Improvements in Samba to take advantage of OneFS

What is OneFS?

- File System & Operating System
- Based off FreeBSD
  - ifs.ko
- POSIX & Windows Semantics
- Clustered File System
  - Store single volume, across multiple disks, across multiple machines
  - Think RAID across machines instead of disks

Isilon OneFS Cluster

- NAS file server
- Single file system (3.45 PB)
- Fully symmetric peers
  - No metadata servers
  - 3 to 96+ nodes
- Fast intra-cluster network
  - InfiniBand
- Each Node:
  - CPU
  - Memory
  - Disks (12 to 36)
  - Commodity hardware

What Makes Isilon OneFS Special?

- Single pool of storage
- Granular data protection
  - » Down to the file level
  - » 8x mirroring or +4 recovery
- Easy to manage and grow
  - » Add additional nodes in 60 seconds
  - » Automated data balancing
- Extreme performance for concurrent access

Samba in OneFS

Used over 6 years – **2.2** through **3.4**

Getting Smarter

- 3 large code merges
- On top of 3.0.24
  - 35,000 lines of diff
  - No defined patch stack
- Isilon specific code was not well abstracted
- Have a problem? Add a new parameter!

OneFS File System Features

- NTFS ACL storage & enforcement
- NTFS Alternate Data Streams
- Zero-copy writes - like Linux splice()
- Snapshots with Volume Shadow Copy Service (VSS)
- Bulk Directory Enumeration - readdirplus() syscall

Cluster Coherent

- Win32 like CreateFile() syscall
  - Oplocks
  - Share mode locks
  - Delete on close
- Windows Byte Range Lock semantics
- Kernel based Change Notification

Utilizing OneFS in Samba

VFS - Virtual File System

- VOP - Virtual OPerations
- Function dispatch table
- Shared object library - onefs.so
  - source3/modules/vfs_onefs.c
  - source3/modules/onefs_*.c
  - 31 operations defined

VFS - Access Control Lists

onefs.so

- source3/modules/onefs_acl.c
- 3 VFS functions
  - SMB_VFS_FGET_NT_ACL()
  - SMB_VFS_GET_NT_ACL()
  - SMB_VFS_FSET_NT_ACL()
    - onefs_fset_nt_acl()
      - ifs_set_security_descriptor()
- Wrappers around OneFS syscalls
- Plus some helper functions

VFS - Byte Range Locks

onefs.so

- source3/modules/onefs_cbrl.c
- Added 3 new VFS functions
  - SMB_VFS_BRL_LOCK_WINDOWS()
  - SMB_VFS_BRL_UNLOCK_WINDOWS()
  - SMB_VFS_BRL_CANCEL_WINDOWS()
    - onefs_brl_cancel_windows()
      - ifs_cbrl()
- Moved old calls to default implementations
- Wrappers around OneFS syscalls
- Plus some helper functions

VFS - Change Notify

onefs.so

- source3/modules/onefs_notify.c
- Only 1 VFS call!
  - SMB_VFS_NOTIFY_WATCH()
- 6 methods to implement all functionality
  - Register watch request
    - event_add_fd()
  - Kernel event handler
  - Notification dispatcher
  - Translate Win notify mask to IFS mask
  - Translate IFS notify mask to Win mask
  - Destructor

VFS - Open Path

onefs.so

- source3/modules/onefs_open.c
- SMB_VFS_CREATE_FILE()
  - New VOP
  - Allows atomic implementation of:
    - Access checks
    - Share mode lock acquisition
    - Oplock acquisition
    - Setting security descriptor at create time
    - Setting DOS attributes at create time
- Abstracted and simplified the very complex open() path

VFS - Bulk Directory Enumeration

onefs.so

- source3/modules/onefs_dir.c
- Cadillac vs Camry
  - SMB_VFS_FIND_FIRST()
  - SMB_VFS_FIND_NEXT()
- TRANS2 -> POSIX VFS -> ReadDirPlus()
  - Bulk -> Singular -> Bulk
- SMB_VFS_READDIR()
  - OPENDIR, SEEKDIR, TELLDIR, REWINDDIR, CLOSEDIR
- 25% decrease in latency for SMB_FIND_FILE_BOTH_INFO

VFS - Snapshots

onefs_shadow_copy.so

- source3/modules/vfs_onefs_shadow_copy.c
- source3/modules/onefs_shadow_copy.c
- 41 VOPs
- Need second module because we can’t define multiple functions per VOP within one module
- Translate @GMT Windows names to OneFS snapshot path

VFS - Performance Counter

perfcount_onefs.so

- Track SMB performance statistics
  - Count SMB packet types
  - Max/Min/Avg size/latency
  - Calculate rates: ops/sec, bytes/sec
  - Track per-user statistics
- source3/modules/perfcount_onefs.c
  - OneFS syscalls
  - Stats stored in kernel memory
  - Stats retrieved through CLI program
- source3/modules/perfcount_test.c
  - Output to smb.log
- perfcount module = pc_test

VFS - Performance Counter

isi statistics

COUNT	IMIN	IMAX	IAVG	OMIN	OMAX	OAVG	LAVG	OPRATE	QUERY
158262	76	266	141.16	0	39	0	157.1	2.22	1.cifs.namespace_read.trans2:qpathinfo
57954	59	59	59	0	382	0.05	572.85	0.81	1.cifs.read.readx
48709	86	300	191.5	0	107	105.94	1081.27	0.68	1.cifs.create.ntcreatex
47954	41	41	41	39	39	39	801.22	0.67	1.cifs.file_state.close
23960	77	116	96.5	0	0	0	144.51	0.34	1.cifs.namespace_write.trans2:setfileinfo
12000	97	370	97.14	51	51	51	103.75	0.17	1.cifs.write.writex
745	133	133	133	101	101	101	467.91	0.01	1.cifs.session_state.negprot
741	266	348	345.79	204	210	209.98	73779.63	0.01	1.cifs.session_state.sesssetupx
741	266	348	345.79	52	58	57.98	39355.48	0.01	1.cifs.session_state.tconx
739	35	35	35	39	39	39	1432.39	0.01	1.cifs.session_state.tdis
203	86	290	191.01	0	0	0	38074.88	0	1.cifs.namespace_read.trans2:findfirst
37	72	72	72	0	0	0	540.59	0	1.cifs.namespace_read.trans2:qfileinfo
4	148	148	148	0	0	0	403295.75	0	1.cifs.namespace_read.trans2:findnext
3	84	84	84	0	0	0	115.67	0	1.cifs.file_state.nttrans:notify_change
1	84	84	84	0	0	0	238	0	1.cifs.unimplemented.nttrans:ioctl

Oplock Improvements

source3/smbd/oplock_onefs.c

- CreateFile() syscall acquires oplock
- Utilize OneFS event channel for oplock breaks
  - smbd listens on fd for kernel events
- No VFS interface for kernel oplocks, yet.
- Added support for level 2 kernel oplocks
- Added spanning level 2 oplock breaks
- Added flags to specify kernel level oplock capability
  - KOPLOCKS_LEVEL2_SUPPORTED
  - KOPLOCKS_DEFERRED_OPEN_NOTIFICATION
  - KOPLOCKS_TIMEOUT_NOTIFICATION
  - KOPLOCKS_OPLOCK_BROKEN_NOTIFICATION

Non-Modular

smbd code

- A few things we couldn't modularize
  - struct file_id

```
struct file_id {
        uint64_t devid;
        uint64_t inode;
        uint64_t extid;
        /* Support systems that use an extended
           id (e.g. snapshots). */
};
```

Other Additions

Micellaneous

- smbtorture tests
- sharesec utility
- Access based share enumeration
- Coverity fixes
- Backtrace symbols for FreeBSD
- Allow setting of create time via SMB
- Refactored a lot of common helper functions

Conclusions

Be smarter

- A lot of modularity in Samba
  - Take advantage of it!
- Easy to add OS specific features through VFS
- Push as much of your work upstream as possible
  - Less merge headache
  - Better compilation and testing coverage
- Thorough code audit found a lot of small improvements
  - SENDFILE/RECVFILE VOPs didn’t actually work
  - Detect stream names properly

Questions?

Contact

- Steven Danneman
  - » steven.danneman@isilon.com
  - » steven@samba.org