HDInsight (Hive, Pig, etc.)</th>
<th>Azure Data Lake Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Latency</td>
<td>Low</td>
<td><strong>Low</strong></td>
<td>Low</td>
<td><strong>Medium (Tez), High (MapReduce)</strong></td>
<td>High</td>
</tr>
<tr>
<td>Durability</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Data Volume</td>
<td>Up to 60 TB</td>
<td>*nodes</td>
<td>*nodes</td>
<td>*nodes</td>
<td>*vertices</td>
</tr>
<tr>
<td>Managed</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Storage</td>
<td>SQL DB</td>
<td>Blob, ADLS</td>
<td>HDFS</td>
<td>ADLS, Blob</td>
<td>ADLS, Blob</td>
</tr>
<tr>
<td>SQL Compatibility</td>
<td><strong>High</strong></td>
<td>Low (SparkSQL)</td>
<td>Medium</td>
<td>Medium (HiveQL)</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Reference: “Architect Robust Big Data Solutions with Azure Data Lake” – Matt Winter, Ignite Australia 2017
Agenda

3  Q & A
References

- Map Reduce - [https://en.wikipedia.org/wiki/MapReduce](https://en.wikipedia.org/wiki/MapReduce)
- Spark (core, streaming, ML, graphX) - [https://en.wikipedia.org/wiki/Apache_Spark](https://en.wikipedia.org/wiki/Apache_Spark)
- Avro - [https://wiki.apache.org/hadoop/Avro/](https://wiki.apache.org/hadoop/Avro/)
- ORC - [https://orc.apache.org/docs/](https://orc.apache.org/docs/)
- SPARK vs IMPALA – which and when - [https://learning.naukri.com/articles/spark-vs-impala/](https://learning.naukri.com/articles/spark-vs-impala/)