HARDWIRED:
An SMB Offload Engine

Christopher R. Hertel
Samba Team
SambaXP
- virtually Göttingen -
May, 2020
Quick Introductions
Me:

- Samba Team Elder
- Data Storage Geek
- Writer / Developer

The opinions expressed are my own and not necessarily those of:

- My spouse, my children, my dog, my colleagues,
- my spirit familiar, the Internet Voices,
- the monster under the floor,
- the basement mice, etc.
Introductions

ACK:
The work I am presenting parallels some ongoing work by others. Kudos to them. I'm not trying to compete. This is just what intrigues me.
SmartNICs
SmartNICs

Network

Bus

Memory

BRAINS
It's a network computer on a card.

- Sorta like a directly attached Raspberry Pi on Steroids with a Jet Engine and a sugar buzz.
SmartNICs

"Storage is the 'Killer App' for SmartNICs."

Hmmm...
What could we do with that?
SMB Offload
SMB Offload

- Marshalling and Unmarshalling
  - Packing and unpacking of packets
  - Compression / Decompression
- Host-provided State
  - Signing and Sealing keys
- Zero Copy I/O
  - Fast path for Read, Write, and Flush
- SMB2, SMB3, **no SMB1**
SMB Offload

How would such a thing fit into an SMB2/SMB3 implementation?
SMB Offload Stack

- Semantic Layer: SMB Server
  - Semantics and Metadata

- Fuzzy Layer: Driver/Library
  - Offload Engine Interface

- Syntactic Layer: Offload
  - In-Memory Layout

- Network Layer: Transport
  - Wire Format
The Semantic Layer

...is where the serious work gets done:

- Manage Windows FS Semantics
  - Locking, Identity, EAs, etc.
- Local Filesystem Interface
  - E.g.: POSIX Layer
  - Sync'd Access (Local, NFS, Object)
- Metadata Management
  - ACLs, Attributes
- Cluster Support
The Semantic Layer

...is not part of the offload engine.

- Must keep it in mind, though.
- The API needs to be useful.
- Different implementations should be able run over the same API.
- Even run different implementations in parallel.
The Fuzzy Layer

...is not (yet) well defined. It provides the interface between the Server and the Offload Engine.

- **Shared State:**
  - Encryption keys.
  - Sessions, Tree Connects, and Open Files.
- **Communicate with the NIC.**
  - Tell it what to do.
The Fuzzy Layer
...is not (yet) well defined. Here's what we need:
● A rational, well documented API.
● A stackable low-level for adding new dialects and capabilities.
● State management.
● Device Driver / Library / Toolkit?
The SmartNIC Layer
...is the *raison d'être* for this effort.
- Offload encryption & compression.
- Handle message syntax errors.
- Support SMB3 Multichannel.
- Support multiple transports.
- Hide those details from the upper levels of the stack.
Yet Another Project
Yet Another Project

Zambezi

- https://gitlab.com/ubiqx/zambezi

- LGPL
- Only code that's ready
- ...and excessively well documented.
For convenience, messages are listed in six debatably semi-logical categories:

<table>
<thead>
<tr>
<th>Managing Connections</th>
<th>Share Access</th>
<th>Open/Close, Lock/Unlock</th>
</tr>
</thead>
<tbody>
<tr>
<td>● NEGOTIATE (0x0000)</td>
<td>● TREE_CONNECT (0x0003)</td>
<td>● CREATE (0x0005)</td>
</tr>
<tr>
<td>● SESSION_SETUP (0x0001)</td>
<td>● TREE_DISCONNECT (0x0004)</td>
<td>● CLOSE (0x0006)</td>
</tr>
<tr>
<td>● LOGOFF (0x0002)</td>
<td></td>
<td>● LOCK (0x000A)</td>
</tr>
<tr>
<td>● ECHO (0x000D)</td>
<td></td>
<td>● OPLOCK_BREAK (0x0012)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fundamental I/O</th>
<th>Metadata Query and Set</th>
<th>Odds and Ends</th>
</tr>
</thead>
<tbody>
<tr>
<td>● READ (0x0008)</td>
<td>● QUERY_DIRECTORY (0x000E)</td>
<td>● CANCEL (0x000C)</td>
</tr>
<tr>
<td>● WRITE (0x0009)</td>
<td>● CHANGE_NOTIFY (0x000F)</td>
<td>● SMB2 Error Response</td>
</tr>
<tr>
<td>● FLUSH (0x0007)</td>
<td>● IOCTL (0x000B)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● QUERY_INFO (0x0010)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● SET_INFO (0x0011)</td>
<td></td>
</tr>
</tbody>
</table>
Several have the same basic format:

typedef struct
{
    uint16_t StructureSize;
    uint8_t   Reserved[2];
} smb2_BaseMsg;

- LOGOFF Request/Response
- TREE_DISCONNECT Request/Response
- ECHO Request/Response
- CANCEL Request
- LOCK Response
- FLUSH Response

Parse/pack code for all 9 types is complete.
Consider **SMB2 Echo**

- In SMB2/3, Echo is only valid within a Session.
  - No Payload
  - No Repeats

- Does it ever need to leave the NIC?
  - Is the SMB2 Server still Running?
  - Is it still serving the Session?

- The SMB2 Server must respond to the Offload with A-OK.
Where else might this be useful?

- Software Defined Network Devices
- Proxy and Cache Servers
- WAN Accelerators
- Remote Access Portals
Goals:

★ Git 'er done.
★ Work with the SNIA
  ○ Standardize the API
  ○ Fork a reference implementation under an additional license
★ Partner with others to implement on SmartNICs
★ Find new and interesting uses
The End
Questions?