SerNet

Samba Scalability and Performance Tuning

September 17, 2013

Volker Lendecke

SerNet GmbH, Göttingen - Berlin

What can impact Samba performance

SerNet

- SMB/CIFS: Strict request/response pattern
  - For example OS/X Finder is really slow with listing directories
- Delays can be located in Samba itself
- Delays outside Samba
  - Saturated disk, network, RAM, CPU
- Other applications running on the server
  - Backup, rsync, updatedb (locate)

Key components of a file server

SerNet

- Network, CPU, RAM, Disk subsystem
  - Bottleneck could be in any component
- Kernel
  - Samba can't change the kernel, only its use of it
  - syscalls are cheap, but not free
- libc: malloc is expensive
- Modern X86 CPUs are just silly fast, ARM/MIPS not so
  - Optimizations don't apply everywhere

Identify bottlenecks

SerNet

- It's not easy to figure out where smbd spends its time
  - valgrind callgrind, Linux perf
- strace -ttT -o strace.out
  - `06:10:07.030176 getdents(3, /* 126 entries */, 32768) = 4328 <0.000709>`
  - grep -v 0.00 strace.out finds long syscalls
  - poll usually means waiting for the client
- Slows things down

Blaming others

SerNet

- vfs_time_audit intercepts all VFS calls
- If call takes longer than defined threshold it will output a warning message, including operation, file and time spent
- Good to detect sporadic hangs or overload situations
- As it intercepts all VFS calls, interpretation is sometimes difficult (e.g. create_file vs open)
- `WARNING: VFS call "unlink" took unexpectedly long (13.238542 seconds) -- Validate that file and storage subsystems are operating normally`

talloc_pooled_object

SerNet

- talloc: hierarchical memory allocator with destructors
  - C++ for those who don't want to read >1000 pages language spec
  - based on malloc
- talloc_pool: One malloc with a simple incrementing pointer
  - Ideal for request/response
  - One large malloc per SMB
  - SMB is done, one object to free

talloc_pooled_object

SerNet

- Samba has a lot of internal „objects“
  - smb_filename: one struct, 1-3 substrings as talloc children
  - tevent_req: one struct, 2 substructs
- talloc does individual mallocs on structs and substructs
- talloc_pooled_object: one malloc for all
- ```
  out = talloc_pooled_object(mem_ctx, struct smb_filename,
              num, base_len + stream_len + lcomp_len);
  out->base_name = talloc_memdup(out, in->base_name,
                                 base_len);
  ...
  ```
  ```

The SMB protocol

SerNet

- Historically grown protocol
- Traces back to MS-DOS system calls
- Semantic roots back in single tasking DOS
- „Drive D:“ on a different machine

SMB 1, 2, 3

SerNet

- SMB1: Only protocol up to Vista
- Evolutionary steps from DOS to Windows 2003
  - e.g. NTFS semantics, Unicode file names
- Hundreds of client and server implementations
  - More if you count the rootkit script kiddies
- SMB2: Vista, new implementation
- SMB3: Windows 8 / Windows 2012
  - Scalability, Availability

Open a file in Posix

SerNet

- Check file path
  - Do the directories in the path exist?
  - x-Bit permissions on the whole path?
- File exists, sufficient permissions?
  - File open!
- Opening existing files just read from disk
- Every process / node can do this simultaneously
- No other process needs to be told

Open a file in SMB

SerNet

- Similar operations to Posix
  - Path operations, permissions
- BUT: Share modes
- Windows CreateFile API, parameter dwShareModes
  - If this parameter is zero and CreateFile succeeds, the file or device cannot be shared and cannot be opened again until the handle to the file or device is closed.

Samba Architecture

SerNet

- Single Threaded, multi-process File Server
- Threads for Async pread/pwrite (and creat)
- One TCP connection per client
- One smbd per TCP connection
  - Multichannel will change this
- Many users and shares per TCP connection

Share Modes

SerNet

- Every Samba process knows about all open files
- TDB files provide a shared memory area
- Multi-Writer key/value
- Share modes database (locking.tdb) indexed by device and inode
- Open handle info held in a record: Share Modes, transient time stamps, delete on close flag

tdb

SerNet

- Trivial Database, 1st implementation less than 1000 lines
- Shared memory with mmap(2), pread/pwrite fallback
- Hash table: Linear list (Hash Chain) for collisions
- Memory management via simple free list
- Hash chains are protected by fcntl locks
- Transactions: Changes protected by msync/fsync

fcntl byte range locks

SerNet

- ```
  short l_type Type of lock; F_RDLCK, F_WRLCK, F_UNLCK.
  off_t l_start Relative offset in bytes.
  off_t l_len
  Size; if 0 then until EOF.
  ```
- Arbitrary file ranges can be locked, independent of file size
- Advisory locking, read/write is not blocked
  - Just an IPC mechanism
- Automatic cleanup at process exit
- almost ideal for tdb
  ```

fcntl Scalability

SerNet

- Recover a highly loaded cluster
- ctdb takes an fcntl lock across the whole database
  - allrecord lock
- Thousands of smbds wait for recovery to finish
- Allrecord Unlock: Thousands of processes woken up
- Hey, we got 100.000 hash chains, so where's the problem?

/usr/src/linux/fs/locks.c

SerNet

```
static DEFINE_SPINLOCK(file_lock_lock);
void lock_flocks(void) {
        spin_lock(&file_lock_lock);
}
```

- All fcntl operations need to go through one single spinlock
- Leftover of the Big Kernel Lock
- Deadlock detection across multiple files
- Thousands of processes waiting for a single spinlock
  - 32 CPUs creating a thundering herd

pthread_mutex_t

SerNet

- Basic locking for threads
- glibc under Linux: Atomic CPU operations
  - Only a contended mutex does a syscall
- Coordination within a process
- pthread_mutexattr_setpshared:
  - Mutexes across process boundaries

Robust mutexes

SerNet

- Mutexes ususally must not be cleaned up
  - A crashed thread blows away the whole process
- pthread_mutexattr_setrobust:
  - Automatic cleanup (EOWNERDEAD)
  - „modern“ Linux and OpenSolaris
- Linux is not 100% robust, but good enough
- No global spinlock

Questions?

SerNet

Volker Lendecke, VL@samba.org, VL@sernet.de

SerNet GmbH

Bahnhofsallee 1b
37081 Göttingen

Schützenstr. 18
10117 Berlin

tel +49 551 370000-0
fax +49 551 370000-9

+49 30 5 779 779 0
+49 30 5 779 779 9

http://www.sernet.de