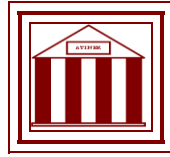


**Athens Institute for Education and Research**  
**ATINER**



**ATINER's Conference Paper Series**  
**COM2012-0019**

**Using Spatial Data in Windows Azure  
(SQL azure and azure blob storage)**

Mihail Mateev  
Chief Assistant Professor  
UACEG, Faculty of Structural Engineering,  
Department of Computer-Aided Engineering  
Bulgaria

Athens Institute for Education and Research  
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece  
Tel: + 30 210 3634210 Fax: + 30 210 3634209  
Email: [info@atiner.gr](mailto:info@atiner.gr) URL: [www.atiner.gr](http://www.atiner.gr)  
URL Conference Papers Series: [www.atiner.gr/papers.htm](http://www.atiner.gr/papers.htm)

Printed in Athens, Greece by the Athens Institute for Education and Research.  
All rights reserved. Reproduction is allowed for non-commercial purposes if the  
source is fully acknowledged.

ISSN **2241-2891**

## An Introduction to ATINER's Conference Paper Series

ATINER started to publish this conference papers series in 2012. It includes only the papers submitted for publication after they were presented at one of the conferences organized by our Institute every year. The papers published in the series have not been refereed and are published as they were submitted by the author. The series serves two purposes. First, we want to disseminate the information as fast as possible. Second, by doing so, the authors can receive comments useful to revise their papers before they are considered for publication in one of ATINER's books, following our standard procedures of a blind review.

Dr. Gregory T. Papanikos  
President  
Athens Institute for Education and Research

This paper should be cited as follows:

**Mateev, Mihail (2012) "Using spatial data in windows azure (SQL azure and azure blob storage)" Athens: ATINER'S Conference Paper Series, No: COM2012-0019.**

Using spatial data in windows azure  
(SQL azure and azure blob storage)

Mihail Mateev  
Chief Assistant Professor  
UACEG, Faculty of Structural Engineering,  
Department of Computer-Aided Engineering  
Bulgaria

Abstract

Windows Azure is Microsoft's implementations of Cloud Computing. Geographic Information Systems are one of the areas that can take advantage of Windows Azure, because they need to use huge amount of data, computing resources, communication, scalability and accessibility from different places over the world. This article is about the spatial data - the foundation of Geographic Information Systems.

One of the key points in any GIS architecture is how to store spatial data. Windows Azure proposes new opportunities that could be used from GIS systems. This paper is focused on two solutions: using spatial data in SQL Azure and maintain spatial data from Windows Azure Blob Storage. SQL Azure provides opportunities to use spatial data in a relational database (with the possibility of migration of data from MS SQL Server). Azure Blob Storage solution is to use file-based spatial data such as shapefiles, WKT, XML.

In this research different solutions that use spatial data in Windows Azure are compared with traditional client-server and multi-tiered solutions. As a result of the comparison is a technical and economic substantiation on the advantages of Cloud Computing in particular Windows Azure in Geographic Information Systems.

Keywords: Cloud Computing, Windows Azure, Spatial Data, SQL server, Windows Azure SQL Database, Blob Storage

Acknowledgements:

- Assoc. Prof. Dr. Eng. Guergana Mollova, Faculty of Structural Engineering, Department Computer-Aided Engineering - head of department
- Jason Beres - VP, Product Management, Community and Evangelism at Infragistics

Contact Information of Corresponding author:  
Mihail Mateev

Chief Assist. Prof. Eng. Mihail Mateev  
UACEG, Computer-Aided Engineering  
Senior Technical Evangelist at Infragistics

E-mail:

[michael@mateev.net](mailto:michael@mateev.net)

[mikeamm\\_fce@uacg.bg](mailto:mikeamm_fce@uacg.bg)

[mmateev@infragistics.com](mailto:mmateev@infragistics.com)

Mobile:

+359887524762

## INTRODUCTION

### Spatial Data

Spatial data describes the position, shape and orientation of objects in space. Data, describing the position and shape of objects on the earth is also known as geospatial data. Spatial data have a versatile use in modern life – from complex analyzes in the field of building construction, logistics and financial analysis to everyday needs like search of address.

Spatial data have many applications in real life

Finance:

- Analyzing sales trends

Construction:

- Analyzing the best placement depending of different criteria
- Seismic analysis

Logistics and Transportation:

- Navigating to a destination using a GPS device.
- Allowing customers to track the deliveries
- Finding the optimum route for transportation tasks
- Reporting geospatial information on the map rather than in a tabular or chart format

### Geospatial Features

When we talk about geospatial features there are two basic models for digitally storing geographic data: vector data and raster data. More details about geospatial features could be found in [1] and [2]

### Spatial Reference System, Datum, Reference Ellipsoid, Geographic and Projected Coordinate Systems

The **spatial reference systems** are the ways of describing positions in space.

The earth is a very complex shape. On the surface, we can see by looking around us that there are irregular topographical features such as mountains and valleys. Geophysicists have a specific word to describe the shape of the earth – the **geoid**.

Spatial reference systems normally define positions on the earth's surface based on a simple model that approximates the geoid. This approximation is called a **reference ellipsoid**.

The simplest reference ellipsoid that can be used to approximate the shape of the geoid is a perfect sphere.

The most important property of a spheroid is that, unlike the geoid, a spheroid is a regular shape that can be exactly mathematically described by two parameters – the length of the semi-major axis (which represents the radius of the earth at the equator), and the length of the semi-minor axis (the radius of the earth at the poles). The flattening ratio of an ellipsoid,  $f$ , is used to describe how much the ellipsoid has been “squashed”, and is calculated as:

$$f = (a - b) / a$$

where  $a$  = length of semi-major axis,  $b$  = length of semi-minor axis.

### **Regional Variations in Ellipsoids**

There is not a single reference ellipsoid that best represents every part of the whole geoid. Some ellipsoids, such as the World Geodetic System 1984 (WGS84) ellipsoid, provide a reasonable approximation of the overall shape of the geoid. Other ellipsoids approximate the shape of the geoid very accurately over certain regions of the world, but are much less accurate in other areas. An ellipsoid is just an abstract mathematical shape; in order to use it as the basis of a spatial reference system; we need to correlate coordinate positions on the ellipsoid with real-life locations on the earth. We do this by creating a frame of reference points. When packaged together, the properties of the reference ellipsoid and the frame of reference points form a geodetic **datum**.

**Datum** is a set of reference points on the Earth's surface against which position measurements are made, and (often) an associated model of the shape of the earth (reference ellipsoid) to define a geographic **coordinate system** (Fig. 3).

### **Geographic Coordinate System**

In a geographic coordinate system, any position on the earth's surface can be defined using two coordinates: latitude and longitude. It defines positions on a three-dimensional, round model of the earth [2].

Since a map is a flat, two-dimensional surface, we can then describe positions on the plane of that map using familiar two-dimensional Cartesian coordinates in the  $x$  and  $y$  axes. This is known as a **projected coordinate system** [2].

**Projected coordinate systems** describe the position of points on the earth surface as they lie on a flat, two-dimensional plane [2].

### **Spatial Reference Identifiers (SRIDs)**

Every time we state the latitude and longitude, or  $x$  and  $y$  coordinates, that describe the position of a point on the earth, we must also state the associated spatial reference system from which those coordinates were obtained.

Without the extra information contained in the spatial reference system, a coordinate tuple is just an abstract set of numbers in a mathematical system. The spatial reference system takes the abstract coordinates from a geographic or projected system and puts them in a context so that they can be used to identify a real position on the earth's surface.

Various authorities allocate easily memorable, unique integer reference numbers that represent all of the necessary parameters of a spatial reference system. These reference numbers are called spatial reference identifiers (SRIDs).

One authority that allocates SRIDs is the European Petroleum Survey Group (EPSG), and its reference identification system is implemented in SQL Server 2012.

Some examples of SRIDs assigned by the EPSG that can be used in SQL Server are:

- 4269 (North American Datum 1983)
- 32601 (UTM Zone 1 North)

- 4326 (World Geodetic System 1984)
- 32136 (Tennessee State Plane Coordinate System)

### **SQL Server Spatial**

Microsoft SQL Server supports spatial data from SQL Server 2008. The first implementations of SQL Azure were on SQL Server 2008.

This article discusses the current version of SQL Server 2012 and the implementation of SQL Azure.

### **Geospatial Data Types**

With the launch of SQL Server 2008 developers have access to two spatial data types that are included with the product: **Geometry** and **Geography** (Fig.1). These types are supported in SQL Server 2012 and SQL Azure

- GEOMETRY - This data type stores data in projected planar surfaces
- GEOGRAPHY - This data type stores data in an ellipsoidal model.

The Geometry spatial data type in SQL Server is used to represent information in a uniform 2-dimensional plane.

The SQL Server Geography data type uses Latitude and Longitude angles to identify points on the Earth. Latitude measures how far North (or South) of the Equator a point is, while Longitude measures how far East (or West) of a Prime Meridian a point is.

Classes, used in Geometry and Geography are the same. That is the reason to describe a Geometry classes hierarchy: (Figure 1). More information about spatial types you could find in [1] and [2].

### **Exchange spatial data with SQL Server**

#### **Well-Known Text of a Spatial Reference System**

SQL Server maintains a catalog view, `sys.spatial_reference_systems`, in which it stores the details of all 392 supported geographic spatial reference systems. The parameters of each geographic spatial reference system in `sys.spatial_reference_systems` are stored in the `well_known_text` column using the Well-Known Text (WKT) format, which is an industry-standard format for expressing spatial information defined by the Open Geospatial Consortium (OGC) [1] and [2].

### **SPATIAL DATA AND WINDOW AZURE**

#### **SQL Azure and Spatial Data**

SQL Azure is based on SQL Server 2012. It supports all spatial features that you have in SQL Server.

The Geometry and Geography data types are implemented as User Defined Types (UDT) written in .NET. They are automatically installed with the server, and are available for use by any SQL Server 2008 or newer database.

**Migrating Spatial Data to SQL Azure:** When we talk about data migration usually it means migration from local SQL Server to SQL Azure. It is also possible to have spatial data in specific file formats (shapefile). In this case you could use the same conversion tools to import spatial data from files for both: local SQL Server and SQL Azure.

- **Migrating Spatial Data to SQL Azure**



If you are using Microsoft SQL Management Studio for SQL Server 2008 R2 it could be not possible to maintain spatial data in SQL Azure. The reason is that this version of SQL Management Studio is released when SQL Azure didn't support spatial data and now they are no updated to support this feature.

One possibility is to use SQL Azure Migration Wizard:  
<http://sqlazuremw.codeplex.com/>

Another solution is to use SQL Management Studio for SQL Server 2012.

### **SQL Azure Performance:**

Windows Azure (and SQL Azure) won't make your app automatically scalable if it wasn't before. All the good practices and recommendations for high performance and scalability are still valid and should be used [6].

### **RESEARCH**

The objective of this study is to examine and compare the possibilities of using spatial data in Windows Azure. The study compares similar systems using spatial data in Windows Azure and those using local servers.

#### **1. SQL Server vs. SQL Azure**

It is difficult to compare SQL Server vs. SQL Azure performance. When you have a select query, that returns data most of the time is because of traffic latency.

It is possible to measure time to receive results after query has executed in two different ways:

1. Using SET STATISTICS TIME ON  
 SET STATISTICS TIME ON  
 go  
 SELECT \* from [Database].[User].[Table];
2. Using variables to get time before and after the query  
 declare @t1 datetime, @t2 datetime  
 set @t1=getdate()  
 SELECT \* from [Database].[User].[Table];  
 set @t2=getdate()  
 select @t2-@t1

In the research there is a comparison between queries with spatial data, executed on local SQL 2008 Express, SQL Azure (1GB Web Database) and SQL Azure (10GB Business Database) – **Figure 2**.

Results show that the main time is because of traffic data. When you change database type and size in SQL Azure the time is almost the same doesn't matter of database parameters. SQL Azure performance can be improved using the same guidelines like in SQL Server when you have a single database. You could improve performance using scaling-out - [6] and [7].

The time for transportation has the biggest impact on performance. SQL Azure scalability could has impact on amount of data. The performance depends on other components of the system [3]

- **Scaling-Out with SQL Azure: Federations**

Federations make it easier to build, deploy and manage scalable database tiers by allowing server side support for horizontal partitioning and sharding within SQL Azure. Simply, Federations provide elasticity as well as increased performance and scale by horizontally partitioning set of objects of a database across multiple databases. Each physical database in this architecture is considered to be a federation

member (a.k.a a shard). A federation can contain one or typically many members, and administrators can add or remove federation members dynamically and repartition and redistribute application workload without any downtime. Federations could be used with spatial data as well as with an attribute data. More details about federations you could find in [6].

### **Migrating Applications with Spatial Data (SQL Spatial) to Windows Azure**

If you have web applications based on Microsoft .NET related technology: (ASP.Net, Silverlight) that use Spatial Data from SQL Server you need to follow the steps above to publish these applications to Windows Azure

1. Create a new SQL Azure database
2. Migrate a sample database – SqlSpatialDemo from Local SQL Server to SQL Azure
3. Create a new Windows Azure Storage Account and a Hosted Service
4. Add a Web Role project in Solution
5. Edit settings in Web.config and ServiceReferences.ClientConfig
6. Publish the Windows Azure Cloud Service:

#### **1. Spatial Data and Windows Azure Blob Storage**

When we talk about spatial data in Windows Azure is usually understood SQL Azure. Windows Azure presents other options when you file your sources for spatial data is to use Windows Azure Blob Storage. It is possible to use blob storage to store spatial data in different file formats – shapefiles, WKT, XML etc. In this paper and research are used shapefiles.

- Windows Azure Storage Abstractions

Accessing the Windows Azure Storage is done via a storage account. A storage account can have many blob containers (Figure 3).

- Windows Azure Blobs

A container is a user-defined set of blobs that has only properties, which include a list of the blobs it contains. Containers don't store data directly.

The word blob stands for binary large objects. Blobs store binary data, such as images, XML documents, compressed (zipped or gzipped) files, and other content as an arbitrary array of bytes within a container that's associated with a storage account.

Blob URL format is:  
 http://<Account>.blob.core.windows.net/<Container>/<BlobName>

Example:

- Account – sally
- Container – music
- BlobName – rock/rush/xanadu.mp3
- URL: <http://sally.blob.core.windows.net/music/rock/rush/xanadu.mp3>

Two Types of Blobs:

##### **1. Block Blob**

A Block blobs is a collection of blocks, where each block has a block ID (of 64 bytes). A Block Blob can be up to 200GB. You can upload a maximum of 64MB of data in a Put Blob operation to create a blob. If it is larger than 64MB it must be sent in multiple chunks of no more than 4MB, with Put Block operations.

##### **2. Page Blob**

A Page Blob can be up to 1TB is size and is organized into 512 byte pages within the block. Any point in the blob can be accessed for read or write operations by using the offset from the start of the blob (Figure 3.1).

- Spatial Data in Windows Azure Blob Storage

Spatial data in Azure Blob Storage could be any kind of file based spatial data: as shapefiles, WKT, XML.

Blob storage can be accessed via abstractions account, container and blob name scalability is supported “out of the box”. Data is not accessed via “file structure” and when you have more space on different servers the system supports scalability. Applications, created to access small amount of data could be used in the same way with a huge data. Data could be accessed from different clients via WEB services, hosted in WEB roles. It is possible when the system receives many requests to use many instances of the WEB Roles behind a load balancer.

To implement an application using Spatial Data in Windows Azure Blob Storage you need:

- Have an appropriate spatial data
- Implement a WCF to upload and download files, using Windows Azure Blob Storage.
- Implement a client application that could communicate with WCF services and render the spatial data.

More details about how to use spatial data from Windows blob storage you could find in [9].

**Prototypes:**

In this research are used two prototypes:

- Silverlight Application that uses spatial data from SQL Azure
- Silverlight Application that renders spatial data from shapefiles. Shapefiles are placed Windows Azure Blob Storage.

Both prototypes use Infragistics XamMap component to render spatial data.

Prototype that uses data from Windows Azure Blob Storage implements shapefiles downloading via WCF services in Isolated Storage. Stored shapefiles are loaded and rendered via XamMap component.

**Results:**

As a result of the research are different real workable solutions to spatial data in Windows Azure was compared with traditional client-server and multi-tiered solutions. As a result of the comparison is a technical and economic substantiation on the advantages of Cloud Computing in particular Windows Azure in Geographic Information Systems.

Figure 2 shows comparison when you have local SQL Server and different SQL Azure databases. The most time consuming element is time to receive data. Solution that uses data from blob storage is a slower when you have a small amount of data, but it could be faster with a huge amount of data or via data size and data transfer optimization.

Figure 4 demonstrates results when test Windows Azure applications are scaled for different number of spatial features.

**CONCLUSION**

With Windows Azure it is possible to use spatial data in several ways:

- SQL Azure
- Windows Azure Blob Storage
- Windows Azure Table Store

The current article covers the first two approaches: using SQL Azure and Windows Azure Blob Storage.

Solutions with SQL Azure are easier to implement and faster for small amount of data. Blob storage offer more flexibility and scalability when you have a huge amount of data. In this case you could use many web role instances to improve performance when there are many requests to services.

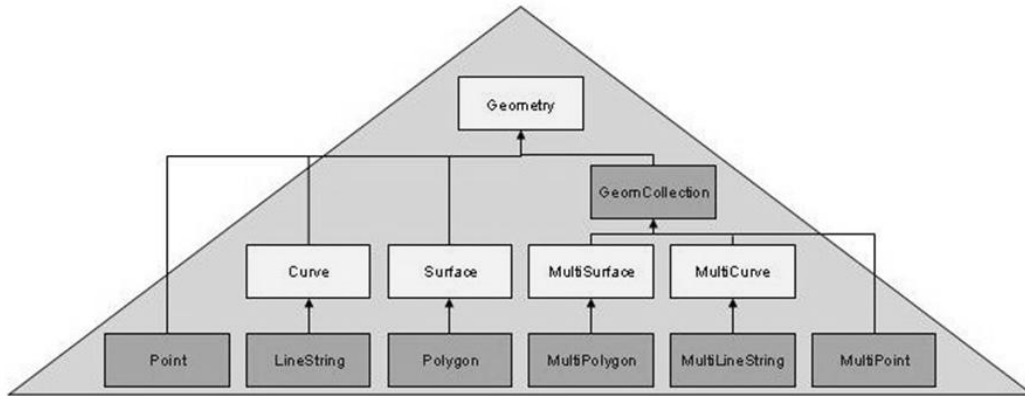
It is possible to have multi-tier applications with SQL Azure using WEB services. In this case it is possible to use web role scalability and improve performance in some cases. When you access spatial data from remote servers obviously Windows Azure offers better performance and scalability. When Azure applications are compared with solutions in the local network time to transfer the spatial data has the biggest impact on performance.

## REFERENCES

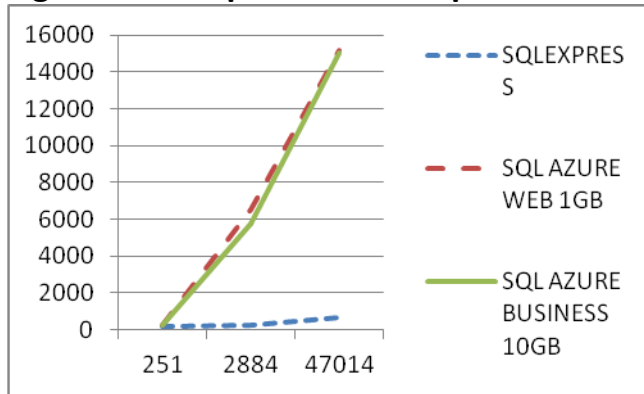
1. Alastair Aitchison (2012) Pro Spatial with SQL Server 2012, Apress: Springer Science + Business Media
2. Alastair Aitchison (2009) Beginning Spatial with SQL Server 2008 , Apress: Springer Science + Business Media
3. V. Subramanian, L. Wang (2010) 'Rapid Processing of Synthetic Seismograms Using Windows Azure Cloud'. Paper presented at the 2nd International Conference on Cloud Computing 2010, Nov 30 - Dec 3, 2010, Indianapolis, USA
4. INTERNATIONAL STANDARD ISO/IEC 13249-3 Fourth edition (2011-04-15) Information technology — Database languages — SQL multimedia and application packages — Part 3: Spatial, International Electrotechnical Commission
5. Bob Beauchemin (2009) Visualizing Spatial Data, MSDN Magazine, November 2009. Available at <http://msdn.microsoft.com/en-us/magazine/ee335706.aspx> [November 2009]
6. Shaun Ziyang Xu (2012) SQL Azure Federation – Introduction, geekswithblogs.net. Available at <http://geekswithblogs.net/shaunxu/archive/2012/01/07/sql-azure-federation-ndash-introduction.aspx> [7 January 2012]
7. Trace Young (2011) SQL Azure Performance and Elasticity Guide, Microsoft TechNet. Available at <http://social.technet.microsoft.com/wiki/contents/articles/sql-azure-performance-and-elasticity-guide.aspx> [21 Jun 2011]
8. Z. Hill, J. Li, M. Mao, A. Ruiz-Alvarez, and M. Humphrey (2010). Early observations on the performance of windows azure. In the 1st Workshop on Scientific Cloud Computing (ScienceCloud 2010). ACM, 2010.
9. Mihail Mateev(2011) Infragistics Community Site. Using spatial data in the Windows Azure Blob Storage. Available at [http://blogs.infragistics.com/blogs/mihail\\_mateev/archive/2011/04/17/using-spatial-data-in-the-windows-azure-blob-storage.aspx](http://blogs.infragistics.com/blogs/mihail_mateev/archive/2011/04/17/using-spatial-data-in-the-windows-azure-blob-storage.aspx) [17 April 2011]

**Figure 1. Geospatial Data Types**

The Geometry Classes define a hierarchy as follows:

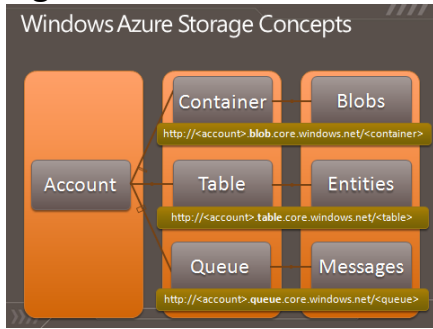


**Figure 2. Comparison of the performance of SQL Server and SQL Azure**

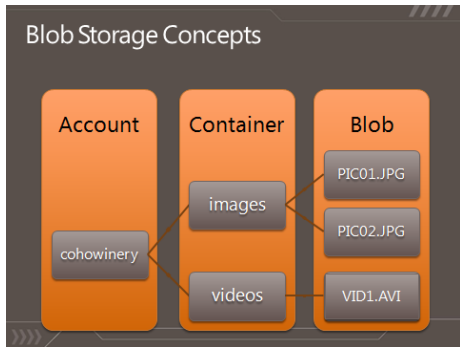


NUMBER OF ITEMS	251	2884	47014
SQL EXPRESS	181.6	242.4	682.8
SQL AZURE WEB 1GB	269	6510	15201.2
SQL AZURE BUSINESS 10GB	211.4	5786.2	15049.8

**Figure 3. Windows Azure Blob Storage Architecture**



**3.1 Windows Azure Blobs**



**Figure 4. Execution time and resources according to the number of objects**

