INTRODUCE IN-KERNEL SMB3 SERVER CALLED CIFSD

Namjae Jeon
Samsung Electronics
Hyunchul Lee
LG Electronics
• Introduction
• Architecture
• Performance/Stability/Compatibility
• SMB over RDMA
• Plans
INTRODUCTION
Fill the blank of kernel SMB Server

KERNEL

USERSPACE

NFS

NFSD

CIFS

SMB SERVER

CIFSD

NFS Ganesha

SAMBA

2019 Storage Developer Conference. © Samsung Electronics. All Rights Reserved.
Full SMB stack support

- SMB1 ~ 3.1.1 version support
- SMB1, 2.0 optional and disabled by default
- Pending SMB2 notify
  - fsnotify functions are not exported
  - raise a request to export them
Supported Features

- Authentication support
  - NTLM/NTLMv2

- Performance features
  - Oplock/lease, compounding, copy offload and SMBDirect

- Security features
  - Signing (HMAC-SHA256, AES-CMAC)
  - Encryption (AES-CCM/GCM)
**linux-cifsd project**

- Github Repos
  - [https://github.com/cifsd-team/cifsd](https://github.com/cifsd-team/cifsd)
  - [https://github.com/cifsd-team/cifsd-tools](https://github.com/cifsd-team/cifsd-tools)
- Mailing-list
  - [linux-cifsd-devel@lists.sourceforge.net](mailto:linux-cifsd-devel@lists.sourceforge.net)
- Supported Linux kernel versions
  - Linux 4.1 kernel or later
Test automation

- Use Travis CI
  - Github integration to test per pull request
- Build test
  - Linux 4.1 kernel and The latest Linus-tree
- Stability test
  - Xfstests(72)
  - Smbtorture(180)
- Need minimum implementation
  - Commands needed to work as file server
- DCERPC engine is located in userspace
- DCE/RPC response is prepared from the user space
- passed over to kcifsdsd kernel thread
Configuration

- User configure parameters in smb.conf
- Format is compatible with samba’s one
- Minimum implementation
  - Select what user generally use a lot
  - Add more parameters users requested
- List-up supported parameter list in smb.conf.example in cifsd-tools
SMB1 disabled by default

- Keeping for clients which doesn’t switch to SMB2 yet
- Planning to remove SMB1 in cifsd when contributing to mainline
OpenWRT Built-in cifsd

- OpenWRT is wireless router open source project
- Built-in cifsd as file sharing function
- OpenWRT had built-in Samba, Why?
  - Easy cross compile without special handling
  - Small binary size (page 16)
  - Low CPU usage and better performance
KEY CONCEPTS
In-kernel server

- No system call cost
- Shorter path to use VFS and network stack
- No duplicate memory allocation (Filesystem metadata)
Optimized SMB over RDMA
Oplock lease/better handling

**Kernel Space SMB Server (CIFSD) (Uniform Address Space)**

<table>
<thead>
<tr>
<th>Client 1</th>
<th>Client 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>kcifsd/1</td>
<td></td>
</tr>
<tr>
<td>kcifsd/2</td>
<td></td>
</tr>
</tbody>
</table>

**Open “File”**

- Open OK – Oplock granted
- Oplock Break Request
- Oplock Break Response

**User Space SMB Server (SAMBA)**

<table>
<thead>
<tr>
<th>Client 1</th>
<th>Client 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>smbd/1</td>
<td></td>
</tr>
<tr>
<td>smbd/2</td>
<td></td>
</tr>
</tbody>
</table>

**Open “File”**

- Open OK – Oplock granted
- Oplock Break Request
- IPC Message – Oplock Break Request
- IPC Message – Oplock Break Done
- Open OK – Oplock granted
Minimal SMB3 server

Code line count comparison

- samba 4.10.6
- nfsd+lockd+sunrpc
- nfsd+lockd
- cifsd
- cifsd(only SMB3)

The line of code
**Lightweight file share**

- Reported by Andy Walsh (OpenWRT)

<table>
<thead>
<tr>
<th>Binary Size</th>
<th>Main</th>
<th>Extra</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>cifsd</td>
<td>128KB (cifsd kmod, tools)</td>
<td>61KB (crypto kmods) + 872KB (glib2)</td>
<td>1061KB</td>
</tr>
<tr>
<td>samba4</td>
<td>6MB (samba libs, server package)</td>
<td>64KB (libtirpc, etc)</td>
<td>6064KB</td>
</tr>
</tbody>
</table>
ARCHITECTURE
User and Kernel thread

**ucifsd (User-level thread)**
Work related to non-performance, more memory/more resource feature. i.e. DCERPC, user/share management, configuration.

**kcifsd (Kernel-level thread)**
Work related to performance, It contain SMB engine, transport, server handler, oplock/lease.
IPC between kernel/user

- Use Netlink interface
- Specify a few command-set

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFSD_EVENT_HEARTBEAT_REQUEST</td>
<td>Monitor cifsd is alive</td>
</tr>
<tr>
<td>CIFSD_EVENT_STARTING_UP CIFSD_EVENT_SHUTTING_DOWN</td>
<td>Transfer the initial information necessary for the start and shutdown</td>
</tr>
<tr>
<td>CIFSD_EVENT_LOGIN_REQUEST CIFSD_EVENT_LOGIN_RESPONSE</td>
<td>Transfer the user account / password information necessary for login</td>
</tr>
<tr>
<td>CIFSD_EVENT_SHARE_CONFIG_REQUEST CIFSD_EVENT_SHARE_CONFIG_RESPONSE</td>
<td>Transfer the share configuration</td>
</tr>
<tr>
<td>CIFSD_EVENT_TREE_CONNECT_RESPONSE CIFSD_EVENT_TREE_DISCONNECT_REQUEST</td>
<td>Transfer the tree connect info</td>
</tr>
<tr>
<td>CIFSD_EVENT_RPC_REQUEST CIFSD_EVENT_RPC_RESPONSE</td>
<td>Transfer DCERPC requests</td>
</tr>
</tbody>
</table>

Generic Netlink Socket Usecase
Architecture

User Space
- cifsuseradd
- ucifsd

Kernel Space
- kcifsd/0 (forker thread)
- kcifsd/1
- kcifsd/2
- kcifsd/N

NETLINK/SYSFS Interface
- DCE/RPC
- Share configuration
- ID/PW configuration

Authentication
- NTLM
- NTLMv2
- Kerberos

Dialects
- SMB1
- SMB2
- SMB2.1
- SMB3
- SMB3.1.1

SMB1, SMB2, SMB2.1, SMB3, SMB3.1.1

Local Filesystem

CIFS

Client
- SOCKET (445)
Test Environment

- Direct connection on TWO PC
- Tools: IOZONE, fileop, bench-oplock
- Client: CIFS Client(SMB3.11)
- Share: tmpfs
- SMB Server: samba 4.7.6 and cifsd
Single Read/Write iozone throughput

![Graph showing throughput vs record length]

- **Kilobyte/second**
- **Record length (KB)**: 4, 8, 16, 32, 64, 128
- **Throughput (KB/s)**: 0, 10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000, 90000, 100000

- **samba read**
- **cifsd read**
- **samba write**
- **cifsd write**
Multi Read/Write iozone throughput

![Graph showing Multi Read/Write iozone throughput](image-url)
Fileop throughput

Ops/second

- mkdir
- chmod
- rmdir
- create
- open
- read
- write
- close
- stat
- access
- chmod
- readdir
- link
- unlink
- delete

<table>
<thead>
<tr>
<th>Ops/second</th>
<th>samba</th>
<th>cifsd</th>
</tr>
</thead>
<tbody>
<tr>
<td>mkdir</td>
<td>500</td>
<td>13%</td>
</tr>
<tr>
<td>chdir</td>
<td>1000</td>
<td>16%</td>
</tr>
<tr>
<td>rmdir</td>
<td>1500</td>
<td>22%</td>
</tr>
<tr>
<td>create</td>
<td>2000</td>
<td>28%</td>
</tr>
<tr>
<td>open</td>
<td>2500</td>
<td>18%</td>
</tr>
<tr>
<td>read</td>
<td>3000</td>
<td>13%</td>
</tr>
<tr>
<td>write</td>
<td>3500</td>
<td>11%</td>
</tr>
<tr>
<td>close</td>
<td>4000</td>
<td>10%</td>
</tr>
<tr>
<td>stat</td>
<td>4500</td>
<td>9%</td>
</tr>
<tr>
<td>access</td>
<td></td>
<td>12%</td>
</tr>
<tr>
<td>chmod</td>
<td></td>
<td>28%</td>
</tr>
<tr>
<td>readdir</td>
<td></td>
<td>9%</td>
</tr>
<tr>
<td>link</td>
<td></td>
<td>11%</td>
</tr>
<tr>
<td>unlink</td>
<td></td>
<td>12%</td>
</tr>
<tr>
<td>delete</td>
<td></td>
<td>10%</td>
</tr>
</tbody>
</table>
File lookup time (ls -l)

![Graph showing file lookup time for samba and cifsd. The graph compares the time (in milliseconds) for different numbers of files. The x-axis represents the number of files, ranging from 1000 to 50000. The y-axis represents the time, ranging from 0 to 1400 milliseconds. The graph indicates a 44% improvement for cifsd compared to samba at certain file counts.]
Bench oplock

Ops/second

samba

cifsd

+47%
Playing a 4K movie
mpstat –p ALL 2

CPU Usage

- cifsd
- samba
CPU Usage

-32% decrease in CPU usage while playing a 4k movie.

- Samba
- Cifsd

CPU usage Avg (%)

Playing a 4k movie
## Compatibility

<table>
<thead>
<tr>
<th>SMB CLIENT VERSIONS</th>
<th>CIFSD SUPPORTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows XP (SMB 1.0)</td>
<td>✓</td>
</tr>
<tr>
<td>Windows Vista (SMB 2.0)</td>
<td>✓</td>
</tr>
<tr>
<td>Windows 7 (SMB 2.1)</td>
<td>✓</td>
</tr>
<tr>
<td>Windows 8 (SMB 3.0)</td>
<td>✓</td>
</tr>
<tr>
<td>Windows 10 (SMB 3.1.1)</td>
<td>✓</td>
</tr>
<tr>
<td>MacOS (~ High Sierra)</td>
<td>✓</td>
</tr>
<tr>
<td>Ubuntu File Explorer</td>
<td>✓</td>
</tr>
<tr>
<td>Linux CIFS Client (linux v5.3-rc6)</td>
<td>✓</td>
</tr>
</tbody>
</table>
Stability

- **XFSTESTS**
  - Total: 82, PASS: 82
  - [Detail test result link](#)

- **SMB TORTURE**
  - Total: 311, PASS: 205, FAIL: 11, N/S: 95
  - [Detail test result link](#)
Catching up cifs’s new features

- SMB Direct
- Posix extension mode bit
- GCM encryption
- rsize and wsize increase
- Test Automation
Hardware Acceleration

- Performance advantage through HW acceleration
- Accelerate encryption (AES-GCM) performance with AES-NI support in kernel

![Bar Graph]

```
dd if=/dev/zero of=aes.bin bs=128k count=1024
```
SMB over RDMA
- New transport protocol for supporting SMB over RDMA
  - Introduced in SMB 3.0 (Windows Server 2012)
- Operates over Reliable RDMA transports
  - Infiniband, RoCE, iWARP
- Defines transport framing for SMB exchange
SMB Direct

- Negotiate capabilities
  - Version, message size, credits
- Supports datagram-type send / receive for the exchange of SMB messages
  - RDMA send / receive
  - Fragmentation / Reassembly
  - Credits management
- Supports remote direct data placement for moving data between memory buffers on each peer
  - RDMA read / write
  - Memory registration and advertisement of the location to another peer
  - Remote memory invalidation (SMB 3.02)
SMB Direct and TCP

client

TCP (445)

kcifsd/1

SMB Direct (5445)

kcifsd/2

SMB Engine
Send SMB3 Response

Client

1. Wait for send credit & Segment

2. Gather and post RDMA sends at once

3. Reassemble

Server

1. Wait for send credit & Segment

2. Gather and post RDMA sends at once

SMB

SMB Direct
Transfer File Data

Client

1. Advertise memory location

SMB

message

data buffer

Server

2. Post RDMA reads

SMB Direct

message

data buffer

1. Advertise memory location
Test Environment

- Client / Server PC
  - Intel i7-3770k 3.90GHz
  - 16GB RAM
  - Mellanox ConnectX-2 10G
- Direct connection on two PC
- Linux kernel v5.2 cifs client (SMB 3.02)
- fio
Write Throughput
Read Throughput

![Graph showing read throughput with various line colors representing different values of 1, 4, 16, 64, and 256. The graph plots throughput in MB/s against I/O size in KB or MB.]
Plans

- Windows Protocol Testsuite
- SMB2 notify
- WinACL support
- Check compatibility with different NICs
- Support Multichannel
The end, Thank you!
Any question?