ORACLE

ZFS STORAGE APPLIANCE

An Oracle Technical White Paper

August 2014

SMB 2.0 Update for the Oracle ZFS Storage Appliance

ORACLE

Table of Contents

Introduction

The Oracle ZFS Storage Appliance combines advanced hardware and software architecture for a multiprotocol storage subsystem that enables users to simultaneously run a variety of application workloads and offer advanced data services. First-class performance characteristics are illustrated by the results of the industry standard benchmarks like SPC-1, SPC-2 and SPECsfs.

The Oracle ZFS Storage Appliance product line is known as Unified Storage, as it provides storage services to both file and block protocols and networks. The focus of this paper is on the file protocol Server Message Block (SMB) and the functionality added to the SMB service provided by the Oracle ZFS Storage Appliance with the introduction of Oracle ZFS Storage Appliance Software version 2013.1.

SMB was originally designed by IBM in the 1980s to allow file sharing between workstations and was adopted by Microsoft. Microsoft made many modifications and merged SMB with LAN Manager which was being co-developed with 3COM and then continued to be developed as part of Windows for Workgroups c. 1992 and in subsequent Windows versions.

SMB clients and servers negotiate the features they can use in the negotiation phase at the beginning of a session so that each knows what to expect in the subsequent transactions. In this way, SMB2.0 servers still need to support SMB1.0.

SMB2.0 requires the use of Transmission Control Protocol (TCP) as the transport mechanism and drops support for NetBIOS.

This white paper identifies the differences between SMB1.0 and SMB2.0 as they pertain to the Oracle ZFS Storage Appliance.

SMB2.0 Supported Features

The following list describes the major features of SMB2.0 which are supported by the Oracle ZFS Storage Appliance:

- Reduced command set
- Compound requests
- Asynchronous operations
- Larger IO request sizes
- Durable handles
- Improved message signing
- Improved scalability

The following list describes the optional features defined under the SMB2.0 standard that are not supported on the Oracle ZFS Storage Appliance:

- Compound responses
- Windows symbolic links

Reduced Command Set

SMB1.0 defined over 100 commands – with a good number of duplicated commands differing slightly in semantics or syntax. For example, SMB2.0 has a single `WRITE` operation whereas SMB1.0 has at least 14 distinct `WRITE` variants.

SMB2.0 reduces this set from over 100 to just 19 with no loss of functionality.

The following list defines the command set supported by the Oracle ZFS Storage Appliance under SMB2.0:

```
SMB2_NEGOTIATE                  SMB2_SESSION_SETUP
SMB2_LOGOFF                     SMB2_TREE_CONNECT
SMB2_TREE_DISCONNECT            SMB2_CREATE
SMB2_CLOSE                      SMB2_READ
SMB2_WRITE                      SMB2_FLUSH
SMB2_QUERY_DIRECTORY            SMB2_QUERY_INFO
SMB2_SET_INFO                   SMB2_LOCK
SMB2_IOCTL                      SMB2_CHANGE_NOTIFY
SMB2_CANCEL                     SMB2_ECHO
SMB2_OPLOCK_BREAK
```

Compound Requests

The SMB2.0 commands just shown are primitive commands, only requiring a single action to take place. SMB1.0 commands are relatively complex in comparison.

As mentioned earlier, SMB1.0 has 14 distinct `WRITE` commands. While these operations essentially request the same action, there are additional subactions included – for example, `WRITE_AND_UNLOCK`.

SMB2.0 caters to multiple actions being sent as a single request to allow these complex SMB1.0 commands to be emulated but also to reduce the network traffic between server and client. Under SMB2.0, a compound command to emulate the SMB1.0 `WRITE_AND_UNLOCK` could be one containing an `SMB2_WRITE` and `SMB2_LOCK` command.

Compounding commands in this way allows a chain of unrelated commands to be executed asynchronously or related commands to be executed synchronously, depending on the context or requirement of the client – thus offering flexibility unavailable under SMB1.0.

Related compound requests (also known as chained requests) must be executed sequentially in the order they occur within the command from the client and can be used for command pipelining, where the output from one command is used as the input to another in the chain.

Unrelated compound requests may be executed in any order. Windows clients do not currently send unrelated compound requests under SMB2.0, but the functionality is required in any SMB2.0 server implementation.

Asynchronous Operations

Under SMB1.0, a command was issued by the client, which then waited for the server to send the response. Only then could the client process then proceed to the next command. This synchronous communication constrained the performance of the system as a whole.

Under SMB2.0, commands can be sent without requiring a response before sending the next command. Ultimately, all server responses have to be accounted for, but this allows for increased concurrency in programs and libraries that have been coded with this functionality in mind.

The following figure shows a stream of SMB1.0 and SMB2.0 commands. In this figure C*n* represents the client commands, A*n* represents the corresponding action carried out on the server and R*n* represents the result for the action which is sent back to the client for processing.

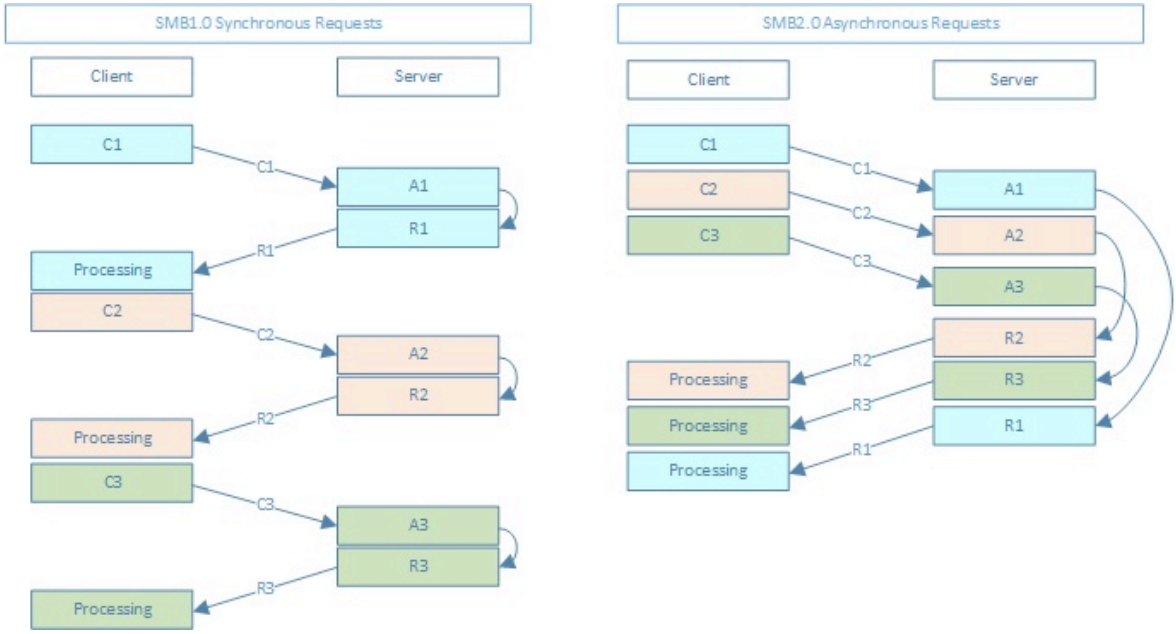

Figure 1. SMB1.0 synchronous compared to SMB2.0 asynchronous requests

Overlapping these commands can lead to an overall reduction in response times, improving client response overall. Figure 1 shows sequential request processing but there is no restriction in the order or parallelism in which these commands are processed. For example, A1 and A2 could be processed simultaneously and the results sent in a different order.

Larger IO Request Sizes

SMB1.0 uses 16-bit data addressing which limits block sizes on requests to 64 K, among other restrictions. SMB2.0 allows for 32-bit or 64-bit storage fields and 128-bit file handles, removing the previous constraints on block sizes and ultimately improving performance with large file transfers over fast networks.

These larger IO requests will improve response times in high-latency networks by reducing the number of commands that have to be sent and the number of responses that need to be received.

Durable Handles

Durable handles are designed to allow transparent reconnection upon momentary network outages. Under SMB1.0 file handles would be invalidated and any pending IO requests would fail in the event of network outage. The inclusion of durable handles in SMB2.0 was designed to improve accessibility on temporary disconnection such as when moving between wireless network access points.

Durable handles allow the client and server to renegotiate access to shares at the protocol layer without requiring any additional application awareness. Applications will just continue to perform IO requests to the files they have handles for and the underlying protocol deals with any housework required after the network outage – such as re-opening files, re-locking files, and so on.

Applications do not need to be rewritten to take advantage of this feature as it is handled in a lower layer of the communication.

Improved Message Signing

Message signing is a feature by which communications between the client and server can be digitally signed. Digital signing enables the recipient of the messages to confirm their point of origin and their authenticity. This form of security mechanism in SMB2.0 helps avoid issues like modifying packets and "Man in the Middle" attacks. Message signing was available under SMB1.0 but has been enhanced for security and performance under SMB2.0. Under SMB1.0, messages had MD5 hashes attached; these have been replaced with HMAC SHA-256 hashes under SMB2.0.

Digital signing is turned on automatically on Active Directory (AD) domain controllers but it is not on by default on client configurations. SMB is used to download Group Policy configurations, and SMB signing provides a method by which clients can ensure they are receiving genuine Group Policy.

Microsoft does not recommend modifying the settings for message signing for anything other than the default – AD domain controller-originated traffic being signed and client-originated traffic being unsigned. This is due to the additional processing load imposed in checking hash values for incoming messages and also calculating hash values for outgoing messages. Enabling for all traffic will have an impact on SMB performance but may be necessary in secure environments where client-server traffic must be completely trusted.

Under SMB2.0, the SMB packet signing is negotiated at session creation time. It is possible to have signed messages originating only from the client, only from the server, or from both.

Controlling digital signing is done on the Oracle ZFS Storage Appliance from the SMB service page or through the registry or AD Group Policy on the Windows clients.

Improved Scalability

With the simplification of the command set under SMB2.0, the processing load has been reduced on servers. This reduction improves response times and, with compound requests, offers flexibility unavailable under SMB1.0 to create command pipelines where all processing is carried out on the SMB server with responses returned to the clients where appropriate.

Increased addressing allows for the number of concurrent users, shares, and open files to be increased, and reduced network traffic can result in a further increase in concurrent users.

Additionally, improvements in Windows client implementations through SMB2.0 and the networking stack, along with improved performance provided by caching of file handles and properties, further increases the performance advantages of SMB2.0

Conclusion

The SMB2.0 upgrade to the Oracle ZFS Storage Appliance provides increased functionality and reduced protocol overheads, allowing for more efficient use of the storage resources and services, while increasing the number of users without adding any additional hardware components to the Oracle ZFS Storage Appliance.

SMB2.0 Update for the Oracle ZFS Storage Appliance
August 2014 Version 1.0
Author: Oracle Application Integration Engineering, Andrew Ness

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Oracle is committed to developing practices and products that help protect the environment

Hardware and Software, Engineered to Work Together