Bulk Data Service - 60MB/sec “NFS”

Larry McVoy

lm@sgi.com

Silicon Graphics Engineering

Outline

How fast is NFS?

How fast should it be?

Using SGI technology for higher bandwidth

User level client & server implementation and performance

Kernel level client

BDS technology and NFS

Portability & Availability

How fast is NFS?

NFS V2/UDP - about 2.5MB/sec

NFS V3/UDP - up to 18MB/sec over Hippi

- That’s real NFS too, XDR & all
- Up to 30MB/sec in loopback

How fast should it be?

Goals

- Replace Cray disk farms with SGI disk farms
- Crays must be able to read at 50+ MB/sec
- At least 4 streams concurrently (200MB/sec sustained)

What technology do we have?

Local XFS file system w/ O_DIRECT

- Up to 500MB/sec for one reader
- Up to 250MB/sec for one writer

TCP over Hippi

- 65 MB/sec when touching the data, 92MB/sec max

How do we get high bandwidth to remote data?

XFS is fast, TCP is fast

- Glue ’em together

BDS is the result

- User level server
  - Sort of like ftpd
- User level client library
  - Portable from Linux on PCs to Unicos on Crays
- Kernel level client
  - Implements NFS security

BDS - Bulk data service

Big block remote I/O protocol

- No fixed block sizes
- Limit is the DMA transfer size on server host
- Block size is whatever passed to read/write

Only implements a few interfaces

- open/read/write
  - all other interfaces provided by NFS
- read/write calls include seek pointers
- read/write calls may be asynchronous

User level BDS

Both client and server code

Client code is a library

Server code is a daemon a la FTP

User level BDS client side

User level library

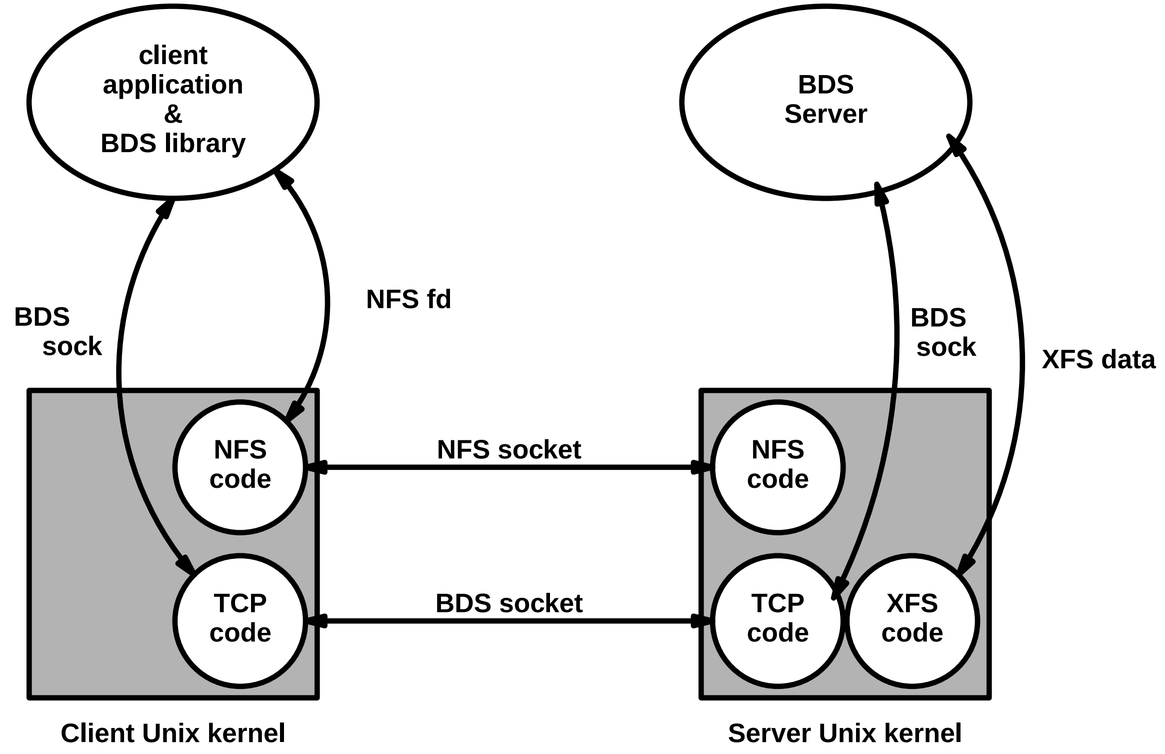
- Catches calls to open/read/write
- BDS protocol turned on by O_DIRECT
- Establishes socket to BDS server
- Has size & alignment restrictions like XFS
- Remaps I/O calls to remote I/O calls via BDS protocol

User level BDS server side

FTP-like daemon

- Forks a new process for each open
- Does all direct I/O
- Does simple read ahead
- Uses very little CPU
- Uses a lot of XFS and network bandwidth

User level BDS enhanced NFS



User level BDS client read example

read(nfsfd, buf, nbytes)

remapped to

- send read request on socket
- read response (includes byte count)
- read data

error conditions indicated by a closed socket

User level BDS server read steady state

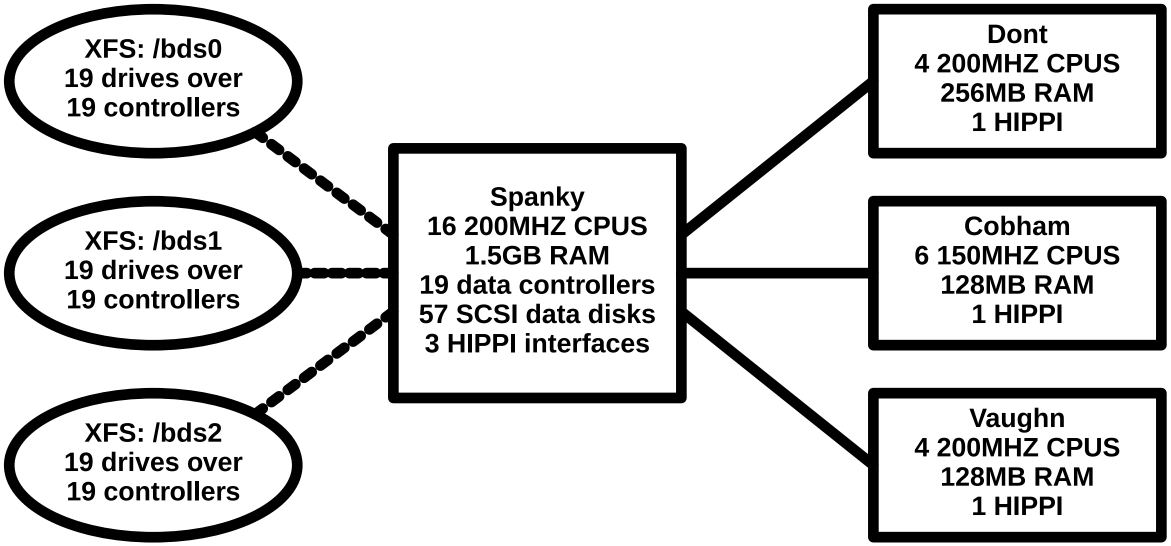
get request

match request offset & length to read ahead

send read ahead buffer

do the next read ahead

User level performance configuration



User level results

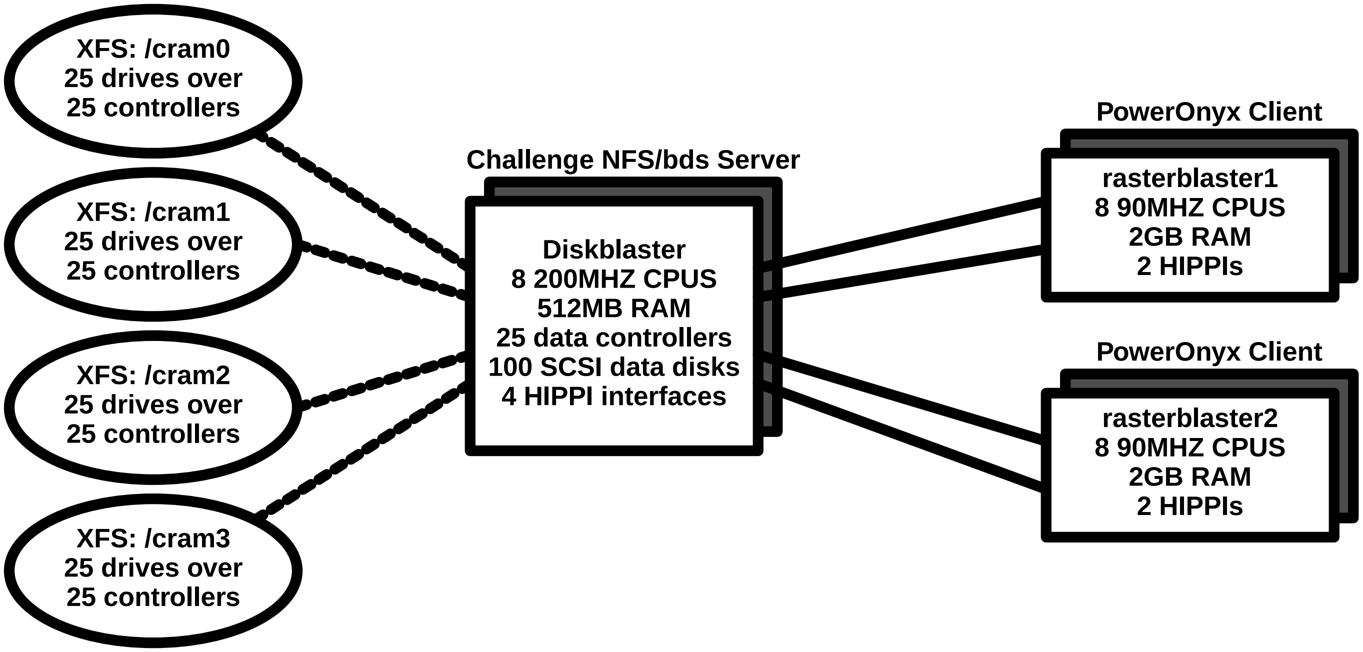
Reads with 1.5MB read requests

- 1 client: 67MB/sec
- 2 clients: 2 * 63 = 126MB/sec
- 3 clients: 3 * 61 = 183MB/sec

Reads vs file size

- Includes all start up overhead
- 3MB file: 10MB/sec in 300 milliseconds
- 6MB file: 15MB/sec in 400 milliseconds
- 12MB file: 28MB/sec in 428 milliseconds
- 25MB file: 45MB/sec in 555 milliseconds

Super Computing ’95 configuration



Super Computing ’95 demo

Power Wall demo

- 4 parallel data streams driving 4 frame buffers
- 4 projectors form a single large screen
- Data streams held in sync

New for ’95

- BDS used instead of local disks
- Nobody could tell

Client side kernel BDS

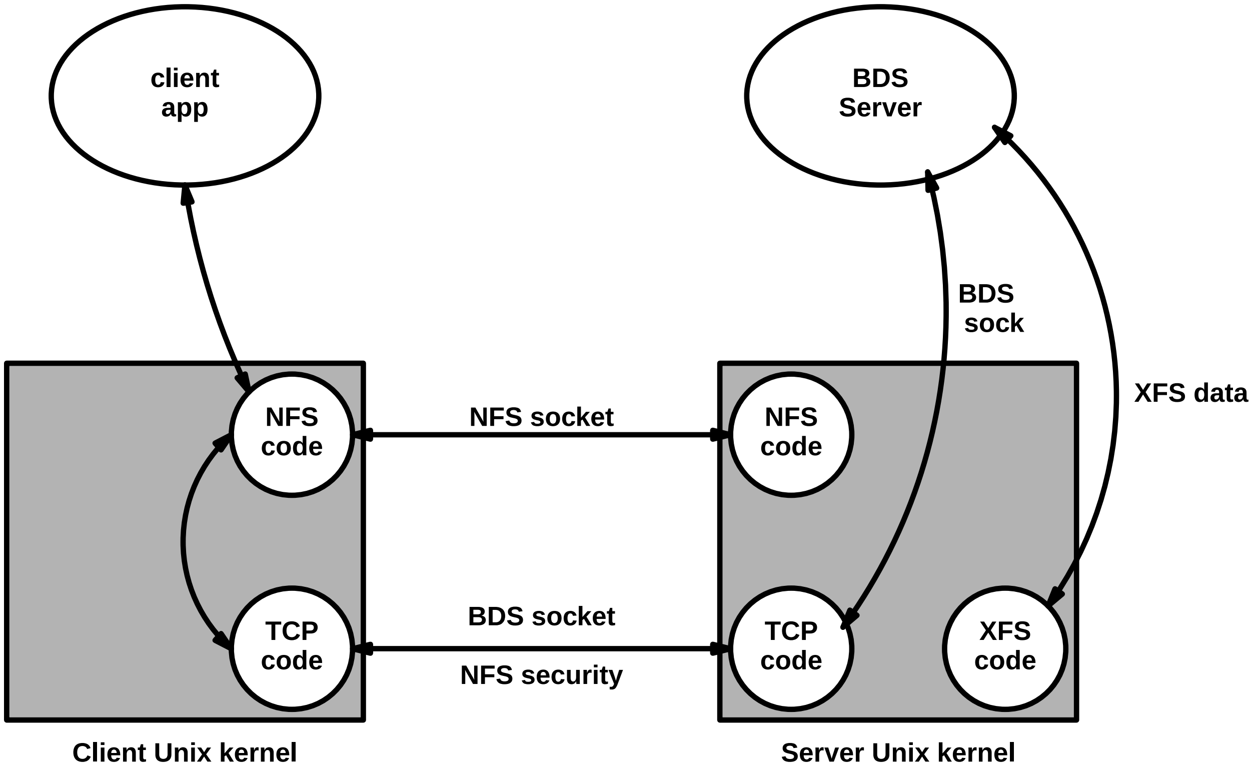
Small modification to client side NFS VFS

Changes in open/close/read/write

remapping of syscall -> bds_syscall done in kernel

Needed for security

Kernel BDS picture



How is BDS different from NFS v3?

One socket & process per open file

Very fast path to network

- read -> nfs_rdwr -> bds_read -> sosend

Simple packet format

- All one size
- All fields 64 bits wide

No server or client side caching

- All I/O is O_DIRECT

Why does BDS exist outside of NFS?

Time to market

- User level implementation is small, simple

Has to scale to 500MB/sec for Super Hippi

Has to handle failover cases

Has to be portable to other OS's now

- User level makes that much easier

Will SGI NFS get as fast as BDS?

In some cases NFS is faster due to caching

NFS v3 is catching up - 18MB/sec today

NFS will absorb most of the BDS technology

- NFS will eventually approximate BDS performance

Portability

User level client & server are available now to alpha testers

- Several SGI customers using it today
- Runs on Linux, IRIX, and Unicos

Kernel level client

- Code will come standard in future IRIX versions
- Reference port will be distributed for free in Linux
- RFC forthcoming

Availability

Should be available as a patch to 6.2 within 6 months

GPLed version available for free

- No support for this version

Normal business style license also available from SGI

- This is a supported product

Bulk Data Service - 60MB/sec “NFS”

Larry McVoy

lm@sgi.com

Silicon Graphics Engineering