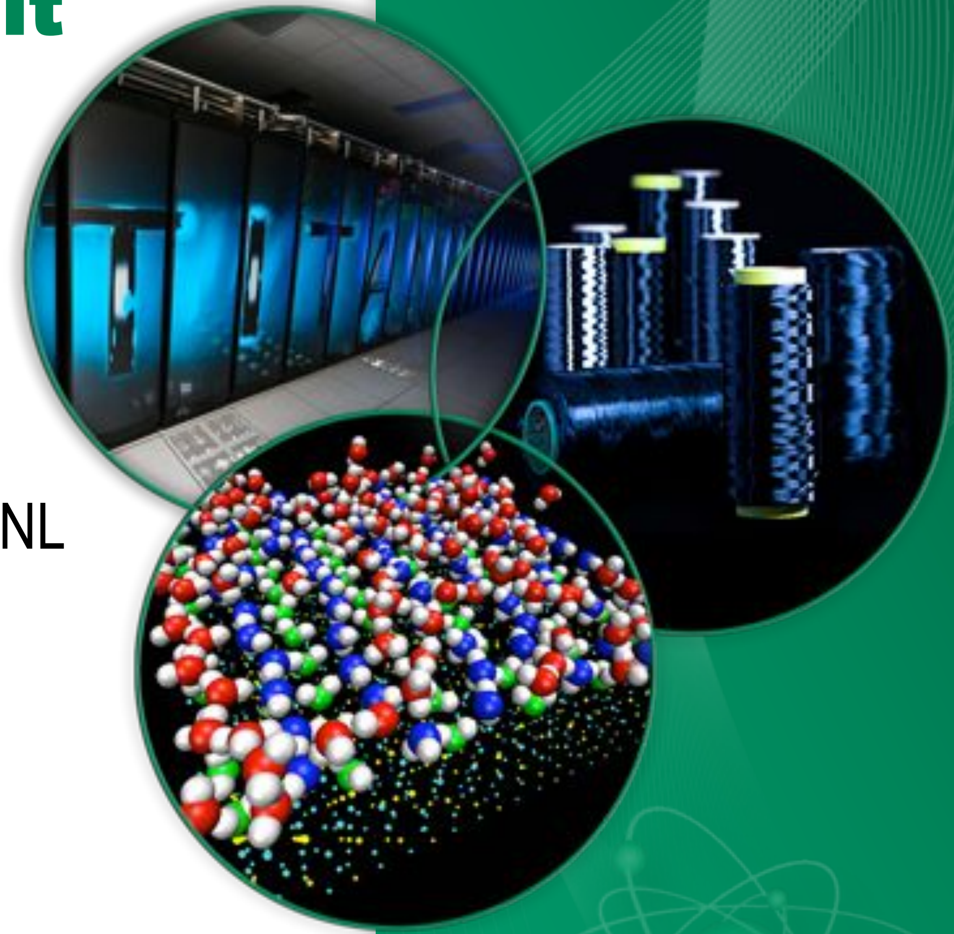
# An Adaptive End-to-End Approach for Terabit Data Movement Optimization

**Galen Shipman,** Scott Atchley,
Youngjae Kim, Geoffroy Vallee - ORNL

**George Bosilca**, Thomas Herault,
Stephanie Moreaud - UTK



# Leading research requires the use of large-scale resources across DOE

- Computational facilities – ALCF, NERSC, and OLCF
- BES facilities – SNS/Neutrons, APS/hard x-rays, ALS/soft x-rays, LCLS/coherent beams, many others..
- BER data centers – ARM, CDIAC, CMIP-5 (ESGF), many others..
- Multiple petabytes stored across these geographically distributed resources today, quickly approaching an Exabyte
- Many examples of coupling data today:
  - Experimental data from SNS coupled to models run at the OLCF & NERSC
  - Climate simulations run at ALCF and OLCF validated with BER data sets at ORNL data centers
  - Real-time visualization of tomography data taken at APS

# The Problem Space

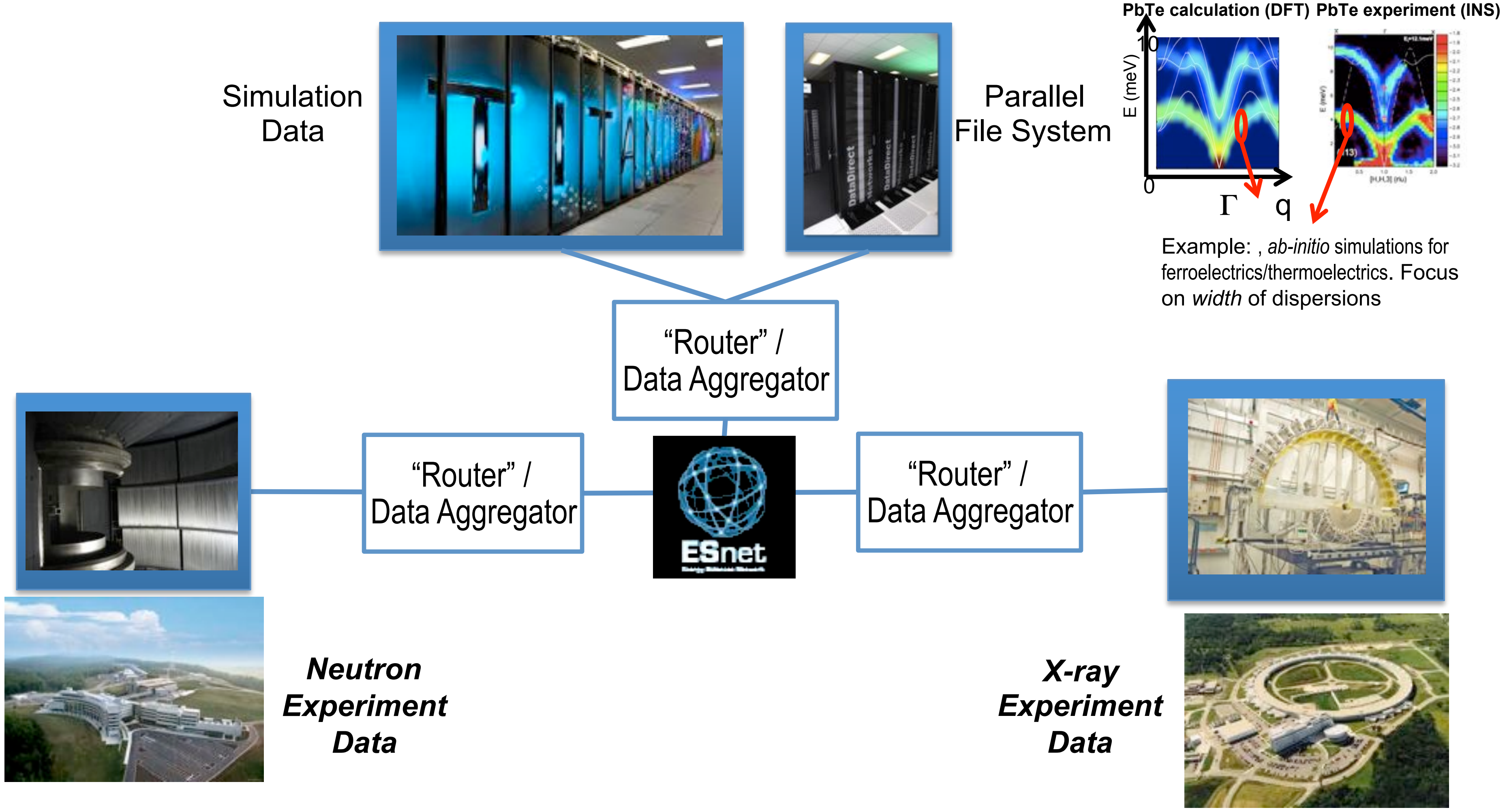
- Heterogeneous network infrastructure
  - Different fabrics for IPC, SAN, LAN, and WAN
- Variety of data sources
  - Difficult to optimize performance for data movement
- Competing workloads
  - Data movement to a remote site could choke out a 300,000 core simulation

# Our Work

- A portable high-performance network overlay transport - **CCI**
  - single API for network communication
  - optimized for the WAN
- An optimized I/O scheduling infrastructure – **Z Scheduler**
  - Optimized for common file systems, parallel and single system
- A workload aware scheduling infrastructure – **Z Scheduler**
  - Arbitrates competing workloads on the end-to-end path

# Our Vision

- *An optimized end-to-end data path to couple facilities*



Simulation Data

Parallel File System

Example: , *ab-initio* simulations for ferroelectrics/thermoelectrics. Focus on *width* of dispersions

"Router" / Data Aggregator

"Router" / Data Aggregator

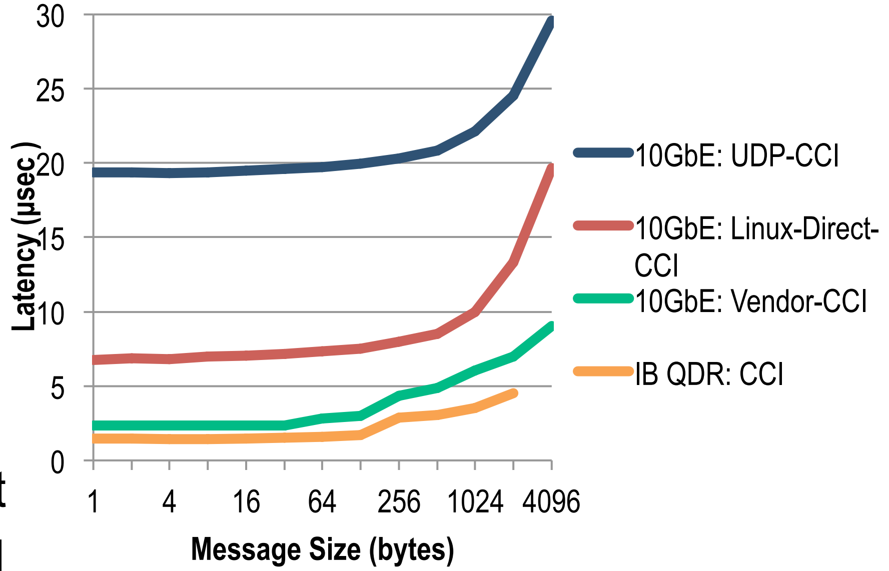
"Router" / Data Aggregator

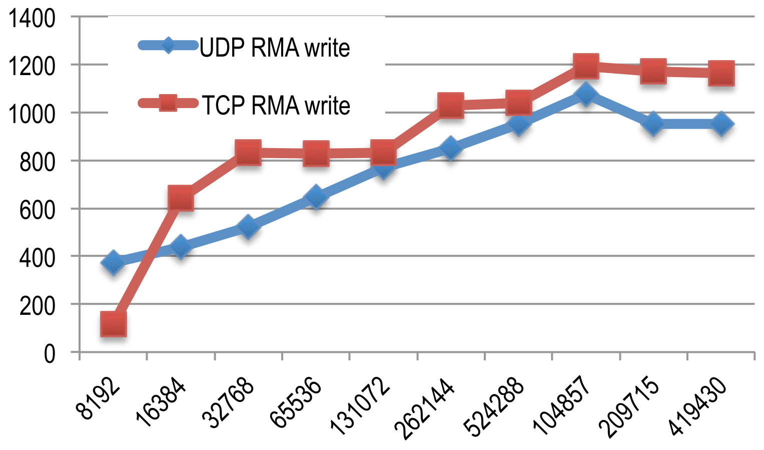
***Neutron Experiment Data***

***X-ray Experiment Data***

# Common Communication Interface

- What is CCI?
  - A generic, communication abstraction layer
    - Platform for experimentation and production
  - Supports a number of networks (SAN, LAN, WAN)
    - InfiniBand, Gemini, Ethernet, etc.
- New features
  - Completed initial WAN optimized CCI transport
  - Adding support for "virtual fabric support" (CCI Routed)
    - Provides end-to-end virtual fabric over heterogeneous networks via CCI over CCI!
- CCI is now a native transport within Open MPI
  - Allows quick prototyping and testing of complex use-cases.



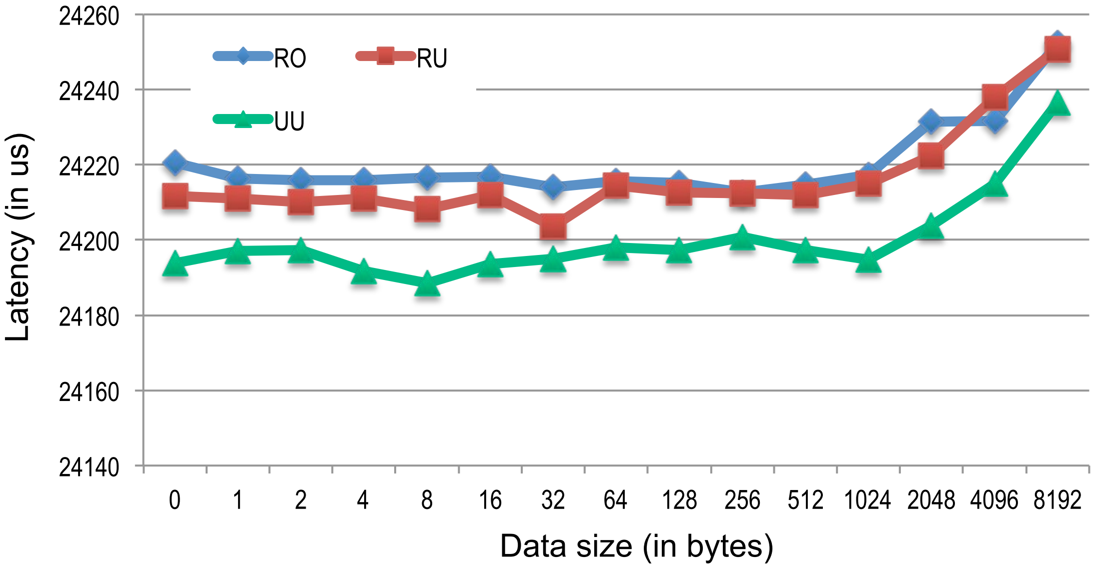


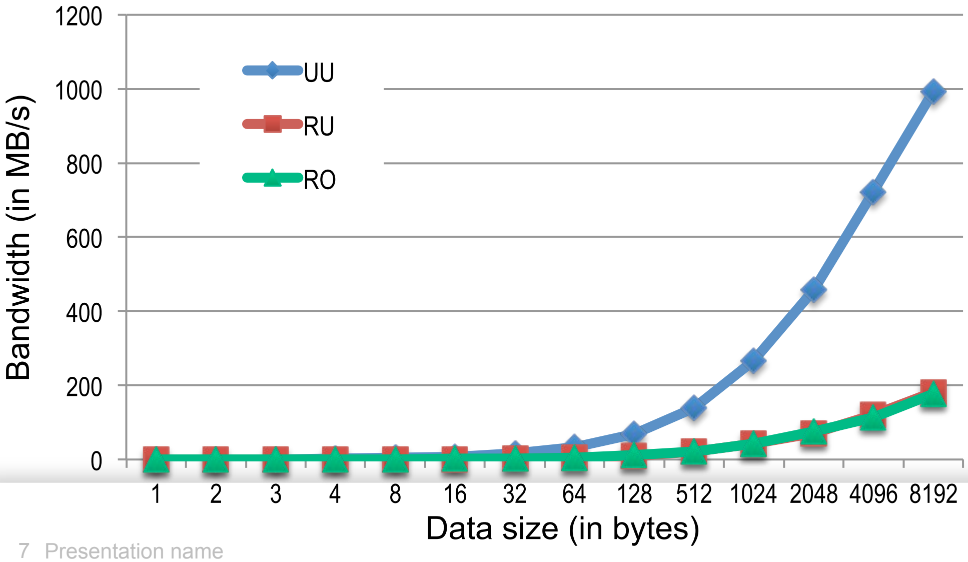
# Experimentation

- CCI WAN Evaluation
- ESnet 100G testbed – DOE Advanced Networking Initiative (ANI)
  - Results based on communications over ANI
  - 10 Gb Ethernet Myricom cards
  - 9000 MTU

# Small messages

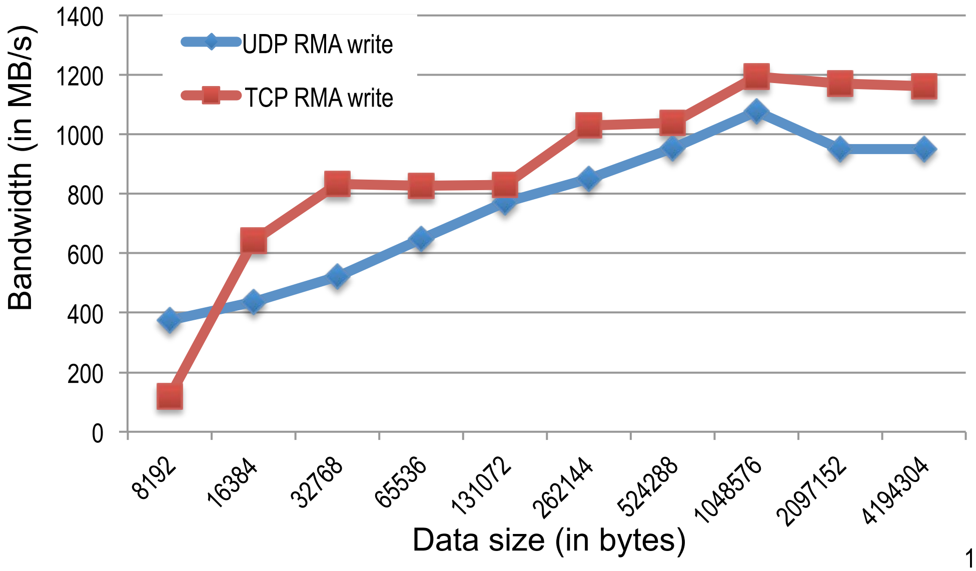
- UDP transport
- Pingpong latency
- Reliable-ordered (RO), reliable-unordered (RU), unreliable-unordered (UU)





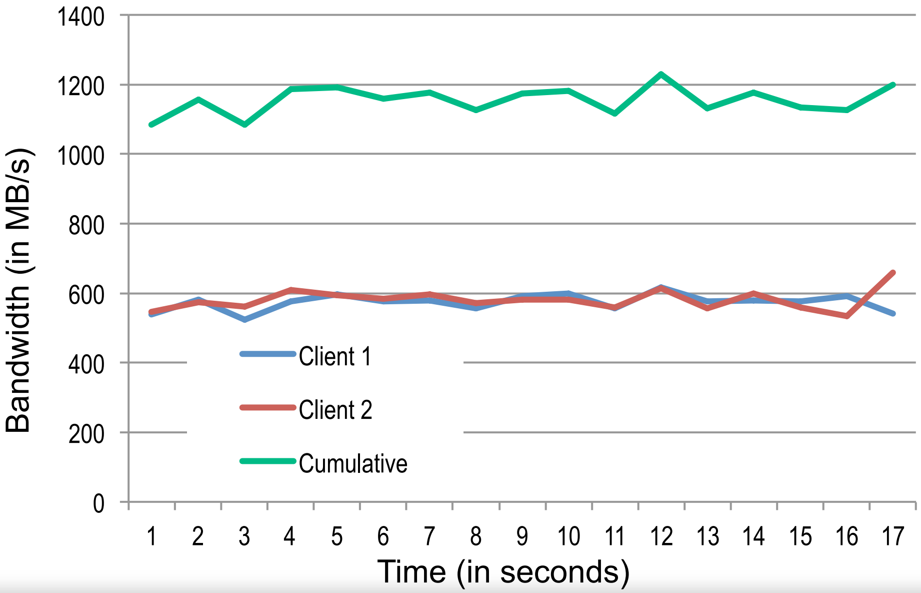
- UDP transport
- Stream between two endpoints
- Near line rate bandwidth with no reliability

# RMA operations



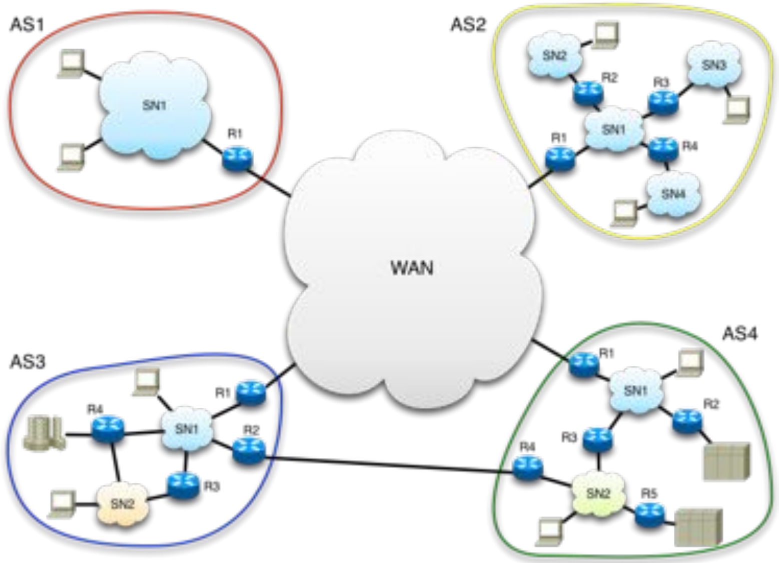
- Pipeline of RMA write operations
- One-to-one operation
- Near line rate bandwidth with data size > 10MB

- CCI Semantics allow servers to “self flow-control”
- UDP transport
- Many-to-one: 2 clients perform RMA write operations to a single target



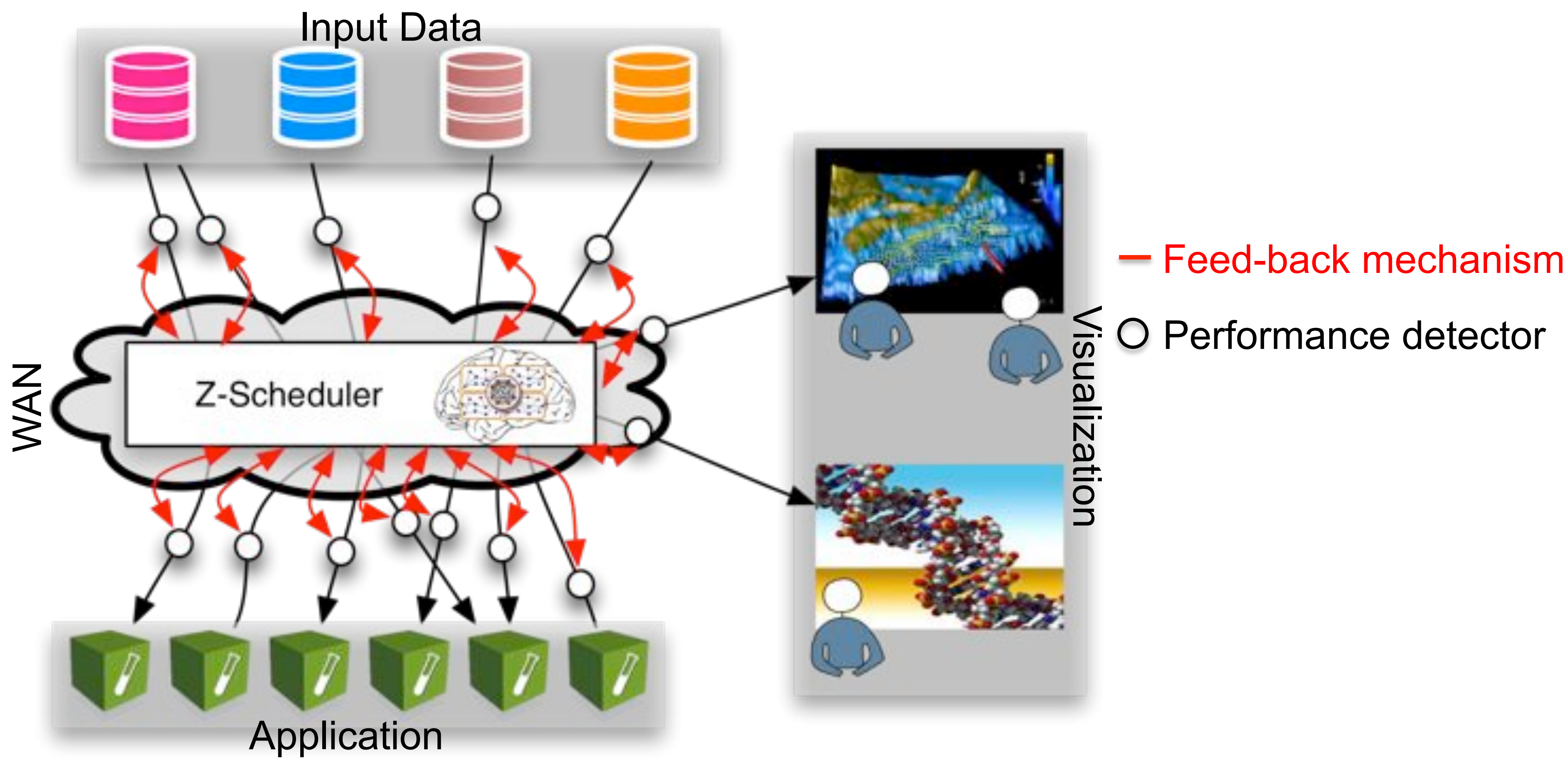
# Next steps - CCI Routing

- Allow routing across heterogeneous networks
  - InfiniBand, Cray Gemini, Ethernet
  - Within a site and between sites
- Same CCI interface
  - Messages and Remote Memory Access
- Enables best performance of network
- Routing Daemons
  - Bridge two or more networks
  - Route discovery and resources
  - Forwards MSGs and RMA
  - Multiple metrics (throughput, latency, hop count)
- Client routing transport
  - Manages end-to-end state
  - Uses native transports for communication
  - Load-balancing when multiple routers present



- Status
  - Design completed see http://cci-forum.com
  - Routing Daemon code in development
  - Next steps
    - Complete router discovery and mapping
    - End-to-end protocol
    - Client transport

# End-to-end Data Transfer Orchestration



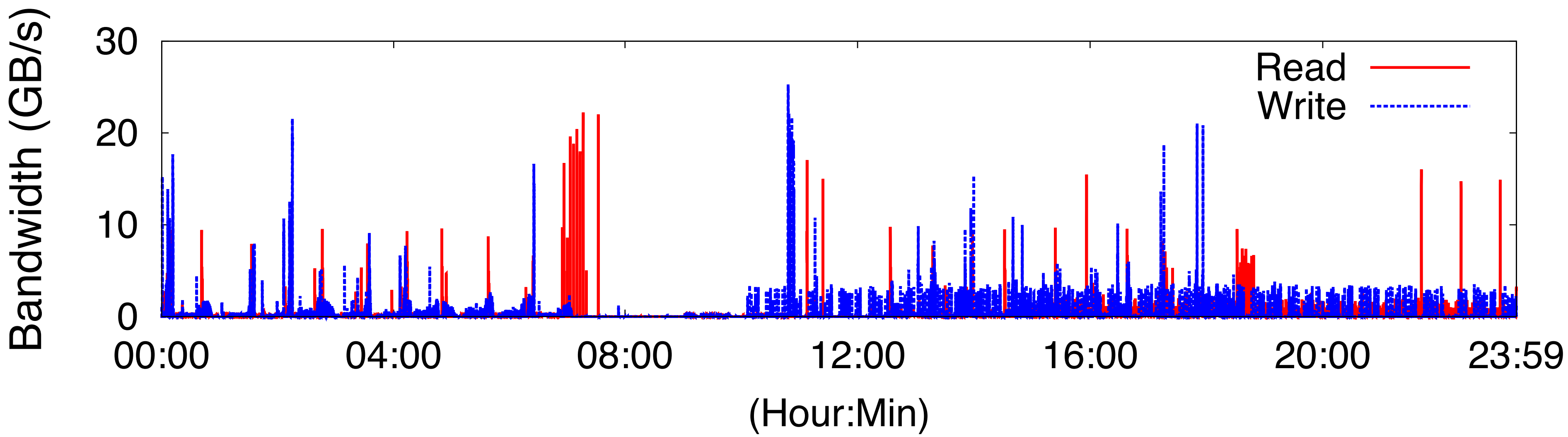
- Goal: Minimize the end-to-end transfer time
- Online predict the file system and network capacity and adapt the bandwidth

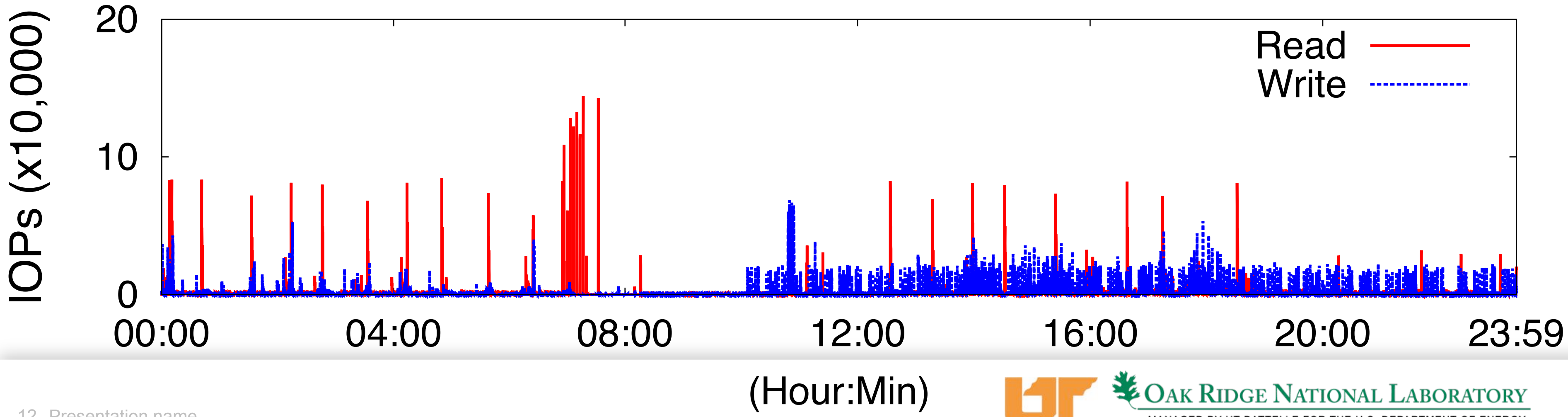
# I/O (File System) Scheduling

- Optimize IO accesses for parallel file systems
  - Investigate strategies for minimizing impact of intermittent congestion on storage servers
  - Require different strategies for read vs write, and large vs small files
  - Minimize accesses to the metadata server(s)
- Design Goal
  - Exploit I/O parallelism in Parallel File Systems
  - Adapt I/O loads on storage servers to minimize the impact of I/O congestion on disks and other systems (Titan)
- Disk layout-aware algorithm
  - Source Algorithm
    - Segment files into chunks (stripe unit in PFS)
    - Spawn multiple threads for I/Os
    - Use *pread()* to read specific chunks of a file
    - Schedule chunks based on congestion “feedback”
  - Sink Algorithm
    - Each “disk/lun” has a separate queue
    - Maintain a thread pool, and threads are engaged to work on uncongested OSTs
    - Avoid congested storage servers

# Jan 30, 2013

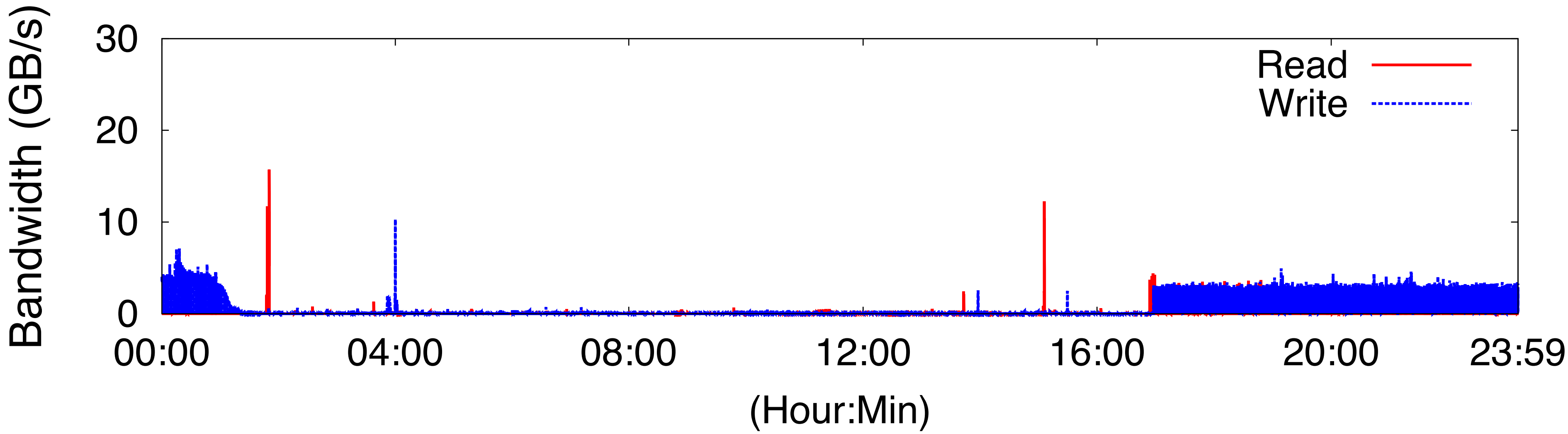
- I/O usages in Bandwidth (GB/s) and in IOPs for Jan 30, 2013 (for 24 hours)

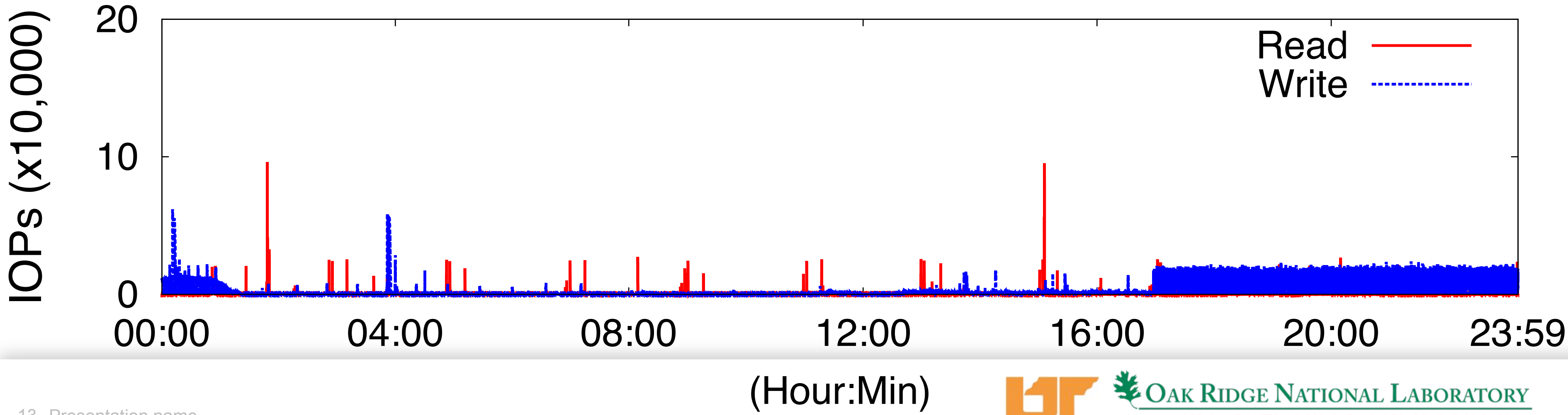




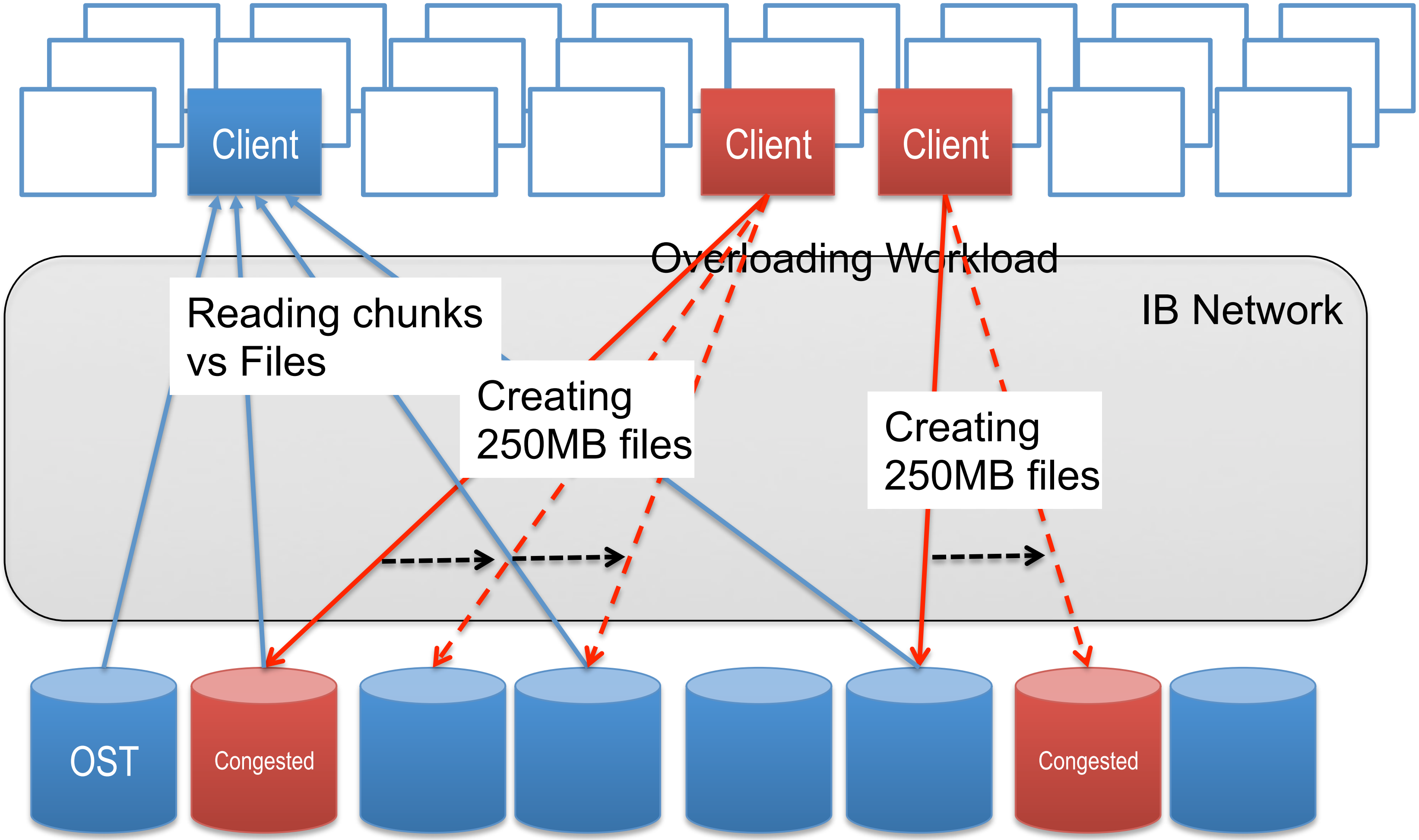
# Feb 18, 2013

- I/O usages in Bandwidth (GB/s) and in IOPs for Feb 18, 2013 (for 24 hours)





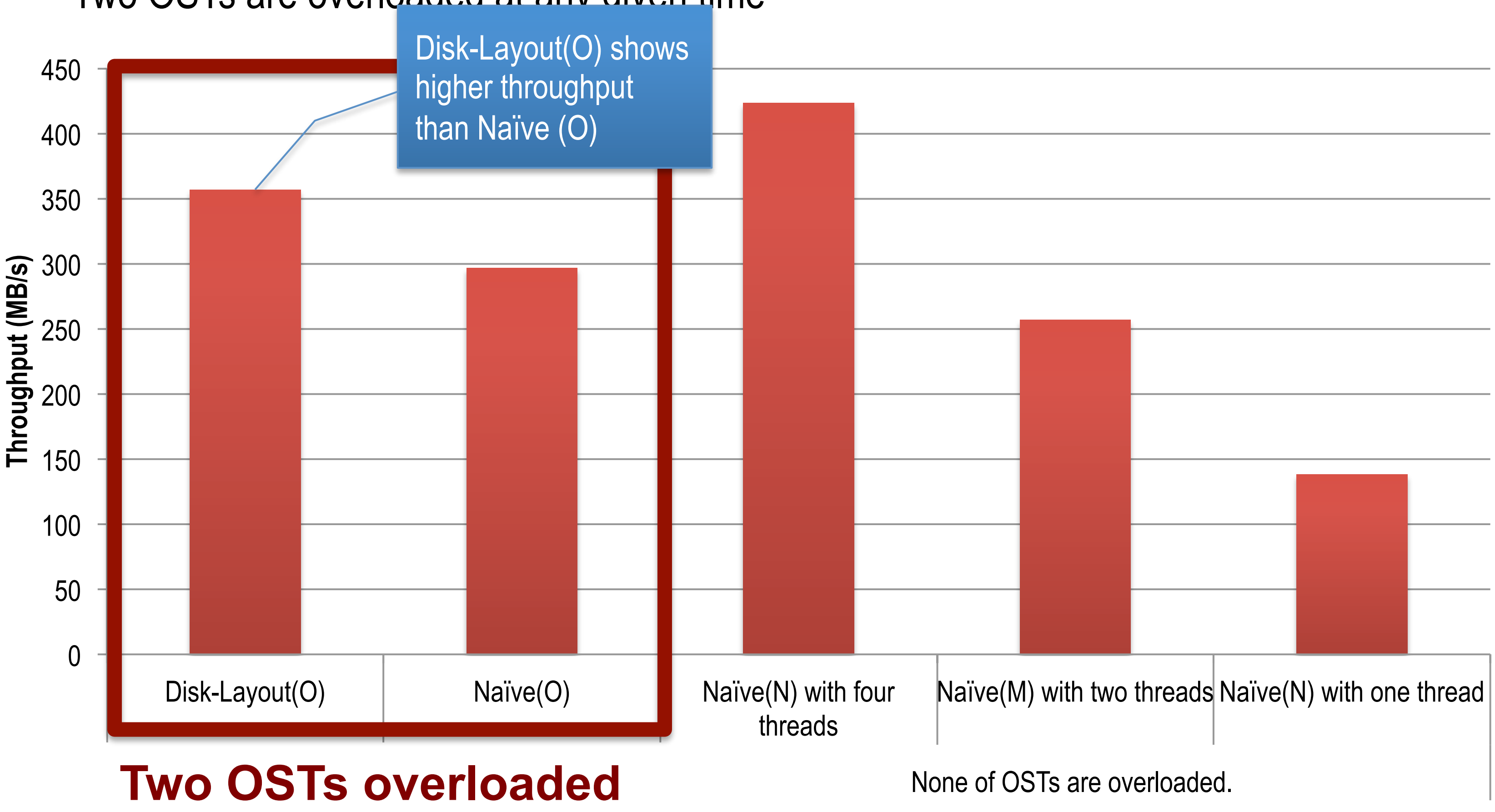
# Test Environment



- OST= RAID-0 of 3 HDDs

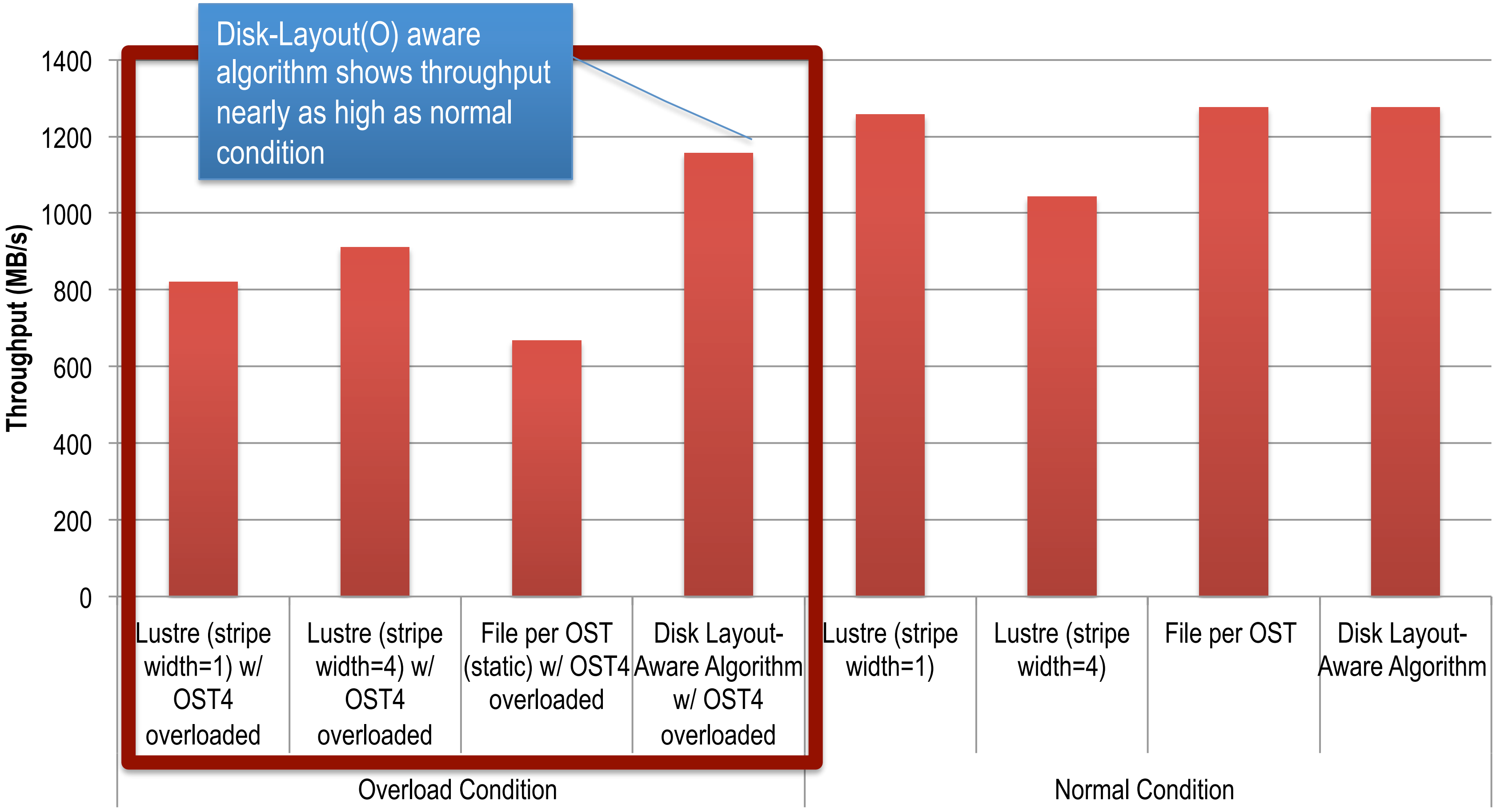
# Evaluating Source Algorithm

- Disk Layout-Aware Algorithm vs. Naïve
  - Four I/O threads reading 20MB files, and Eight OSTs for Lustre
  - Two OSTs are overloaded at any given time



# Evaluating Sink Algorithm

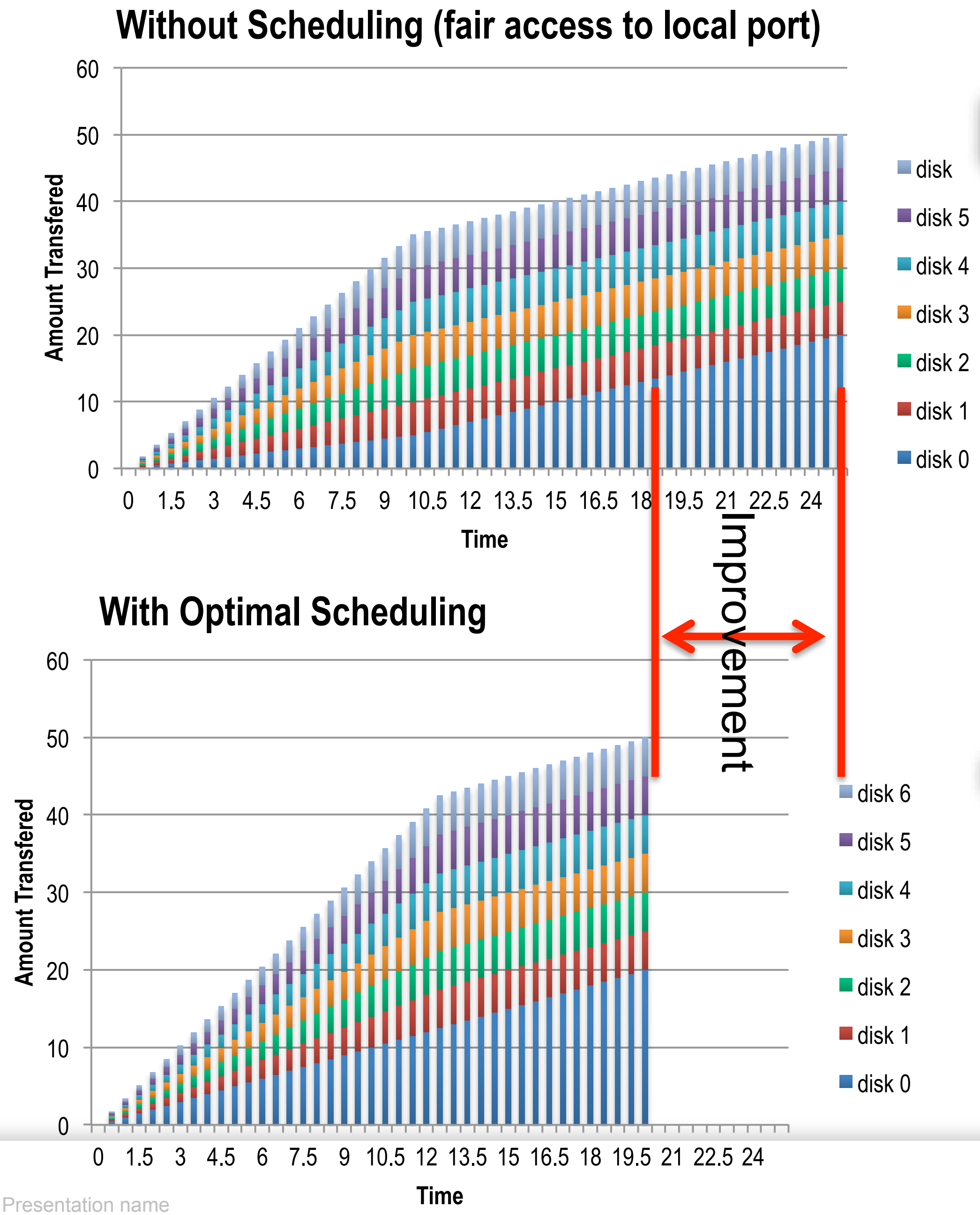
- Disk Layout-aware Algorithm vs. Naïve access
  - Eight I/O threads writing 10MB files, and Eight OSTs for Lustre
  - Only OST4 is overloaded



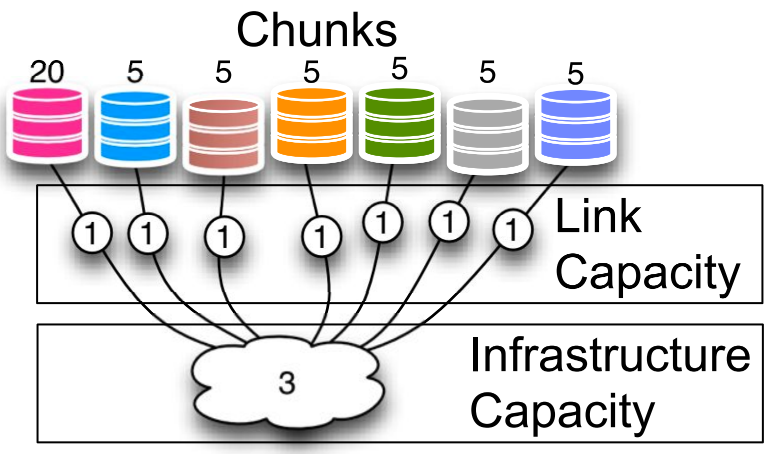
Disk-Layout(O) aware algorithm shows throughput nearly as high as normal condition

**One OST overloaded**

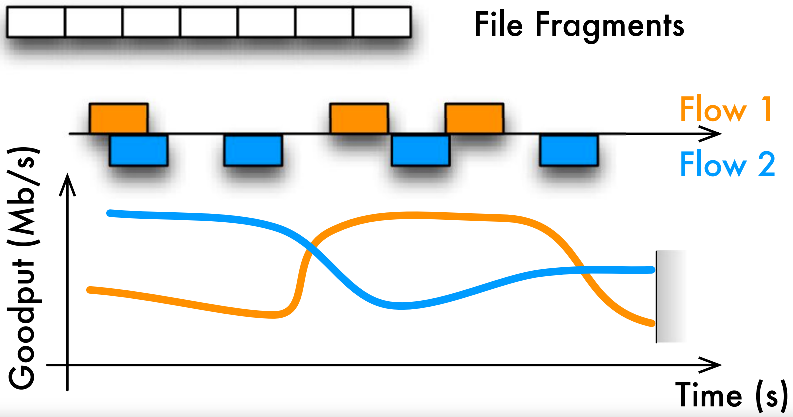
# Towards Orchestrated Scheduling

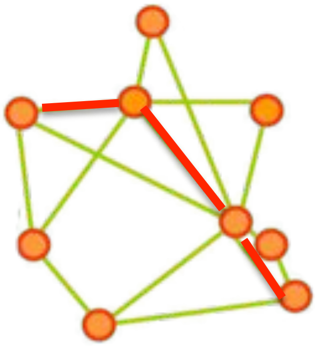

Without Scheduling (fair access to local port)

With Optimal Scheduling

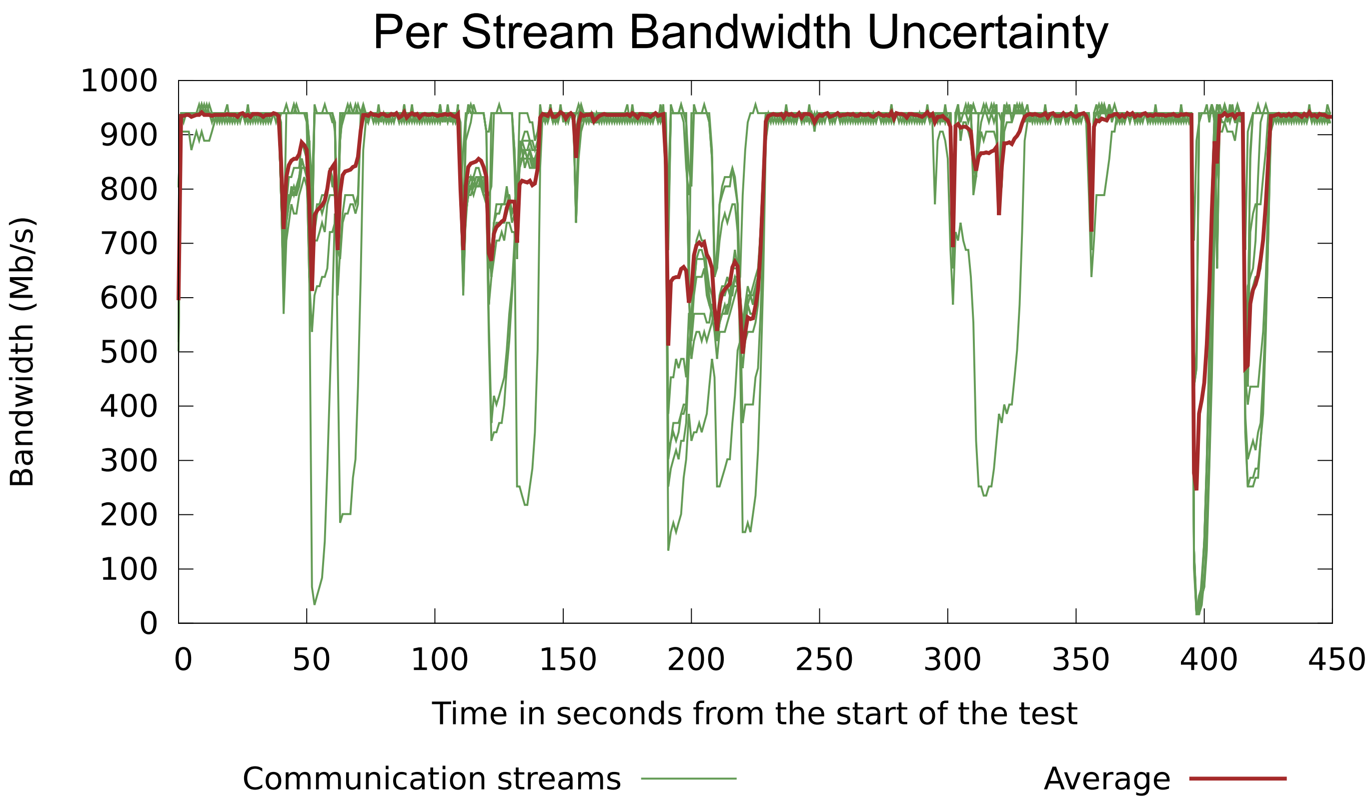


Scheduling may limit the goodput based on the amount of data to be transferred per link, not only on the total available bandwidth

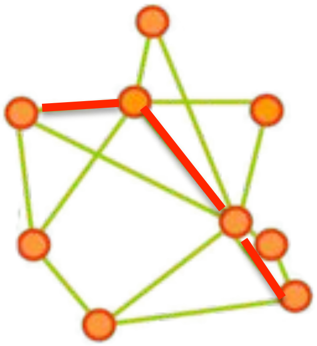




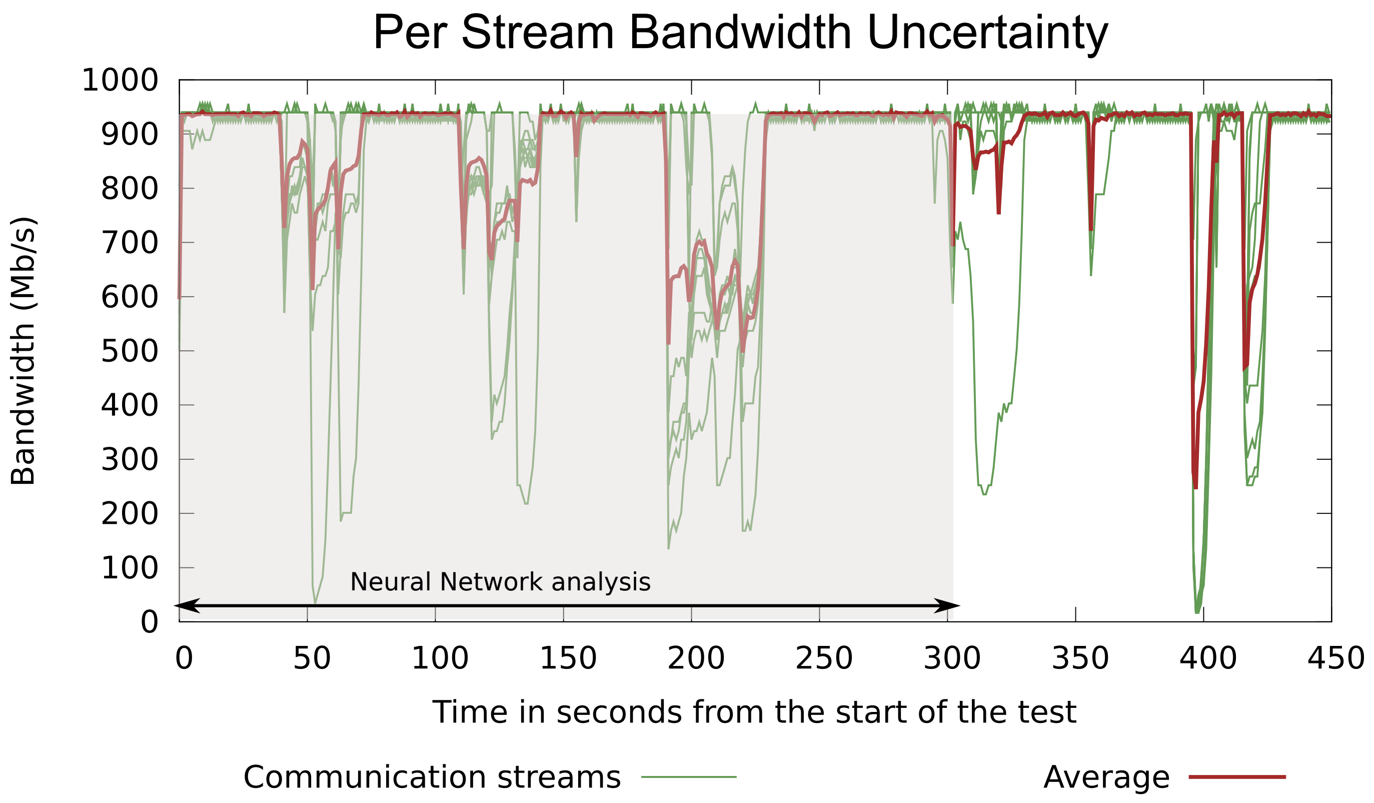
# WAN: TCP bandwidth under stress



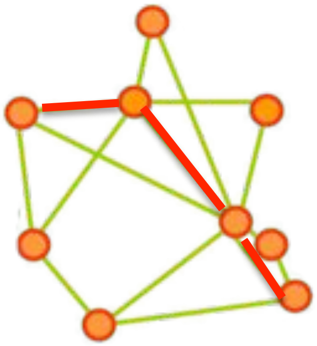
Under intense bandwidth stress, random data streams will be affected, and their potential bandwidth drastically decreased (no fairness guaranteed)



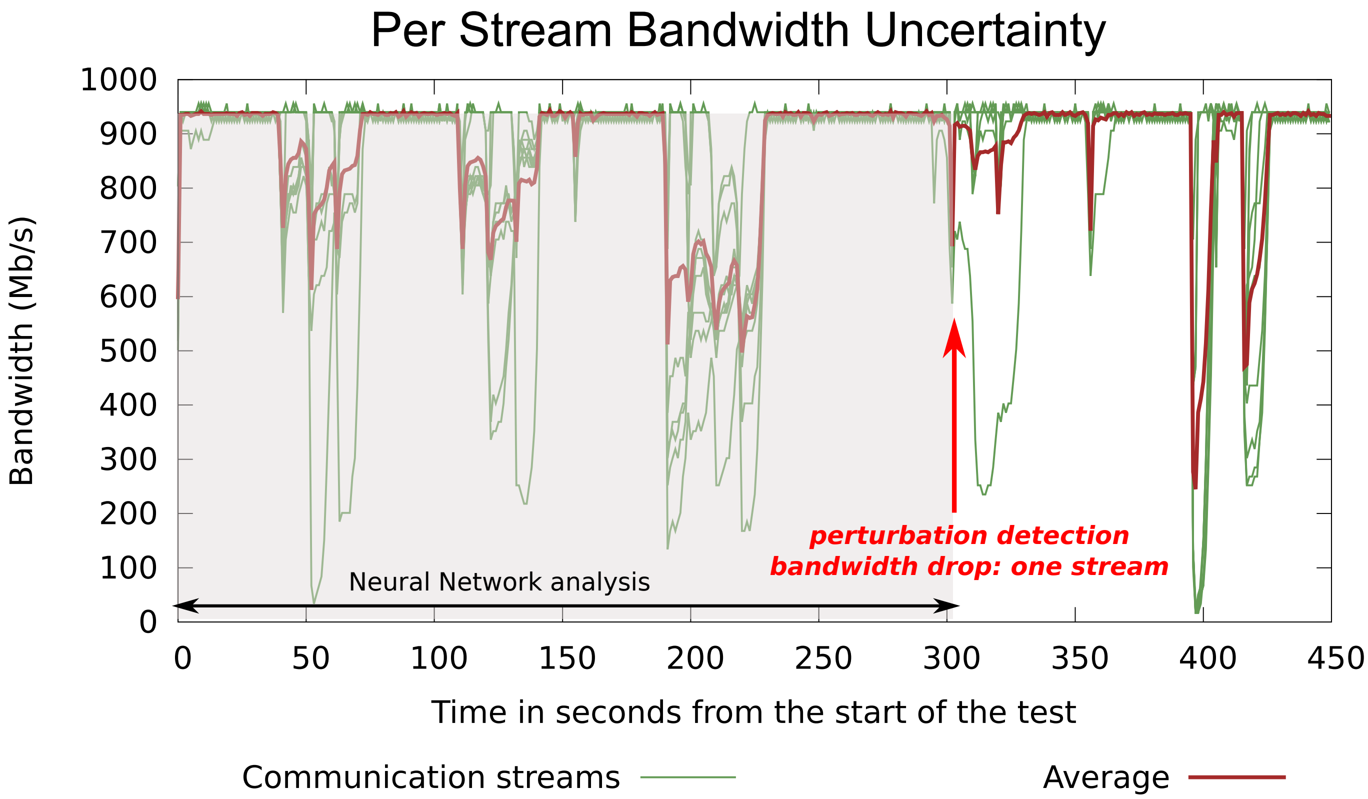
# WAN: TCP bandwidth under stress



Performance feed-back mechanism is used to learn about specific classes of perturbations and their impact (duration and weight) on the data transfers.

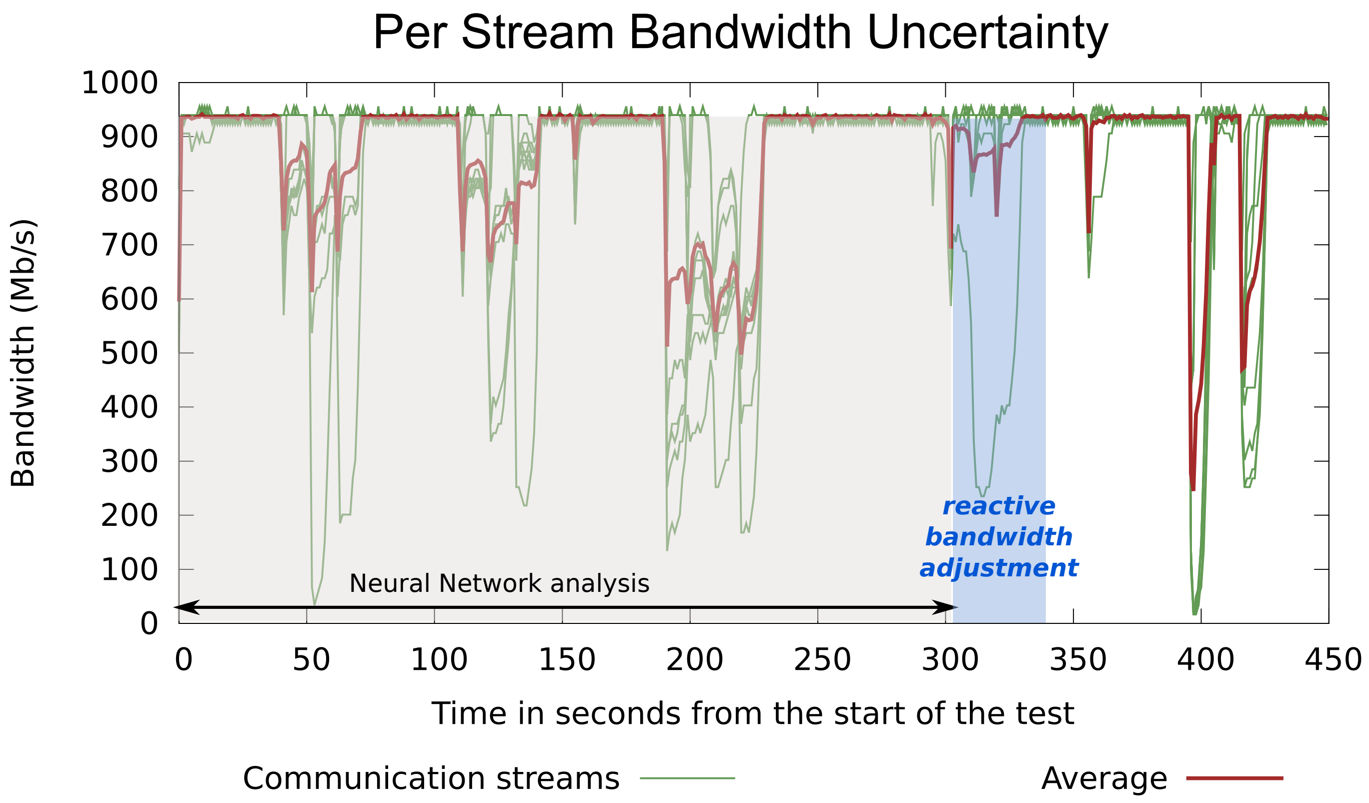


# WAN: TCP bandwidth under stress


Per Stream Bandwidth Uncertainty

New perturbations will be predicted and classified. The training will continue during the entire transfer.  Traces will be used to predict characteristics of the traffic based on temporal properties.

# WAN: TCP bandwidth under stress

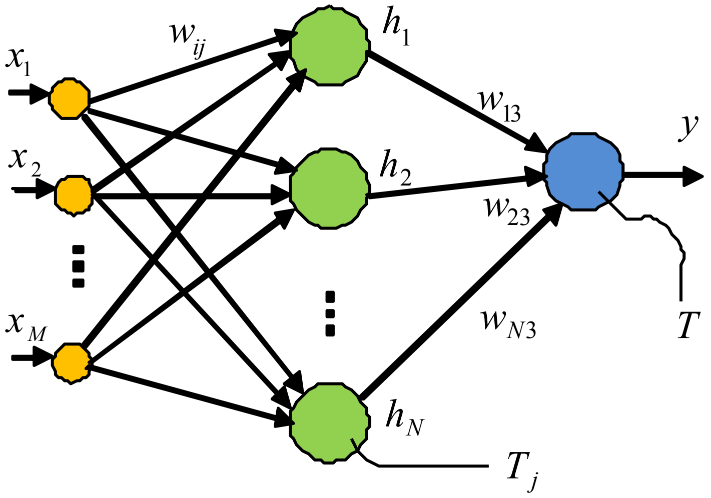


The bandwidth will be dynamically adjusted to satisfy the requirement determined by the Z-scheduler.

# Neural Networks

- excellent abilities to model difficult nonlinear systems
- very good alternative to traditional methods for solving complex problems in many fields, including image processing, pattern recognition, robotics, etc
- solvers of intelligent tasks - prediction, classification, recognition, optimization, etc
- good generalized properties, self-adaptation allow considering NNs as universal tools

**Multi-layer perceptron N-M-1**



Output value of the perceptron

$$y = F_3\left(\sum_{j=1}^{N} w_{j3}\left(F_2\left(\sum_{i=1}^{M} w_{ij}x_i - T_j\right)\right) - T\right)$$

Sigmoid activation function of hidden layer neurons

$$F(x) = \frac{1}{1+e^{-x}}$$

Sum-Squared Error

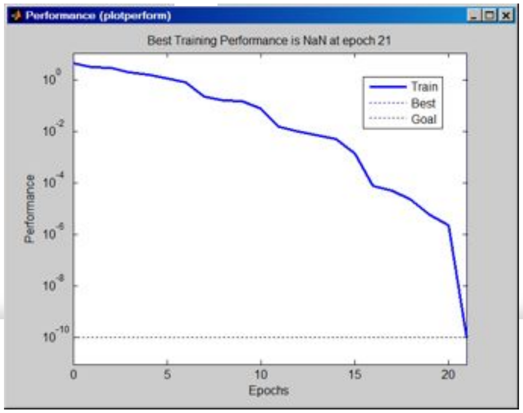
$$E = \frac{1}{2}(y_O^k - d_O^k)^2$$

Goal of training: $E(t) \rightarrow \min$

Modification of weights and thresholds

$$\Delta w_{ij}(t) = -\alpha \frac{\partial E^p(t)}{\partial w_{ij}(t)} \qquad \Delta T_j(t) = -\alpha \frac{\partial E^p(t)}{\partial T_j(t)}$$

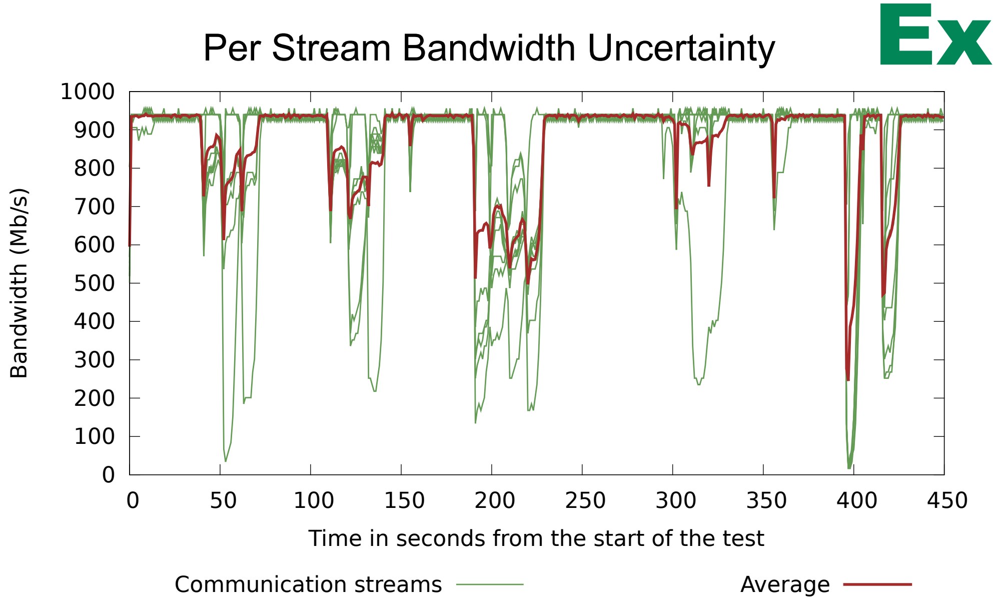
Training process over time: decreasing the Sum-Squared Error


Performance (plotperform)

Best Training Performance is NaN at epoch 21

The training algorithms: Back Propagation Error and Multiple Propagation Error

# Experimental results



Input data : 5 events with 30 sec and 60 sec duration, total of 10 events

Input data : 103 vectors, 63 for the training and 40 for the testing

Ideal result of the NN: to classify types of the events only (5 events) and type & duration of the events (10 events) with a detection rate **100%** ON the first timestamp (on the THIRD SECOND from the START of the EVENT)

Architecture of NN: Multi-layer perceptron, 10 input, 10 hidden and 1 output neuron, sigmoid activation

Training of NN: Back-propagation error, SSE=0.048, 400000 training iterations, moving simulation mode, 200000 training iteration of re-training, 14 seconds of re-training on Inter Core2

Results on the training set (63 vectors):

- detection rate to classify only type of the event (5 events p1-p5) - **93,7%**
- detection rate to classify type&duration of the event both (10 events p1-p5 with 30 sec and 60 sec) - **73.1%**

Results on the testing set (40 vectors):

- detection rate to classify only type of the event (5 events p1-p5) - **40,0%**
- detection rate to classify type&duration of the event both (10 events p1-p5 with 30 sec and 60 sec) - **25.0%**

**Results showed good adaptability and good potential capabilities of NNs to solve this task**

# Future Work

- Investigate hierarchical storage awareness
  - How to utilize NVM devices in the orchestration framework?
  - Investigating the use of NVM devices for extended memory buffer space using *NVMAlloc* library
- Integrate statistics on I/O from Titan in the prediction framework
  - What predictions can be made on the application access pattern and on the file system?
  - Can we match the prediction of the application access pattern with the prediction about the network status?
  - Can we minimize the impact of data movement to large-scale simulations?
- Improve the detection rate of the classification by the Neuron Network
- Complete the implementation of the stream middleware, allowing end-to-end traffic management

# Questions?

- Please see http://cci-forum.com/ for papers, presentation and source code related to this project

This work is sponsored in part by the Office of Advanced Scientific Computing Research (ASCR); U.S. Department of Energy. The work was performed in part at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. De-AC05-00OR22725.

This research used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

U.S. DEPARTMENT OF ENERGY

Office of Science
U.S. Department of Energy