Windows Azure Storage
Scaling Cloud Storage

Andrew Edwards
Microsoft
Agenda: Windows Azure Storage

- Overview
- Architecture
- Key Design Points
Windows Azure Storage

- Cloud Storage - Anywhere and anytime access
  - Blobs, Disks, Tables and Queues
- Highly Durable, Available and Massively Scalable
  - Easily build “internet scale” applications
  - 10+ trillion stored objects
  - 1M+ request/sec on average
- Pay for what you use
- Exposed via easy and open REST APIs
- Client libraries in .NET, Java, Node.js, Python, PHP, Ruby
Abstractions – Blobs and Disks

- **Blobs** – Simple interface to store and retrieve files in cloud
  - Data sharing – share documents, pictures, video, music, etc.
  - Big Data – store raw data/logs and compute/map reduce over data
  - Backups – data and device backups

- **Disks** – Network mounted durable disks for VMs in Azure
  - Mounted disks are VHDs stored in Azure Blobs
  - Move on-premise applications to cloud
Abstractions – Tables and Queues

- Tables – Massively scalable and extremely easy to use NoSQL system that auto scales
  - Key-value lookups at scale
  - Store user information, device information, any type of metadata for your service

- Queues – Reliable messaging system
  - Decouple components/roles
    - Web role to worker role communication
    - Allows roles to scale independently
  - Implement scheduling of asynchronous tasks
  - Building process/work flows
Windows Azure Storage
Windows Azure Data Storage Concepts

Account

Container

Blobs

https://<account>.blob.core.windows.net/<container>

Table

Entities

https://<account>.table.core.windows.net/<table>

Queue

Messages

https://<account>.queue.core.windows.net/<queue>
How is Azure Storage used by Microsoft?

**Xbox:** Uses Blobs, Tables & Queues for Cloud Game Saves, Halo 4, XBox Music, XBox Live, etc.

**Skype:** Uses Blobs, Tables and Queues for Skype video messages and to keep metadata to allow Skype clients to connect with each other.

**Bing:** Uses Blobs, Tables and Queues to provide a near real-time ingestion engine that consumes Twitter and Facebook feeds, indexes them, which is then folded into Bing search.

**SkyDrive:** Uses Blobs to store pictures, documents, videos, files, etc.
For More Information

- Joe’s Talk (1:00pm today)
  - How to use Windows Azure Storage
  - Best Practices

- Windows Azure Developer Website

- Windows Azure Storage Blog

- SOSP Paper/Talk
Architecture
Design Goals

Highly Available with Strong Consistency
- Provide access to data in face of failures/partitioning

Durability
- Replicate data several times within and across regions

Scalability
- Need to scale to zettabytes
- Provide a global namespace to access data around the world
- Automatically scale out and load balance data to meet peak traffic demands

Additional details can be found in the SOSP paper:
Windows Azure Storage Stamps

Access blob storage via the URL: http://<account>.blob.core.windows.net/

- Intra-stamp replication
- Inter-stamp (Geo) replication
Architecture Layers inside Stamps

- **Front-end Layer**
  - REST front-end (blob, table, queue)
  - Authentication/authorization
  - Metrics/logging

- **Partition Layer**
  - Understands our data abstractions, and provides optimistic concurrency
  - Massively scalable index
  - Log Structured Merge Tree
    - Each log (stream) is a linked list of extents

- **Distributed File System Layer**
  - Data persistence and replication (JBOD)
  - Data is stored into a file called extent, which is replicated 3 times across different nodes (UDs/FDs)
  - Append-only file system
Key Design Points
All writes are appends to the end of a log, which is an append to the last extent in the log.

Write Consistency across all replicas for an extent:
- Appends are ordered the same across all 3 replicas for an extent (file)
- Only return success if all 3 replica appends are committed to storage
- When extent gets to a certain size or on write failure/LB, seal the extent’s replica set and never append anymore data to it

Write Availability: To handle failures during write
- Seal extent’s replica set
- Append immediately to a new extent (replica set) on 3 other available nodes
- Add this new extent to the end of the partition’s log (stream)
- Better availability, durability compared to Quorum-based approach
Availability with Consistency for Reading

- Read Consistency: Can read from any replica, since data in each replica for an extent is bit-wise identical.

- Read Availability: Send out parallel read requests if first read is taking higher than 95% latency.
Dynamic Load Balancing – Partition Layer

- Spreads index/transaction processing across partition servers
  - Master monitors traffic load/resource utilization on partition servers
  - Dynamically load balance partitions across servers to achieve better performance/availability
- Does not move data around, only reassigns what part of the index a partition server is responsible for
Dynamic Load Balancing – DFS Layer

- DFS Read load balancing across replicas
  - Monitor latency/load on each node/replica; dynamically select what replica to read from and start additional reads in parallel based on 95% latency

- DFS write load balancing
  - Monitor latency/load on each node; seal the replica set with an overloaded node, and switch to a new extent on another set of nodes to append to

- DFS capacity load balancing
  - Lazily move replicas around to ensure the disks and nodes have equal amount of data on them
  - Important for avoiding hot nodes/disks
Append Only System

- **Benefits**
  - Simplifies the replication protocol
  - Easier diagnosis and repair of software bugs
  - Erasure coding (only recode on GC)
  - Keep snapshots at no extra cost
  - Works well with future drive technologies and sizes

- **Tradeoff**
  - GC overhead
Our Approach to the CAP Theorem

- Layering and co-design provides extra flexibility to achieve “C” and “A” at same time while being partition/failure tolerant for our fault model
  - **Stream Layer**
    - For Availability, if write failure create new extent on the available nodes and keep on appending to it
    - For Consistency, replicas are bit-wise identical up to the commit length so can read from any replica
  - **Partition Layer**
    - For Consistency, uses the streams such that it always has a consistent view of the data (see SOSP paper)
    - For Availability, RangePartitions can be served by any partition server and are moved to available servers if a partition server fails
- Designed for specific classes of partitioning/failures seen in practice
  - Process to Disk to Node to Rack failures/unresponsiveness
  - Node to Rack level network partitioning
Lessons Learned

- Automatic load balancing
  - Quickly adapt to various traffic conditions
    - Need to handle every type of workload thrown at the system
  - Built an easily tunable and extensible language to dynamically tune the load balancing rules
  - Need to tune based on many dimensions
    - CPU, Network, Memory, tps, GC load, Geo-Rep load, Size of partitions, etc
- Achieving consistently low append latencies
  - Ended up using SSD journaling
- Efficient upgrade support
- Pressure point testing
Erasure Coding

- Implemented at the DFS layer
  - All data is committed to 3 copies for a fast ack back to clients
  - Lazily the data is erasure coded, and the 3 copies are deleted
- Papers/Presentations:
  - [https://www.usenix.org/conference/usenixfederatedconferencesweek/erasure-coding-windows-azure-storage](https://www.usenix.org/conference/usenixfederatedconferencesweek/erasure-coding-windows-azure-storage)
Azure VM’s Persistent Disks

- VHDs for persistent disks are directly stored in Windows Azure Storage as blobs
- Can access your VHDs via REST
  - Easy to upload/download your own VHDs and mount them
  - REST writes are blocked to VHD blob while mounted to a VM
- Disks Snapshot are supported
- Geo replication is supported
 Separating Compute From Storage

- Allows compute and storage to be scaled separately
  - Lowers costs because we can balance ratio of compute and storage based on demand
- Provide Flat Network Storage
  - We use a Quantum 10 network architecture
  - Fully non-blocking 10Gbps based fully meshed network, providing an aggregate backplane in excess of 50Tbps of bandwidth for each data center
Architecture Summary

- **Durability**: All data stored with at least 3 replicas
- **Consistency**: All committed data across all 3 replicas are identical
- **Availability**: Can read from any 3 replicas; If any issues writing seal extent and continue appending to new extent
- **Performance/Scale**: Retry based on 95% latencies; Auto scale out and load balance based on load/capacity

Additional details can be found in the SOSP paper:
Resources

- Joe’s Talk (1:00pm today)
  - How to use Windows Azure Storage
  - Best Practices

- Windows Azure Developer Website

- Windows Azure Storage Blog

- SOSP Paper/Talk