



DMLite GridFTP frontend

Andrey Kiryanov
IT/SDC

13/12/2013



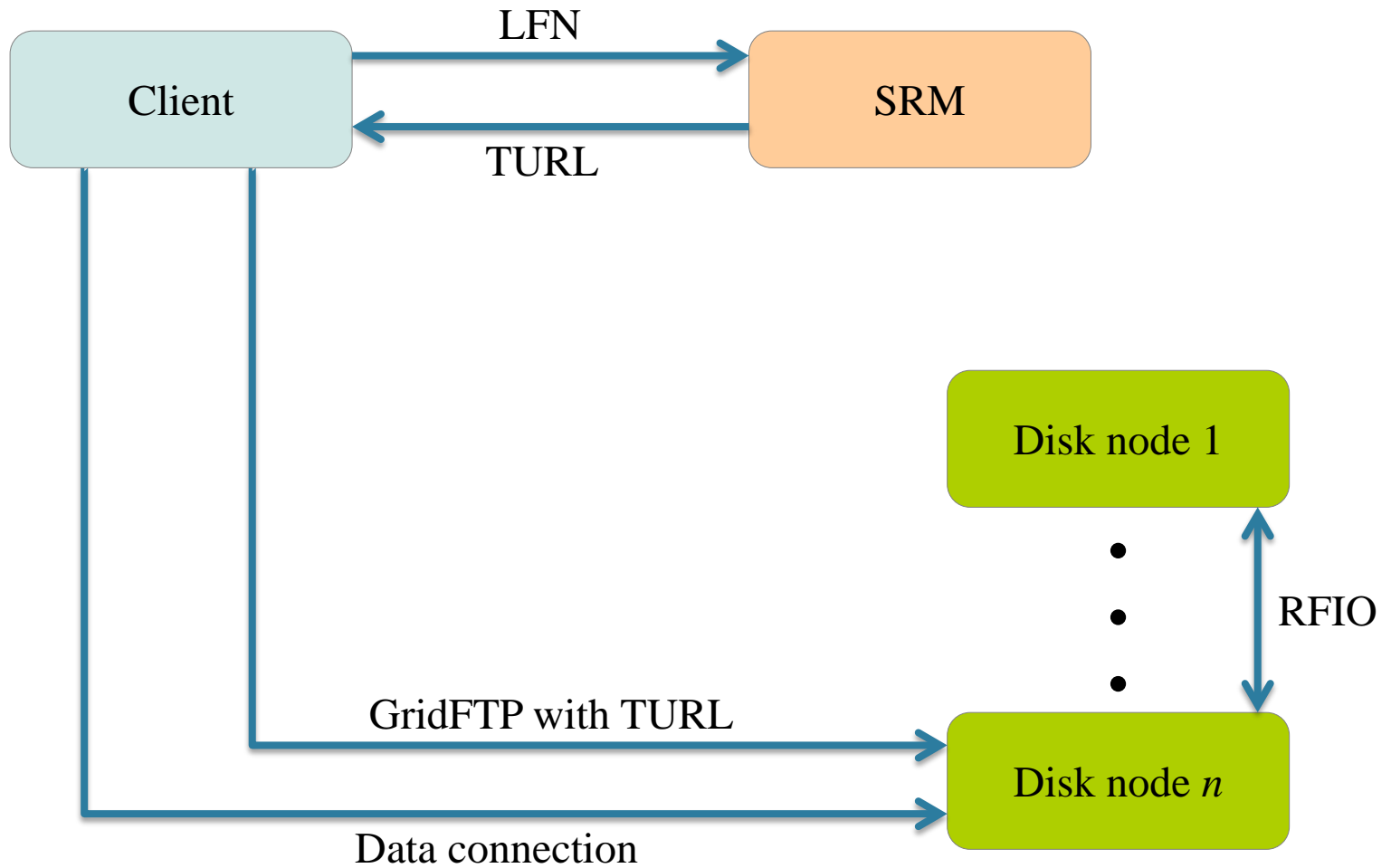
How GridFTP works

- Basically, it's a standard FTP with GSI on control channel and a bunch of extensions.
- A separate connection(s) is used for data transfer. Endpoint parameters are negotiated via control channel.
- Two standard FTP data connection modes: **Passive** (initiated by client) and **Active** (initiated by server).
- GridFTP v2 adds many extensions, in particular a third data connection mode: **Delayed Passive**.

How it was integrated with DPM

- Globus' GridFTP server supports pluggable backend modules: DSIs (Data Storage Interfaces).
 - No need to implement a full FTP server from scratch, just provide some basic file I/O callbacks specific to your backend.
- DPM interaction was implemented with a DSI.
- Clients contact disk nodes directly. Advance SRM call is necessary to get a TURL with a hostname of specific disk node.
- If a file is not available locally RFIO is used for transparent staging (slow).

GridFTP in DPM



Transition from DPM to DMLite

- GridFTP frontend is still based on Globus' server and a DSI plugin.
- Most parts of the plugin were completely rewritten using DMLite API (= expect new bugs).
- DSI functionality was extended with support of redirection.
 - It is understood that there's not much interest in SRM outside of HEP community.

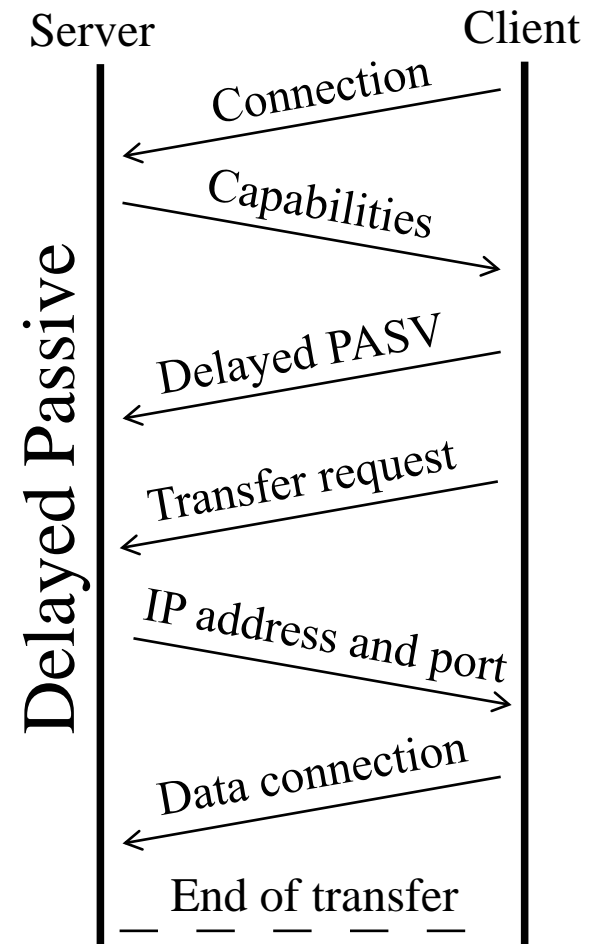
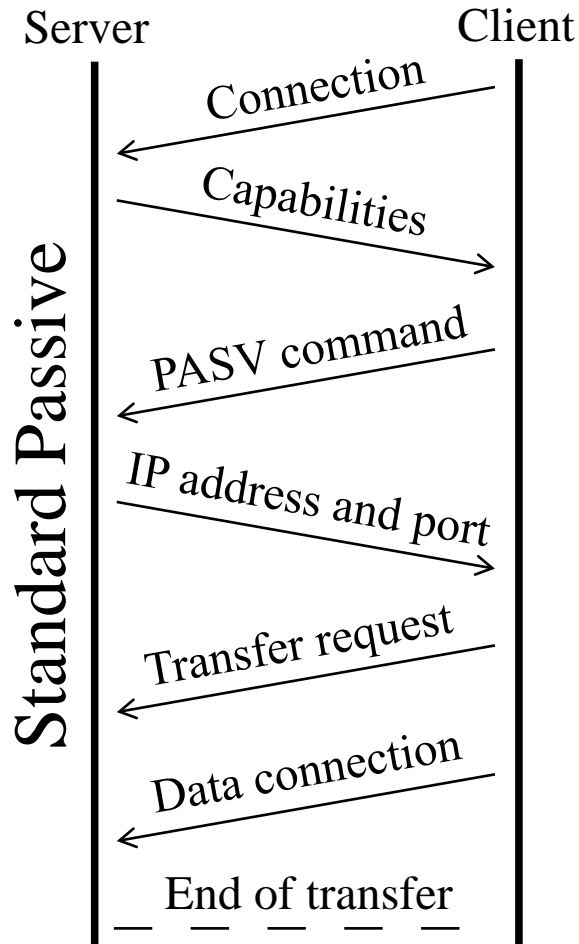
Default deployment scenario

- Almost no visible changes.
- DPM DSI is replaced with DMLite DSI.
- Clients still need to contact SRM prior to disk nodes.
- Needs proper configuration of DMLite.
 - Namespace operations can go directly through DMLite MySQL plugin (faster, no need to contact DPNS).

GridFTP redirection

- Data transfer channel is negotiated and established separately, not necessarily with the same server (third party transfers rely on this).
- There's a shortcoming in standard FTP finite-state machine: in Passive mode server address needs to be supplied to a client in advance, before the transfer command when file name is not yet known.
- Globus' GridFTP provides a third connection mode that works around this problem: **Delayed Passive**. It delays server address response until a file name is known to the server.

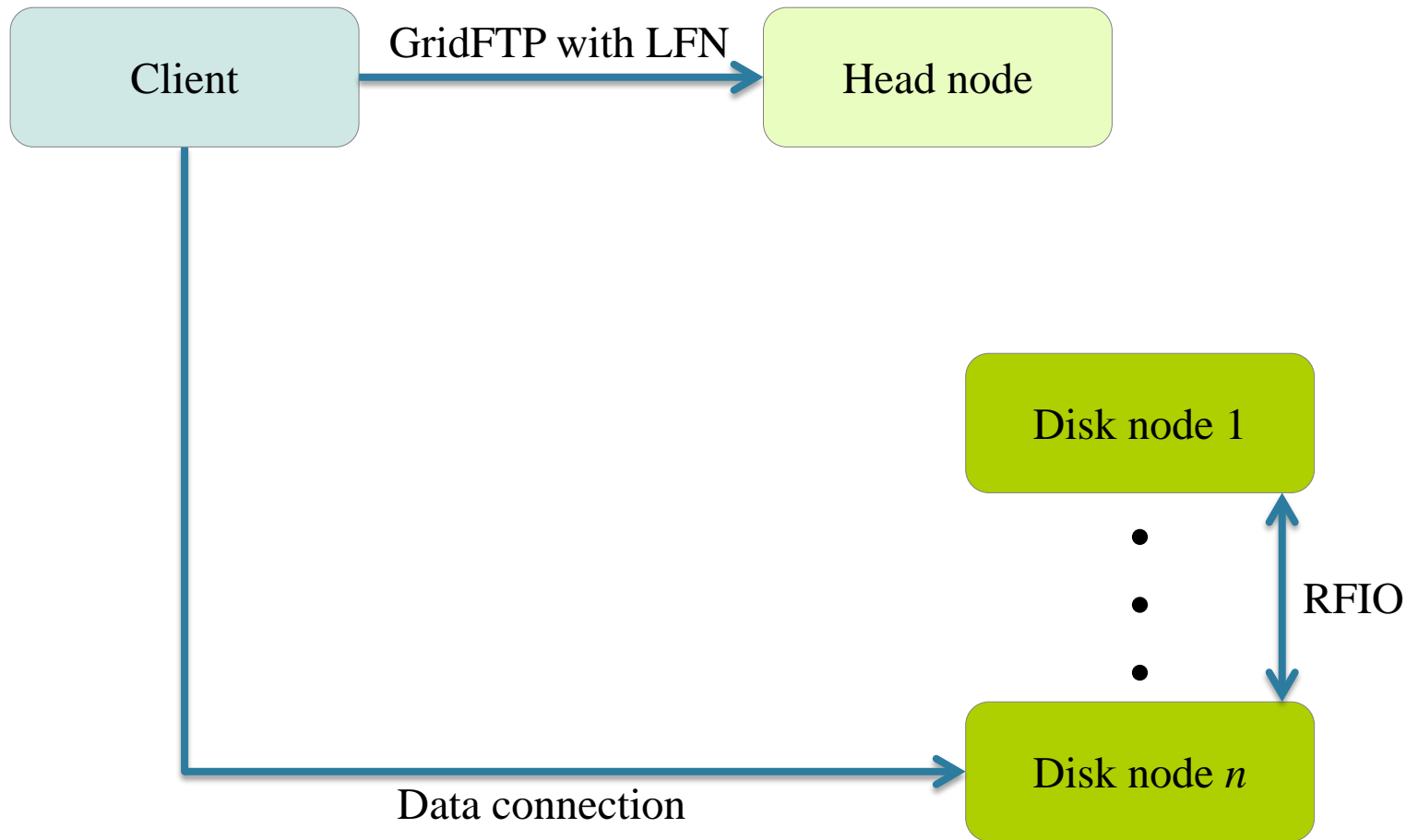
Standard Passive vs. Delayed Passive



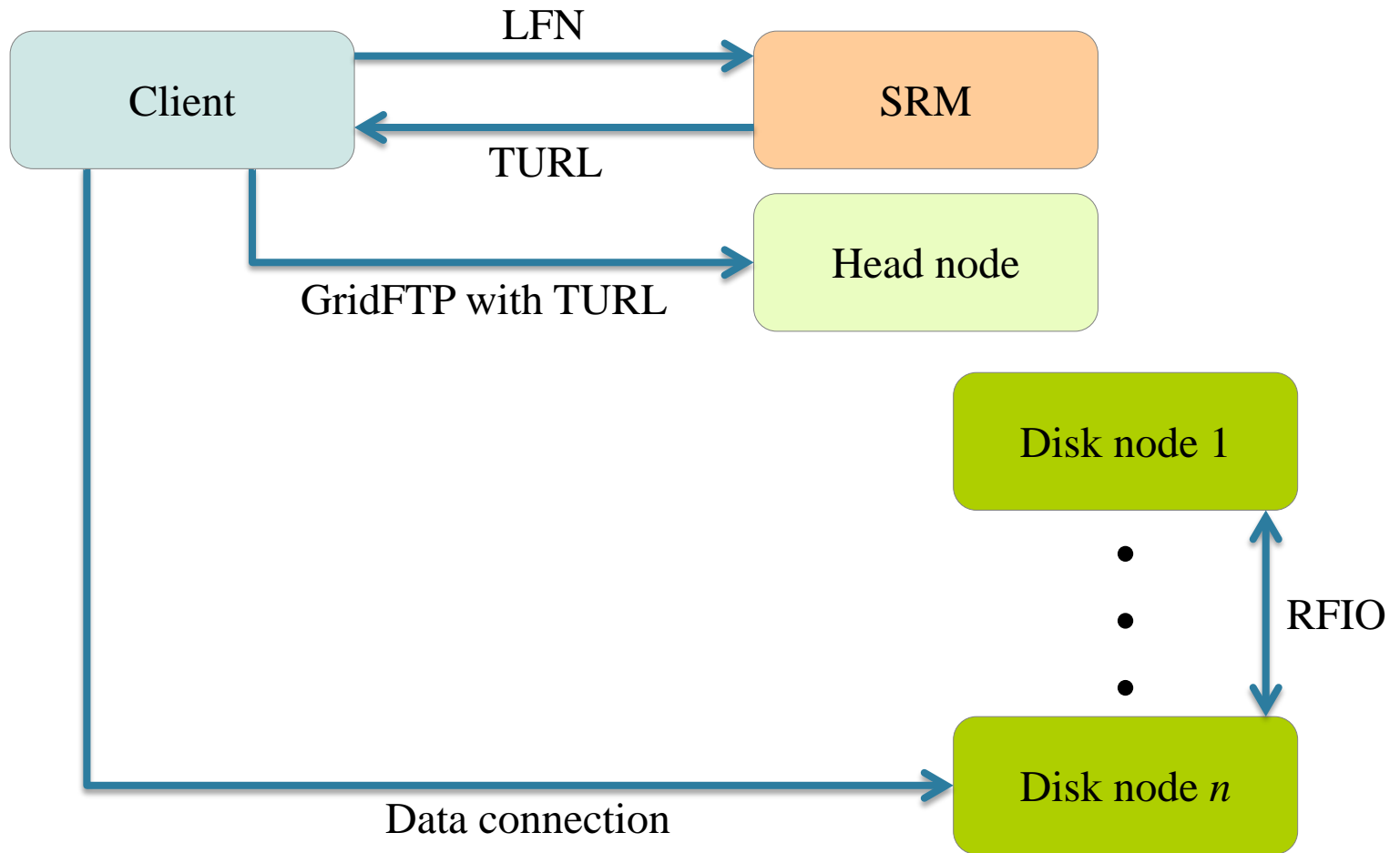
Future deployment scenario

- GridFTP is configured differently on a head node and on a disk servers.
- Clients always contact the head node. Direct control connections to disk nodes are no longer supported.
- Clients do not need to contact SRM, but if they do, a proper TURL with a head node is given.
 - “FTPHEAD” option in */etc/shift.conf* should point to the head node.
- Clients need to support Delayed Passive mode for optimal performance.
 - Older clients connecting to the head node without SRM will end up on a random disk node. A file transfer will be done with transparent RFIO staging to the right disk node (slow).

Redirecting GridFTP in DMLite



Redirecting GridFTP with SRM



Interoperability

- Current versions of GFAL2 and FTS3 already support **Delayed Passive** mode.
- DPM already has all the necessary bits to support GridFTP redirection with SRM.
- Clients that rely on Globus libraries can easily be modified to support **Delayed Passive** mode:
 - Use the `globus_ftp_client_operationattr_set_delayed_pasv()`, Luke!
- Third party clients (e.g. uberftp) that do not support **Delayed Passive** mode will work with non-optimal performance involving RFIO fallback.
 - One can switch it off completely in DMLite configuration.

Thank you!

