Mitglied der Helmholtz-Gemeinschaft

UNICORE

JÜLICH
FORSCHUNGSZENTRUM

File transfer in UNICORE
State of the art

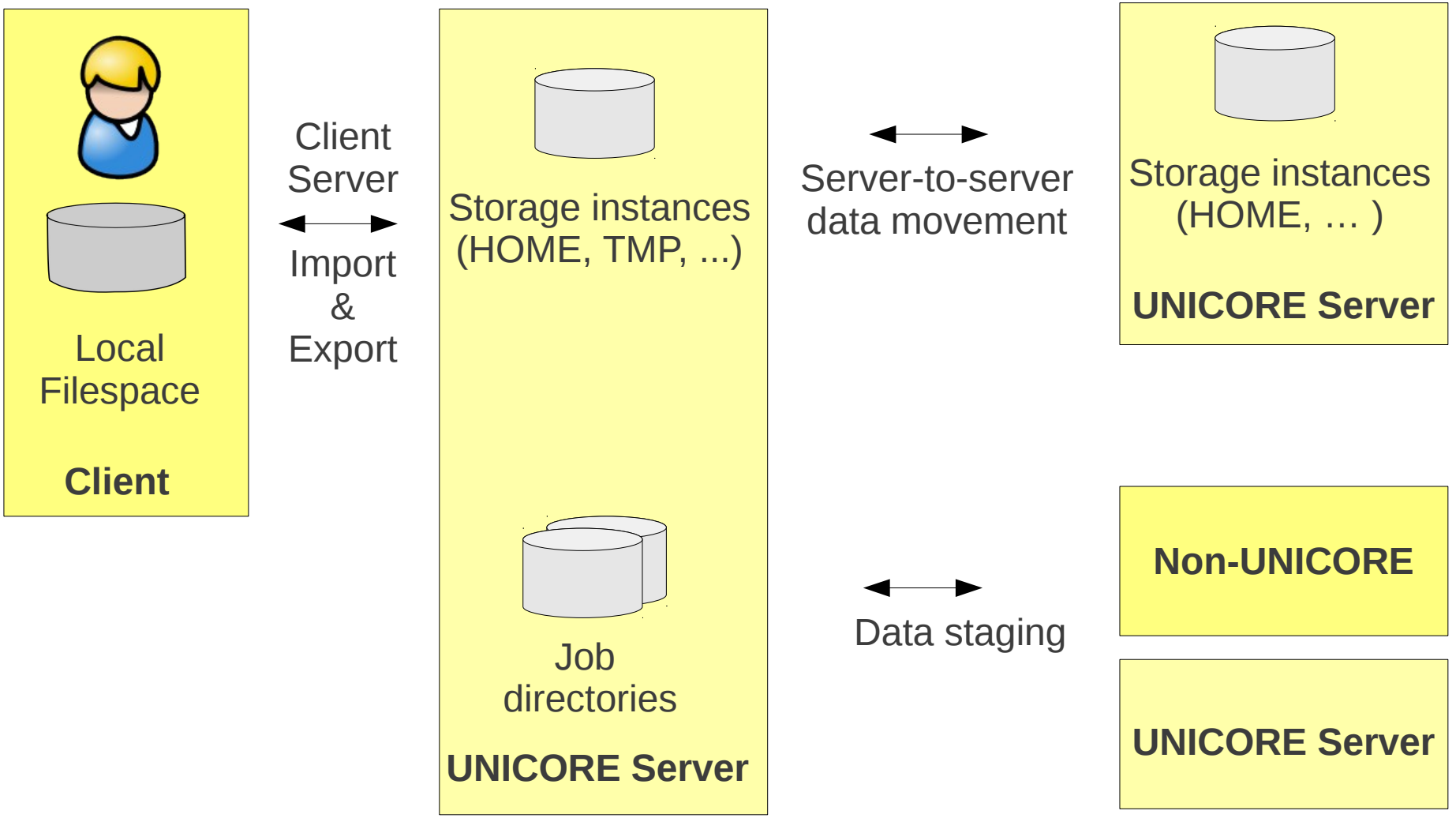
Bernd Schuller, Björn Hagemeier, Michael Rambadt
Federated Systems and Data division
Jülich Supercomputer Centre
Forschungszentrum Jülich GmbH

31 May 2012, UNICORE Summit 2012, Dresden

Outline

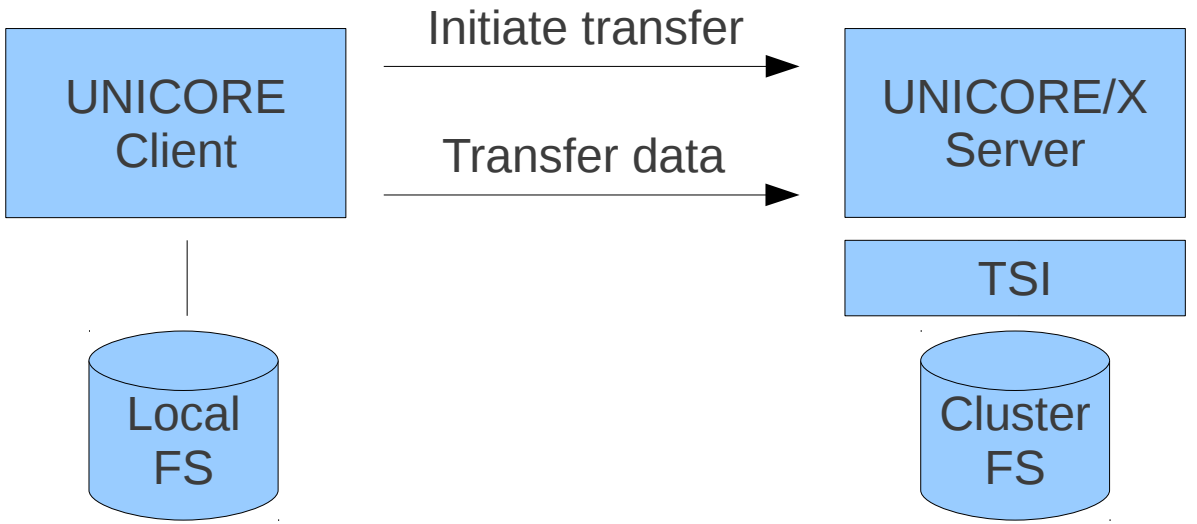
- Filetransfer in UNICORE: a (short) review
- UFTP
  - Performance tests in a 100 Gbit/s network
  - Tests in PRACE
- GridFTP: recent developments
- Recent work on reliable filetransfer

Three types of data movement in UNICORE



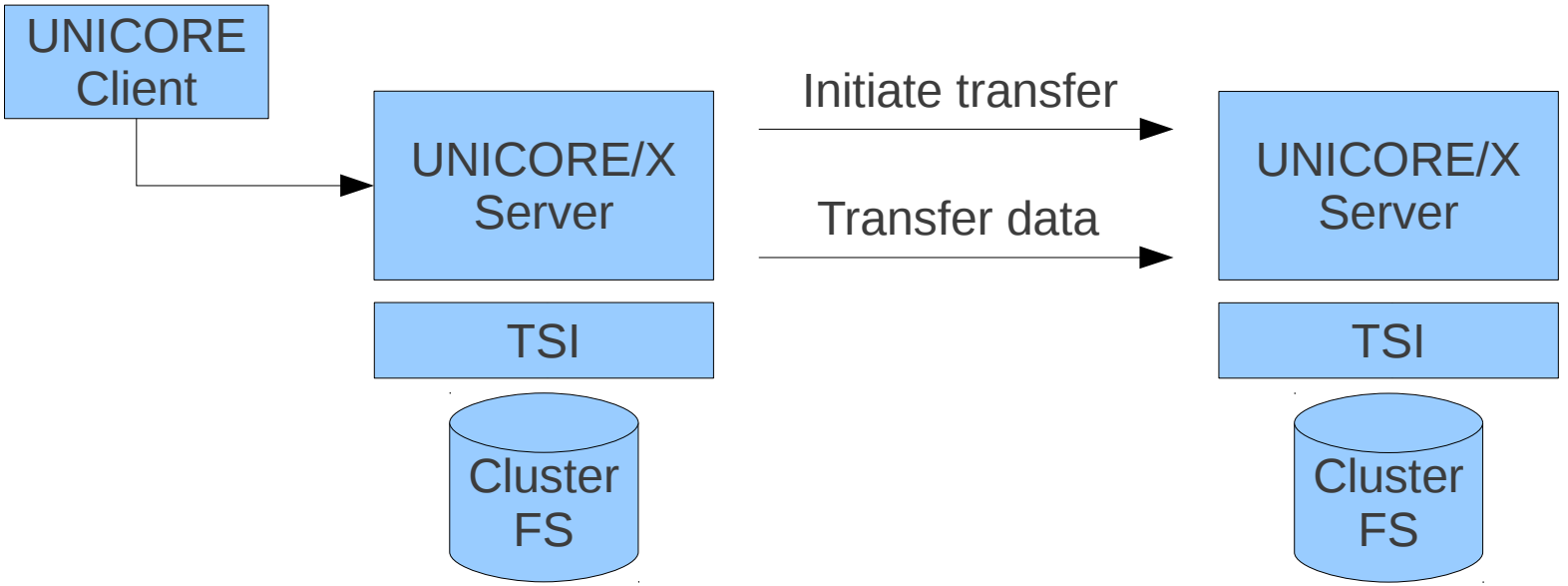
Client-Server transfers

- Web service layer integration
- Client-initiated by invoking SMS ImportFile / ExportFile
- Client-side protocol handler is usually Java code, integrated into URC/UCC
- Server-side protocol handler
  - Can run in UNICORE/X or directly on the resource (e.g.UFTP)



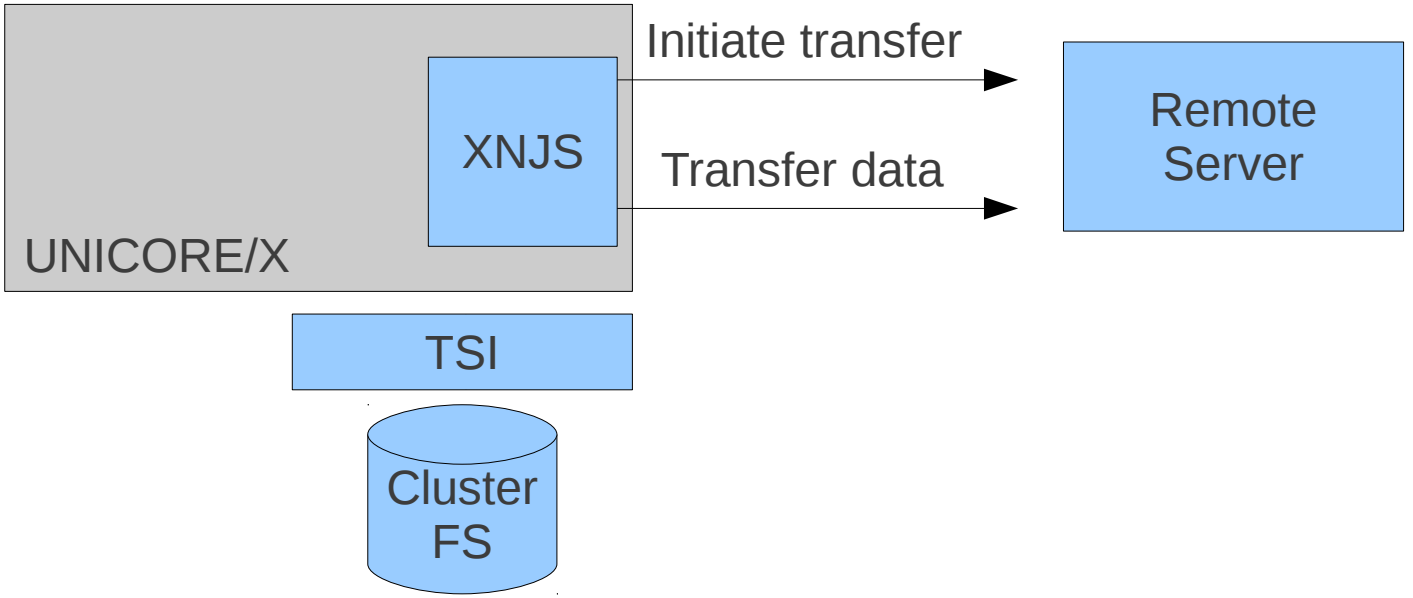
Server-server transfers

- Web service layer
- UNICORE server acts as client
- Client initiated by invoking SMS SendFile / ReceiveFile
  - Transfer start can be scheduled
- Same code, features and set of protocols as client-server



Data staging

- Part of job processing (XNJS level integration)
  - JSDL DataStaging element: file name and source/target URL
- Peer need not be a UNICORE server
- UNICORE protocols (BFT, …) plus additional ones (scp, ftp, xtreemfs, mailto, file, gsiftp …)



Protocol overview

	BFT (https)	ByteIO	UFTP	GridFTP	http(s), ftp	scp
WS layer integration	Yes	Yes	Yes	No (1)	No	No
Data staging	Yes	Yes	Yes	Yes	Yes	Yes
Direct FS to FS	No	No	Yes	Yes	No (2)	Yes
Efficiency	Fair	Bad (3)	Very good	Very good	Good	Good
Security	Full U6	Full U6	Full U6	Proxy in SOAP header	User/Pass In JSDL	User/Pass In JSDL

Notes:
1) Work on better GridFTP support is in progress
2) http/ftp can be configured to use wget / curl
3) ByteIO performance would benefit of MTOM support

UFTP

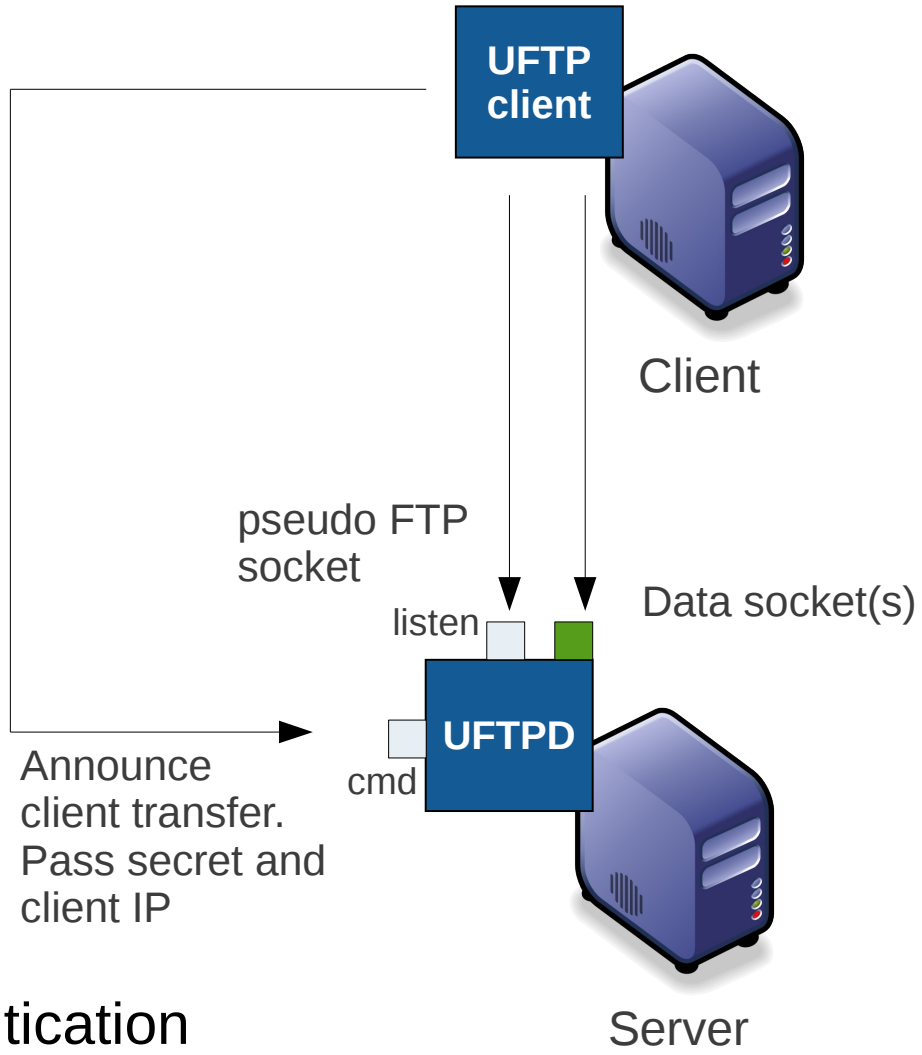
What is UFTP?

JÜLICH FORSCHUNGSZENTRUM

- File transfer library based on „passive FTP“ ideas
- Single „FTP“ port. Dynamic opening of firewall ports for data connections
- Multiple TCP streams per transfer. Optional data encryption
- Pure Java (plus a bit of native code for Unix user/group ID switching)

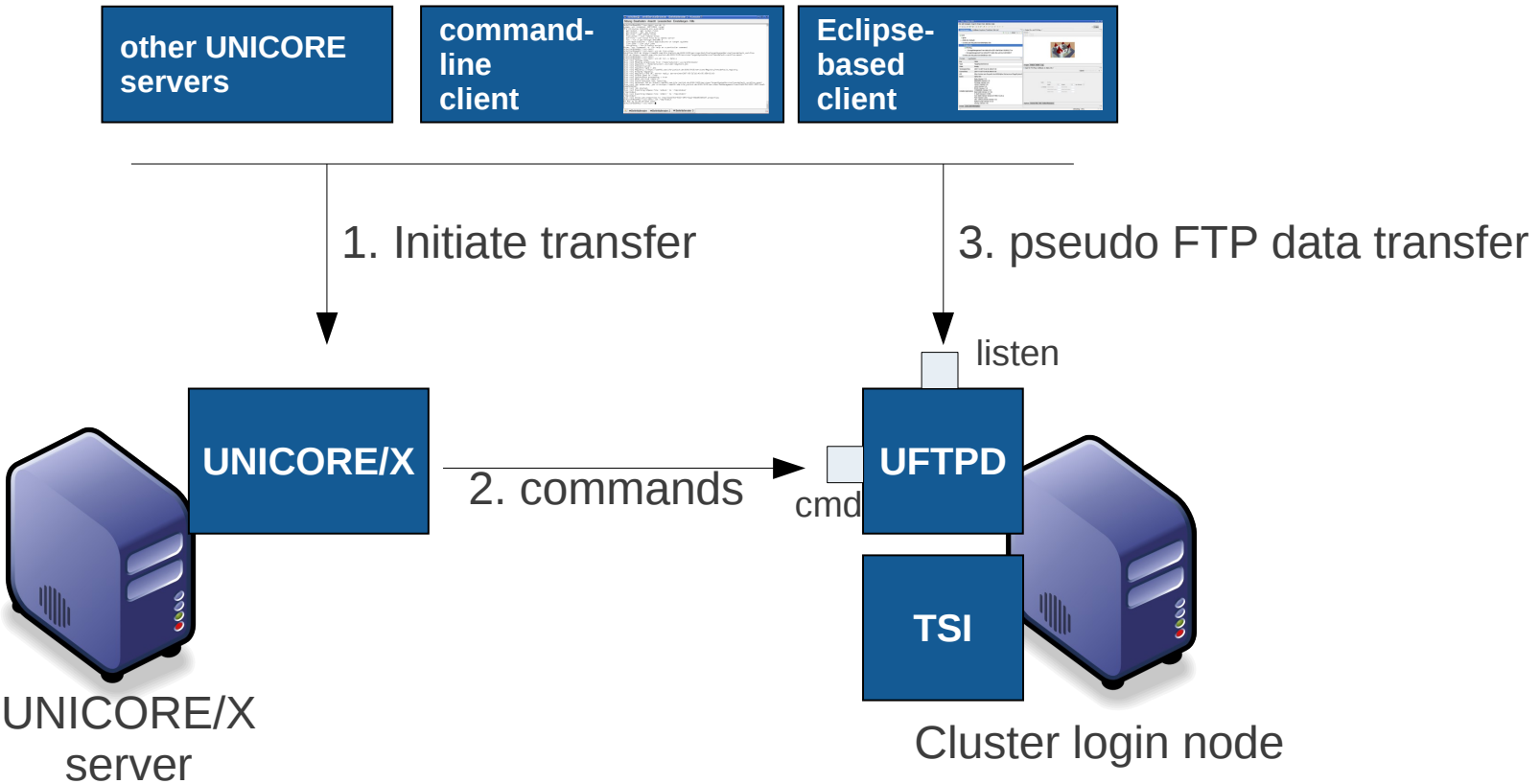
Basic UFTP

- UFPTD server
  - Pseudo-FTP port (open in firewall)
  - Local command port (can be SSL protected)
- UFTP client
  - Connects to pseudo-FTP port
  - Open required data connections
  - Send/receive data



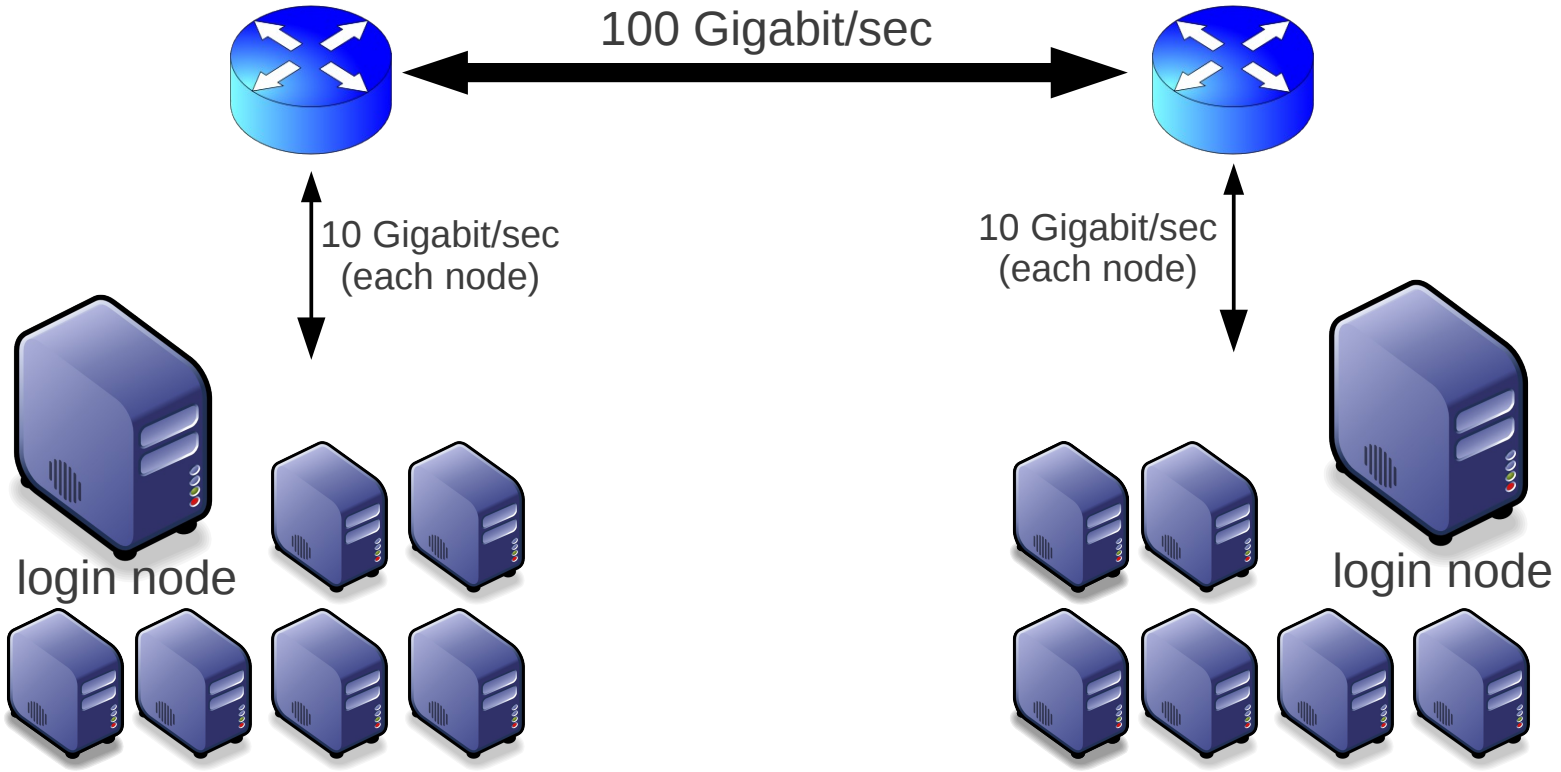
- **Secure channel** required for authentication and authorisation

UFTP integration



Reference:
B. Schuller, T. Pohlmann: UFTP - high performance data transfer for UNICORE, UNICORE Summit 2011, Proceedings, 7-8 July 2011, Torun, Poland, ed.: / M. Romberg, P. Bala, R. Müller-Pfefferkorn, D. Mallmann, Forschungszentrum Jülich, 2011, IAS Series Vol. 9, ISBN 978-3-89336-750-4, pp. 135-142

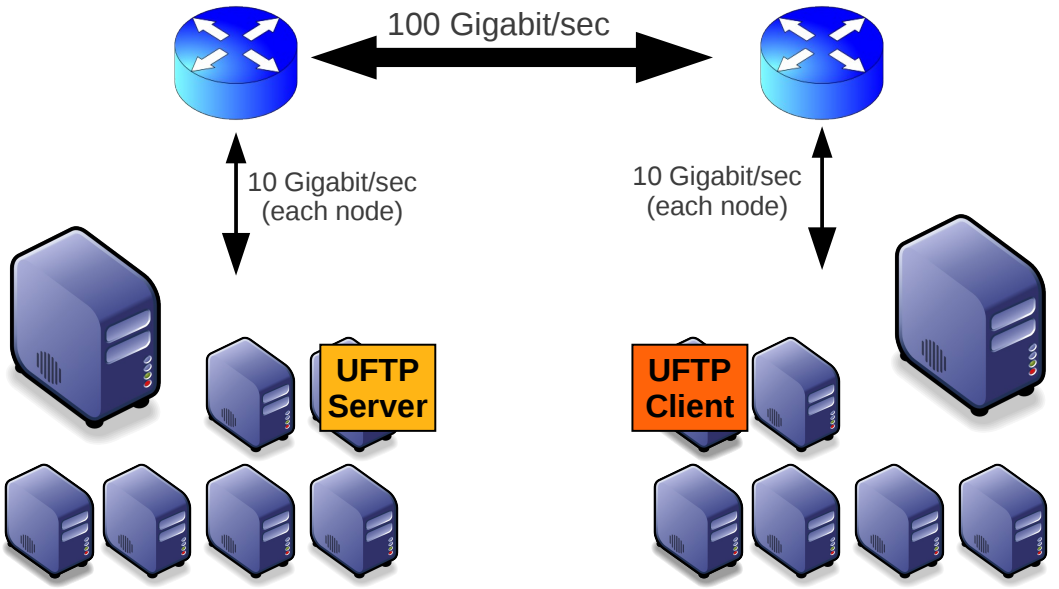
100 Gigabit/sec testbed
TU Dresden – TU Freiberg

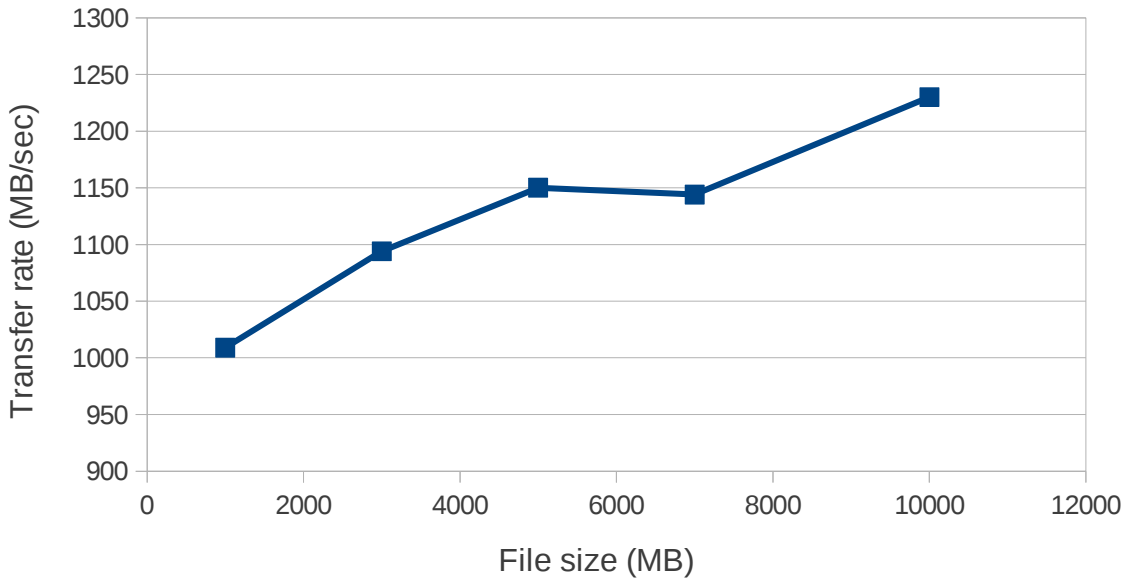


- Up to 10 GBit/sec from/to each cluster node
- Up to 100 GBit/sec aggregated transfer rate (using multiple nodes)

Single client, single server

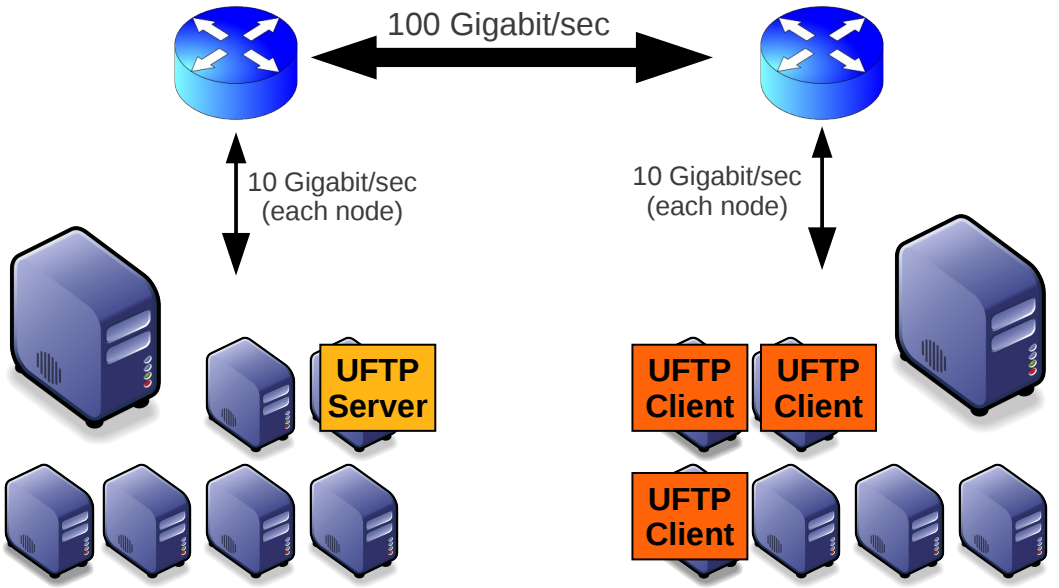
- Up to 1.2GB/sec
- 98% of line rate

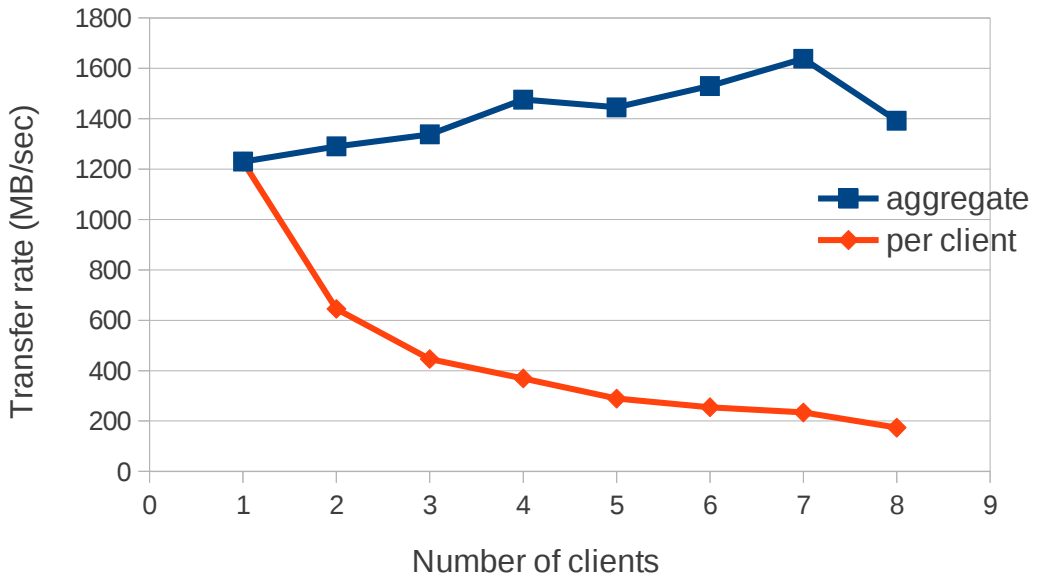




Multiple clients, single server

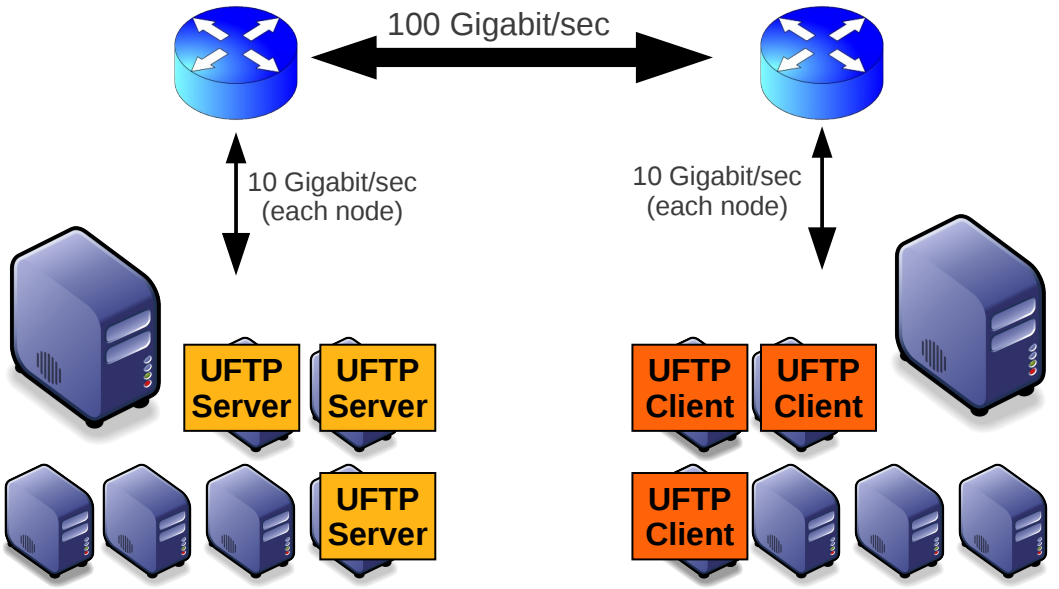
- Up to 8 clients
- (roughly!) parallel transfers (50GB each)

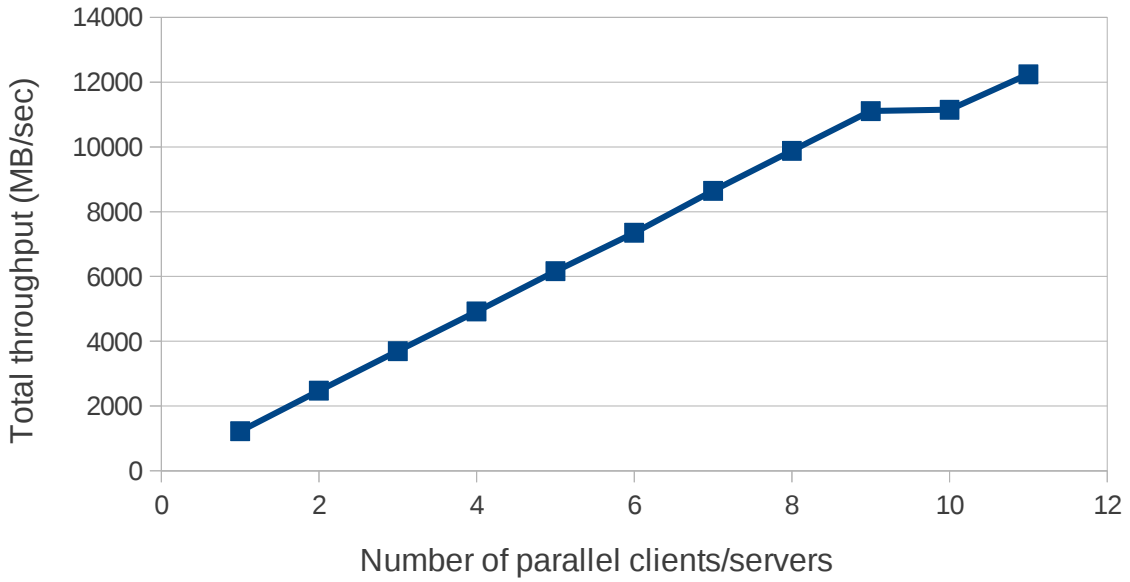




Multiple client/server pairs

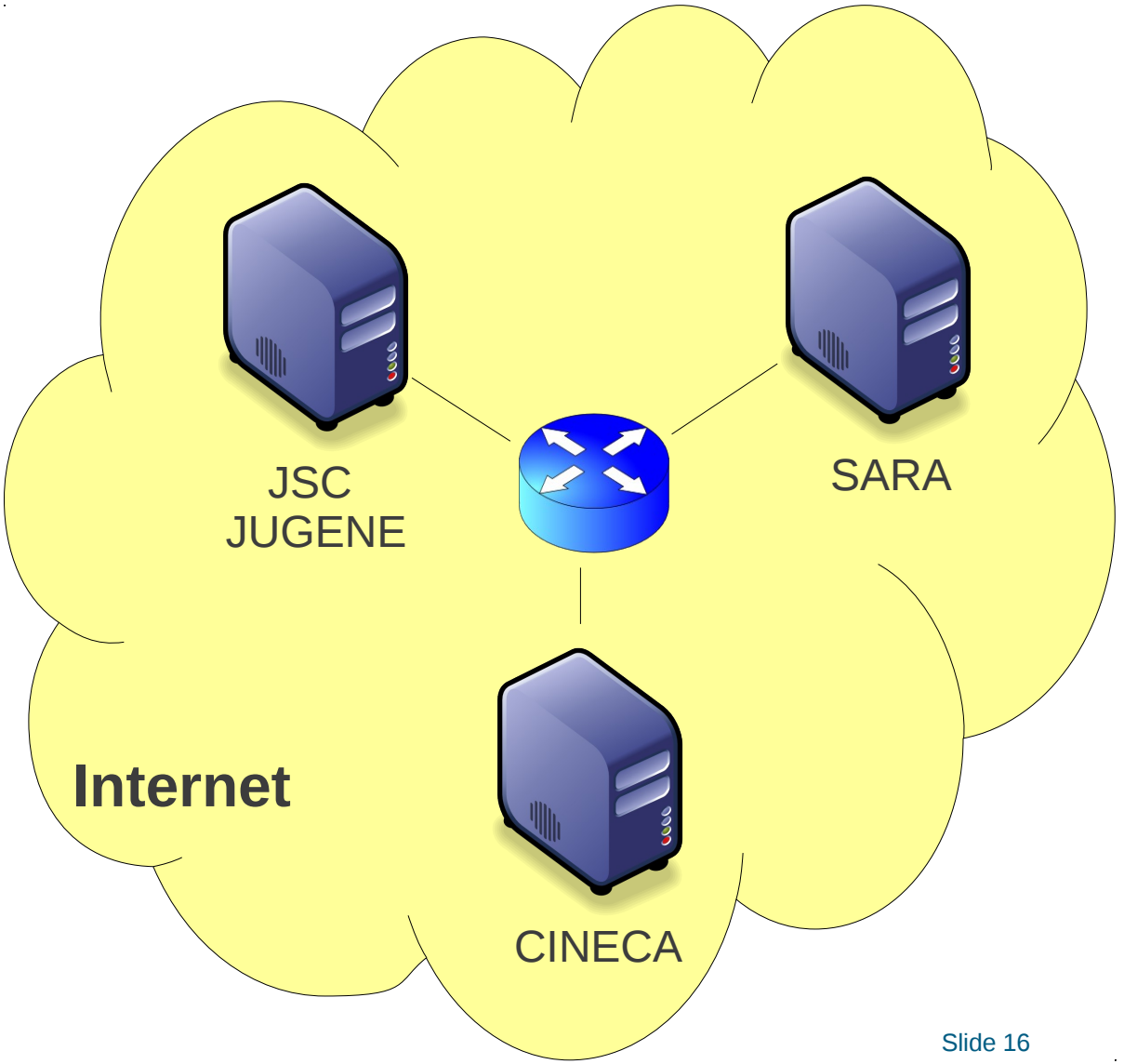
- Up to 11 (roughly!) parallel transfers (50GB each)
- 12 GB/sec
- 98% of line rate





UFTP tests in PRACE

- UFTP is being evaluated
- Small testbed: JSC, SARA and CINECA sites
- Resources available via two network interfaces (internet and internal high-speed)
- For tests, UFTP server is deployed at JSC. Client (UCC) at SARA and CINECA



UFTP tests in PRACE – results

- Results depend on filesystems and the network interface that is used!

- JSC – CINECA (1 Gbit/s link, 1GB file size)
  - JSC tmp → CINECA tmp : 70 MB/s
  - JUGENE tmp → CINECA dev/null: 128 MB/s

- JSC – SARA (10 Gbit/s link, 10 GB file size)
  - SARA Home → JSC Home: 72 MB/s
  - SARA Scratch → JSC tmp: 126 MB/s

- For comparison (internal 10 Gbit/s link):
  JSC JUROPA Home → JUGENE Home : 258 MB/s

GridFTP

GridFTP developments

- Status
  - Data staging only. Callout to existing `globus-url-copy` tool
  - Proxy cert placed in SOAP header by the client.

- Recent development work (motivated by the EMI project)
  - Use JGlobus libs, updated to not conflict with UNICORE libs
  - In principle, this allows to integrate gridftp clients into UNICORE clients
  - Use existing proxy cert generated via `voms-proxy-init` (`myproxy` should be the same)
  - Prototype module for the HiLA library, used with the EMI services prototypes

GridFTP outlook

- Investigate full WS level integration (client, SMS) of GridFTP
- The proxy cert problem ...
- See also SRM/LFC work (→ C. Loeschen)
- User/infrastructure requirements increase developer motivation :-)

Current and future work on UNICORE file transfer features

Current file transfer pain points

- Reliability
  - For server-server transfers
    - Status: no retries, no proper restart of failed transfers
    - Goal: server-server transfer is 100% reliable
  - For client-server transfers
    - Status: failed transfers need to be re-done
    - Goal: continue aborted transfer
- Scalability
  - Current: one filetransfer „process“ per file
  - Goal: more efficient transfer of many files
  - Ambitious goal: full „rsync“ (→ efficient replication of full storages)

File transfer: work in progress

JÜLICH FORSCHUNGSZENTRUM

- Work on supporting partial downloads (byte ranges)
  - Required for efficient, reliable, scalable transfer
  - Allows to download same file from multiple sources
  - Failure of of individual downloads is „cheaper“
- UFTP protocol extensions
  - Sessions
    - Get/put multiple files in a single UFTP session (as in ftp)
    - Byte ranges
    - Support `ls`, `cd`, `mkdir` and all that. Rsync?
  - Will be available in UFTP 2.0

File transfer: work in progress - II

- Reliable and scalable server-server transfer using a well-known method
  - Target server downloads file in chunks
  - Keep download state on the file system to allow restarts without loss of data
  - Combine chunks as the last step
  - In 6.5.0 already „almost“ possible with BFT („UNICORE https“)

Questions?

- Thanks
  - Andy Georgi, Stefan Höhlig (TUD)
    for generously providing access to the 100Gbit/s testbed
  - Andreas Landhäußer, Gert Ohme (T-Systems SfR)
    for initiating the UFTP / 100Gbit testing