A short introduction into selecting a NoSQL database for deployment into the Windows Azure Platform.
Introduction
This document provides some background into selecting a NoSQL database for the Windows Azure Platform. It starts by providing some questions to consider when considering which NoSQL database to use. It continues with brief introductions to the main types of NoSQL database - Hadoop, key-value, column store, and graph – along with an example of each that is deployable to Windows Azure. Each introduction concludes with a statement of the typical use case for each type of database. Finally, the document cautions about the lack of quantitative data for Windows Azure deployments of these NoSQL databases.

Questions When Selecting a NoSQL Database
Big Data is often discussed in terms of the 4 Vs.

- Volume – How much data is there?
- Variety – what are the sources of data?
- Velocity – How fast is the data generated?
- Variability – How much does the meaning of the data vary?

Other questions that guide the choice of database are:

- Consistency – What is the consistency model for the application data?
- Analytics – What are the data analysis needs for the data?
- Queries – What is the nature of the dominant queries on the data?

Another consideration when selecting a NoSQL database is whether or not there are appropriate client libraries?

Hadoop
Hadoop is the canonical NoSQL Big Data solution. The name refers to a platform comprising several components including:

- HDFS – the distributed file system used by Hadoop
- Map/reduce – the Java programming environment for Hadoop
- Pig – a programming environment for manipulating HDFS datasets
- Hive – a programming environment for analyzing HDFS datasets using a SQL-like language
- Sqoop – a tool allowing data to be transferred back and forth between a relational database and HDFS.
- Flume – an application to pipe data from various sources into HDFS.
- Zookeeper – the distributed coordination service for Hadoop.
- Mahout – a machine learning library for HDFS.
Hadoop clusters can handle extremely large quantities of data.

Pig and Hive were developed to provide high-level programming abstractions that removed the need to write map-reduce jobs in Java. Programs written using these tools create and invoke map-reduce jobs automatically. Pig provides a convenient way to handle data variety by manipulating it into standardized form. Hive is used to provide a schema for the data and then query it using a SQL-like schema.

The Hadoop on Azure preview supports small Hadoop clusters in which data can be stored either in HDFS or in Windows Azure Blob Storage. It supports both Pig and Hive and also provides an extension allowing Hadoop jobs to be invoked from Excel. Sqoop can be used to transfer data to and from Windows Azure SQL Database, although care has to be taken since it does not handle transient connection failures. Flume is typically used to aggregate log files from disparate sources and persist them into HDFS. However, it can’t be used to persist data into HDFS in the Hadoop on Azure preview, and there is no Flume sink that persists data into Blob Storage. The significance of this is mitigated by the fact that cloud services can write directly to Blob Storage which Hadoop on Azure can use.

The core use of Hadoop is for the batch-oriented analysis of high-scale data. Hadoop comprises a sophisticated ecosystem of applications that allow data scientists to analyze the data without developer interaction. Hadoop is not intended to be the backing database for a real-time application, although it can be used as a destination for log data from the application.

Key-Value: Windows Azure Table Service
A key-value store is a NoSQL database that associates one or more values with a primary key.

The Windows Azure Table Service is a managed key-value store provided as part of the Windows Azure Platform. A single table can hold up to 100TB of data that, with appropriate choice of partition key, autoscales across storage nodes. The Table Service has limited query capability with only a single index. There is no backup/restore capability, so that a backup can only be made by physically retrieving all the data and a restore requires that all the data be reinserted. The Table Service uses strong consistency for all writes.

Typically, key-value stores have limited indexing capability and are good for storing high-scale data where a simple query capability is sufficient. The Table Service benefits from being a managed service.

Column-Store Database: Cassandra
A column-store database is a variant of a key-value store in which each row comprises possibly millions of columns, each of which can contain disparate data structures. Each row can be regarded as representing a map of a hashed key into an array of columns.

Cassandra is a popular column-store database developed initially by Facebook. Cassandra supports millions of columns in each row and, since it is schemaless, each row can have different columns.
Cassandra distributes data among nodes depending on a partitioning algorithm. This distribution is further modulated by both the write and read quorum specified, allowing data consistency to be tuned up to and including full consistency. All nodes in Cassandra are peers, although storing different data ranges, so that there is no primary node that can limit performance.

Cassandra is developed in Java. DataStax provides open-source and commercial versions for both Windows and Linux. While it is possible to deploy Cassandra to a Windows Azure worker role, it is significantly easier to deploy it into multiple Virtual Machine instances. Microsoft has provided a tutorial showing how to do deploy a Cassandra cluster into several Linux Virtual Machines. In Windows Azure IaaS, a Cassandra database can be backed up by taking a snapshot of the blob backing the data drives on each Cassandra node.

Cassandra has client libraries for .NET, Node.js, Java, PHP and Python.

Cassandra has a very high write throughput so is popular for storing data such as time-series data, where data can be stored as columns. This allows the efficient querying of time-range data. Its initial use at Facebook and other high-profile companies such as Netflix has made Cassandra a popular NoSQL database.

**Document Database: MongoDB**

A document database stores complex documents or data structures.

MongoDB is a popular document database that stores structured data in BSON format (the binary form of JSON). MongoDB supports sharing of data across multiple nodes. It also provides high availability through replica sets. MongoDB uses a primary node for all writes, which can then be replicated asynchronously to one or more replicas. It can also be configured for full consistency. It does not provide its own backup/restore capability but this can be handled by taking a snapshot of the blob backing the data drive on each node of the MongoDB database.

10gen and Microsoft have collaborated in providing templates allowing the deployment of a MongoDB cluster to both Windows Azure PaaS and IaaS.

MongoDB is a good fit when highly-structured objects must be serialized. It performs best when all the data can reside in memory.

**Graph Database: Neo4J**

A graph database is focused on storing relationships between items. It provides native support for more complex relationships than simple foreign-key relationships. Datasets in graph databases are typically non-tabular.
Neo4J is the “world’s leading graph database.” It is implemented in Java and can be deployed into a Windows Azure PaaS worker role or IaaS. There are RESTful client APIs for Java, .NET, Node.js and Python.

Graph databases are useful for storing networks such as a social graph.

**Performance in Windows Azure**

The Windows Azure Storage Team has published scalability targets for the Windows Azure Storage Service including the Table Service. There don’t appear to be any published statistics providing comparative performance of other NoSQL databases hosted in Windows Azure.

Specifically, there doesn’t appear to be any equivalent to the performance statistics that Netflix is publishing for Cassandra on EC2. For example, the smallest instance size that Netflix tests for Cassandra on EC2 has 15GB of RAM. Netflix primarily uses an m2.4xlarge EC2 AMI, which has 68GB of RAM and 8CPU threads.

For some NoSQL databases, performance may be negatively implicated by the limited memory available to an instance. Furthermore, the scalability targets for the Storage Service make it possible that optimal performance requires each drive hosting data to be hosted in a different storage account. Given this lack of information it is best to verify the performance of any NoSQL database deployed into Windows Azure.