SMB3 Update

David Kruse
Microsoft

Agenda

- Why 3.02?
- SMB 3.02 Changes
  - Read/Write Flags
  - Asymmetric Shares
- Relevant “Windows 2012 R2” Features
  - Diagnosibility Improvements
- SMB for IPC?
- Q/A

Why 3.02?

- Capabilities are useful in permitting a server to offer a subset of dialect functionality
- Post-introduction of new capabilities risks exposing servers (or clients) to unexpected new wire behavior
- Capability bits are of limited quantity

Dialects vs. Capabilities

- How should a client or server behave if it receives a capability/flag that it does not understand?
  - Asymmetric Shares
  - FILE_ATTRIBUTE_INTEGRITY_STREAM
- How does a server behave when a client attempts to use capabilities that were not available in their negotiated dialect?
  - SMB 2.0 client attempting MC? Or CA?
  - This is a consistency/doc question more than a use case

Dialect Handling

From this point on:

1. A server that receives a request from a client with a flag/option/capability that is not valid for the dialect selected SHOULD ignore (AND off) the flag/option/capability
2. A client that receives a response from a server with a flag/option/capability that is not defined for the dialect selection SHOULD ignore the flag/option/capability.

```
// Indicates that the write request is to be issued write-through even
// if the file was not opened for write-through IO. This flag may only
// be used for unbuffered writes (the file was opened unbuffered or the
// request has the SMB2_WRITEFLAG_UNBUFFERED flag set).
//
#define SMB2_WRITEFLAG_WRITE_THROUGH          (0x00000001)

//
// Indicates that the write request is to be issued unbuffered, regardless
// of how the file was opened.
//
#define SMB2_WRITEFLAG_UNBUFFERED             (0x00000002)

//
// Mask of write request flags that are supported for dialects < 3.0.0
//
#define SMB2_02XX_WRITEFLAGS_MASK (SMB2_WRITEFLAG_WRITE_THROUGH)

//
// Mask of write request flags that are supported for dialect 3.0.0
//
#define SMB2_0300_WRITEFLAGS_MASK (SMB2_02XX_WRITEFLAGS_MASK)

//
// Mask of write request flags that are supported for dialect 3.0.2
//
#define SMB2_0302_WRITEFLAGS_MASK (SMB2_0300_WRITEFLAGS_MASK |
SMB2_WRITEFLAG_UNBUFFERED)
```

Read/Write Changes

- Unbuffered IO flags for better enterprise application support
- Continued improvements of RDMA performance via “remote invalidation”

Unbuffered Read/Write

- SMB 3.0 permitted annotating individual write requests with WRITE_THROUGH
- Various storage components (and the NT IO model) permit marking individual IO's as unbuffered (not cached by system or controller)
- On receipt, server passes this flag on to storage stack
- Adds alignment requirements to IO size and offset (same as issuing IO on an unbuffered handle)

Unbuffered Read/Write


```
//
// Indicates that the read request is to be issued unbuffered, regardless
// of how the file was opened.
//
#define SMB2_READFLAG_UNBUFFERED (0x01)

//
// Indicates that the write request is to be issued unbuffered, regardless
// of how the file was opened.
//
#define SMB2_WRITEFLAG_UNBUFFERED (0x00000002)
```

When set, server MUST:

- Validate offset and length are sector aligned (or fail!)
  - Identical to requirements on file handles opened with NO_BUFFERING, but now enforceable per-IO
- Request the underlying storage system not buffer the data

RDMA Changes

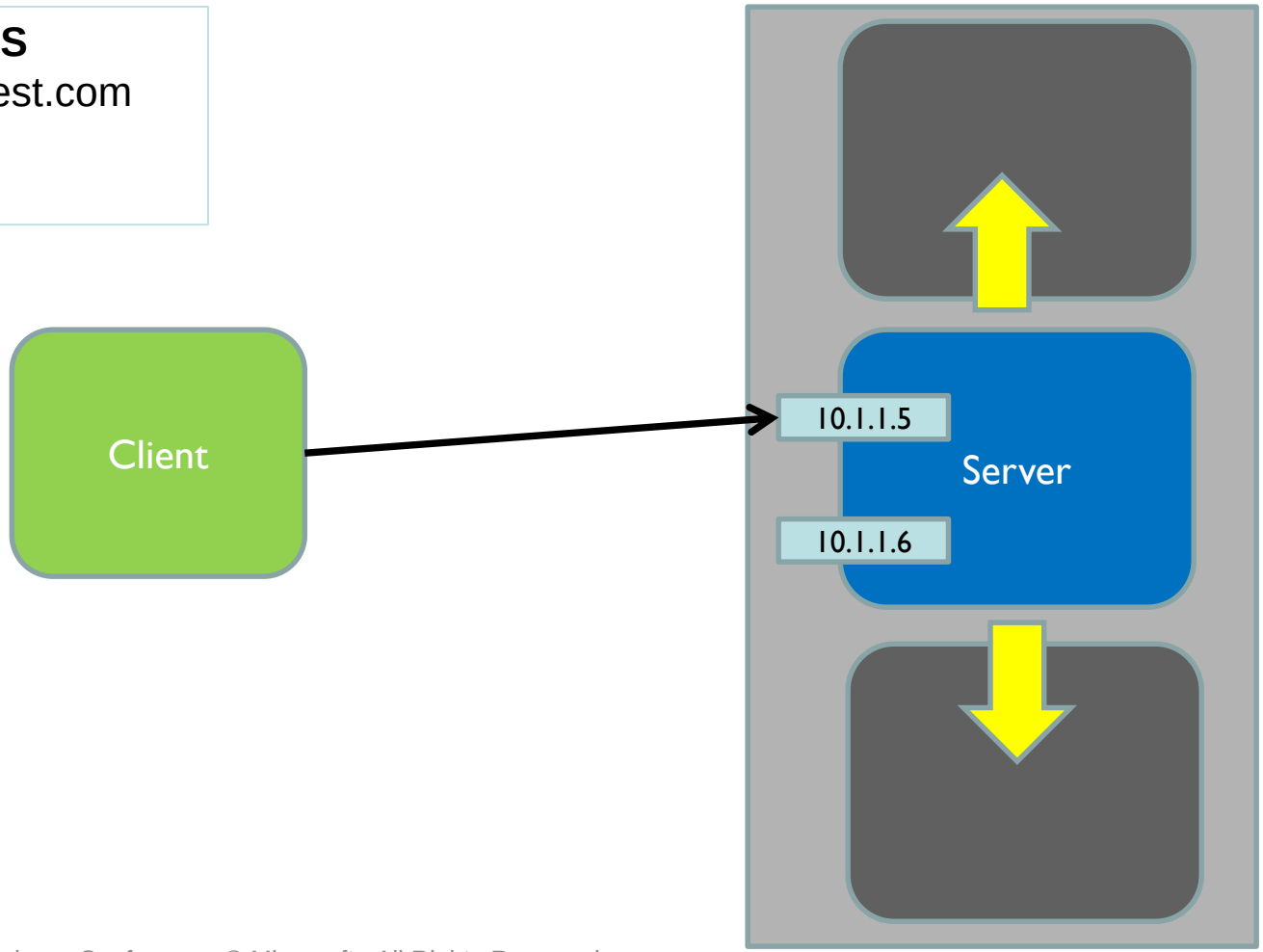
- Come hear Greg Kramer and Tom Talpey tomorrow at 1:00!

"Traditional" Clustered File Server

DNS

smbsrv.test.com

- 10.1.1.5
- 10.1.1.6

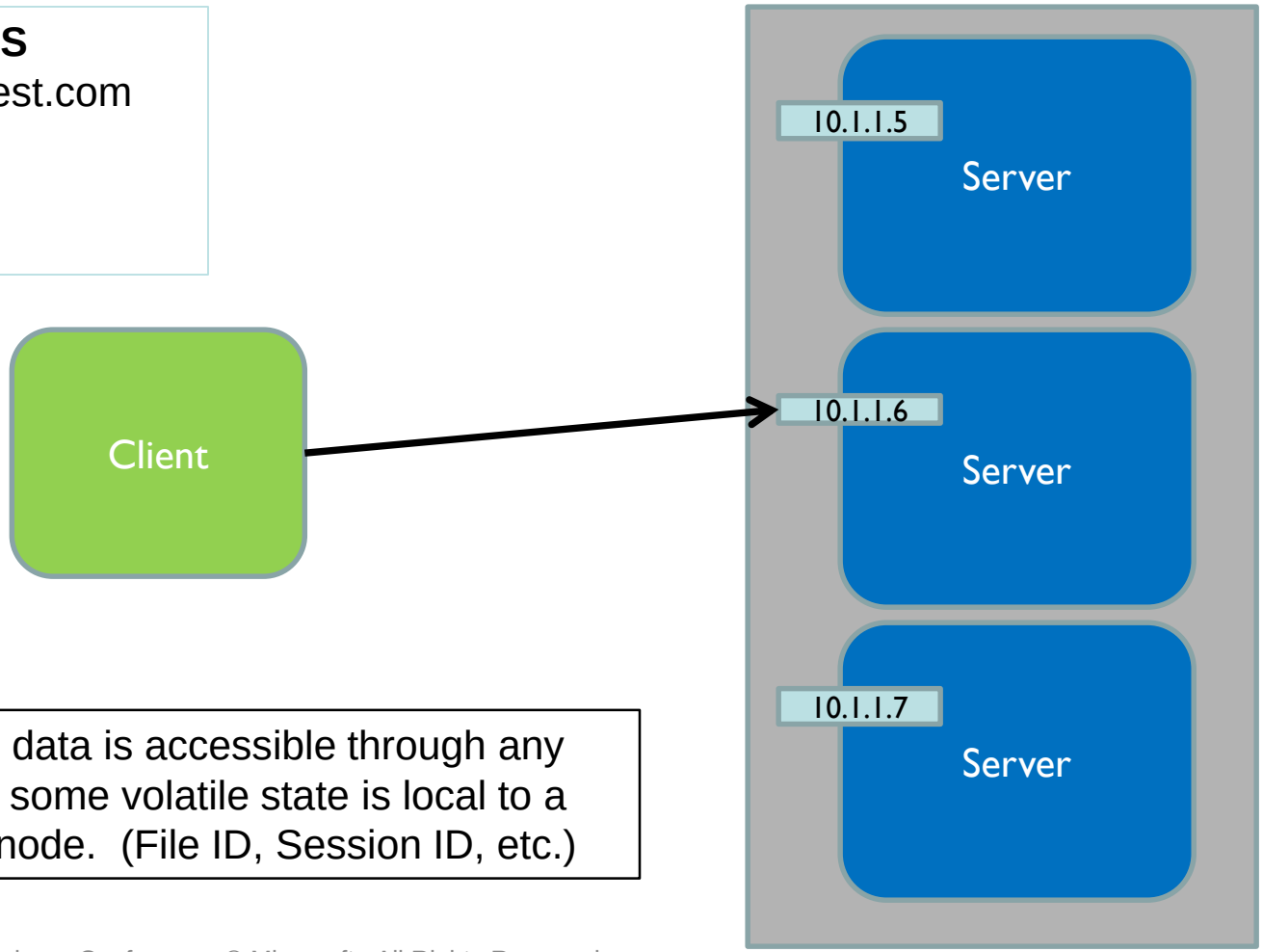


"Scale-Out" Clustered File Server

DNS

smbsrv.test.com

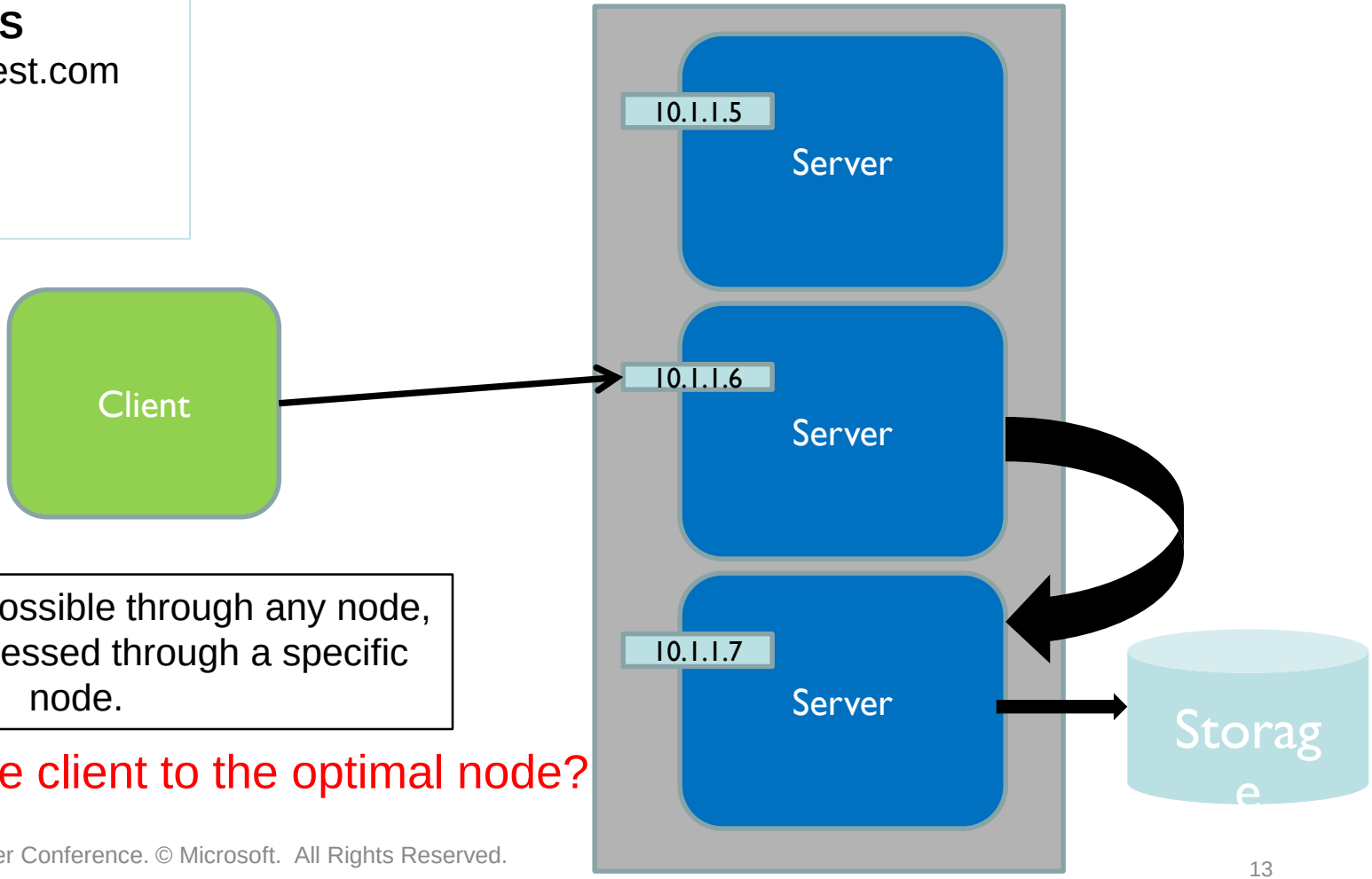
- 10.1.1.5
- 10.1.1.6
- 10.1.1.7



While data is accessible through any node, some volatile state is local to a given node. (File ID, Session ID, etc.)

"Asymmetric Scale-Out" Clustered File Server

DNS
smbsrv.test.com
- 10.1.1.5
- 10.1.1.6
- 10.1.1.7



Data access is possible through any node, but faster if accessed through a specific node.

Can we get the client to the optimal node?

Asymmetric Shares

1. Client receives TREE_CONNECT response that includes SMB2_SHARE_CAP_ASYMMETRIC
2. Client establishes new connection pool for this share. (Connection pool is main authentication + associated multichannel channels)
3. Client registers with witness, receives notification of disk location
4. Client processes share-level “move” in the same fashion as a server-level “move”. (i.e. disconnects, reconnects to IP, rebinds handles)

Asymmetric Shares

- Client will now talk to multiple servers in a scale-out file server for different shares.
- In case of failures, client may reconnect to any available node
- Move-SmbWitnessClient powershell cmdlet will not work with asymmetric connections, as they are already moving to a targeted node

Asymmetric Shares

<Windows Implementation>

By default, Windows Server 2012 R2 only marks shares as Asymmetric if they use Spaces in mirrored configurations. For SAN or iSCSI deployments (where multiple nodes can access the data directly), we do not treat them as Asymmetric*

</Windows Implementation>

* You can override this by settings HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters\(REG_DWORD) AsymmetryMode = 2, in which case all scale out shares will be treated asymmetric

Asymmetric Shares

- Corresponding WitnessRegisterEx call permits client to specify share name in registration.
- Witness protocol also defines “Share Move” request
- Get-SmbWitnessClient PS cmdlet shows share registration. An empty share name implies a server-level registration

Witness Changes

```
DWORD
   WitnessrRegisterEx(
      [in]              handle_t          Handle,
      [out]              PPCONTEXT_HANDLE    ppContext,
      [in]              ULONG             Version,
      [in,string,unique]  LPTSTR              NetName,
      [in,string,unique]  LPTSTR              ShareName,
      [in,string,unique]  LPTSTR              IpAddress,
      [in,string,unique]  LPTSTR              ClientComputerName,
      [in]              ULONG             Flags,
      [in]              ULONG             KeepAliveTimeoutInSeconds
      );
}

#define REGISTER_NETNAME_NOTIFICATION ( 0x1 )
#define REGISTER_SHARE_NOTIFICATION ( 0x2 )
#define REGISTER_MULTICHANNEL_NOTIFICATION ( 0x4 )
```

Witness Changes

- Multichannel notifications offer client insight into arrival/loss of network interfaces.
- Windows clients currently use it for aborting operations for recovery
- In the future, could use arrival to light up new interfaces faster (instead of 10 minute probe)

Witness Keepalive

- RPC over TCP timeouts for async IO are very coarse (sometimes multiple hours to detect lost peer)
- Client optionally provides KeepAlive
- Server will complete pending Notification after KeepAlive with ERROR_TIMEOUT
- Client can assign RPC_C_CALL_TIMEOUT of KeepAlive + 60 seconds
- Guarantes Client and Server discover lost peer in minutes instead of hours

SMB 3.0 Diagnosibility Events
Windows Server 2012 R2

- For SMB3 Server Implementers testing with Server 2012 R2, check out the Microsoft-Windows-SMBClient eventlogs for new always-on events including:
  - Persistent/Resilient handle failures (with reasons)
  - Connectivity Events
  - Authorization errors
  - Transport arrival/removal
  - Multichannel and RDMA related events
  - Negotiate/Signing/Encryption failures
  - Witness Events
- Also available: Microsoft-Windows-SMBWitnessClient

SMB 3.0 Diagnosibility Events
Windows Server 2012 R2

Failed to reconnect a persistent handle.

Error: The transport connection is now disconnected.

FileId: 0x200000D0000029A:0x0

CreateGUID: {5d3a718b-c979-11e2-a085-001ec9fdd176}

Path: \434275K08-C1SOD\434275K08-C1Sh2\AMITKM2N1-5K08-C1H32\SharedDisk22.vhdx:SharedVirtualDisk

Reason: The connection object was suspended by the client

Previous reconnect error: The transport connection is now disconnected.

Previous reconnect reason: The connection object was suspended by the client

Guidance:

A persistent handle allows transparent failover on Windows File Server clusters. This event has many causes and does not always indicate an issue with SMB. Review online documentation for troubleshooting information.

SMB 3.0 Diagnosibility Events
Windows Server 2012 R2

For SMB3 Client Implementers testing with Server 2012 R2, check out the Microsoft-Windows-SMBServer eventlogs for new always-on events including:

- Signing/Encryption/Negotiate Validation errors
- Persistent/Resilient Handle Reconnect failures
- Authentication and Authorization failures
- Anonymous access related failures
- Slow file system operation warnings

Also check out Microsoft-Windows-SMBWitnessServer

What happened to Named Pipes

- Named pipe communication over SMB has been declining in popularity
  - Applications transition towards RPC and web service models
  - Named pipes have implicit performance concerns
    - Serialization of IO degrades high performance, prevents deep pipelines
    - *This is a property of **named pipes,** not of SMB as a transport for IPC*

SMB as IPC

- For an upper layer, SMB provides:
  - Capability discovery and negotiation
  - Authentication and authorization
  - Zero config multi-path discovery and bandwidth aggregation
  - RDMA for high bandwidth, low latency, low CPU
  - Implicit replay/recovery (optionally including cluster failover)
  - Zero config transport level encryption or integrity

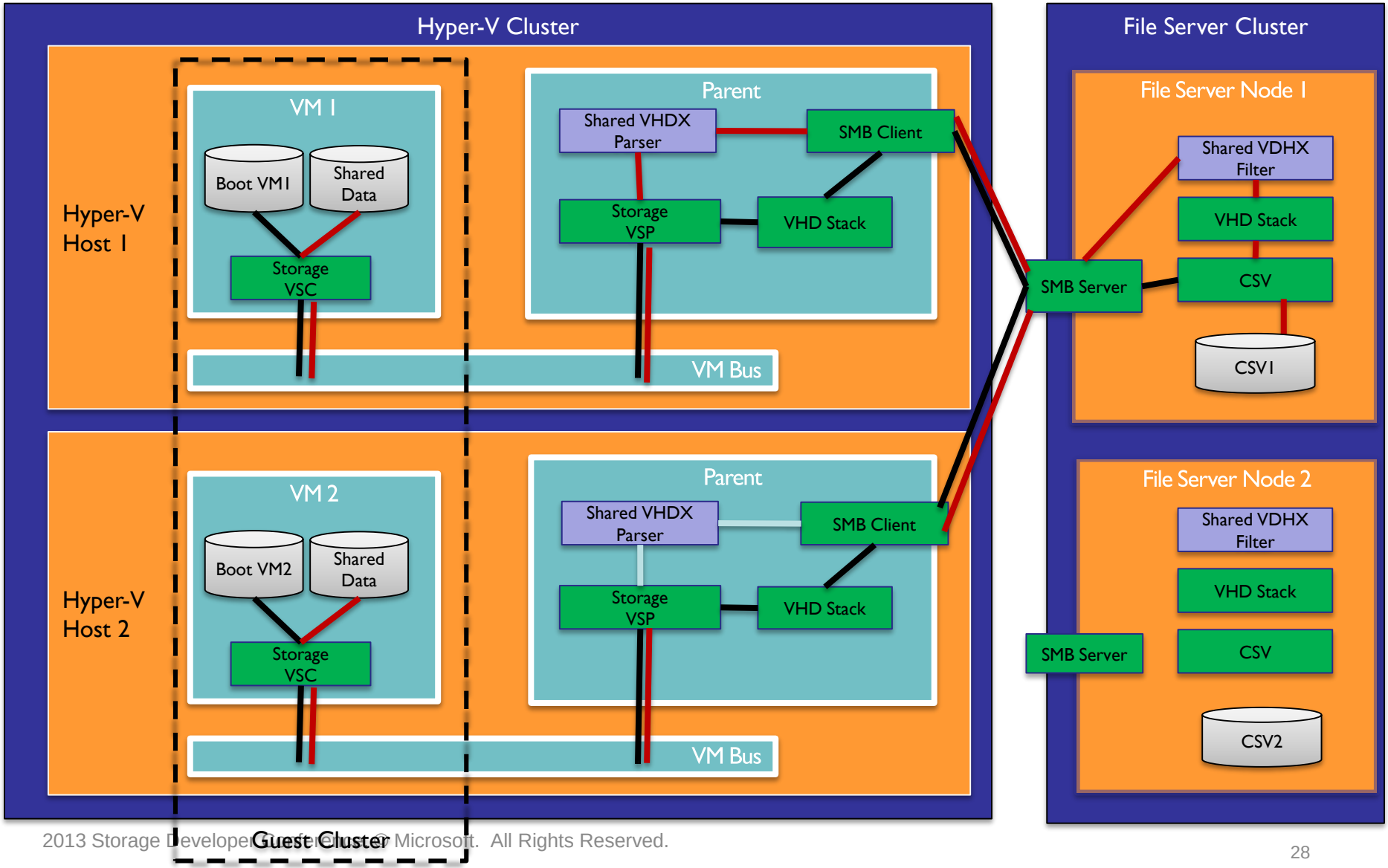
SMB as IPC

- To capitalize on all this, an application needs simply to represent their workload as Create/Read/Write/Close
- Pipe performance issues can be avoided by utilizing offset of read/write into usage
- Other instance specific requirements can be communicated via FSCTL calls

SMB as IPC

- Simplified file-system like interface permits developer to test locally, and then extend to remote (via SMB)
- Transparently utilizes all the current (and future) performance gains in SMB
  - Support for new RDMA or performance features
  - Improved client/server implementations

Remote Shared Virtual Disk Protocol



Remote Shared Virtual Disk Protocol

- Come to Jose's talk later this week to learn more!

Hyper-V Live Migration over SMB

- Hyper-V on destination creates share with virtual file representing VM
- Hyper-V on source opens handle to destination
- VID issues writes (over SMB) to target for memory pages
- With MC, bandwidth aggregation and recovery supported
- With SMBDirect, transfer is true zero copy

Future Thoughts on SMB as IPC

Q&A

Thanks for listening!