PRZECHOWYWANIE I PRZETWARZANIE DUŻYCH DANYCH W CHMURZE OBLICZENIOWEJ

BARTOSZ ZAREMBA
1. Relational database
2. Document database
3. Storage
4. Quick access to data
5. Search
6. Summary and what's next?
Why cloud computing?
1. RELATIONAL DATABASE
## Microsoft Relational Storage Options

<table>
<thead>
<tr>
<th>Physical</th>
<th>Virtual</th>
<th>IaaS</th>
<th>PaaS</th>
</tr>
</thead>
</table>

From private to public Cloud
Server Connectivity

Microsoft Azure

SQL Database Server

Allow Azure Services

Full Access

Selective Access

IP Based Firewall Rules

Azure Services
Point-in-time restore - “oops recovery”

- Auto backups, transactional logs every 5 min
- Backups in Azure Storage and geo-replicated
- Creates a side-by-side copy, non-disruptive
- Backups retention policy: 7, 14 or 35 days
- Automated export of logical backups for long-term backup protection
Geo-restore – Emergency data recovery when you need it most

- Available in all tiers: Basic, Standard and Premium
- Built on geo-redundant Azure Storage
- Recover to any Azure region
Full-Text Search

• Fast and flexible indexing of textual data
• Data types: char, varchar, nchar, nvarchar, text, ntext, image, xml, varbinary(max), or FILESTREAM
• Handles high query volume
• Common use cases:
  • Searching websites, product catalogs, news items and more
  • Document management systems
  • Any applications that need to provide search capabilities over data stored in a SQL Database
XML Indexes

- XML Indexes - improves XQuery-based query performance
- Primary XML Index - speed up access to elements and attributes
  - CREATE PRIMARY XML INDEX XML_Order_Items
  - ON Sales.Order (Items);
- Secondary XML Index – help resolve specific
- XQuery expressions rapidly
Use Familiar Technologies

Transact-SQL
Elastic Database
Elastic Database Pool

Elastic Database

Predictable model for deploying large numbers of databases
Elastic Scale
Canonical cloud app architecture

- Classic 3-tier enterprise architecture:
  - Scale out the front ends to multiple instances is easy
  - Scale the data-tier is more challenging
Vertical: Scale-up or scale-down
Horizontal: Scale-out or scale-in
Elastic Scale architecture
2. DOCUMENT DATABASE
DocumentDB

Fully managed, scalable JSON document database service
Microsoft Azure Data Services

<table>
<thead>
<tr>
<th>SQL Server in a VM</th>
<th>SQL Database</th>
<th>DocumentDB</th>
<th>Tables</th>
<th>Blobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>fully featured RDBMS</td>
<td>transactional processing</td>
<td>fully managed, scalable, queryable, schema-free JSON document database service for modern applications</td>
<td>managed as a service</td>
<td>elastic scale</td>
</tr>
<tr>
<td>rich query</td>
<td></td>
<td></td>
<td>schema-free data model</td>
<td>internet accessible http/rest</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>arbitrary data formats</td>
</tr>
</tbody>
</table>
MongoDB offers:

1. Powerful, expressive query language
2. Rich secondary indexes
3. Strong consistency
4. Flexible data models
5. Vertical and horizontal scalability
6. In-memory performance
MongoDB

MongoDB is the leading Document Database
Gartner the Magic Quadrant for Operational Database Management Systems
MongoDB vs DocumentDB

1. Existing applications which require extra capacity for scale out and can not be migrated
2. Customer has ecosystem of IT resources for support and maintenance
3. Removing CAPEX
4. Mongo MMS compatibility

1. Applications that need managed elastic scale, query over schema free data, native JSON/JavaScript support
2. Customer does not want to add additional IT resources for support and maintenance
3. Avoiding CAPEX and OPEX
4. Built-for-the-cloud database technology
NOSQL vs SQL

1. Unstructured data
2. Collections

1. Structured data
2. Tables
<table>
<thead>
<tr>
<th><strong>NoSQL</strong></th>
<th><strong>SQL</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td><strong>Relational</strong></td>
</tr>
<tr>
<td>Stores data in JSON documents,</td>
<td>Stores data in a table</td>
</tr>
<tr>
<td>key/value pairs, wide column</td>
<td></td>
</tr>
<tr>
<td>stores, or graphs</td>
<td></td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td></td>
</tr>
<tr>
<td>Offers flexibility as not every</td>
<td>Great for solutions where every</td>
</tr>
<tr>
<td>record needs to store the same</td>
<td>record has the same properties</td>
</tr>
<tr>
<td>properties</td>
<td></td>
</tr>
<tr>
<td>New properties can be added on</td>
<td>Adding a new property may</td>
</tr>
<tr>
<td>the fly</td>
<td>require altering schemas or</td>
</tr>
<tr>
<td></td>
<td>backfilling data</td>
</tr>
<tr>
<td>Relationships are often captured</td>
<td>Relationships are often captured</td>
</tr>
<tr>
<td>by denormalizing data and</td>
<td>in normalized model using joins</td>
</tr>
<tr>
<td>presenting all data for an</td>
<td>to resolve references across</td>
</tr>
<tr>
<td>object in a single record</td>
<td>tables</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Good for semi-structured,</td>
<td>Good for structured data</td>
</tr>
<tr>
<td>complex, or nested data</td>
<td></td>
</tr>
<tr>
<td><strong>Schema</strong></td>
<td><strong>Strict schema</strong></td>
</tr>
<tr>
<td>Dynamic or flexible schemas</td>
<td></td>
</tr>
<tr>
<td>Database is schema-agnostic and</td>
<td>Schema must be maintained and</td>
</tr>
<tr>
<td>the schema is dictated by the</td>
<td>kept in sync between</td>
</tr>
<tr>
<td>application. This allows for</td>
<td>application and database</td>
</tr>
<tr>
<td>agility and highly iterative</td>
<td></td>
</tr>
<tr>
<td>development</td>
<td></td>
</tr>
<tr>
<td><strong>Transactions</strong></td>
<td><strong>Supports ACID transactions</strong></td>
</tr>
<tr>
<td>ACID transaction support varies</td>
<td></td>
</tr>
<tr>
<td>per solution</td>
<td></td>
</tr>
<tr>
<td><strong>Consistency &amp; Availability</strong></td>
<td><strong>Strong consistency enforced</strong></td>
</tr>
<tr>
<td>Eventual to strong consistency</td>
<td>Consistency is prioritized</td>
</tr>
<tr>
<td>supported, depending on solution</td>
<td>over availability and performance</td>
</tr>
<tr>
<td>Consistency, availability, and</td>
<td></td>
</tr>
<tr>
<td>performance can be traded to</td>
<td></td>
</tr>
<tr>
<td>meet the needs of the application</td>
<td></td>
</tr>
<tr>
<td>(CAP theorem)</td>
<td></td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td><strong>Insert and update performance</strong></td>
</tr>
<tr>
<td>Performance can be maximized by</td>
<td>is dependent upon how fast a</td>
</tr>
<tr>
<td>reducing consistency, if needed</td>
<td>write is committed, as strong</td>
</tr>
<tr>
<td></td>
<td>consistency is enforced.</td>
</tr>
<tr>
<td></td>
<td>Performance can be maximized</td>
</tr>
<tr>
<td></td>
<td>by using scaling up available</td>
</tr>
<tr>
<td></td>
<td>resources and using in-memory</td>
</tr>
<tr>
<td></td>
<td>structures.</td>
</tr>
<tr>
<td>All information about an entity</td>
<td>Information about an entity</td>
</tr>
<tr>
<td>is typically in a single record</td>
<td>may be spread across many</td>
</tr>
<tr>
<td>so an update can happen in one</td>
<td>tables or rows, requiring</td>
</tr>
<tr>
<td>operation</td>
<td>many joins to complete an</td>
</tr>
<tr>
<td></td>
<td>update or a query</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td><strong>Scaling is typically achieved vertically with more server resources</strong></td>
</tr>
<tr>
<td>Scaling is typically achieved</td>
<td></td>
</tr>
<tr>
<td>horizontally with data</td>
<td></td>
</tr>
<tr>
<td>partitioned to span servers</td>
<td></td>
</tr>
</tbody>
</table>
3. STORAGE
Rodzaje struktur

1. BLOB
2. Files
3. Queue
4. Tables
Azure Storage Architecture

Massive Scale Out & Auto Load Balancing Index Layer

Distributed Replication Layer

REST

Blob Head

Table Head

Queue Head

File Share Head

REST

SMB
Microsoft Azure

Storage Blob
<table>
<thead>
<tr>
<th>Block Blob</th>
<th>vs</th>
<th>Page Blob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Targeted at streaming workloads</td>
<td>1. Targeted at random read/write workloads</td>
<td></td>
</tr>
<tr>
<td>2. Each blob consists of a sequence of blocks</td>
<td>2. Each blob consists of an array of pages</td>
<td></td>
</tr>
<tr>
<td>3. Each block is identified by a Block ID</td>
<td>3. Each page is identified by its offset from the start of the blob</td>
<td></td>
</tr>
<tr>
<td>4. Size limit 200GB per blob</td>
<td>4. Size limit 1TB per blob</td>
<td></td>
</tr>
</tbody>
</table>
Blob Storage Concepts

http://{account}.blob.core.windows.net/{container}/{blobname}
Blob Details – Blob always accessed by name

Can include ‘/’ or other delimiter in name

e.g. /<container>/myblobs/smurf.png

blob name
Sharing Files – The old way

Setup an IaaS VM to host a File Share backed by an IaaS Disk

Write code to find the IaaS File Share from the rest of the VMs in your service.

Write some code to provide high availability

Handle host upgrades, node failures

You can only access the File Share from other VMs
Sharing Files – The old way

IaaS VM

IaaS VM
(Sharing IaaS Disk)

PaaS VM

Backup IaaS VMs
(Mount/Share after failover)
Azure Files

Shared Network File Storage for Azure

Availability, durability, scalability are managed automatically

Supports two interfaces: SMB and REST
Azure Files – Usage

Share data across VMs and applications

Share settings throughout services

Dev/Test/Debug
Microsoft Azure
Storage Queue
Why use a Queue?

1. Queue length reflects how well the backend processing nodes are doing.
2. Decouples the application.
3. Flexibility of efficient resource usage within an application.
4. Absorb traffic bursts and reduce the impact of individual component failures.
Queue Components

Storage Account: All access to Azure Storage is done through a storage account.

Queue: A queue contains a set of messages.

Message: any format and limit to 64KB.
Queue-based Load Levelling Pattern

Producers

$P_1$

$P_2$

Consumers

$C_1$

$C_2$

Queue

4 3 2 1
Queue Considerations

1. Messages are ordered but not guaranteed FIFO
2. Message will be processed at least once
3. Message may be processed more than once
4. DequeueCount increases every time
   -> Processing must be idempotent
Microsoft Azure

Storage Table
Table Storage Concepts

Account
- contoso

Table
- customers
- photos

Entity
- Name = ...
- Email = ...
- EMailAdd=
- Photo ID = ...
- Date = ...
- Photo ID = ...
- Date = ...
Table Storage Details

Entity Properties

1. PartitionKey & RowKey are mandatory properties
2. Composite key which uniquely identifies an entity
3. They are the only indexed properties
4. Defines the sort order
Table Storage Details

1. Supports full manipulation (CRUD)
2. Including Upsert and Entity Group
3. Transactions
4. Tables can have metadata
4. QUICK ACCESS TO DATA
Application Instance A

Shared cache service

Application Instance B

If the shared cache service is unavailable, application instances can continue to function using local, private caches.

If the shared cache service is unavailable, application logic populates the local cache from the database.

SQL
Redis is an open source (BSD licensed), in-memory **data structure store**, used as database, cache and message broker.
REDIS

It supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs and geospatial indexes with radius queries. Redis has built-in replication, Lua scripting, LRU eviction, transactions and different levels of on-disk persistence, and provides high availability via Redis Sentinel and automatic partitioning with Redis Cluster.
5. SEARCH
Azure Search
Azure Search

Perfect for enterprise cloud developers, cloud software vendors, cloud architects who need a fully-managed search solution.
Azure Search

Embed a sophisticated search experience into web and mobile applications without having to worry about the complexities of full-text search and without having to deploy, maintain or manage any infrastructure.
Search Functionality

- Simple HTTP/JSON API for creating indexes, pushing documents, searching
- Keyword search with user-friendly operators (+, -, *, ",", etc.)
- Hit highlighting
- Faceting (histograms over ranges, typically used in catalog browsing)
Search Functionality

- Suggestions (auto-complete)
- Rich structured queries (filter, select, sort) that combines with search
- Scoring profiles to model search result relevance
- Geo-spatial support integrated in filtering, sorting and ranking
Summary
HDInsight
Big Data Insights

Microsoft Azure HDInsight

• Microsoft instance of Hadoop distribution running as service
• Support existing ecosystem from Hive, Pig, etc
• Includes HBase and Storm
• Extended to support Excel and BI tooling
• Integration with diverse sources of data
Big Data Insights

Microsoft Azure Machine Learning

- Democratized platform for Machine Learning
- Fully-managed cloud service for building predictive analytics solutions
- Elastic, pay as you go with low operating costs
- Extend with Power BI, HDInsight and cloud hosted data
- Extensible, supports R and Python
<table>
<thead>
<tr>
<th>Feature</th>
<th>Relational DB</th>
<th>Hadoop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data size</td>
<td>Gigabytes (Terabytes)</td>
<td>Petabytes (Hexabytes)</td>
</tr>
<tr>
<td>Access</td>
<td>Interactive and Batch</td>
<td>Batch</td>
</tr>
<tr>
<td>Updates</td>
<td>Read / Write many times</td>
<td>Write once, Read many times</td>
</tr>
<tr>
<td>Structure</td>
<td>Static Schema</td>
<td>Dynamic Schema</td>
</tr>
<tr>
<td>Integrity</td>
<td>High (ACID)</td>
<td>Low</td>
</tr>
<tr>
<td>Scaling</td>
<td>Nonlinear</td>
<td>Linear</td>
</tr>
</tbody>
</table>
Internet of Things

With Microsoft Azure IoT services, you can monitor assets to improve efficiencies, drive operational performance to enable innovation, and leverage advanced data analytics to transform your company with new business models and revenue streams.
# Azure Stream Analytics

## Event producers
- Applications
- Legacy IOT (custom protocols)
- Devices
- IP-capable devices (Windows/Linux)
- Low-power devices (RTOS)

## Collection
- Cloud gateways (web APIs)

## Ingestor (broker)
- Field gateways

## Transformation
- Stream processing
- Storage adapters

## Long-term storage
- Service bus
- Azure DBs
- Azure storage
- HDInsight

## Presentation and action
- Web/thick client dashboards
- Search and query
- Data analytics (Excel)
- Devices to take action
Iterative exploration - Process key data into business intelligence using Hadoop
QUESTIONS?

Bartosz Zaremba

✉️ bartosz.zaremba@live.com
🔗 @art_base
📞 +48 506 46 46 95