CIFS Geeks in Exile
—or—
What We Did on our Holiday

Christopher R. Hertel
Storage Architect, CIFS Geek
Founder and CTO

SambaXP
May, 2011
Copyright 2011, ubiqx Consulting, Inc.
Introductions
Me

Your Friendly Neighborhood CIFS Geek

- CIFS Author
- jCIFS project co-founder
- Samba Team member since 97/98
- Incurable Idealist
- Etc., etc., ad nauseam

A ruminant mammal (Geekus geekus) with long legs, humped shoulders, and broadly palmated antlers.
Me

Your Friendly Neighborhood CIFS Geek

Tainted!

- Lead author of the Microsoft [MS-CIFS] and [MS-SMB] specifications.
- Access to MS Internal Information.
- Mustn't touch anyone else's CIFS implementation for one year.

That year is now OVER!

A ruminant mammal (Geekus geekus) with long legs, humped shoulders, and broadly palmated antlers.
What We Did on our Holiday
This is my report on what we did on our CIFS holiday.

- **Linux Clusters**
  Worked on GFS2 “virtual clusters”.

- **BITS Protocol**
  Created a BITS client toolkit.

- **MS BranchCache™**
  Studied Microsoft's BranchCache™ system.
Linux Clusters
Linux Clusters with GFS2

Why GFS2?

- In-kernel cluster file system
- Red Hat Cluster Suite
  - Supported in Fedora
- Local (to me)
  - Originally a U of MN project
  - I know these geeks
  - Easy to interact
- Good “community” choice

...but some Samba Team members have reported difficulties configuring and running GFS2-based clusters.
There are several other cluster FS options:

- Ceph – work in progress
- GlusterFS – cache consistency issues
- MooseFS – untested (to my knowledge)
- OCFS – similar to GFS

See Wikipedia for a longer list.
Linux Clusters with GFS2

Short Term Goal:

- Virtual “Cluster in a Box”
- Single server testing cluster
  - Fedora-14
  - KVM/QEMU

Status:

- The cbox cluster-in-a-box script works
- Virtual GFS2 clusters on KVM do not work due to I/O stress causing FS hang
  - A fix is in the works
Linux Clusters with GFS2

Long Term Goal:

- Samba/CTDB/GFS2 HowTo
  - Do-it-yourself virtual clusters
  - “Real” hardware clusters
- Production clusters running Samba and NFS

Status:

- 3 HowTos, need to be combined into one
- RedHat has built working Samba clusters
  - ...but has not yet performed extensive testing
- Focus is on cbox clusters
Why Clusters?

- **Failover**
  - SMB does not handle disconnect/reconnect very well
  - ...but SMB2 does

- **Active/Active load balancing**
  - SMB/CIFS/SMB2 is stateful
  - CTDB provides shared state

- **Scalability**

Are there other, better ways to approach these goals?
“BITS is Earth’s most widely used file transfer service, with more than 600 million unique users across the planet.”

– Vipul Bansal, Microsoft WMI Blog, Jan 2009.
BITS: Background “Intelligent” Transfer Service

“BITS is Earth’s most widely used file transfer service, with more than 600 million unique users across the planet.”

– Vipul Bansal, Microsoft WMI Blog, Jan 2009.

Note Well: nobody cares.
“BITS is Earth’s most widely used file transfer service, with more than 600 million unique users across the planet.”
– Vipul Bansal, Microsoft WMI Blog, Jan 2009.

What does that mean anyway?

It doesn't say “protocol”, it says “file transfer service”.

BITS is the Windows system service used by Windows Update to download patches.

Most users don't even know it's there.
BITS: Background “Intelligent” Transfer Service

BITS Features

- Built into Windows
- Restartable Transfers...but only linearly; does not “patch”.
- Both Download and Upload...and “Upload Reply”.
- Job priority levels
- Senses network traffic to manage impact
BITS: Background “Intelligent” Transfer Service

BITS Download Jobs

- The overwhelming majority of BITS jobs are probably Windows Update downloads.
- BITS Downloads use HTTP/HTTPS.
- Sort of like `uucp`? `wget + batch + nice + diffserv`?

The “special sauce” is the use of network traffic monitoring to limit BITS data transfer rates.
BITS: Background “Intelligent” Transfer Service

BITS Upload Jobs

- Much less common.
- Proprietary extensions to HTTP/HTTPS.
- Only between Windows BITS clients and Windows HTTP[S] servers.
BITS: Background “Intelligent” Transfer Service

BITS Upload Jobs

- Much less common.
- Proprietary extensions to HTTP/HTTPS.
- Only between Windows BITS clients and Windows HTTP[S] servers – Until now!
STiB means:

❓ Slow Transfer in Background?
❓ Silly Technology is Boring?
❓ Sipping Tea in Belgium?
❓ BITS spelled sdrawkcab with a small ‘i’?

STiB: It Is what It Is.

...

...a toolkit for testing BITS Uploads.

...example code for others to read / use.

A CGI script could be written to accept BITS Uploads.
BITS: Background “Intelligent” Transfer Service

BITS Upload Extensions:
- HTTP Extension Method: BITS_POST
- BITS Packet Types
  - Ping
  - Create-Session
  - Fragment
  - Cancel-Session
  - Close-Session
  - Ack

BITS Documentation:
- MSDN: BITS Upload Protocol
- WSPP: [MC-BUP]
BITS: Background “Intelligent” Transfer Service

BITS Upload Extensions:
- HTTP Extension Method: BITS_POST
- BITS Packet Types
  - Ping
  - Create-Session
  - Fragment
  - Cancel-Session
  - Close-Session
  - Ack

BITS Documentation:
- WSPP: [MC-BUP](https://github.com/microsoft/MC-BUP)
Do we care?

Yet Another Windows Protocol

BITS Upload is supported in IIS,

- and in Microsoft's “lightweight” HTTP server.

- It's convenient when working with Windows,

- but not nearly as powerful as, eg., rsync.

MS-BITS, however, also supports BranchCache™, which suggests some very useful testing scenarios.

- Add “Get” support to STiB,

- Include the modified header,

- See what happens!
Prequel
Pay Attention!

This is where it finally gets interesting.
What the heck is Prequel?
Prequel: A project to build an Open Source Implementation of Microsoft's BranchCache™.

So what the heck is BranchCache™?
Prequel

Prequel: A project to build an Open Source Implementation of Microsoft's BranchCache™.

BranchCache™ is a distributed content caching system
- supported in W2K8 servers,
- and Windows7 clients.

Cheap, effective WAN acceleration for SMB2, HTTP, and BITS.
BranchCache Architecture
A quick overview

Content Servers
Have content to share with multiple clients.

Clients
Request & receive content from content servers.

The Cache
A copy of the original content, cryptographically tagged and divided into segments and blocks.
Content Servers:

- Web Servers (HTTP, BITS)
- File Servers (SMB2)

The client must know to ask for content tags instead of actually content.

- If the tags are already calculated, they are returned by the BranchCache™-enabled server.
- Otherwise, the actual content is returned, and the server calculates the tags for next time.
IE 8 indicates support for BranchCache™ by listing “peerdist” as an acceptable encoding.
Distributed Mode

Content Server

WAN Link/VPN

Cache

Prequel

SambaXP 2011
Distributed Mode

Each client keeps a local cache.

A client requests tags from the server, then broadcasts to find the cached content.

If the content is not cached,

- The client requests the content from the content server,
- The client stores both content and tags in its own cache.

Reminiscent of the Browse Service.
Hosted Mode

Content Server

WAN Link/VPN

Cache

SambaXP 2011
Hosted Mode

1. A client request tags from the content server
2. The client then asks the cache server for the content
3. If the content is not cached, the client requests content from the content server
4. The client sends both content and tags to the cache server
5. Content can now be retrieved from the cache server using only tags
Prequel

Content Tags

Blocks
- Are a unit of download (from either content server or cache server)
- Are 64K (or less, for the last block in a file only)
  The block tag is an SHA hash of the block.

Segments
- Are a unit of discovery
- Are 32M == 512 blocks (or less, if the last block is short)
  The segment has is an SHA of the included block hashes.
Prequel Goals

I. Content Server
   - CGI script for Apache that generates correct tags.
   - Server-side code to provide a starting point for Samba implementation.

II. Cache Server
    - Implement a Hosted Cache server.

III. Peer Cache
    - Implement a stand-alone peer caching client.
Other Stuff
The End