Agenda

GGf8 results

- gridftp 1.0 doc is up for public comments
- during ggf8:
    - charter approved
        - goal: incremental improvement of gridftp protocol > gridftp v2.0
    - list of points of improvements is produced

use existing v1.0 gridftp and make improvements
want backwards compatibility with gridftp 1.0 and ietf gridftp protocol.

Overview of proposals: EOF

- EOF in stream mode
    - Help server to distinguish between end of successful stream mode uplaode and premature client termination
    - New command – eof with "commit" semantics
- url: http://www-isd.fnal.gov/gridftp-wg/eof.htm

overview of proposals: get/put

- get/put
    - combine retr/stor with pasv/port into single command so that the server can come up with right data socket address based on file patha nd perhaps other information
- http://www-isd.fnal.gov/gridftp-wg/getput/getput.htm

another proposal: extended Block mode

- modification of extended block mode
- free of "unidirectional" requirement:
    - data must flow in the same direction as data connection: sender always establishes data connection
- allows parallel transfers in both directions in NAT/firewall env
- http://www-i

lets go into some details
eof in stream mode
what happens now when client is terminated prematurely:
client server
STOR file
120

proposal is to modifiy STOR command a little –
client – issues STOR file
server send 220 opening data connection

client closes data channel
end of file on data socket = end of data transmission
EOF
226 data transfer complete

this is rather simple –

if have premature client termination –
no EOF so treat as error

GET/PUT

- RFC959 FTP in passive mode
- Client: PASV
- SERVER 220 OK
- Client: STOR /path/file
- Server: 120 opening data connection

Idea is to introduce new set of commands
Client – GET parameter=value
Server responds 1xx…
1xx ..PORT=(a.b.c.d.e.f)
2xx Data transfer complete

- parameters
    - connection mode (active/passive
    - transfer mode (S,B,E,X…)
    - max # of data channels
    - other transfer parameters

get/put examples –

- passive download
- client issues GET connect=pasv; path=/path/file, mode=s
- active upload
- PUT connect=actv; address=(_);path=/path/file

This simplifies protocol greatly –

eXtended block mode

- modification of extended block mode
- free of the unidirectional transfers requirement
- idea: replace EODC with reliable and explicit data channel establishment/termination
- benefits:
    - no data is sent before data channel opening is confirmed by both peers
    - data sender is assured no data is lost

there is a race condition that we are trying to grapple with here –that requires uni-directional data transfer –

Xmode how does it work?
Use same data channel socket to transfer control information – in th eopposite direction to the data flow

- "ready" – to confirm data channel establishment
- "bye" – to confirm receipt of EOD and data channel termination

passive sender in x mode –

- no need to send "READY" because receiver initiates data connection

appears that this scheme works well – avoids race condition –
benefits:
the passive side – that accepts connections – is capable of controlling bw on the fly –
this modification of the protocol allows passive side to close connection without breaking the protocol –
receiver has to continue accepting data until sender acknowledges by sending "EOD" and receiver acks with a "BYE".

One difficulty: EOF in X mode

- sender sends block with EOF bit set on one or more data channels
- EOF is sent only after all open channels are confirmed with "READY"
- After EOF is sent/received:
    - No new data channels will be initiated/accepted
    - Existing data channels will exist until closed with EOD/BYE sequence

This makes sure that acceptor will not wait forever for new connections. Only after EOF bit is received can acceptor close the socket –
This may be hard to explain – hope that document will explain better –

Summary of X-mode

- explicit and reliable data channel initiation/termination
- no race condition in passive sender/active receiver mode
- result: bidriectional parallel transfers in presence of NAT/firewall
- dynamic data channel opening/closing
    - initiator can open/close data channels
    - acceptor can close data channel

other proposals for gridftp v2.0

- structured directory listing, "stat" functionality
    - adopt ietf draft
- ipv6 support
    - adopt rfc2428

Did we get consensus on the group on these points???
How do we get agreement/get it blessed?

In general:

Thnk the only answer is to write an actual v2 protocol document –

Today we discuss proposals – after that – produce next document for next time –

Take gridftp as basis –
And then

Area director – right approach, getting input, write document.

Bill's concern is # of eyes looking at it –

May get people from IETF –
To review this –
There is a dormant ftp wg in ietf –
Good that we do sanity check between IETF and GGF –
This isn't the only group with this issue –

One concern with this –
We (being ANL) when started out – considered putting gridftp in front of IETF in front of ggf – this isn't what want on general tcp stack – as work around tcp congestion control –
This is designed for engineered semi-private networks –
If start interacting with IETF ftp working group – have had IETF networking guys in our group complaining about us –

Note – for parallel tcp streams – peaks over range of 7-8 streams –
Parallel transfers – most complicated –
But other issues – inherited from IETF ftp – these deserve to be fixed – perhaps provide as feedback to IETF –
Multiple tcp connections – not optimized for fat pipes –

With web 100 – have "what is the network doing while I was doing that" – on oc12, saw that multiple streams gave you nothing whatsoever – longer round trip time, then ran into other problems –

So seems like wg at ietf is probably the right place
So the ietf draft – appears to be advancing (for structured directory listing, stat functionality) –

Ietf – doesn't take input well, but does take collaboration well –try to get advice role, rather than interlocking standards tracks

Just don't say – that we've standardized thing –
This is a well scoped problem – very well scoped, very specific target –

It behooves you to think about this – prior to having it cast as a standard.

Issues without concrete proposals

- control of server feedback
  - performance markers, keep alive noise, etc
- data integrity verification
  - crc over whole file, each block, ..?
- packed transfers
  - tar them up and send as one file
- flexible striping control algorithms
- these issues need volunteers

last 2 – don't require protocol changes, rather are implementation issues

flexible striping control – long term research thing –
currently stripe width is fixed on gridftp –

would like to have a policy engine on the server – and plug in the policy – add one stripe for every 100 meg or whatever… and central point of control knowing about all requests that are queued –
so if have hundreds of kb files, and a 4 TB file, can intelligently tweak stripes.

What is next?

- get proposals for other items
- by ggf10 – produce final draft for gridftp 2.0 document
- create working prototypes
- present results

will want to get in position to have draft for next ggf –

a good sanity check would be working prototypes –

About gridforge –
Idea is to have all activities coordinated via gridforge –

Trying to maintain http://www-isd.fnal.gov/gridftp-wg, but moving to the gridforge pages.

An aside: Bill – should see an alpha 3.2 release in next 4-5 weeks – has significant upgrades
2 versions – wu based – globus url copy can move directories, etc
also complete reimplementation from scratch
no more feature enhancements to wuftp based server everything will be on new server.
Eg made control channel a separate module – so can replace with ogsi control channel –
Moving data internally, started beating it –
And built on xio io system –

Question: gridftp one of few things not written in java –
Xio handles ipv6 – other one – rfc 2428 – port and pasv commands are very ipv4 specific. Eport/epasv – allow you to deal with any protocol – and spell out how to deal with ipv6 –

This will be in the new server in the alpha –

We don't have an ipv6 network to test on though –

*** important point is to be active on the mailing list ***