Capability testing of data transfer tools on a high latency 100 Gbit/s light path

Kees de Jong

University of Amsterdam
MSc System and Network Engineering
Research Project 1 Presentation
Supervisor: dhr. dr. ing. Leon Gommans (KLM)

February 6, 2018

Background

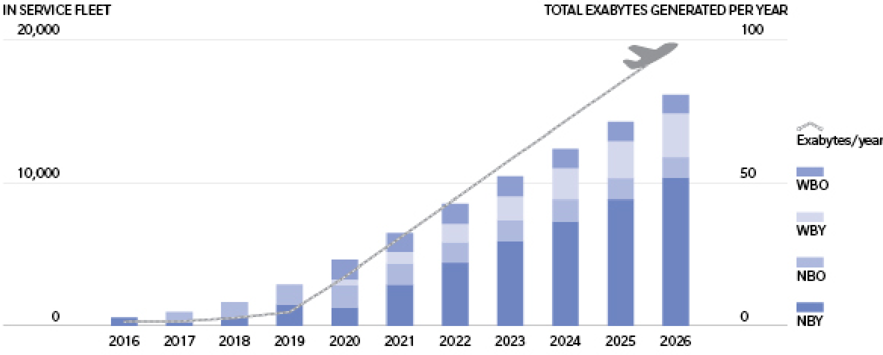
- Airplanes not only transport people and cargo, but also data
  - Sensor readings
  - Engine data
  - And more...
  - Accumulating to several TB's of data per flight

Background (continued)

- Critical to transport data fast, to shorten and improve maintenance
- KLM challenges for the future

DATA GENERATED FROM PROJECTED GLOBAL FLEET

IN 2026, THE GLOBAL FLEET WILL GENERATE 98 EXABYTES OF DATA
(THAT'S 98 MILLION TERABYTES OR 98 BILLION GIGABYTES)



Source: Oliver Wyman Fleet & MRO Forecast, www.planestats.com/betterinsight
www.oliverwyman.com

Background (continued)

- Internet is not private nor fast enough
- 100 Gbit/s path from Amsterdam to Chicago (95 ms RTT)
- Compare capabilities of high performance GridFTP data transfer tools

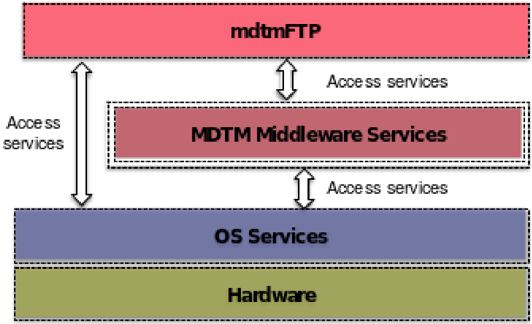
Globus GridFTP

- Globus GridFTP
  - Concurrency (concurrent FTP connections for multiple transfers)
  - Pipelining (latency transparency)
  - Parallelism (divide blocks over multiple transport streams)
  - Third party data transfer

mdtmFTP features

- Build on top of the Globus GridFTP module
- Multicore-Aware Data Transfer Middleware
  - Application level scheduler (mostly independent from OS scheduling)

mdtmFTP features (continued)



mdtmFTP features (continued)

- NUMA: Dedicated NIC and I/O threads + buffers pinned
- Large virtual file mechanism (LOSF)
- Direct I/O (disk –> memory)
- Splice (storage –> NIC)
- *Pipelining*
- *Parallelism*
- *Third party data transfer*

Related work

- mdtmFTP and Globus GridFTP evaluated by L. Zhang et al.
  - Simulated *shared* network loop between Chicago and Oakland
  - RTT 95 ms, 100 Gbit/s
  - Concluded that mdtmFTP was on average 20% to 30% faster
- Globus GridFTP over TCP compared to UDT by John Bresnahan et al.
  - Application level improvement for Globus GridFTP: UDT
  - Tested network with highest latency was 204 ms RTT (ANL to Auckland)
  - "Best of their knowledge" 1 Gbit/s
  - In most cases UDT outperformed TCP (Reno), often by a factor of 3 or 4 in throughput

UDT feature excerpt

- Application level protocol build on top of UDP
- Globus XIO module (substitution of transport protocols)
- Adapts faster to available bandwidth and more features
- Because this is done in the application layer, it consumes more RAM

Research question

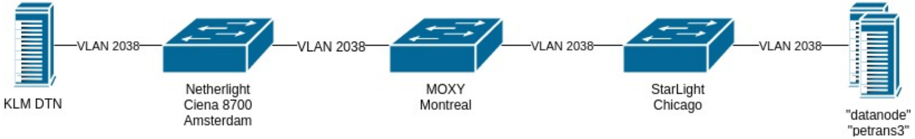
Main research question: "What are the capabilities of mdtmFTP compared to Globus GridFTP on a 100 Gbit/s light path between Amsterdam and Chicago?"

1. Which features and/or design allows optimum throughput?
2. How do these data transfer tools behave with various sets of different file sizes and quantity?
3. Is the conclusion still valid that Globus GridFTP over UDT outperforms TCP on a high latency network? And is it enough to beat mdtmFTP?

Methodology

- Map bottlenecks in the test setup
- Pinpoint the limitations of the data transfer tools
  - Single and concurrent transfer of a large contiguous file
  - Handling LOSF
  - Transfer of KLM flight data
- Measure performance/behavior of throughput
  - Throughput on network level
  - TTC on application level
- Script experiments
  - Drop buffers/caches
  - Repeat tests multiple times (10x)

Network overview



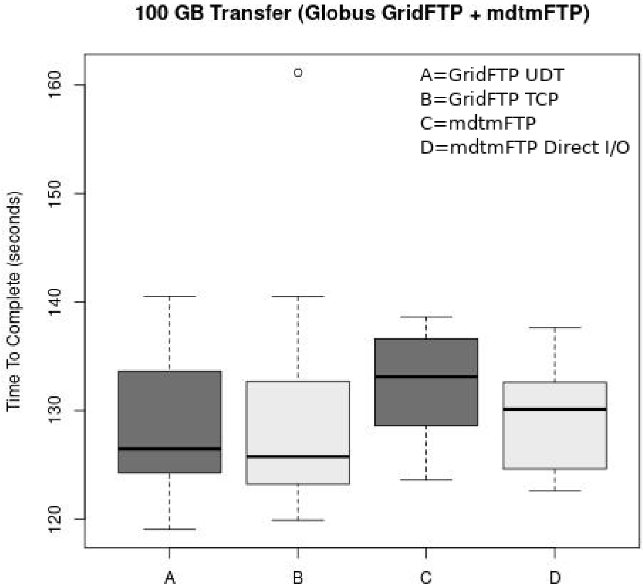
Disk performance

	KLM DTN	**Chicago DTN 1**	**Chicago DTN 2**
Max read speed	∼1500 MB/s	∼1000 MB/s	∼1200 MB/s
Max write speed	∼800 MB/s	∼700 MB/s	∼700 MB/s
# disks	2	6	6

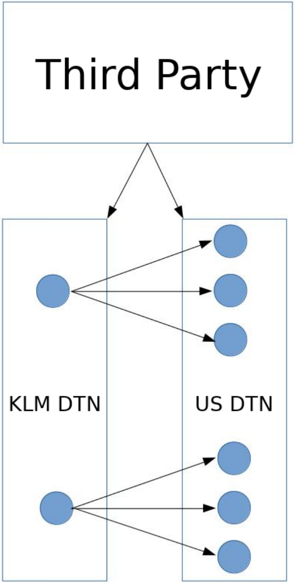
Results: 100GB (node-to-node + 3rd party)

- All experiments were done with 4 parallel data streams
  - Anything above 16 parallel streams is regarded wasteful
  - Initial experimentation verified this
- Globus GridFTP
  - Parallelism
- mdtmFTP
  - Parallelism
  - Direct I/O (disk –> memory)
  - Splice (storage –> NIC)

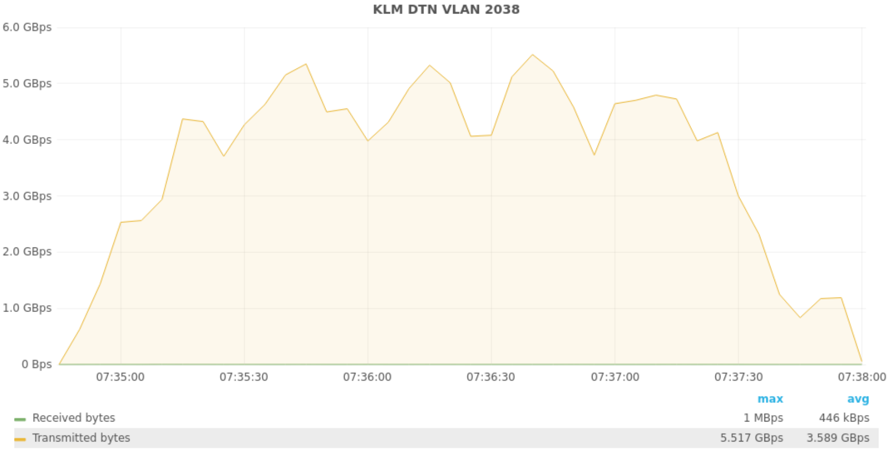
Results: 100GB (node-to-node)



Results: 3rd party 100GB (6*100 GB)

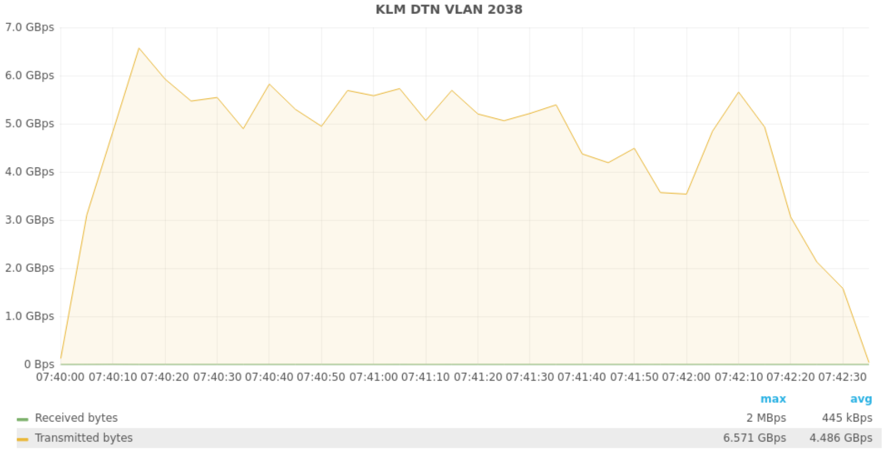


Results: 3rd party, GridFTP TCP, TTC=195 sec.

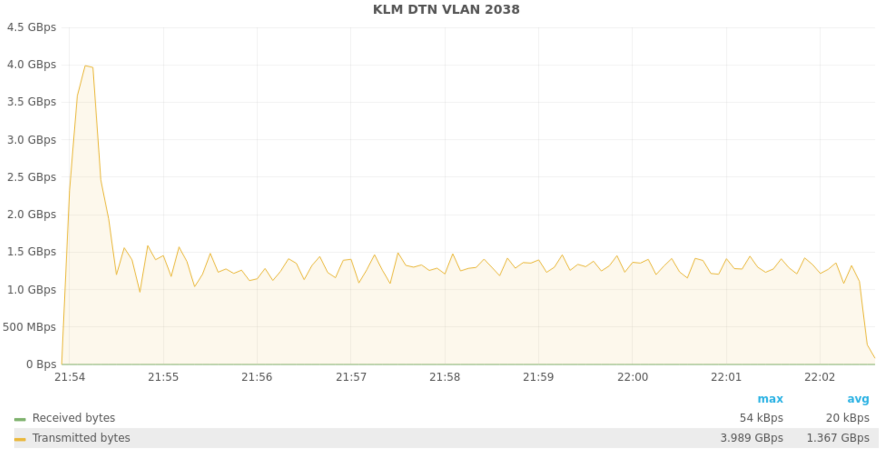


	max	avg
🟩 Received bytes	1 MBps	446 kBps
🟨 Transmitted bytes	5.517 GBps	3.589 GBps

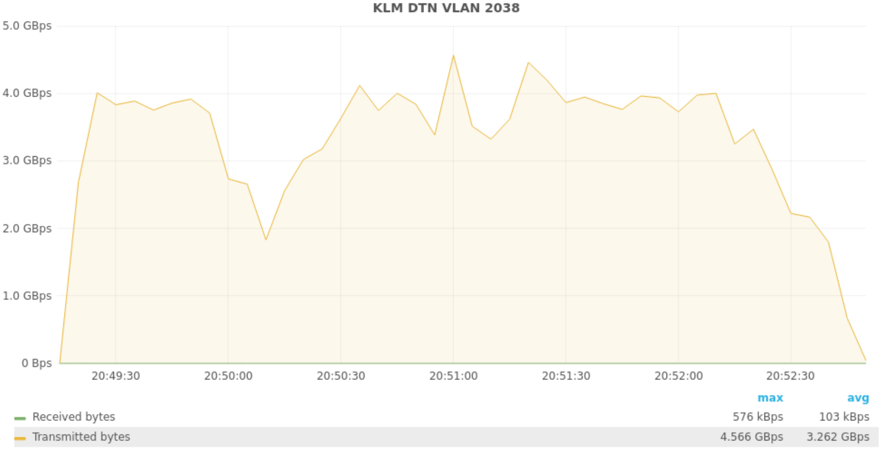
Results: 3rd party, GridFTP UDT, TTC=161 sec., 40% diff.



Results: 3rd party, mdtmFTP, TTC=520 sec.



Results: 3rd party, mdtmFTP, TTC=215 sec., 80% diff.

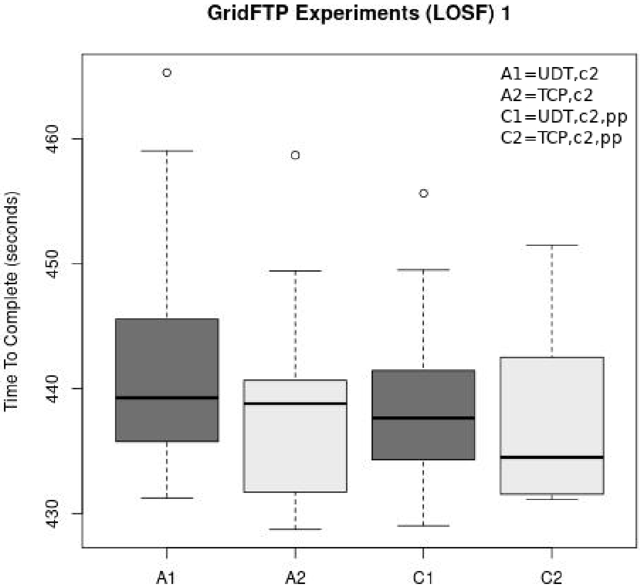


	max	avg
🟩 Received bytes	576 kBps	103 kBps
🟨 Transmitted bytes	4.566 GBps	3.262 GBps

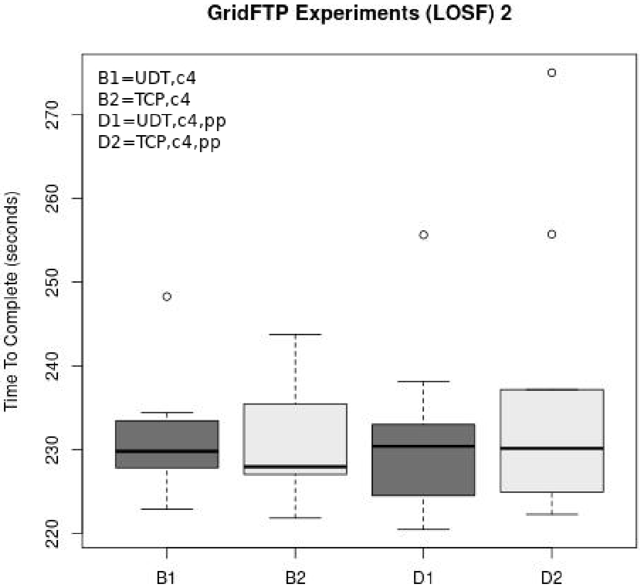
Results: LOSF + KLM

- Node-to-node
  - 3rd party folder transfer only available for mdtmFTP (crashed)
- Globus GridFTP
  - Concurrency
  - Pipelining
  - Parallelism
- mdtmFTP
  - Parallelism
  - Pipelining
  - **Virtual file mechanism for LOSF**

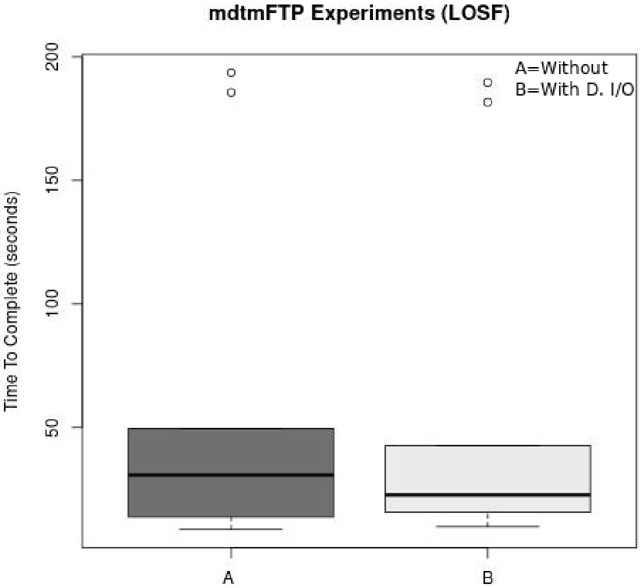
Results: LOSF GridFTP, concurrency 2



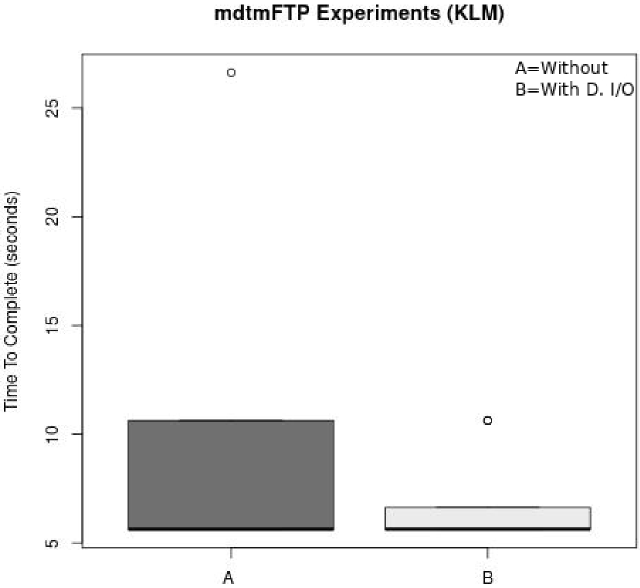
Results: LOSF GridFTP, concurrency 4, 50% diff.



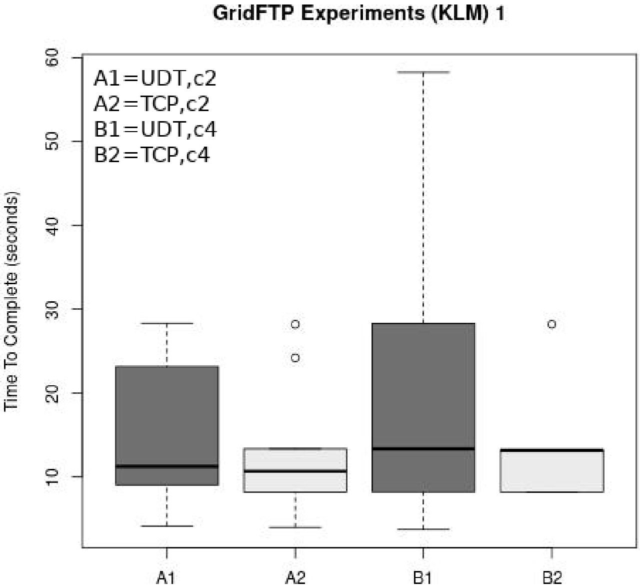
Results: LOSF mdtmFTP, with Direct I/O a 30% diff.



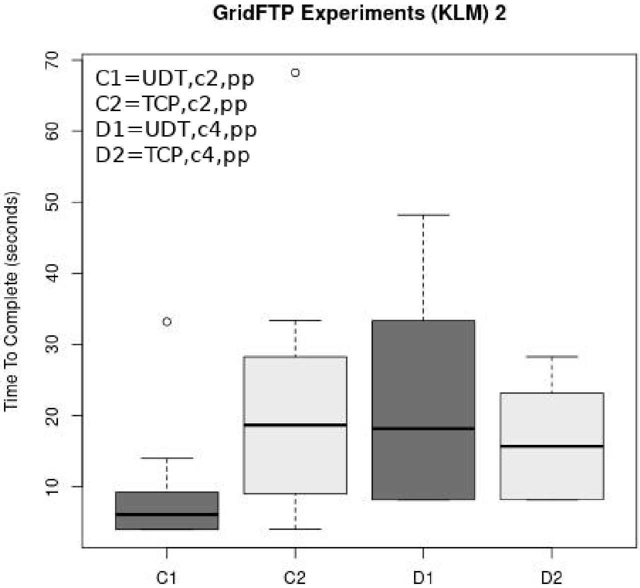
Results: KLM mdtmFTP, with Direct I/O a 65% diff.



Results: KLM GridFTP, without pipelining



Results: KLM GridFTP, with pipelining



Discussion

- mdtmFTP still in development
  - Unclear error messages
  - Limited documentation available
  - Large file performance slow
  - High CPU (90%) usage observed, even when idle
  - Limited testing done in a controlled test environment?
- Large files
  - Globus GridFTP with UDT performed best, 75% faster than mdtmFTP
  - Did not observe more RAM usage
- LOSF/KLM data
  - mdtmFTP's virtual file system greatly benefits performance
  - Globus GridFTP over UDT with concurrency of 2 and pipelining performs equally with KLM data
- Network may not have been fully reserved/stable during testing

Conclusion

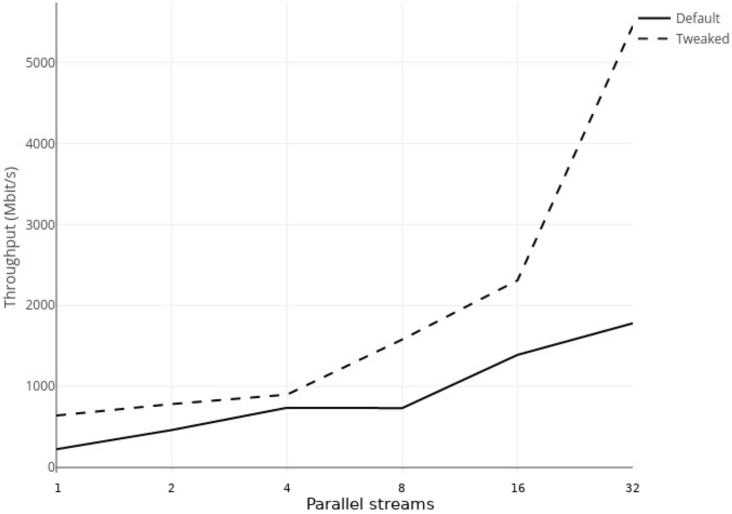
- mdtmFTP is a very promising project
  - Needs more testing and improvements
  - Design is capable of more
  - Performed excellent with LOSF
- Globus GridFTP is here to stay, for now

Future work

- Test Splice and 3rd party folder transfer
- Future testing fo mdtmFTP when it matures
- compare UDT with TCP BBR
- If implemented, test UDT with mdtmFTP
- Redo experiment with a hard network reservation

Questions

Network performance baseline (iPerf)


Default
Tweaked
Throughput (Mbit/s)
Parallel streams