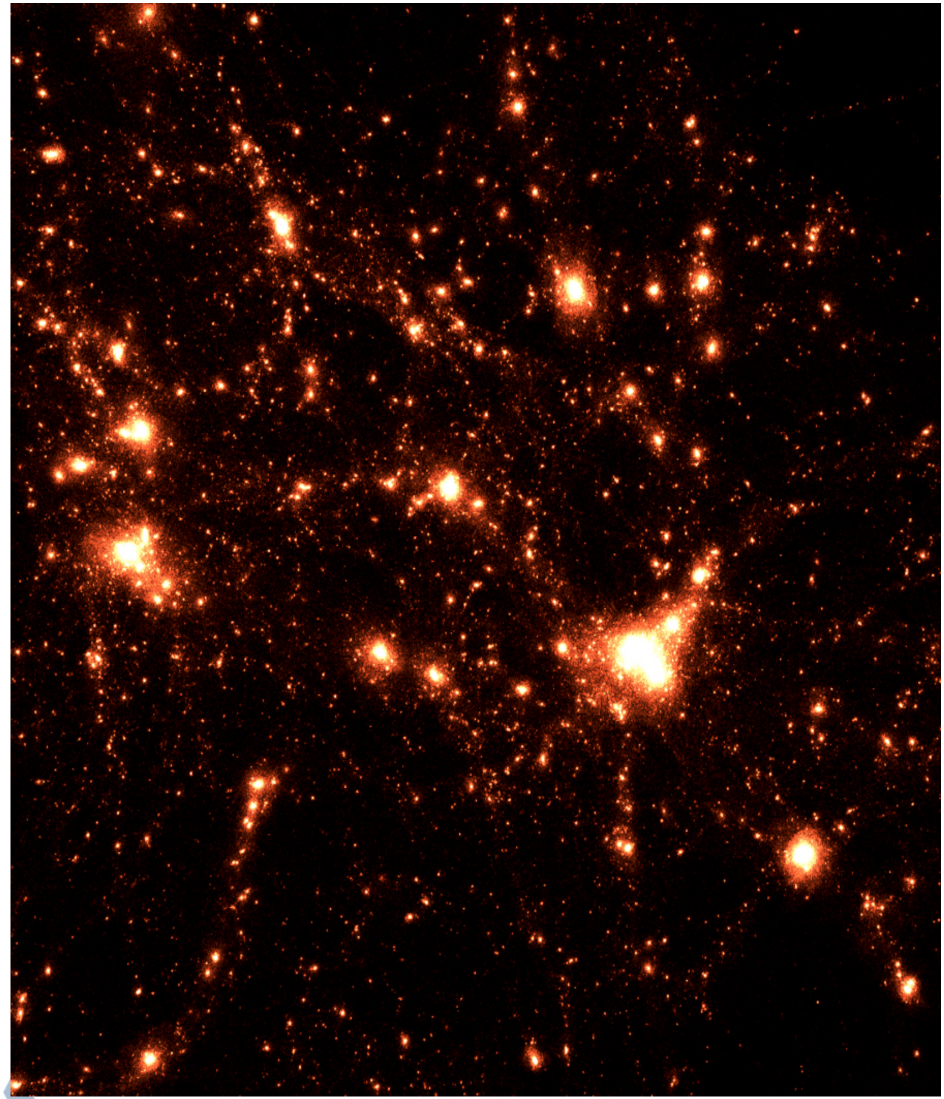
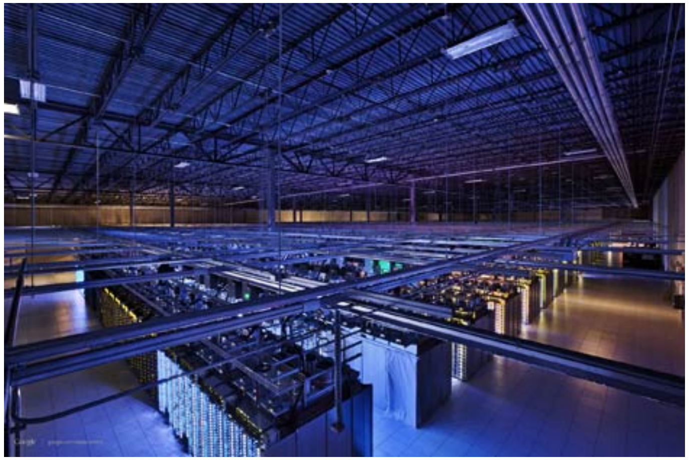
Transferring a Petabyte in a Day

Raj Kettimuthu, Zhengchun Liu, David Wheeler, Ian Foster, Katrin Heitmann, Franck Cappello

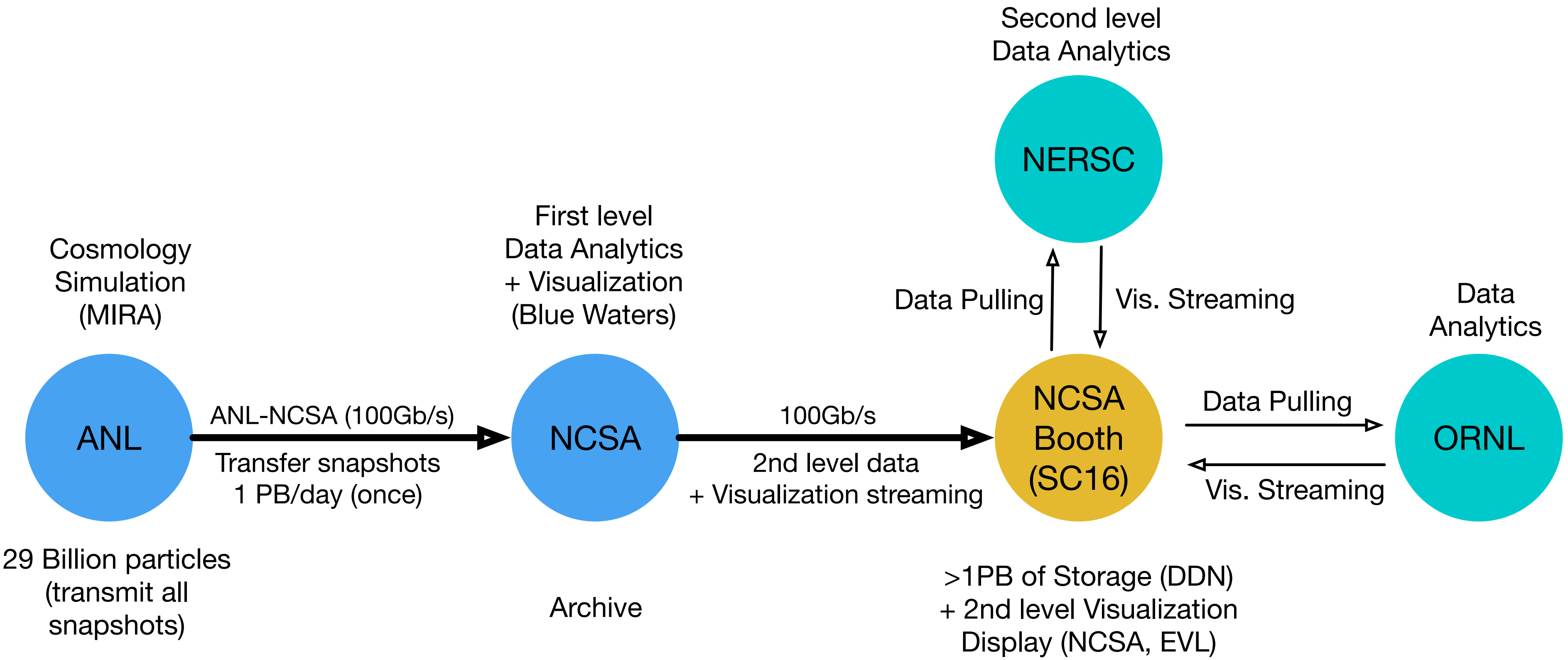
Huge amount of data from extreme scale simulations and experiments



Systems have different capabilities



SC16 demonstration

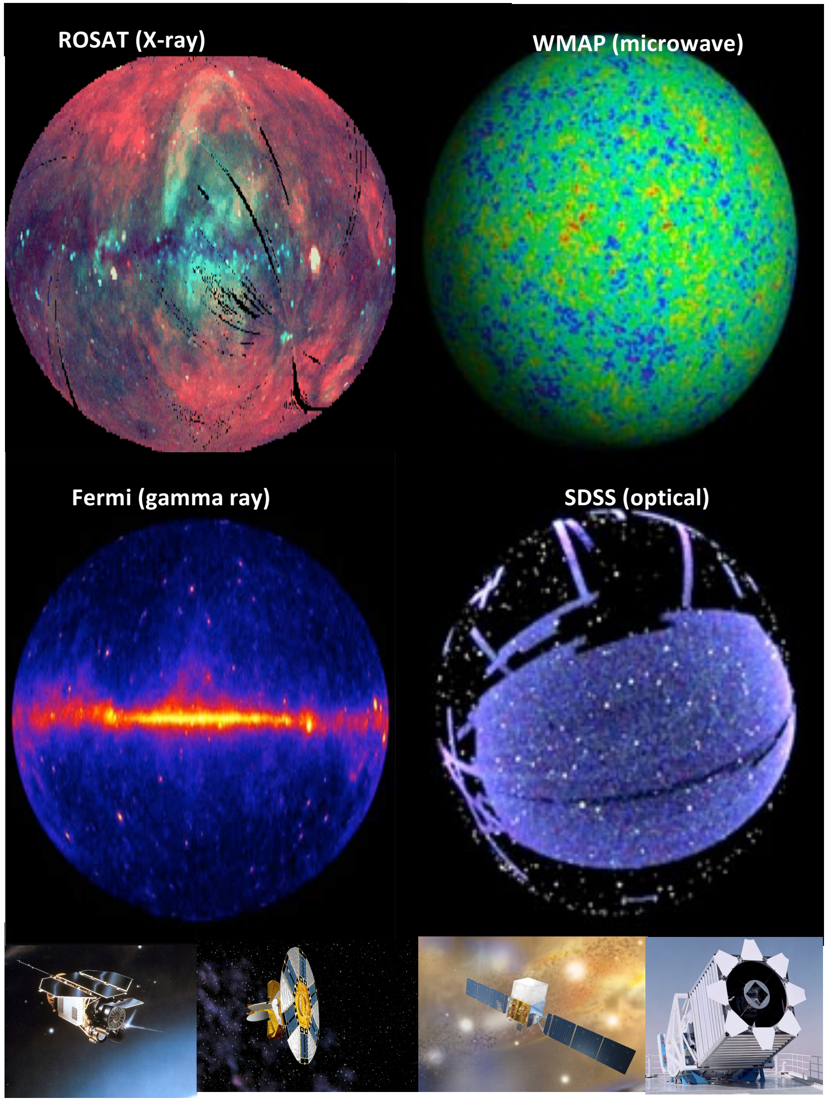


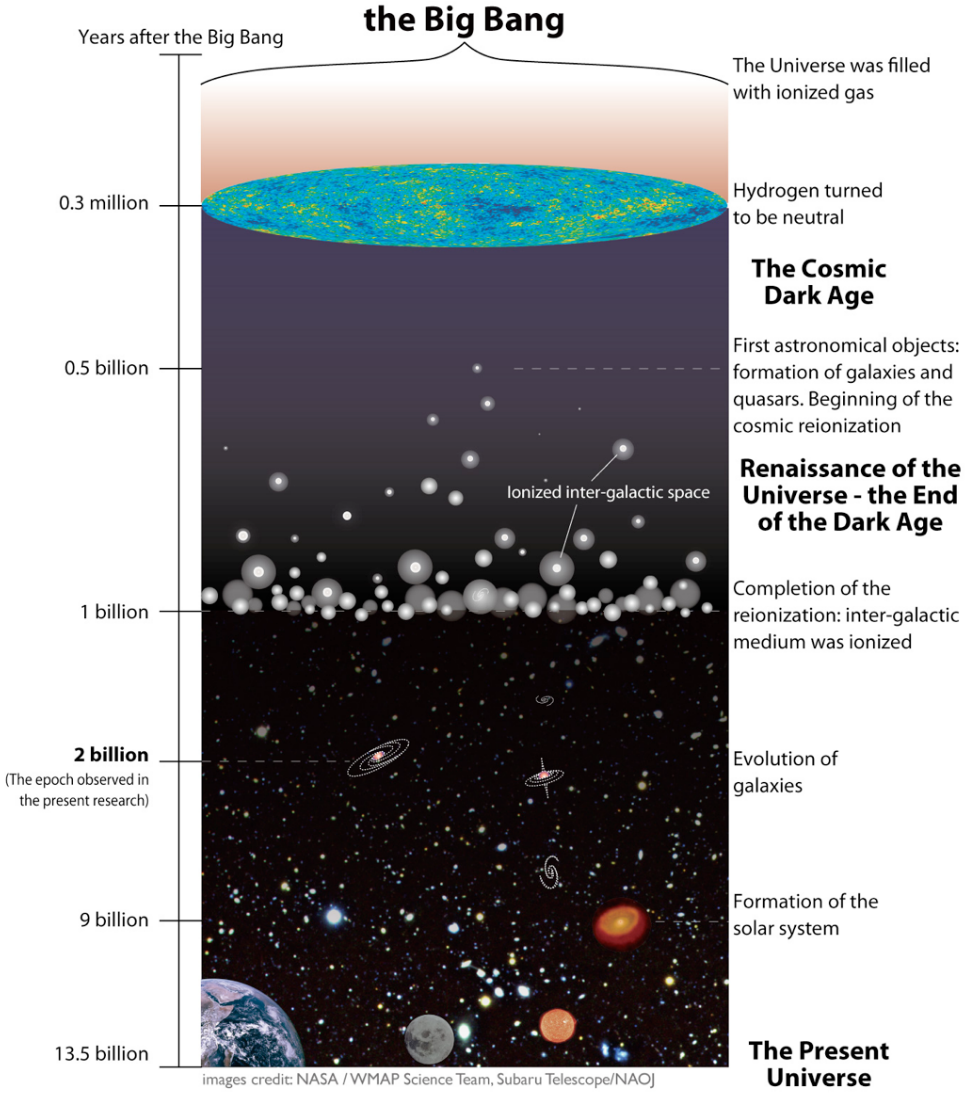
Objectives

- Running a state-of-the-art cosmology simulation and analyzing all snapshots
  - Currently only one in every five or 10 snapshots is stored or communicated
- Combining two different types of systems (simulation on Mira and data analytics on Blue Waters)
  - Geographically distributed, different administrative domains
  - Run an extreme-scale simulation and analyze the output in a pipelined fashion
- Many previous studies have varied transfer parameters such as concurrency and parallelism to improve data transfer performance
  - We also demonstrate the value of varying the file size, which provides additional flexibility for optimization
- We demonstrate these methods in the context of dedicated data transfer nodes and a 100 Gb/s network

Science case

K. Heitmann et al.





the Big Bang

Years after the Big Bang

The Universe was filled with ionized gas

0.3 million

Hydrogen turned to be neutral

The Cosmic Dark Age

0.5 billion

First astronomical objects: formation of galaxies and quasars. Beginning of the cosmic reionization

Ionized inter-galactic space

Renaissance of the Universe - the End of the Dark Age

1 billion

Completion of the reionization: inter-galactic medium was ionized

2 billion
(The epoch observed in the present research)

Evolution of galaxies

9 billion

Formation of the solar system

13.5 billion

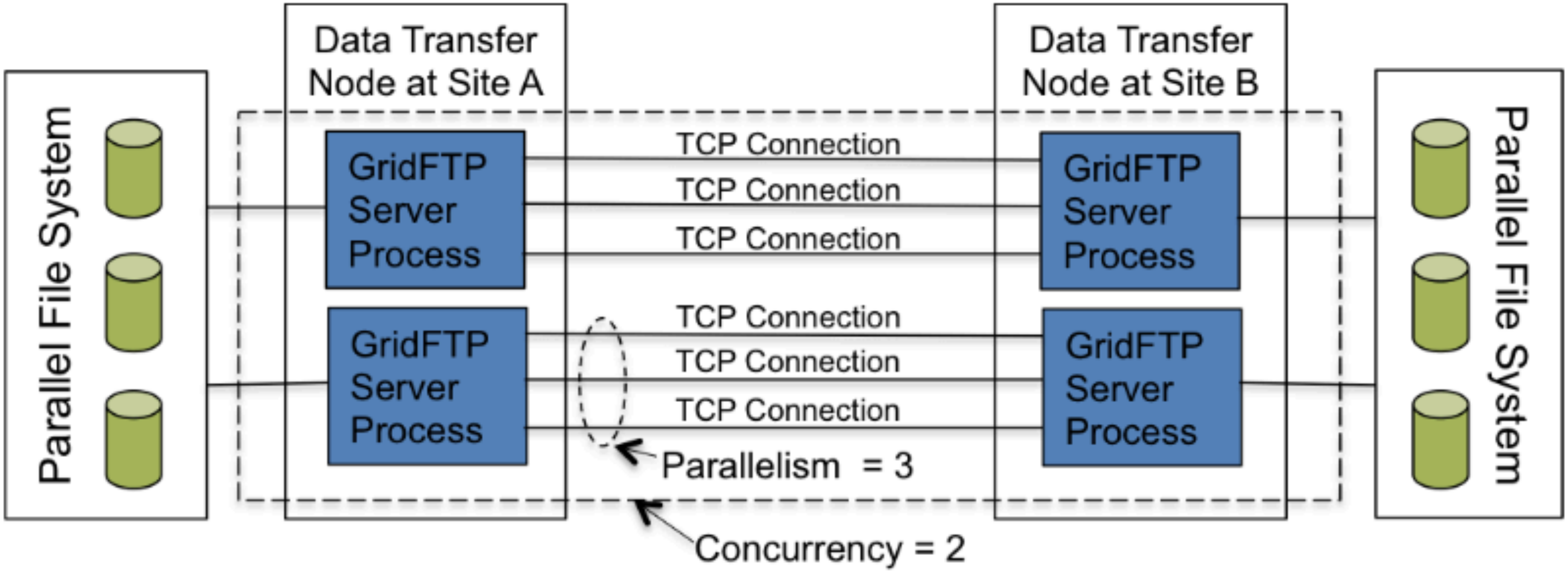
The Present Universe

images credit: NASA / WMAP Science Team, Subaru Telescope/NAOJ

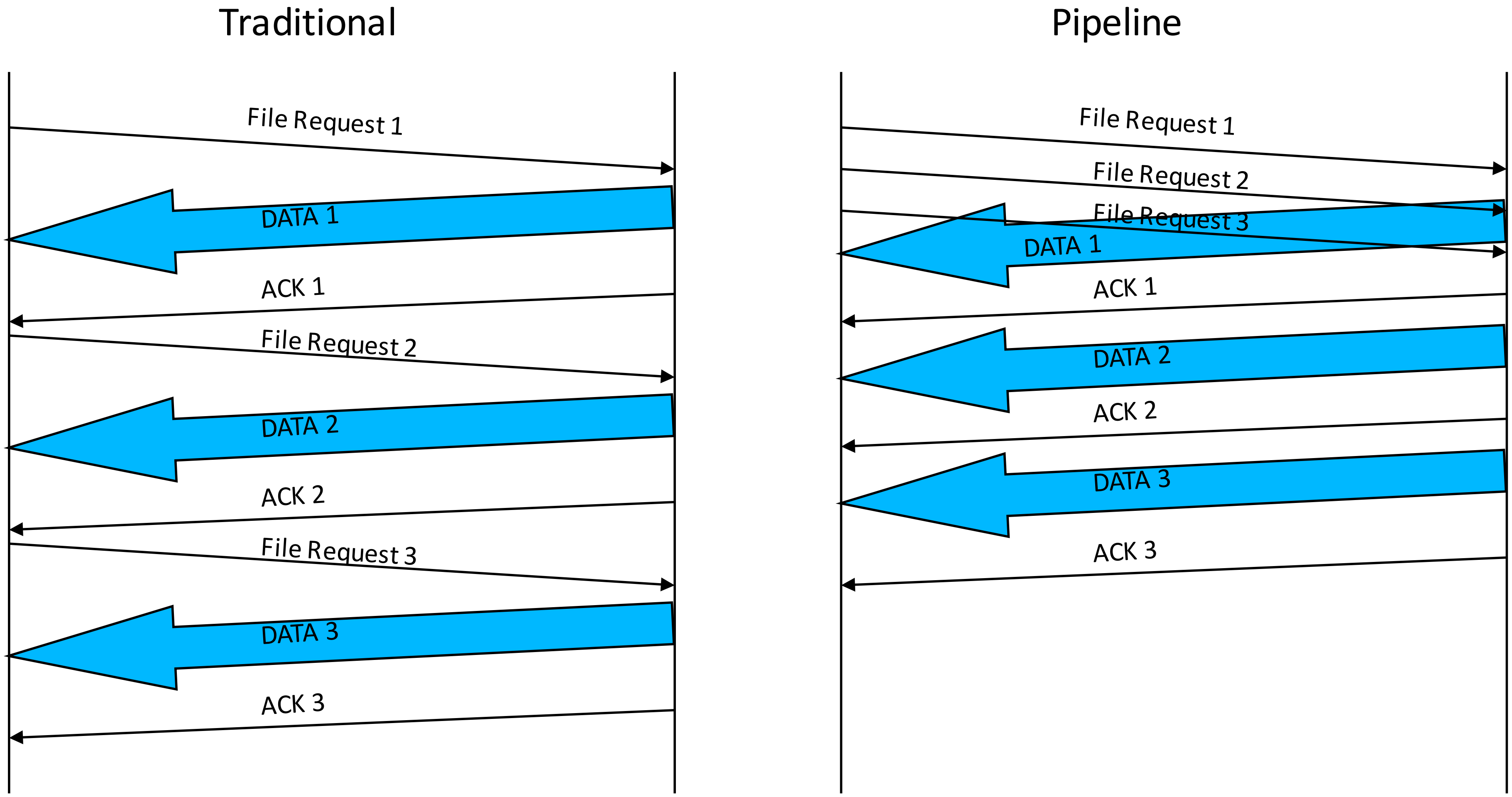
Demo environment

- Source of the data was the GPFS parallel file system on the Mira supercomputer at Argonne
- Destination was the Lustre parallel file system on the Blue Waters supercomputer at NCSA
- Argonne has 12 data transfer nodes (DTNs) dedicated for wide-area data transfer
- NCSA has 28 DTNs
- Each DTN runs a GridFTP server
- Globus to orchestrate our data transfers
  - Automatic fault recovery and load balancing among the available GridFTP servers on both ends.

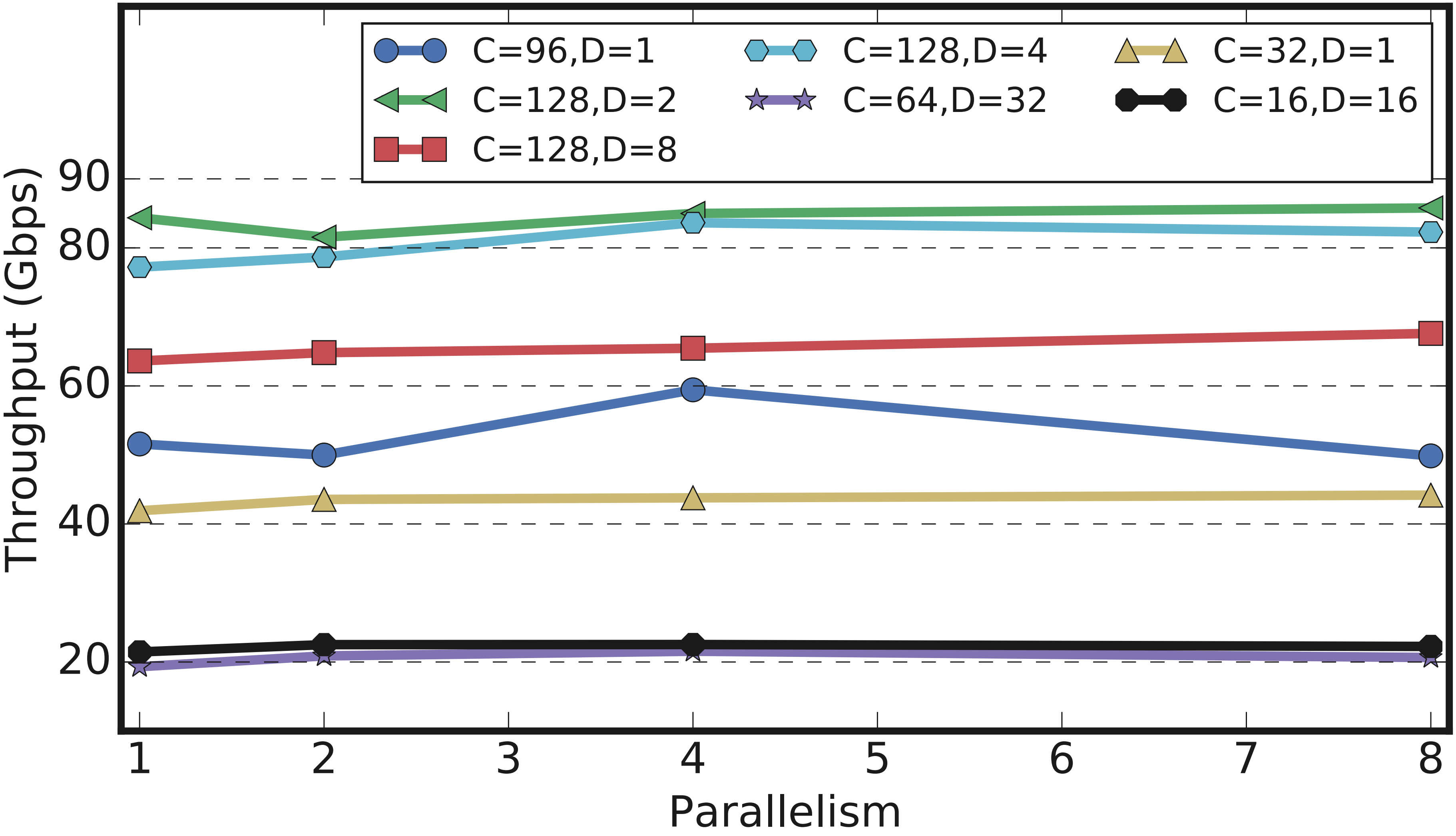
GridFTP concurrency and parallelism



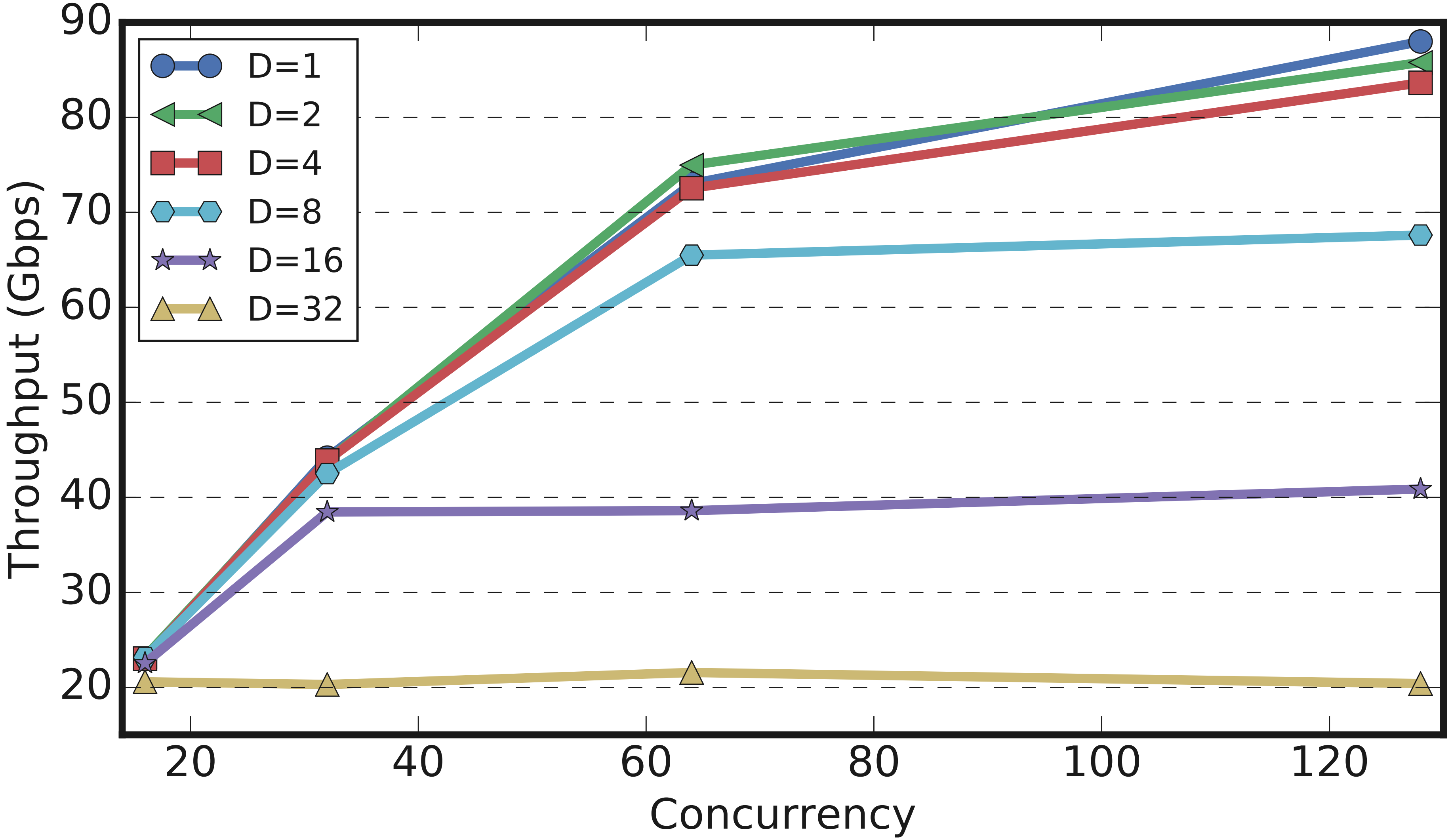
GridFTP pipelining



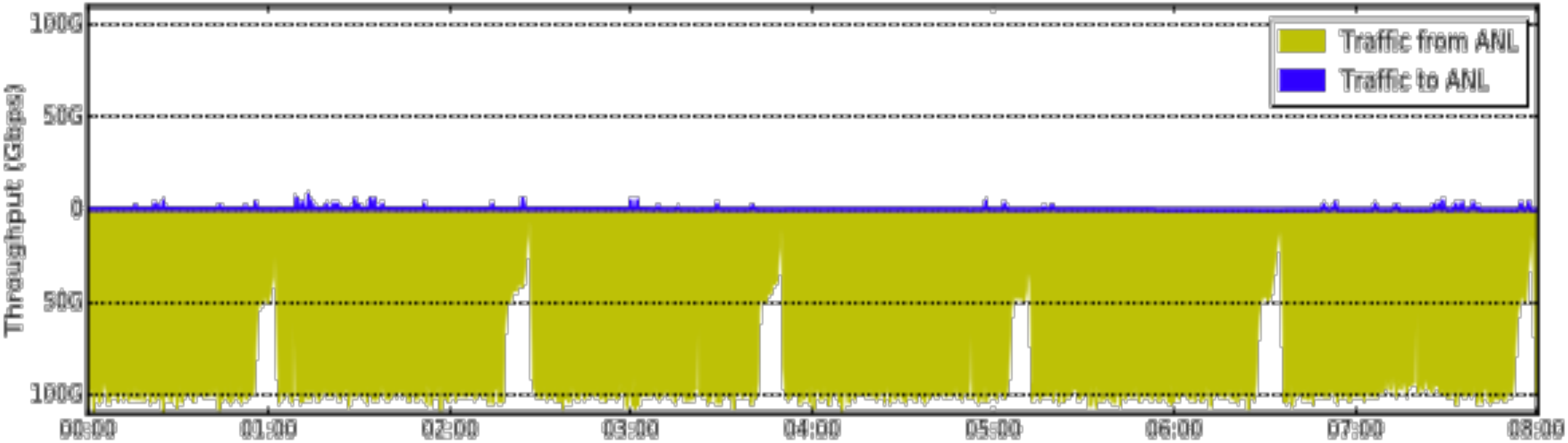
Impact of tuning parameters

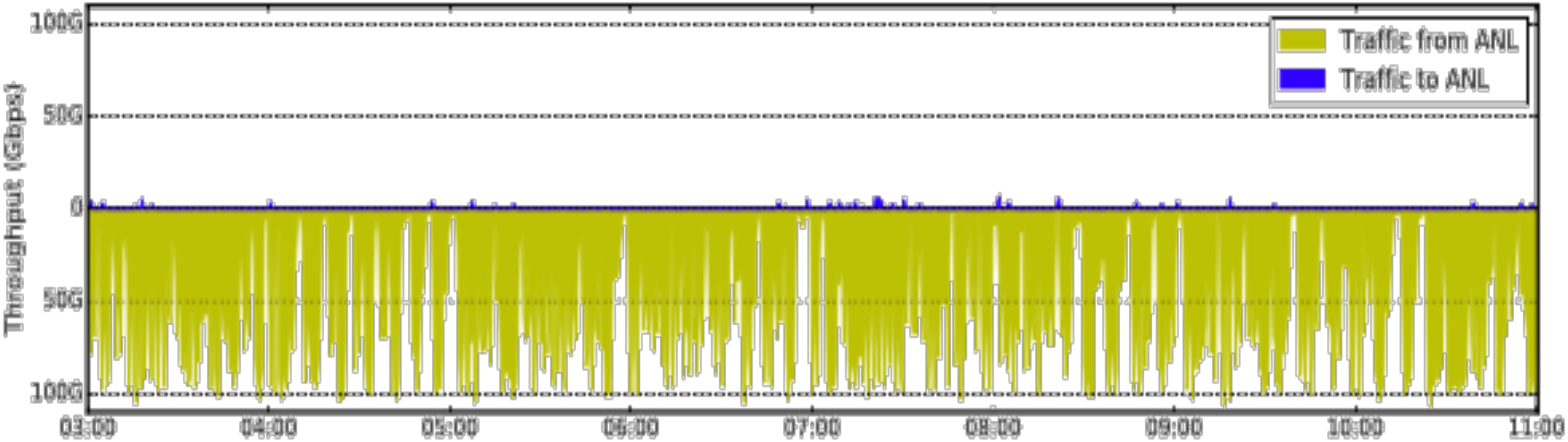


Impact of tuning parameters



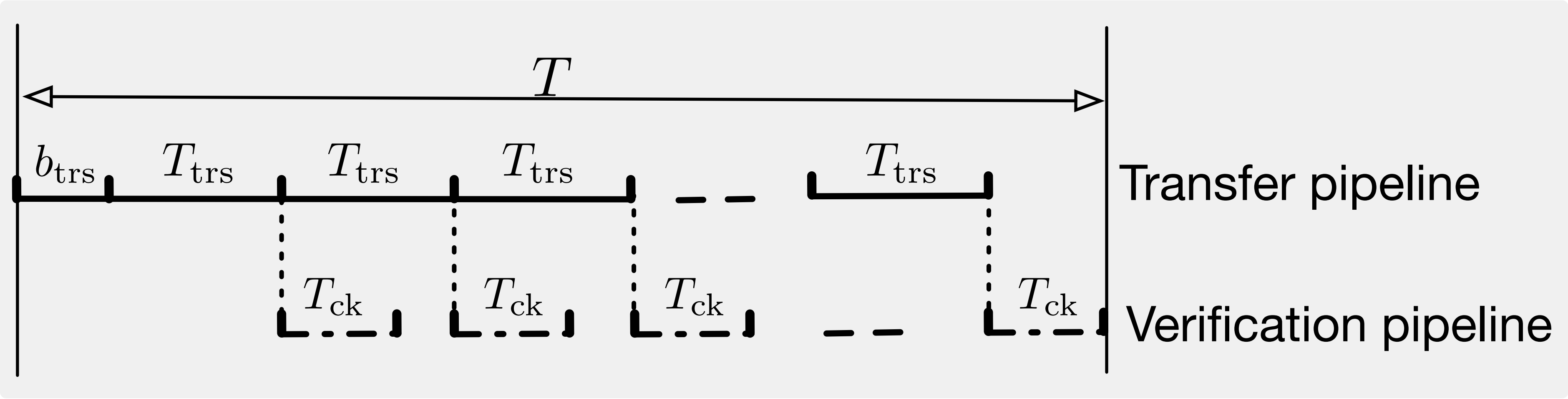
Transfer performance



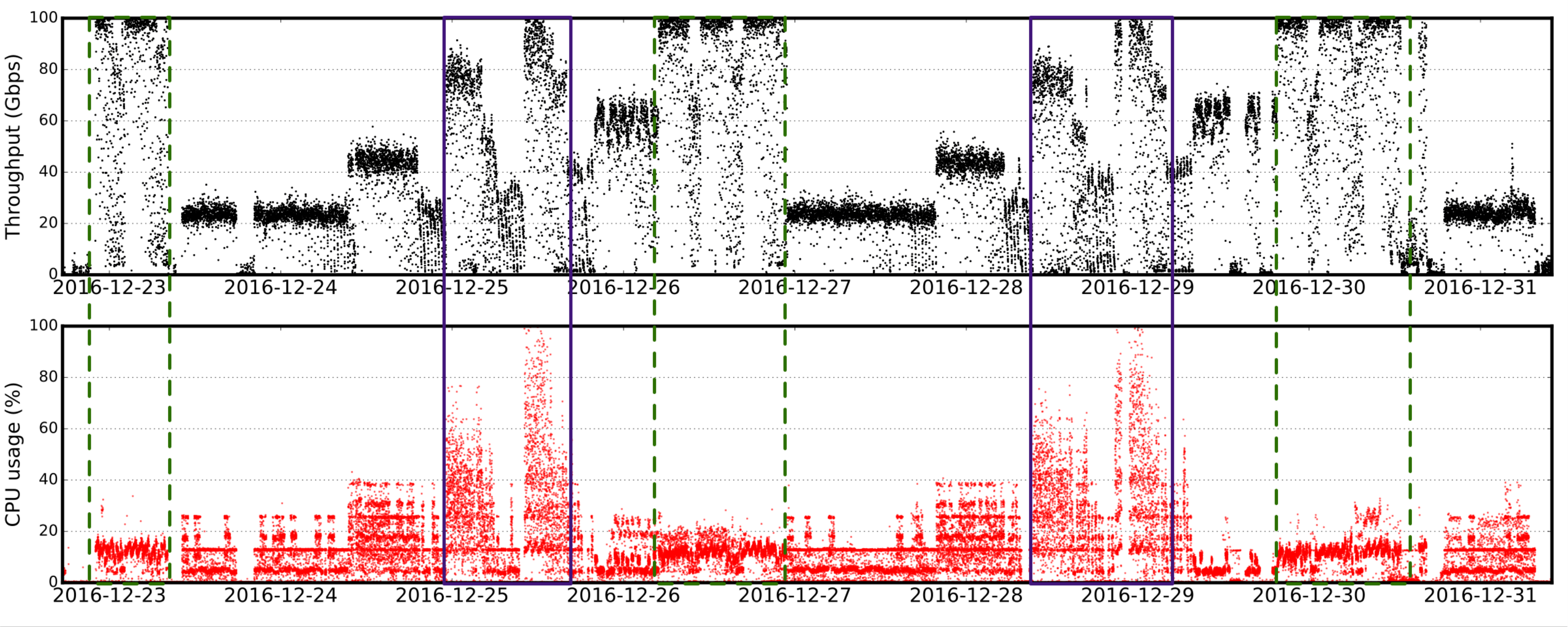


Checksum verification

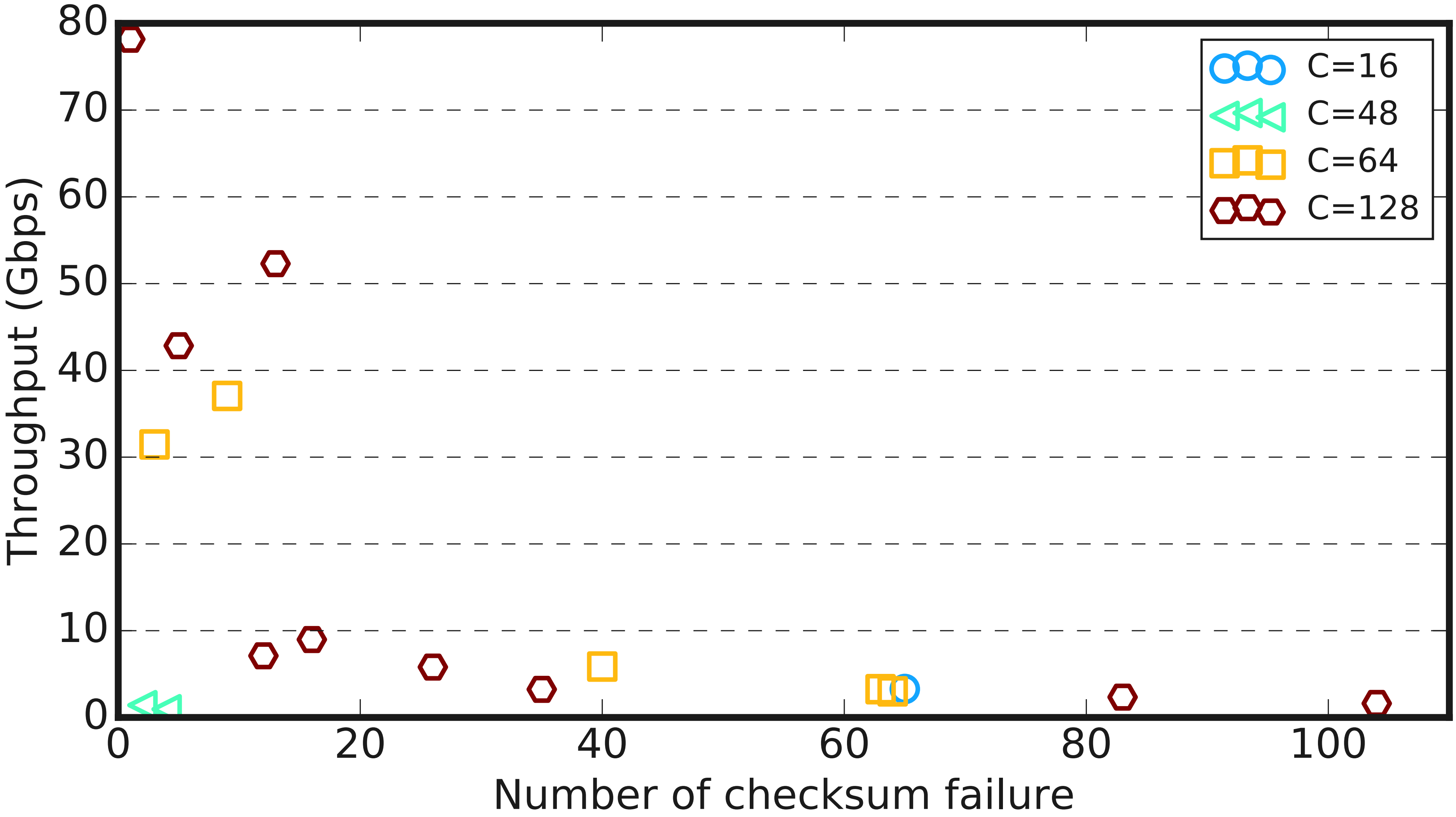
- 16-bit TCP checksum inadequate for detecting data corruption and corruption can occur during file system operations
- Globus pipelines the transfer and checksum computation
  - Checksum computation of the ith file happens in parallel with the transfer of the (i + 1)th file



Checksum overhead



Impact of checksum failures



A model to find optimal number of files

- A simple linear model of transfer time for a single file:

$T_{trs} = a_{trs}x + b_{trs}$; a_{trs} – unit transfer time, x – file size, b_{trs} - startup cost

- $T_{ck} = a_{ck}\, x + b_{ck}$; a_{ck} – unit checksum time, b_{ck} – checksum startup cost

- Assuming that unit checksum time is less than unit transfer time, the total time T to transfer n files with one GridFTP process

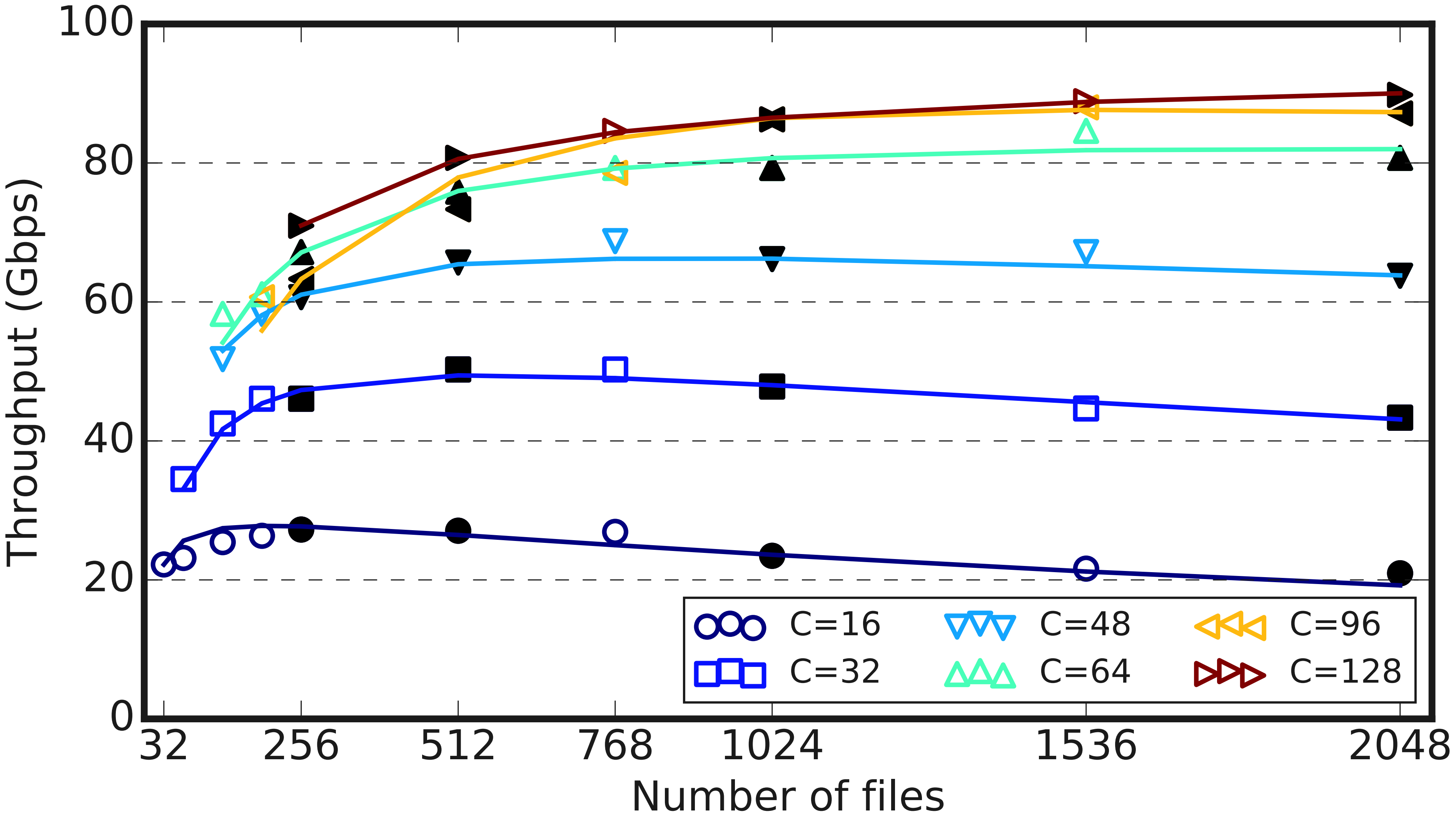
$$T = nT_{trs} + T_{ck} + b_{trs} = n(a_{trs}x + b_{trs}) + a_{ck}\, x + b_{ck} + b_{trs}$$

- S – Total bytes, N – Total files, cc – concurrency; $x = S/N$, $n = N/cc$

- The transfer time T to transfer all N files

$$T\,(N) = S/cc * a_{trs}x + N/cc * b_{trs} + S/N * a_{ck}\, x + b_{ck} + b_{trs}$$

Evaluation of the model



Conclusion

- Our experiences in our attempts to transfer one petabyte of science data within one day
- Exploration to identify parameter values that yield maximum performance for Globus transfers
- Experiences in transferring data while the data are produced by the simulation
  - Both with and without end-to-end integrity verification
- Achieved 99.8% of our one petabyte-per-day goal without integrity verification and 78% with integrity verification
- Finally, we used a model-based approach to identify the optimal file size for transfers
  - Achieve 97% of our goal with integrity verification by choosing the appropriate file size
- A useful lesson in the time-constrained transfer of large datasets.

Questions