SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

Azure File Service: Expectations vs. Reality on the Public Cloud

David Goebel
Microsoft

Azure File Service

Talk Topics:

0. Survey of SMB landscape @MS
1. The features, API surfaces and scenarios enabled
2. The design of an SMB server not backed by a conventional file system
3. What we've learned in the past two years

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

SMB Protocol History

- <= Srv03 was V1.3 and carried **lots** of baggage.
- Vista: SMB 2.02 was a vast simplification to reduce chattiness and map SMB commands to NT Irps and compound commands. Durable handles allowed handles to be reconnect after a network hiccup.
- Win7: SMB 2.1 introduced resilient handles and "leases" which supersede the older oplock scheme taken from 1.3
- Win8: SMB 3.0 added encryption, leases on directory handles, multichannel & RDMA (SMB Direct), and persistent handles to support CA (Continuously Available) shares hosted on SODA clusters.
- Win8.1: SMB 3.0.2 added cluster enhancements.
- Win10: SMB 3.1.1 adds negotiated encryption algorithm, secure negotiate and other security features.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

[MS-SMB2]

- Documents the protocol.
- Very useful, but has its limits.
- Anticipates, though doesn't require, a traditional file system on the other side.
- Azure File Service uses the underlying Azure Tables infrastructure (for metadata) & Page Blobs for file data instead.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

AFS[1] Fundamental Concepts

- AFS is not the Windows SMB server (srv2.sys) running on Azure nodes.
- AFS is a completely new SMB server implementation which uses Azure Tables and Blobs as the backing store.
- AFS leverages the highly available and distributed architecture of Tables and Blobs to imbue those same qualities to the file share.

[1] Azure File Service not CMU's Andrew File System. I wasn't on the naming committee.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Current AFS Status/Limits

- SMB 3.0 with encryption & persistent handles.
- 5TB per share and 1TB per file.
- 1000 IOPS limit per share. ~60 MB/sec typical.
- Some NTFS features not supported – yet (see link to list on the Resources slide).
- Shared namespace with REST imposes some limitation on characters and path lengths due to HTTP restrictions.

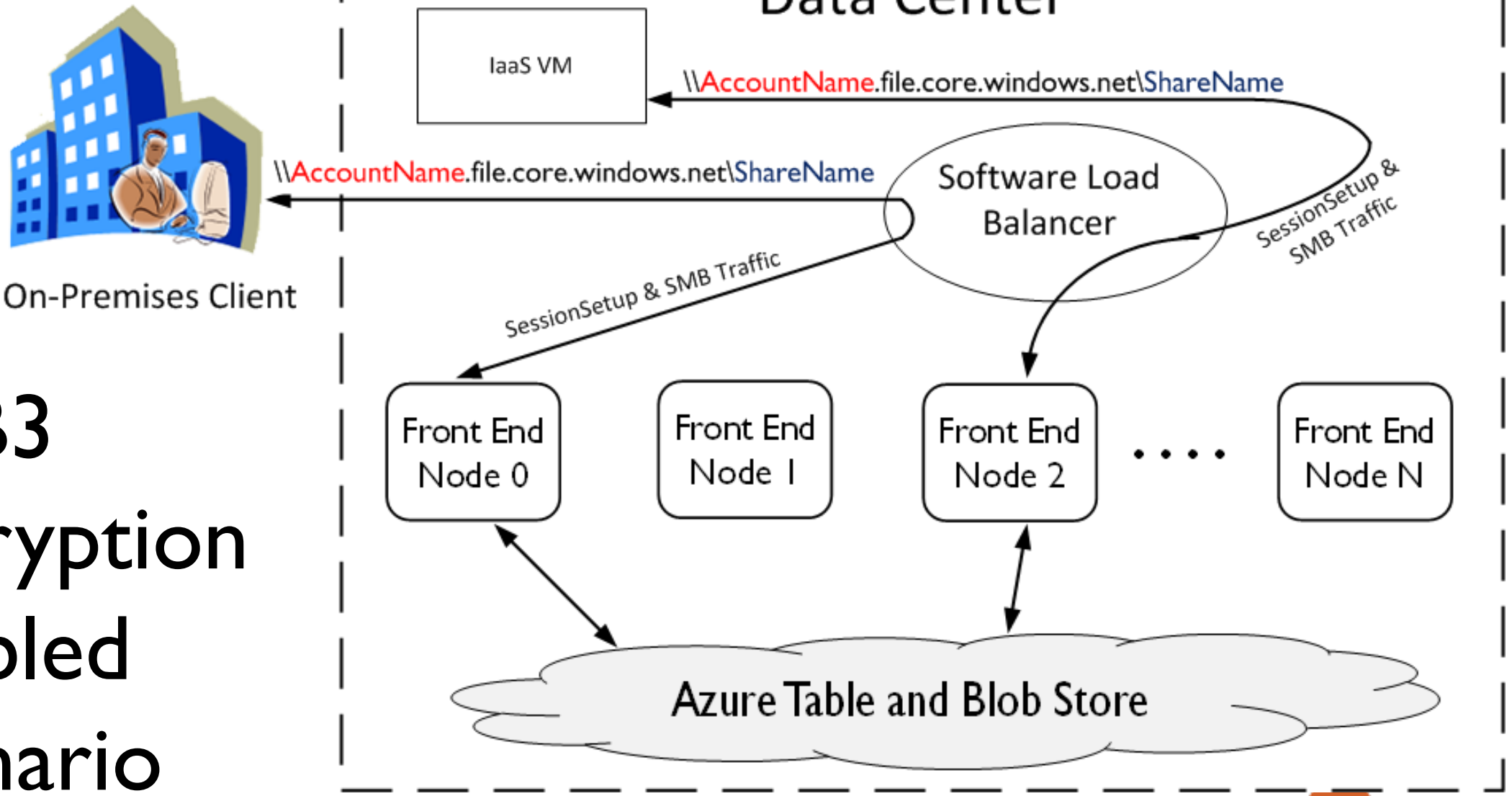
2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Current Linux Support

Linux Distribution	Publisher	Kernel Version	CIFS Version	SMB3 Persistent Handles	SMB3 Encryption
Ubuntu Server 16.04 LTS	Canonical	4.4.0-78-generic	2.08	Yes	No
Ubuntu Server 17.04	Canonical	4.10.0-24-generic	2.09	Yes	Yes
CentOS 7.3	Rogue Wave Software	3.10.0-514.10.2.el7	2.03	No	No
Debian 8	Credativ	3.16.0-4-generic	2.03	Yes	No
Open SUSE Leap 42.2	SUSE	4.4.70-18.9.1	2.08	Yes	No
SUSE Linux Enterprise Server 12 SP2	SUSE	4.4.59-92.20.2	2.08	Yes	No

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

SMB3 Encryption Enabled Scenario



Note: Port 445 outbound must be unblocked.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

DEMO

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Scenarios Enabled By AFS

- Existing file I/O API (Win32, CRT, etc.) based applications, i.e. most business applications written over the last 30 years, should "just work"®. More on this in "Lessons Learned" later.
- A business can stage existing workloads seamlessly into the cloud without modification to mission critical applications.
- Some minor caveats that will become more minor over time.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

What about REST?

If you're a true believer in the benefits of statelessness, SMB and REST access the same data in the same namespace so a gradual application transition without disruption is possible.

- Container operations:
  Create, List, Delete, Get properties, Get/Set metadata
- Directory Operations:
  Create, Delete, Get Properties, List (Same as ListFiles)
- File operations:
  Create, List, Delete, Get/Set properties, Get/Set metadata, Get Contents, Put Ranges, List Ranges

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Leveraging Azure Table Infrastructure

- An Azure Table is a simple NoSQL collection of rows with a common column schema and sorted / searchable by a subset of ordered 'key' columns.
- Internally Azure provides transaction APIs so that rows can be operated on with ACID semantics.
- Tables can be specially associated, extending these semantics to span multiple tables

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

AFS Tables

- AFS shares are 1-1 with Azure containers.
- An AFS share's metadata is stored as a collection of tables, the most notable of which are:

File	A table of all files and directories. It is a hybrid type, keyed by either FileName or FileId (64bit like NTFS).
Page	The allocated file ranges and their backing page blobs.
Handle	All open handles to files and directories.
Lease	All currently active SMB leases.
Change Notify	Registered change notifies.
Byte Range Locks	All currently active byte range locks

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

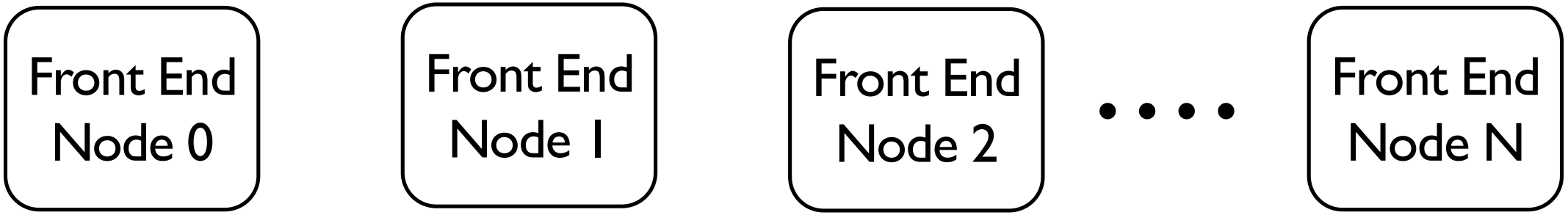
Typical namespace requests

- Namespace oriented requests make the heaviest use of transactions across multiple tables.
- Open/Create/Close will make modifications to at least two tables. Close can be particularly involved.
- Even reads/writes have to look at byte range locks and potentially break leases.
- The built-in transaction support makes this relatively painless.

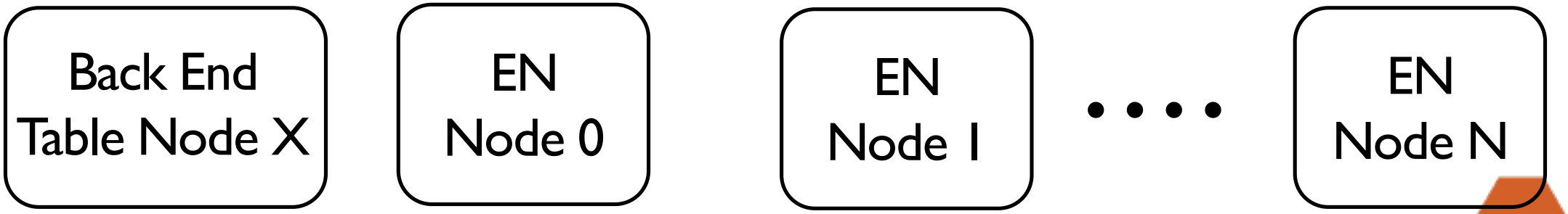
2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Mapping AFS to hardware

FrontEnd nodes receive connections from clients. Any FE node can service any share.



Currently a single share/container is within a single partition which is at any time "owned" by a single BE Table Node. A TableMaster service manages moving partition ownership in the case of BE node failure or for load balancing. Page blobs are managed by EN nodes.



2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Achieving active/active high availability

- For a given share, state is tiered between FE nodes and the BE node depending on its durability requirements. More on this later.
- All state required to correctly handle requests from different clients for the same file is managed by the BE.
- This segregation of state together with table transaction support in Azure enable active/active support with relatively little pain.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Tiering State By Durability Requirement

- A conventional file server treats only actual file data and essential metadata (filesize, timestamps, etc) as needing to be durably committed before an operation is acknowledged to the client (and even then only if opened WriteThrough).
- For true active/active high availability and coherency between FrontEnd nodes, modified state that normally exists only in server memory must be durably committed.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

SMB is a stateful protocol,

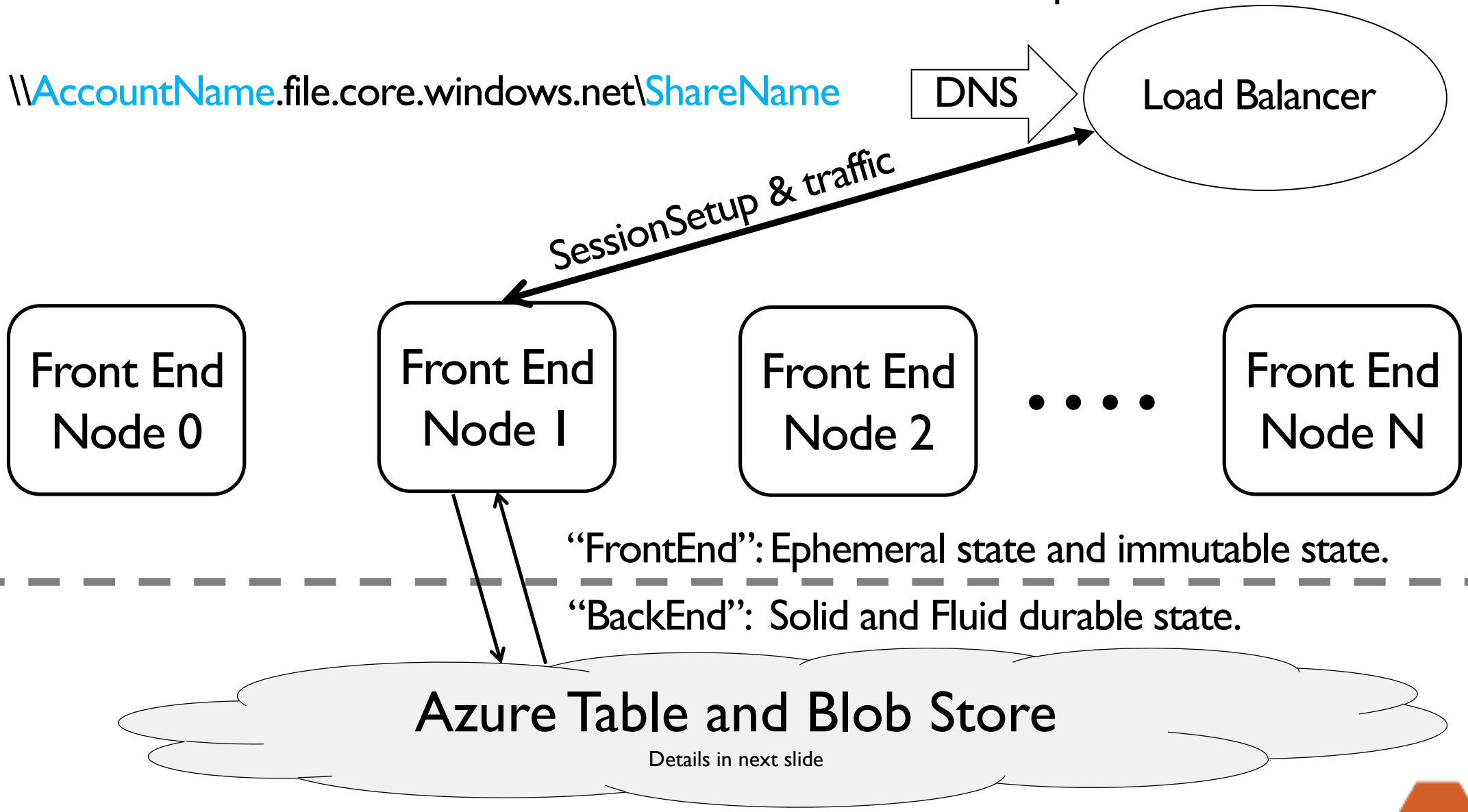
but not all states require expensive distributed transactional semantics

- Some aspects of a file's state are immutable, such as FileId and whether it's a file or a directory.
- Some state is transient, such as open counts, and can be optimized if loss of this state is acceptable in a disaster.
- Some state is also maintained by the client, like CreateGuid, drastically reducing the cost of tracking clients.
- State associated with connection mechanics is ephemeral.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

for example 157.56.217.32:445

\\AccountName.file.core.windows.net\ShareName

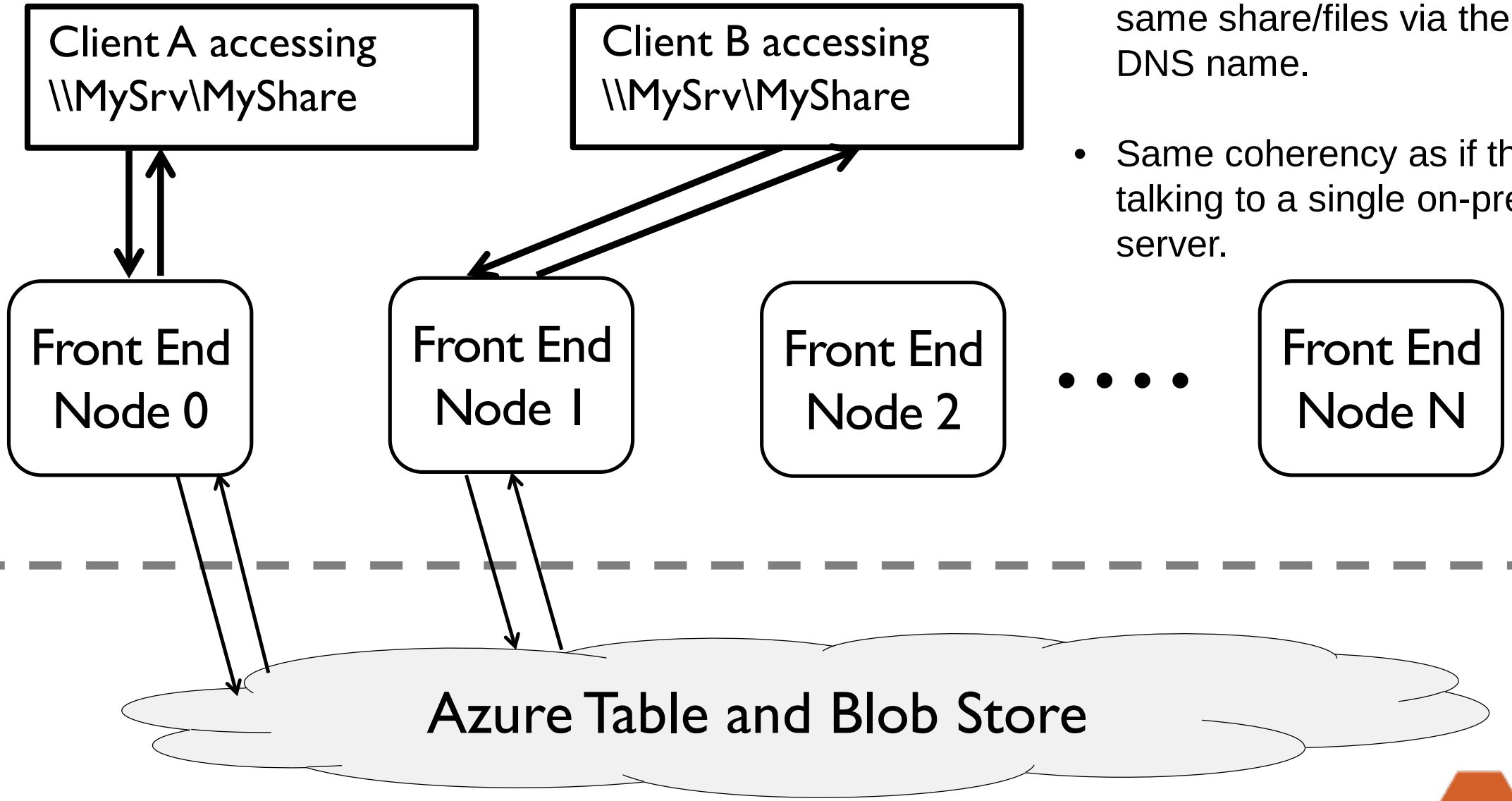


2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Azure Table and Blob Store

- AFS uses the underlying Azure Tables infrastructure to store metadata associated with files/dirs, open handles to them and other state.
- Table operations are transactional within a partition.
- Currently we have a 1-1 relation between share and partition.
- Our tables for the various metadata are specially associated, allowing transactions across them.

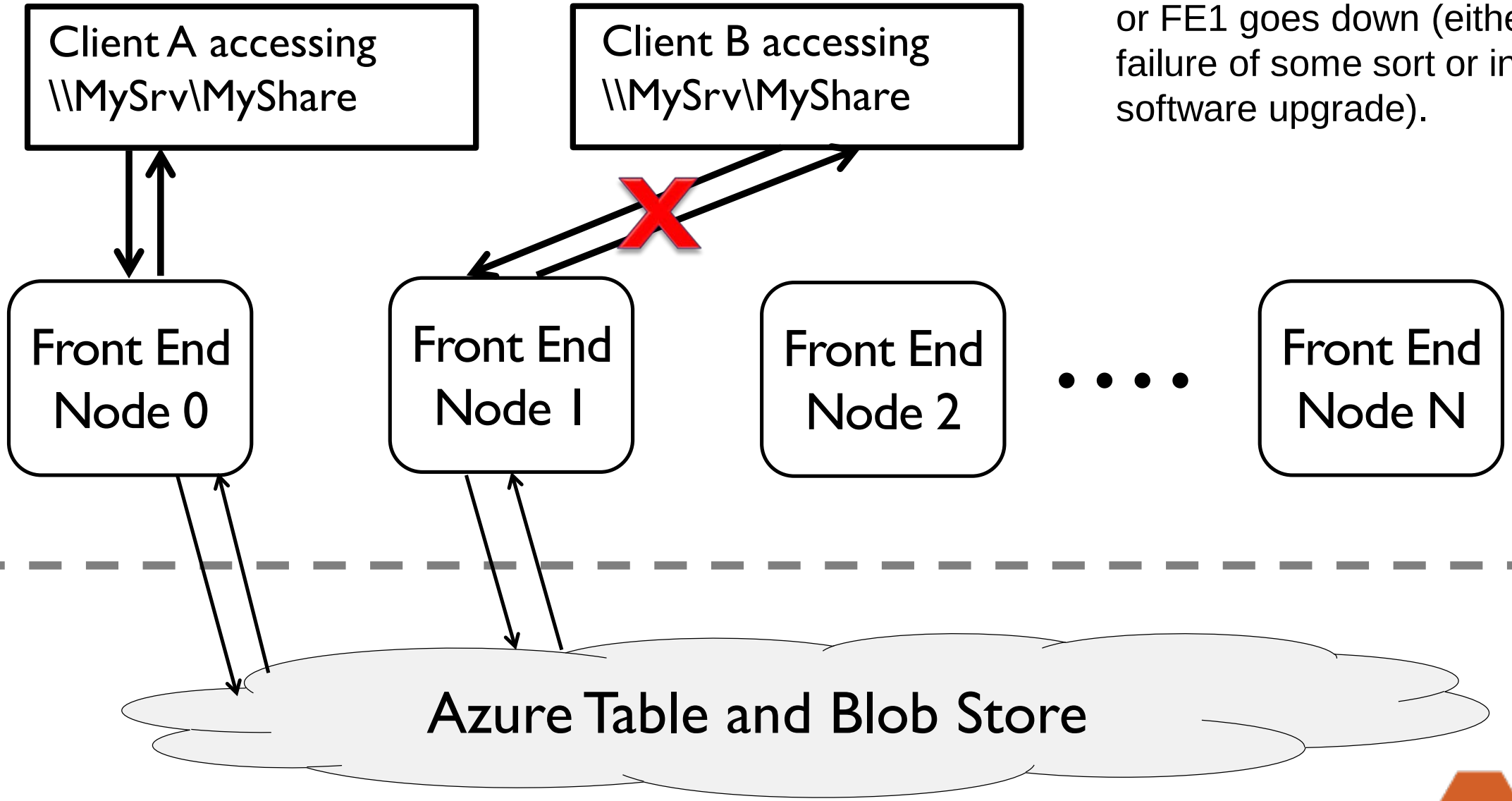
2017 Storage Developer Conference. © Microsoft. All Rights Reserved.



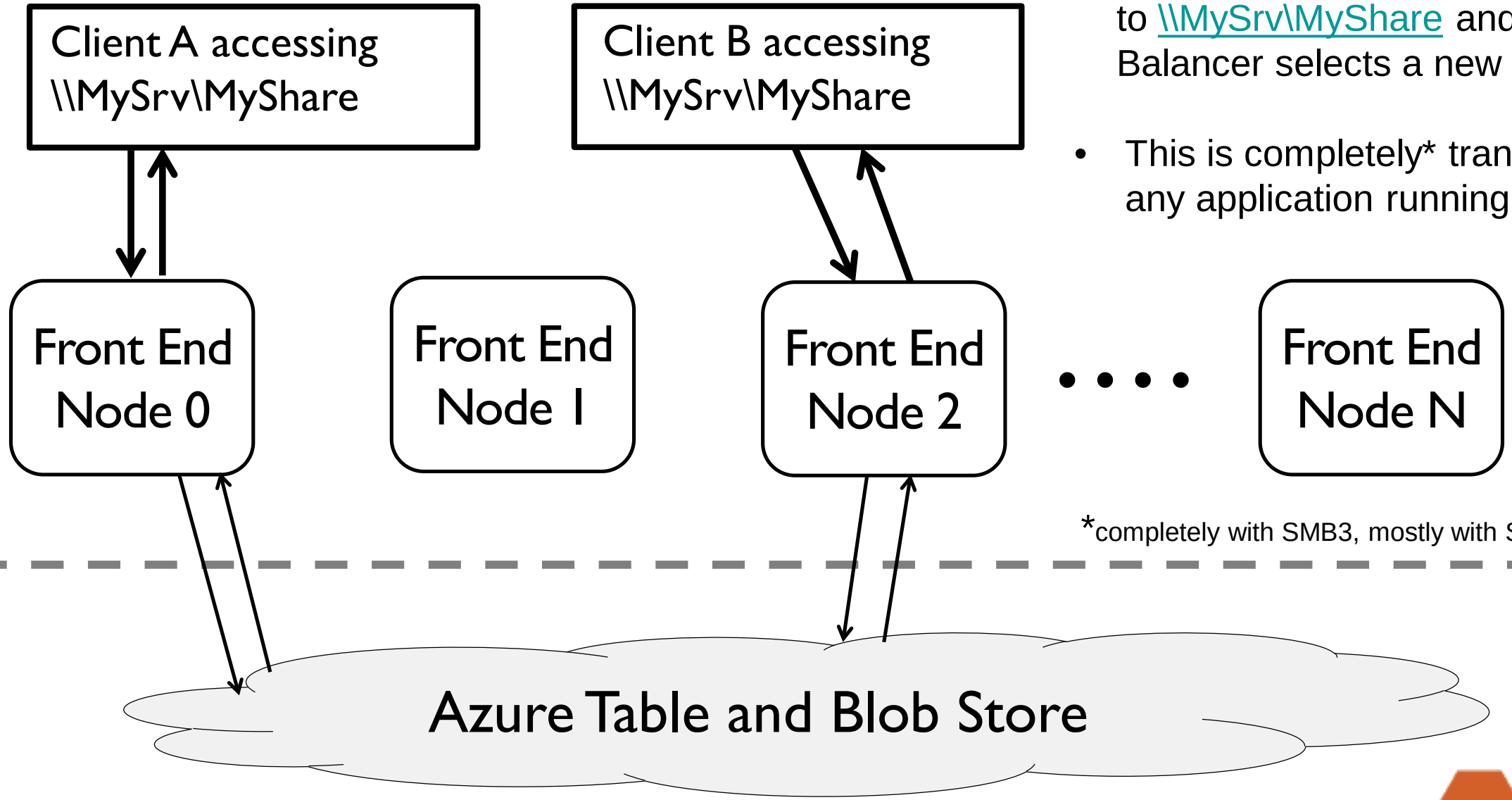
- Clients A & B both accessing the same share/files via the same DNS name.
- Same coherency as if they were talking to a single on-premises server.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

- Clients B loses connection to FE1 or FE1 goes down (either due to a failure of some sort or intentional software upgrade).



2017 Storage Developer Conference. © Microsoft. All Rights Reserved.



- Client B automatically reconnects to \\MySrv\MyShare and the Load Balancer selects a new FE.
- This is completely* transparent to any application running on ClientB.

*completely with SMB3, mostly with SMB2.1

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Examples of state tiering

- Ephemeral state: SMB2_FILEID.Volatile, credits, tcp socket details.
- Immutable state: 64bit actual FileId, IsDirectory
- Solid durable state: SMB2_FILEID.Persistent, SessionId
- Fluid durable state: Open counts, file names, file size, lease levels and many more. This is the largest group of states.

"Solid" here meaning the state is generated by AFS and not generally changeable by normal actions of the client/application while "Fluid" is fully changeable by File APIs.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Example: Durable Handle Reconnect

- Intended for network hiccups as it assumes all state is still valid on the server.
- On AFS this state is durably persisted on our BackEnd so we're able to 'stretch' durable handles to recover from FrontEnd AFS failures (planned or otherwise) since it's transparent to the client.
- This is important as we're continually updating AFS code requiring AFS service restarts.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Example: Persistent Handles

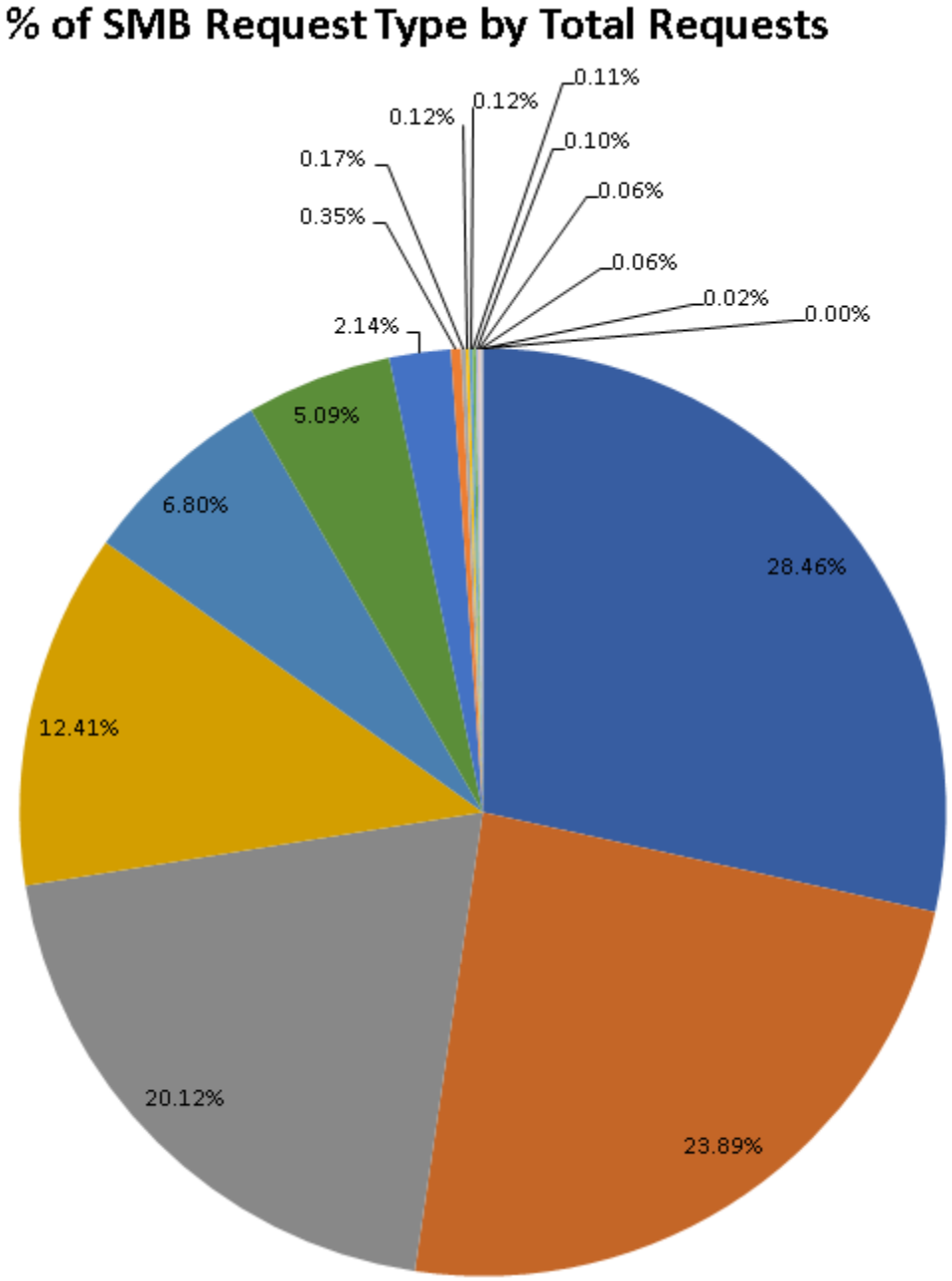
- Unlike Durable Handles, Persistent Handles are actually intended to support Transparent Failover when the server dies.
- Leverages state on the client for replay detection so that 'once only' operations are only executed once.
- More details about create requests durably committed.
- With Durable Handles SMB 2.1 protocol compliance required us to artificially limit our capability. With Persistent Handles we have seamless Transparent Failover.

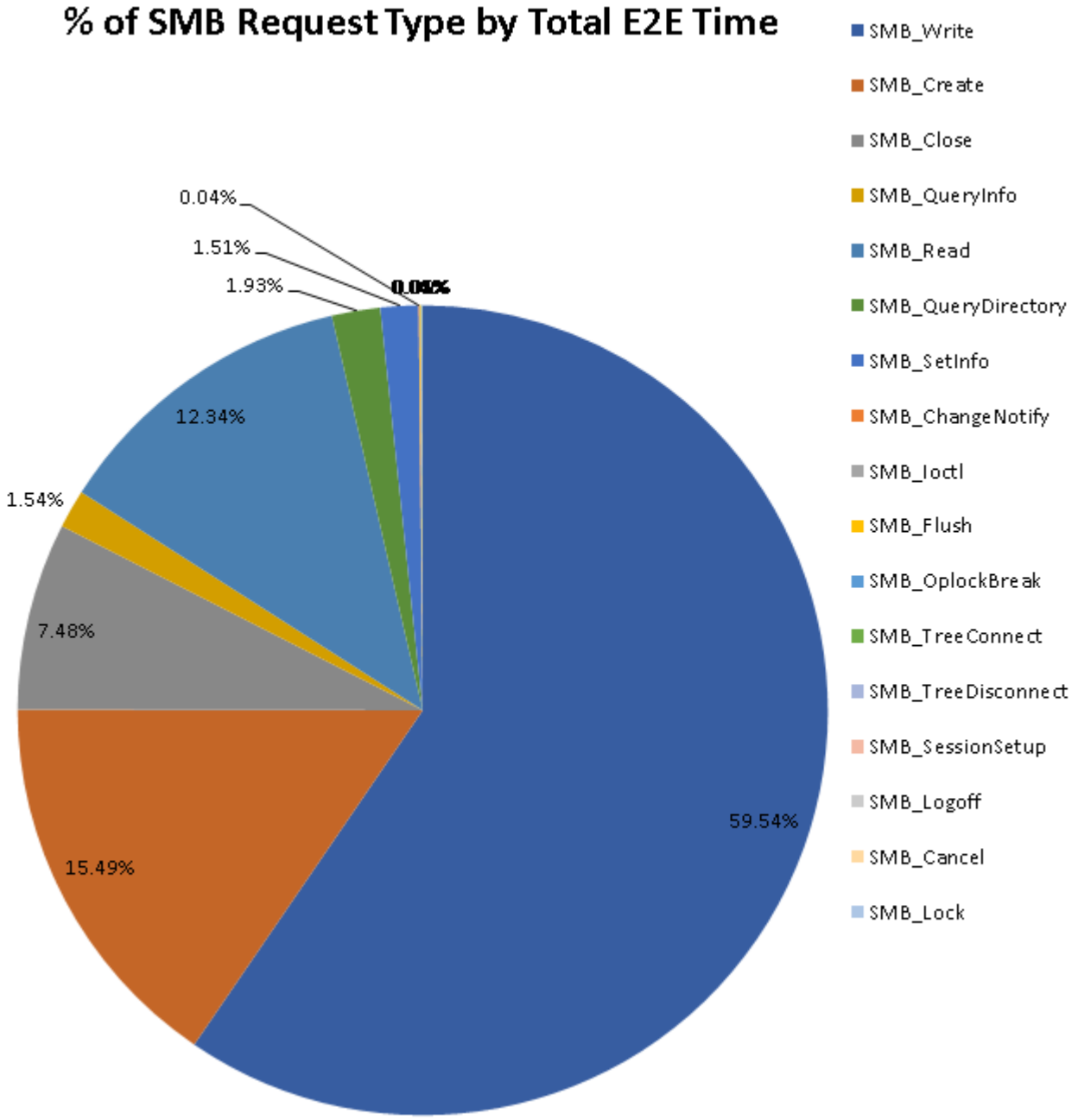
2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Observations and Lessons Learned

- We now have some experience running the world's largest SMB server.
- Metadata operations are unfortunately common and expensive for us.
- Even compared to srv2.sys on-prem, AFS pays a high price for its durability. Open/Close and Write-Only handles are particularly bad.
- Some applications may not be suitable for "lift and shift", especially if they have never even been run against an on-prem file server.
- In terms of total aggregate End-to-End request time, all that matters are Create, Close, Read and Write.

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.





2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Resources:

- Getting started blog with many useful links: http://blogs.msdn.com/b/windowsazurestorage/archive/2014/05/12/introducing-microsoft-azure-file-service.aspx
- Generally Availability announcement: https://azure.microsoft.com/en-us/blog/azure-file-storage-now-generally-available
- NTFS features currently not supported: https://msdn.microsoft.com/en-us/library/azure/dn744326.aspx
- Naming restrictions for REST compatibility: https://msdn.microsoft.com/library/azure/dn167011.aspx

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.

Thank You!

Questions?

2017 Storage Developer Conference. © Microsoft. All Rights Reserved.