Source: LSST

Experimenting with bulk data transfer for LSST

fabio hernandez

CCIN2P3

FJPPL computing workshop, Lyon, Feb 14th, 2018

Contents

- Background
- R&D activities
- Preliminary results
- Summary

Background: LSST data processing at CC-IN2P3

- Main role of CC-IN2P3

  satellite data release production *under NCSA leadership*

  to contribute to the production of the ***annual data release*** *by* ***processing 50% of the raw data***

  both NCSA and CC-IN2P3 to host an ***entire copy of every annual data release*** *both raw and derived data*

  both sites will exchange and validate the data produced by the other party

- Schedule

  commissioning: 2019-2022

  operations: 2022-2032

LSST DATA CENTERS

Data exchange

- CC-IN2P3 capability to transfer data in bulk is essential to fulfil our mission

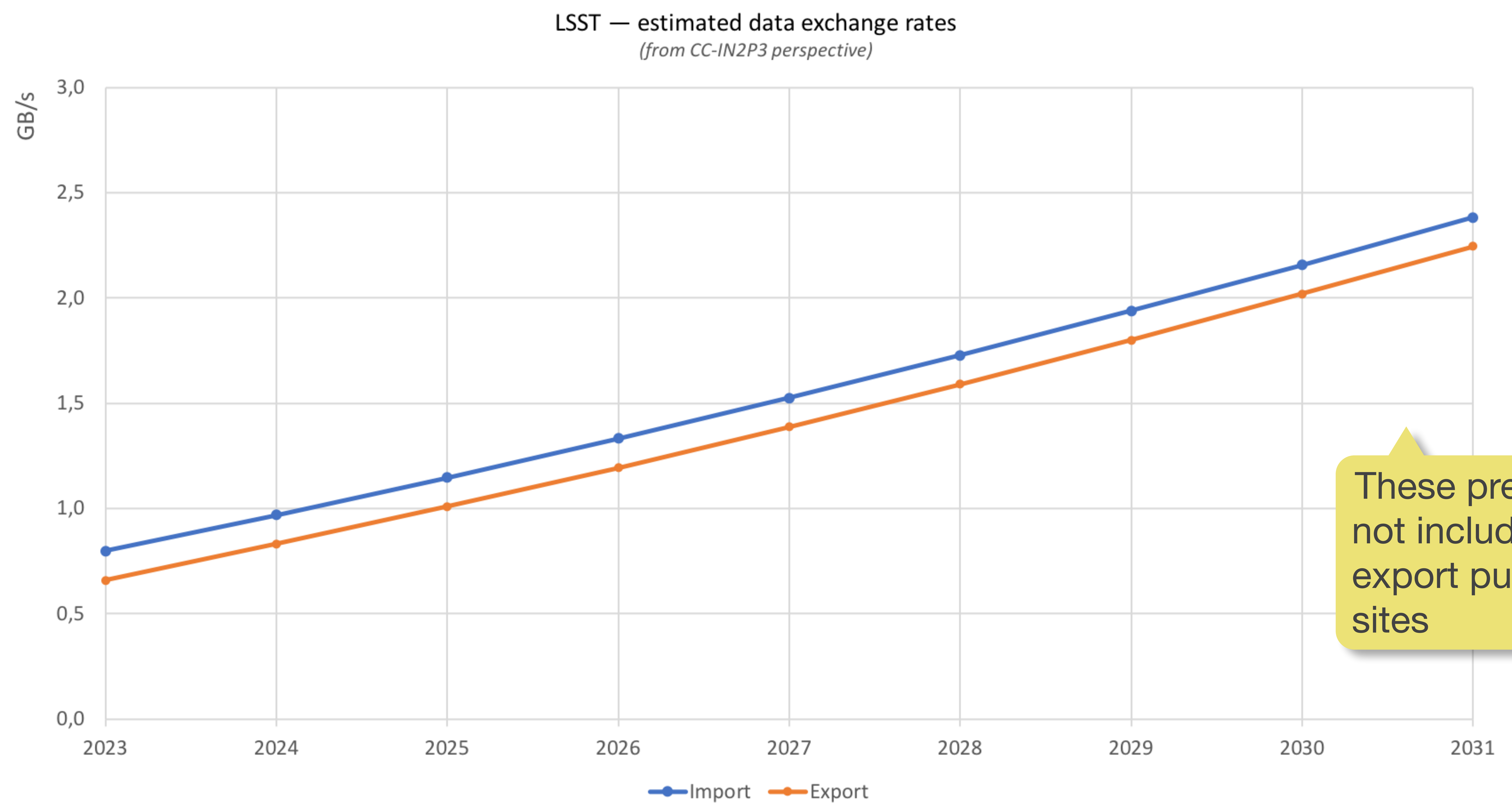
- We need to make sue we are able…

  to import ***raw*** *and* ***derived*** *data from NCSA*

  to export ***derived*** *data to NCSA*

  to export ***released public*** *data to other sites (likely, but not part of any formal agreement)*

Data exchange rates CC-IN2P3 ↔ NCSA



These preliminary figures do not include the rates to export public data to other sites

Data exchange rates (cont.)

	TRANSFER-	STAGING-	DELETION-	CA+	CERN+	DE+	ES+	FR+	IT+	ND+	NL+	RU+	TW+	UK+	US+
TOTAL-	91 % 407 MB/s			97 % 17 MB/s	89 % 126 MB/s	91 % 34 MB/s	90 % 7 MB/s	89 % 59 MB/s	94 % 20 MB/s	90 % 21 MB/s	95 % 16 MB/s	93 % 4 MB/s	89 % 6 MB/s	93 % 27 MB/s	88 % 69 MB/s
IN2P3-CC+	91 % 407 MB/s			97 % 17 MB/s	89 % 126 MB/s	91 % 34 MB/s	90 % 7 MB/s	89 % 59 MB/s	94 % 20 MB/s	90 % 21 MB/s	95 % 16 MB/s	93 % 4 MB/s	89 % 6 MB/s	93 % 27 MB/s	88 % 69 MB/s

ATLAS data import to CC-IN2P3 — observed transfer throughput

(aggregated over full year 2017)

Source: E. Vamvakopoulos, ATLAS dashboard

Data exchange rates (cont.)

origin site	throughput
Tier-1: Fermilab	160 MB/s
Tier-2s: Caltech, Nebraska, Florida	90 MB/s
Aggregated import rate all CMS sites	1200 MB/s

CMS data import to CC-IN2P3 — observed transfer throughput

(aggregated over 26 last weeks)

Source: S. Gadrat

R&D activities

Guidelines

- **Separate storage** infrastructure for import / export needs and for data processing needs

  bulk data exchange and local data processing are two completely different use cases each with its own distinct constraints

  allows for asynchronous import/export and long term storage

  we have been doing tests with object stores, as import / export buffers

- Use **standard transport protocols**, likely to remain relevant for the 2020-2030 period

  we have been doing tests with secure HTTP/1 and HTTP/2, i.e. HTTP over TLS

  TLS standard ensures confidentiality and data integrity

- Optimise for **throughput**, not latency

Benchmarking

- Developed benchmarking tools

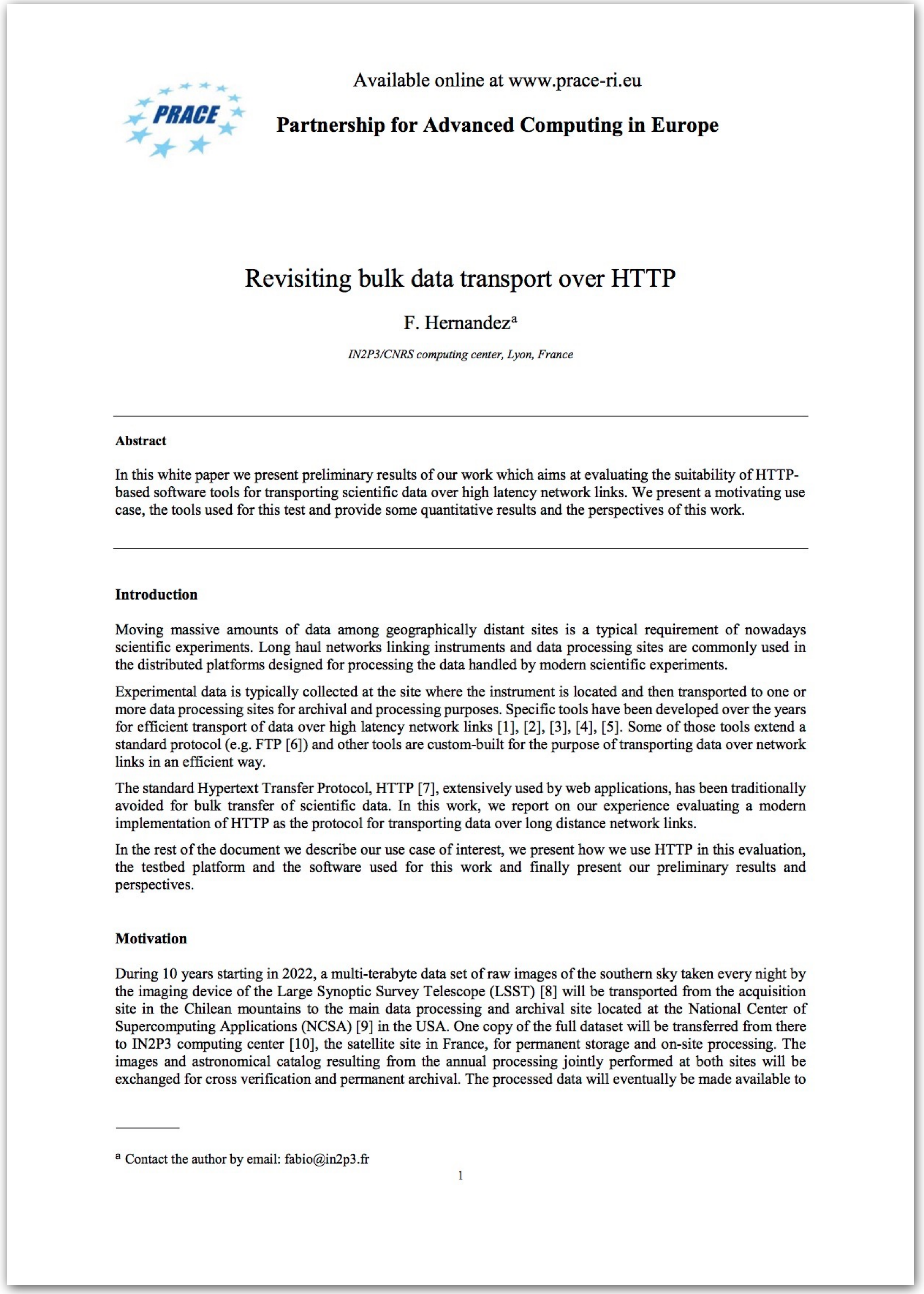
  tlsping, netperf, chasqui

- Publication of first results as a white paper

  contribution to the European project PRACE 4IP

- Go programming language

  built-in concurrency important when transferring data over high latency network links



Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

Revisiting bulk data transport over HTTP

F. Hernandez[a]

IN2P3/CNRS computing center, Lyon, France

Abstract

In this white paper we present preliminary results of our work which aims at evaluating the suitability of HTTP-based software tools for transporting scientific data over high latency network links. We present a motivating use case, the tools used for this test and provide some quantitative results and the perspectives of this work.

Introduction

Moving massive amounts of data among geographically distant sites is a typical requirement of nowadays scientific experiments. Long haul networks linking instruments and data processing sites are commonly used in the distributed platforms designed for processing the data handled by modern scientific experiments.

Experimental data is typically collected at the site where the instrument is located and then transported to one or more data processing sites for archival and processing purposes. Specific tools have been developed over the years for efficient transport of data over high latency network links [1], [2], [3], [4], [5]. Some of those tools extend a standard protocol (e.g. FTP [6]) and other tools are custom-built for the purpose of transporting data over network links in an efficient way.

The standard Hypertext Transfer Protocol, HTTP [7], extensively used by web applications, has been traditionally avoided for bulk transfer of scientific data. In this work, we report on our experience evaluating a modern implementation of HTTP as the protocol for transporting data over long distance network links.

In the rest of the document we describe our use case of interest, we present how we use HTTP in this evaluation, the testbed platform and the software used for this work and finally present our preliminary results and perspectives.

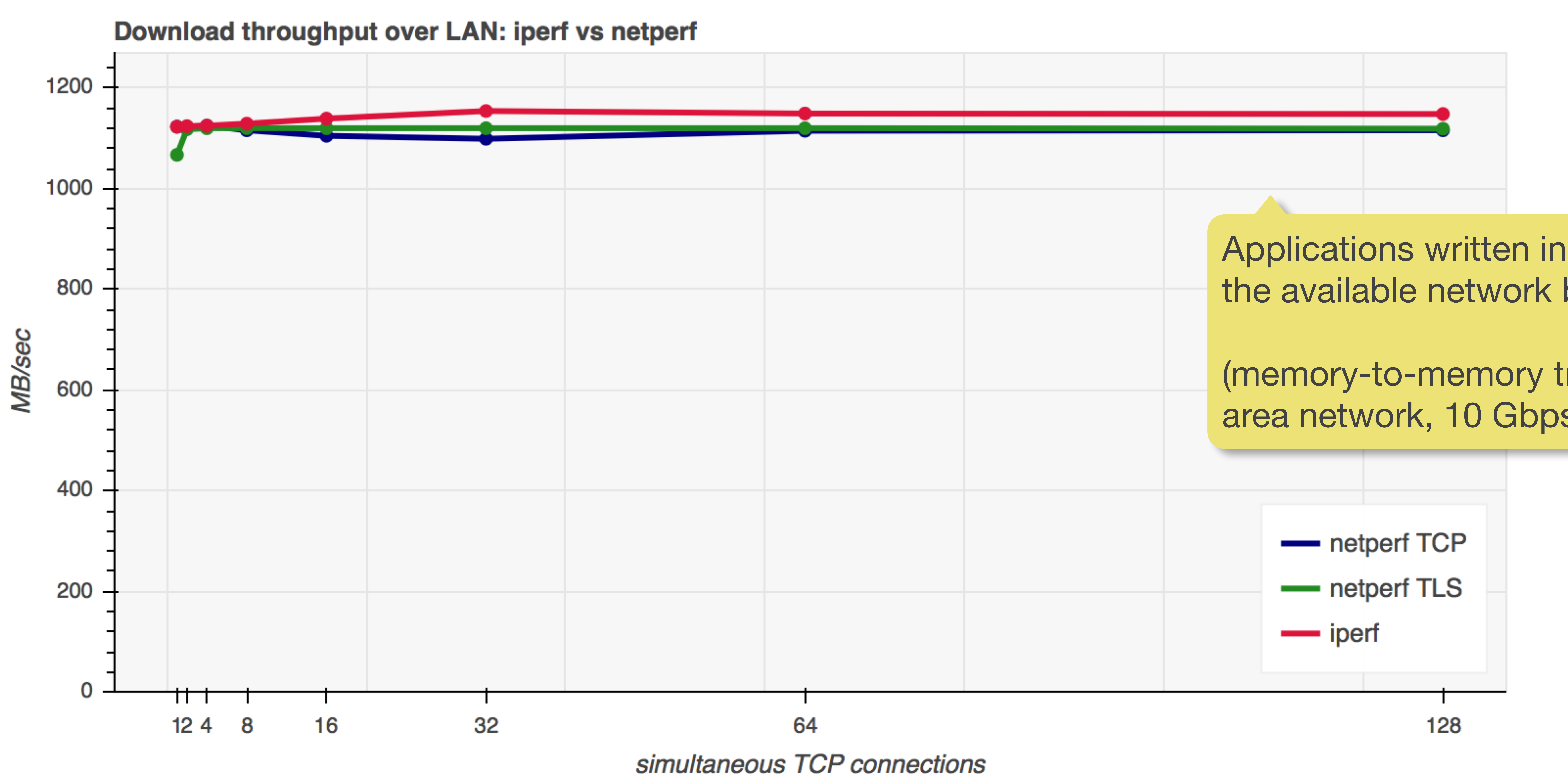
Motivation

During 10 years starting in 2022, a multi-terabyte data set of raw images of the southern sky taken every night by the imaging device of the Large Synoptic Survey Telescope (LSST) [8] will be transported from the acquisition site in the Chilean mountains to the main data processing and archival site located at the National Center of Supercomputing Applications (NCSA) [9] in the USA. One copy of the full dataset will be transferred from there to IN2P3 computing center [10], the satellite site in France, for permanent storage and on-site processing. The images and astronomical catalog resulting from the annual processing jointly performed at both sites will be exchanged for cross verification and permanent archival. The processed data will eventually be made available to

[a] Contact the author by email: fabio@in2p3.fr

1

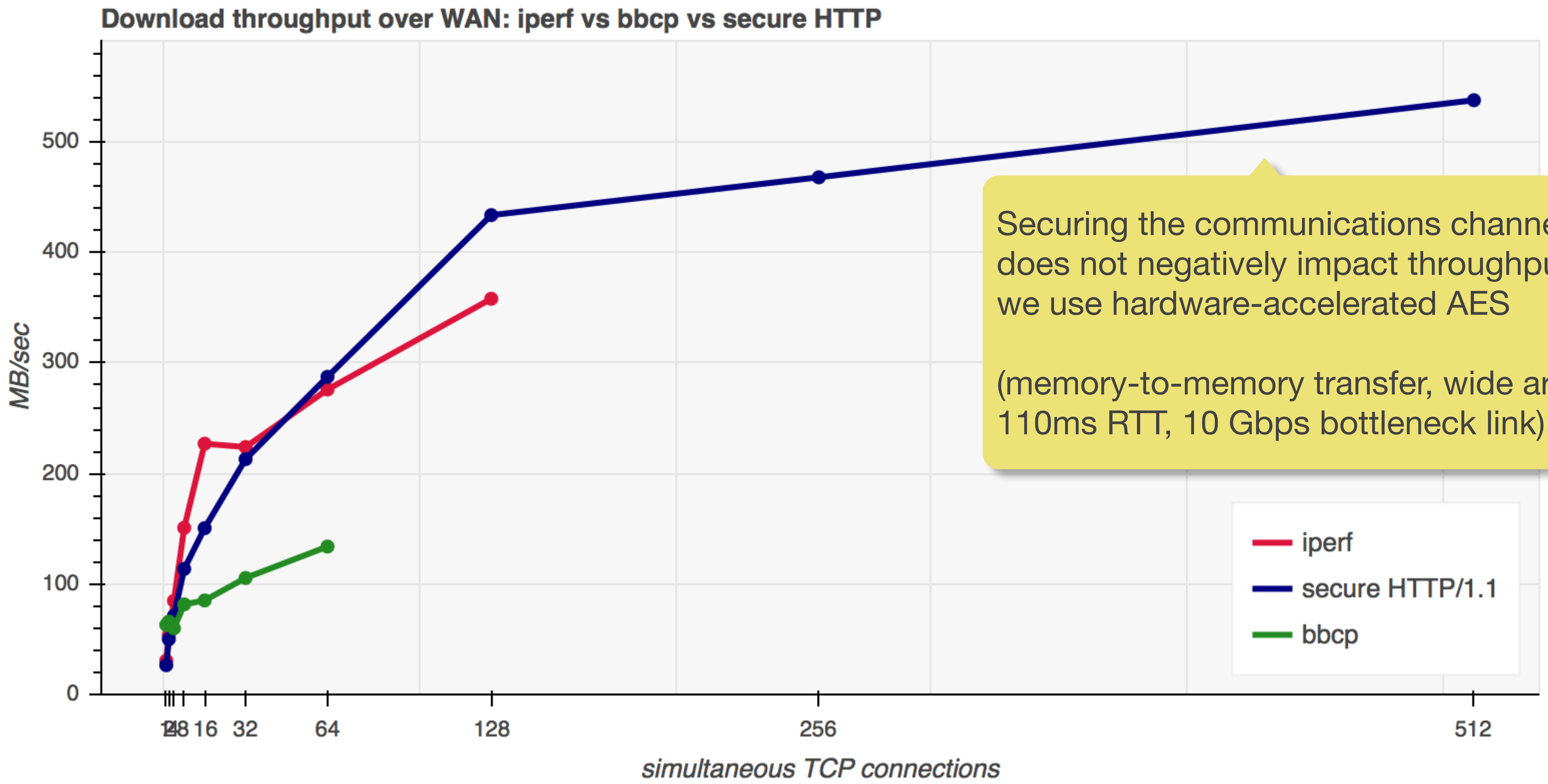
Benchmarking (cont.)



Applications written in Go can exploit the available network bandwidth

(memory-to-memory transfer, local area network, 10 Gbps link)

Benchmarking (cont.)

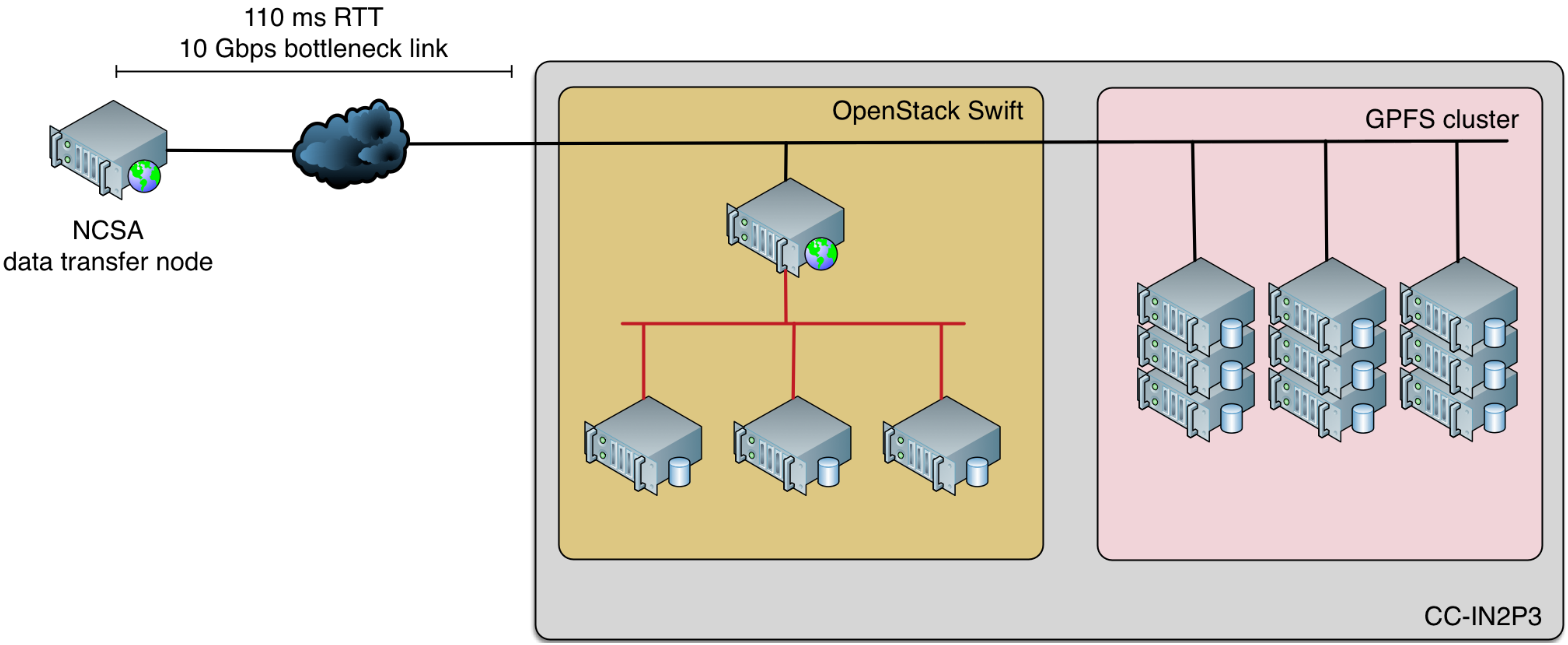


Securing the communications channel using TLS does not negatively impact throughput, provided we use hardware-accelerated AES

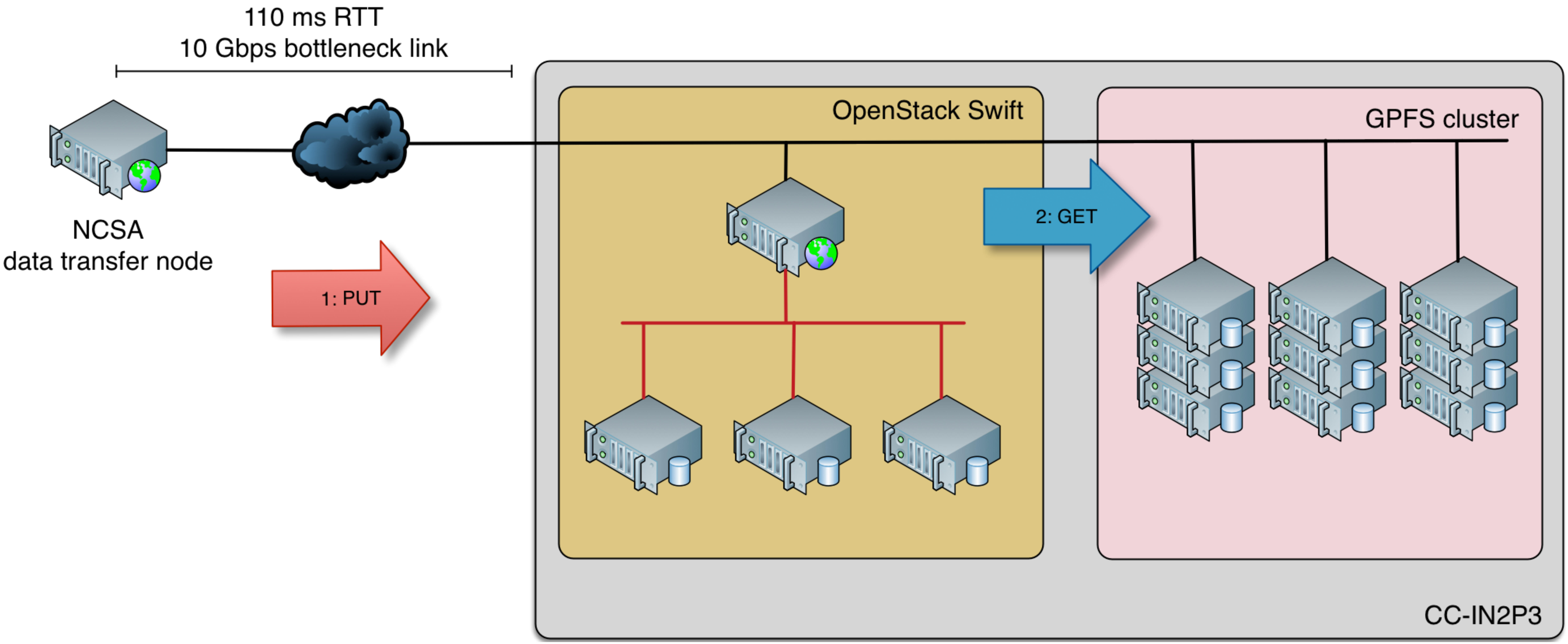
(memory-to-memory transfer, wide area network, 110ms RTT, 10 Gbps bottleneck link)

Transfer model: push over HTTPS

Transit buffer: object store



Transit buffer: object store (cont.)

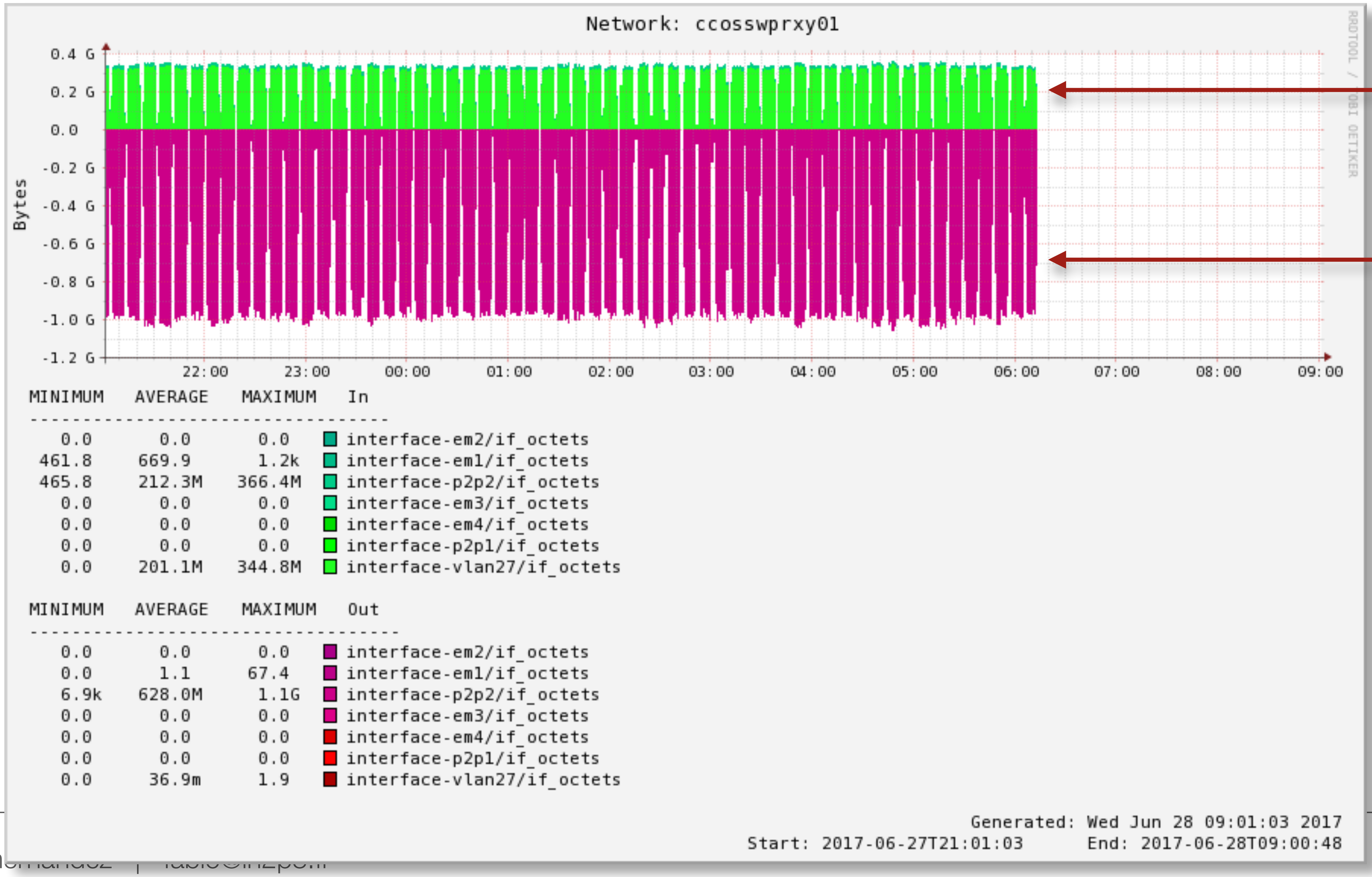


Transit buffer: object store (cont.)

- Step 1: import HSC first public data release from NCSA to Swift

  15 TB, 800K+ FITS files, ~12 hours

  data pushed from NCSA to CC-IN2P3's Swift over a secure channel



NCSA → CC-IN2P3 throughput over HTTPS: **~350 MB/sec**

Data replication to 3 different file servers within Swift

Results obtained without tuning all the components

Some of the limiting factors identified

Transit buffer: object store (cont.)

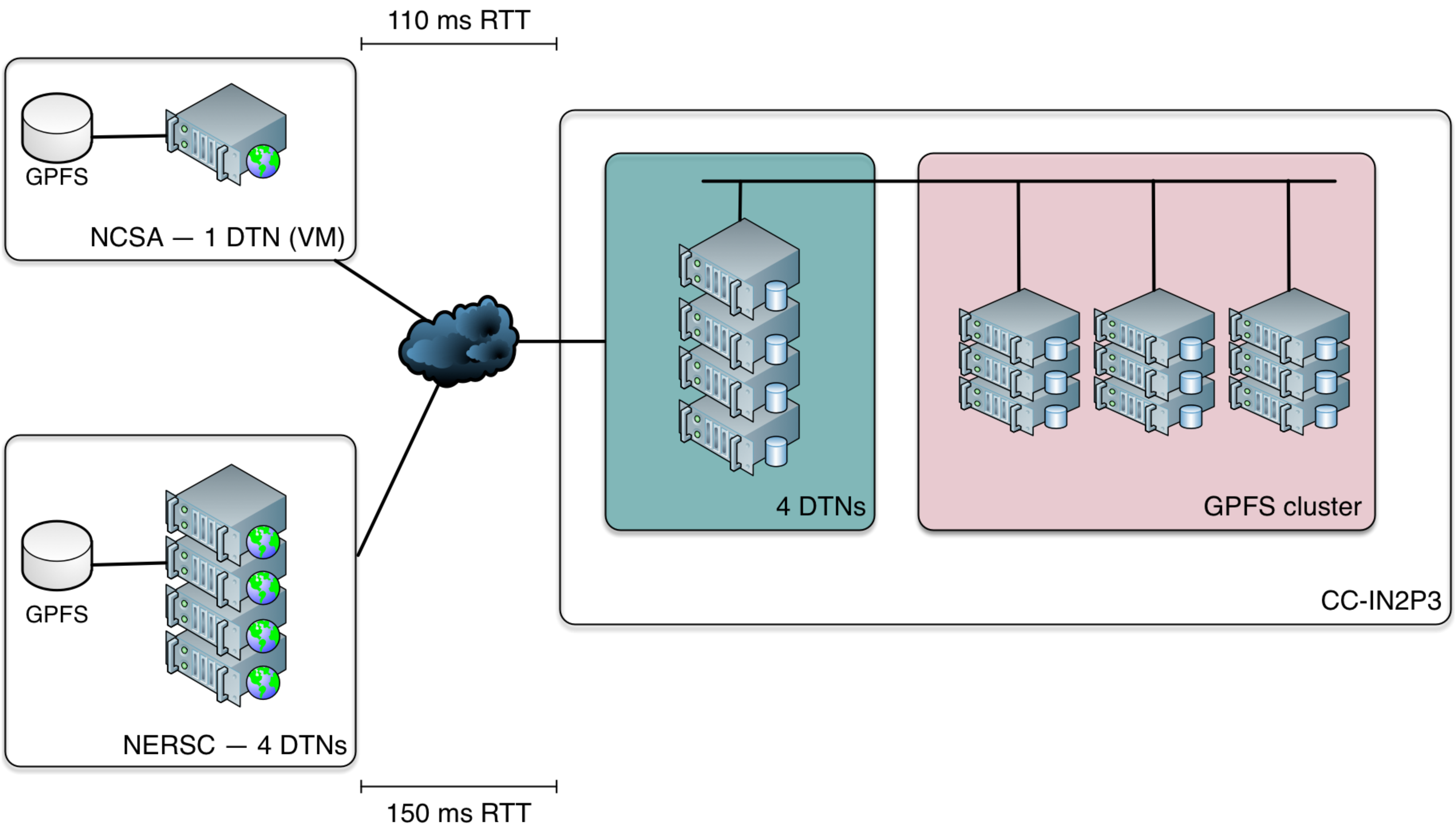
- Step 2: download data from transit buffer (Swift) and save them in permanent location (GPFS)

Data downloaded from Swift over a secure channel and written to GPFS: **~550 MB/sec**

(LAN, bottleneck link 10 Gbps)

Transfer mode: pull over HTTPS

Transfer model: pull, HTTPS



Transfer model: pull, HTTPS (cont.)

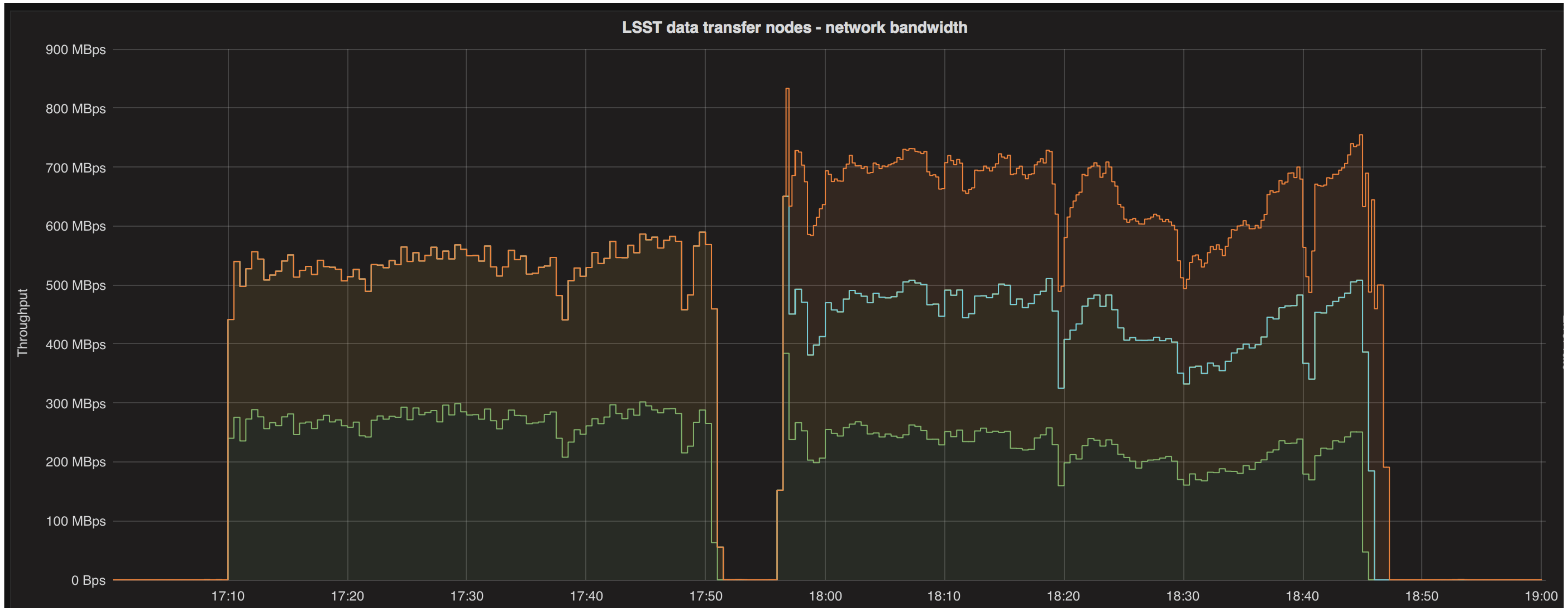
Aggregated application-level throughput: **~360 MB/sec**

(disk-to-memory transfer, wide area network, 110ms RTT, 10 Gbps bottleneck link)

Data flow: **NCSA** (GPFS) → **CC-IN2P3** (memory)

1 server, 4 clients

Transfer model: pull, HTTPS (cont.)



Aggregated application-level throughput: **~606 MB/sec**

(disk-to-memory transfer, wide area network, 150ms RTT)

Data flow: **NERSC** (GPFS) → **CC-IN2P3** (memory)

2 servers, 2 clients then 3 servers, 3 clients

Conclusions

Conclusions

- We have made some progress understanding the problem we need to solve

  a baseline to improve upon is now established

- Pull model on top of HTTPS, combined with transit buffers seems a realistic solution

  nothing LSST-specific

- More engineering work is needed to design a solution at the scale needed for LSST

Questions & Comments