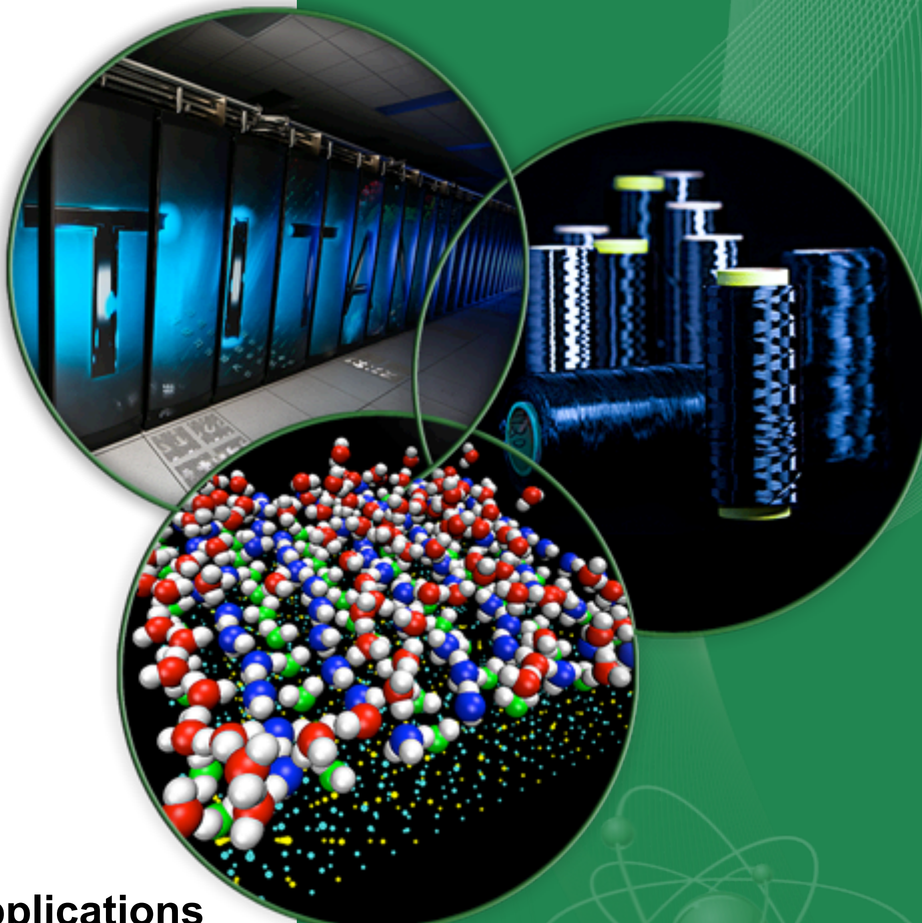
Oak Ridge National Laboratory

Computing and Computational Sciences Directorate

High-Performance Data Flows Using Analytical Models and Measurements



Nagi Rao, Qiang Liu
Oak Ridge National Laboratory

Don Towsley, Gayane Vardoyan
University of Massachusetts, Amherst

Raj Kettimuthu, Ian Foster
Argonne National Laboratory

Brad Settlemyer
Los Alamos National Laboratory

Workshop on
Modeling & Simulation of Systems and Applications
August 10-12, 2016, Seattle, Washington

Sponsored by
RAMSES Project, ASCR, DOE

Outline of Presentation

- Introduction
- Throughput Profiles and Traces
  - TCP, UDT measurements
- Throughput Model
  - Monotonicity and concavity
- File Transfer Rates
  - Lustre and xfs
- Conclusions

Dedicated Network Connections: Increasing Deployments

Dedicated connections becoming more available

- DOE OSCARS provides ESnet WAN connections
- Google SDN dedicated networks

Desirable Features

- dedicated capacity - no competing traffic
- low loss rates – no induced losses from "other" traffic
- no need for "graceful degradation" to accommodate other traffic

Expectations for data transport methods

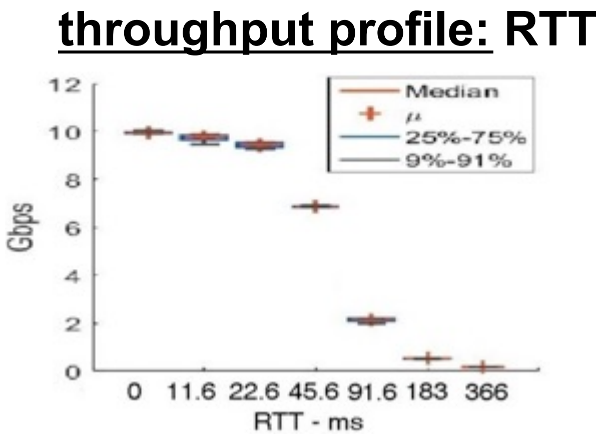
- peak throughput easier to achieve using "simple" flow control
- easier to tune parameters – due to predictable and simple dynamics

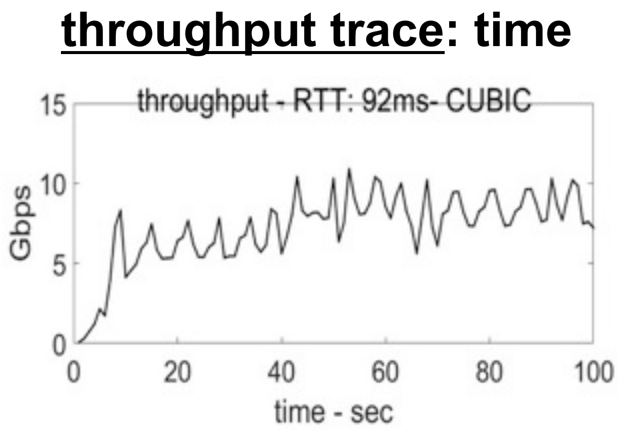
Scenarios:

- memory-to-memory, disk-to-disk, memory-to/from-disk transfers:
  - high bandwidth file transfers
  - data transfers between remote computations
- monitoring and control channels: low loss and jitter requirements
  - computational monitoring and steering
  - remote experimentation – computation controlled

Data Transfers Over Dedicated Network Connections

Performance of data transfer methods





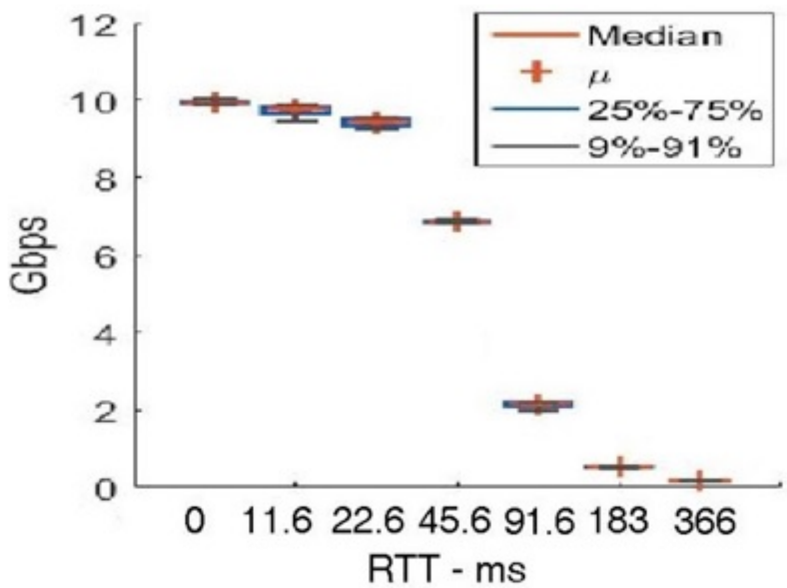
Network Transport Methods

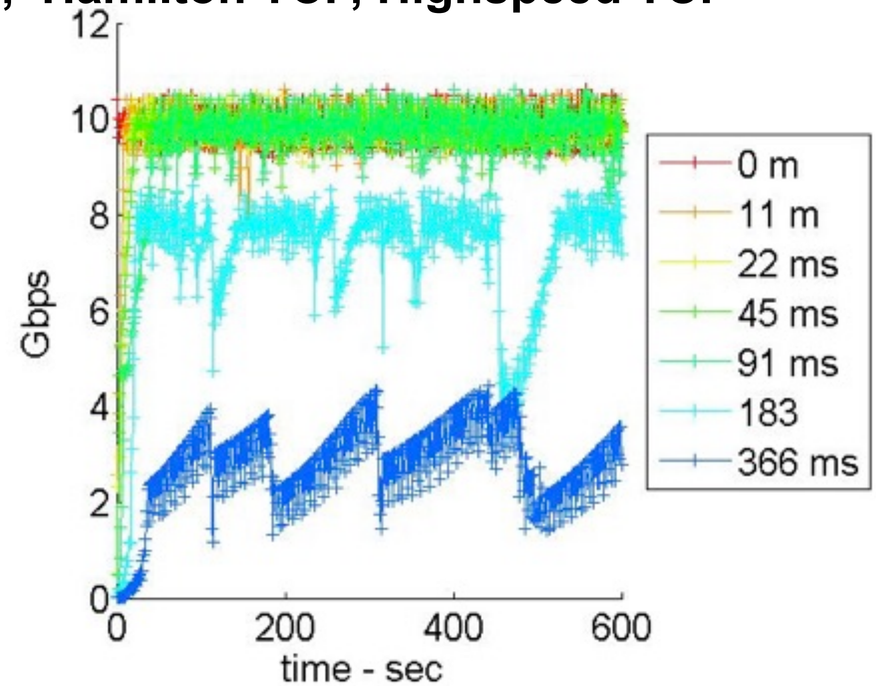
- TCP – widely deployed, including over dedicated connections
  - mechanism: slow-start followed by congestion avoidance
  - expected performance:
    - convex throughput profiles
    - slow-start followed by periodic trajectories
- UDT – UDP-based, particularly well-suited for dedicated connections
  - mechanism: ramp-up followed by stable flow rate
  - expected performance
    - flat throughput profile
    - ramp-up followed by constant trajectory
- ASPERA – commercial UDP-based transport method

OAK RIDGE National Laboratory

TCP Memory-to-Memory Throughput Measurements

Throughput profiles and traces: qualitatively similar across TCP variants CUBIC (Linux default), Scalable TCP, Hamilton TCP, Highspeed TCP





As expected:

- **profile: decreases with RTT;**
- **trace: sort of periodic in time**

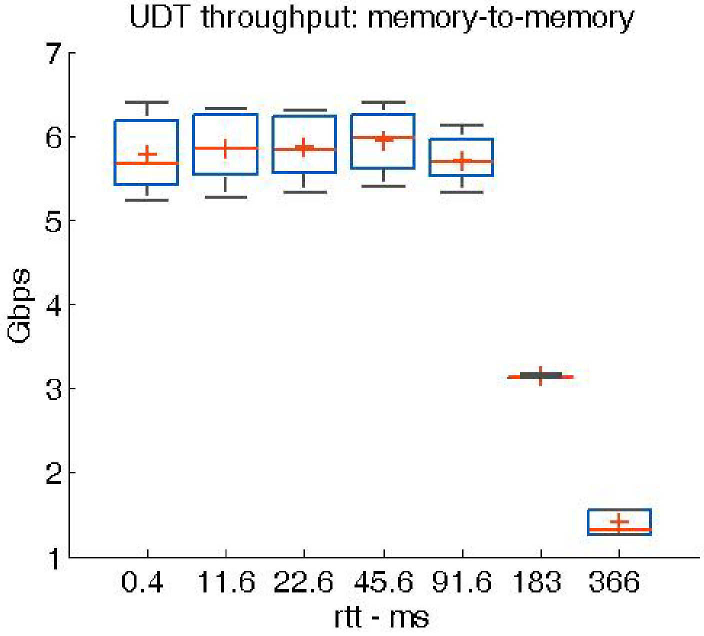
Additional characteristics:

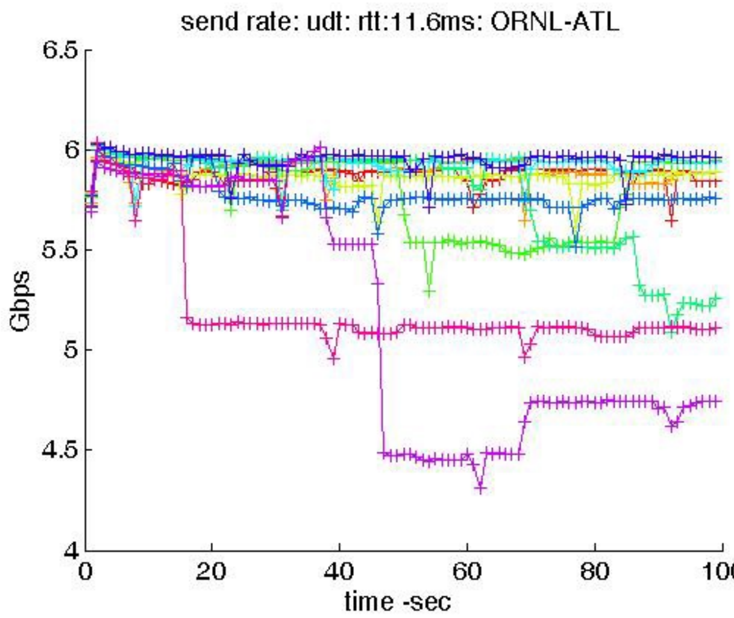
- **profile: concave at lower RTT**
- **trace:**
  - **significant variations**
  - **larger variation at higher RTT**

UDT Memory-to-Memory Throughput Measurements

Implements flow control and loss recovery over UDP

- **application-level – particularly suited for dedicated connections**





Analytical models indicate:

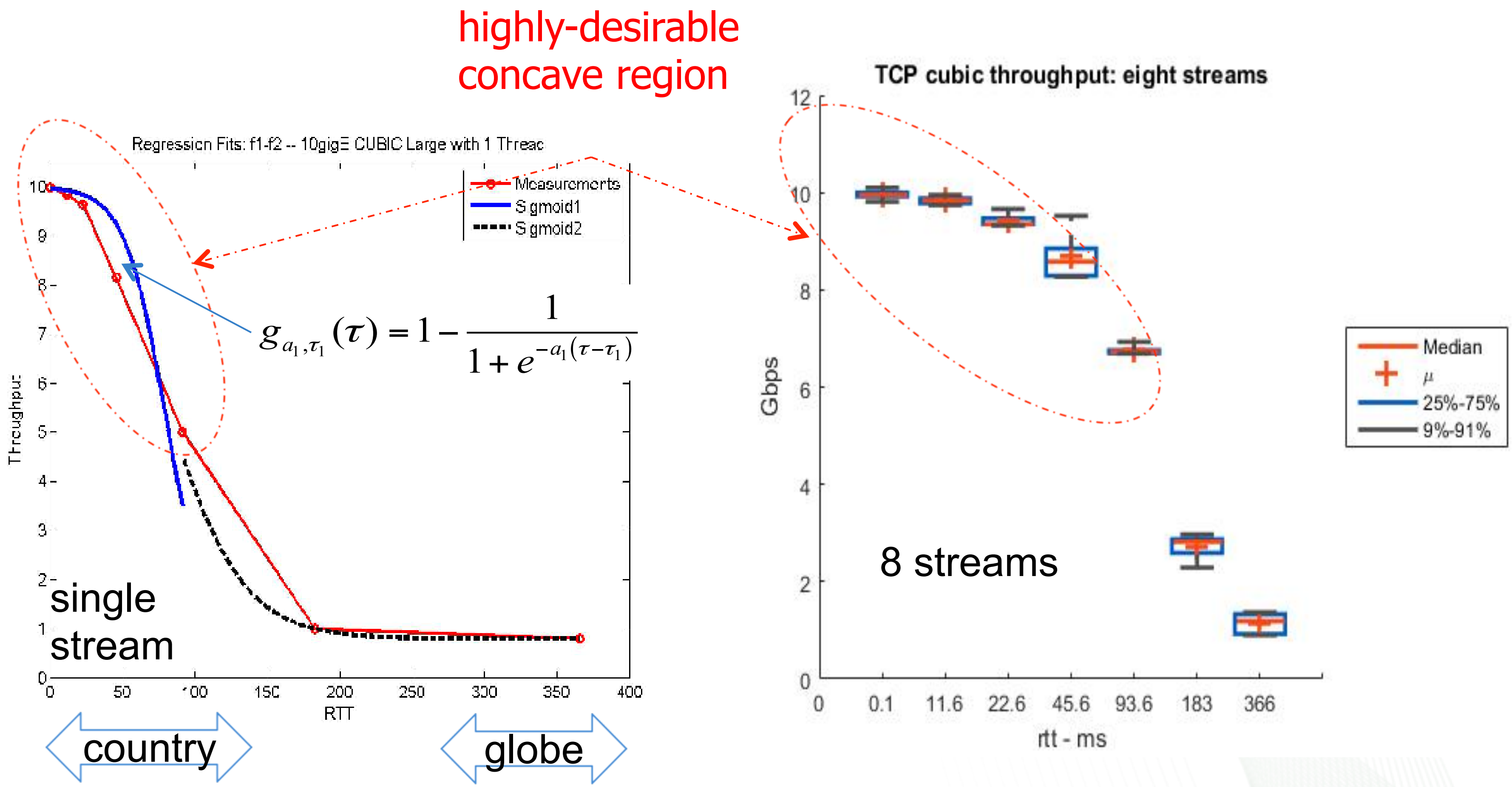
- **profile: flat with RTT**
- **trace: smooth rise and constant**

Measured characteristics:

- **profile: overall decrease with RTT**
- **trace:**
  - **significant variations: same RTT**
  - **repeated – significant drops**

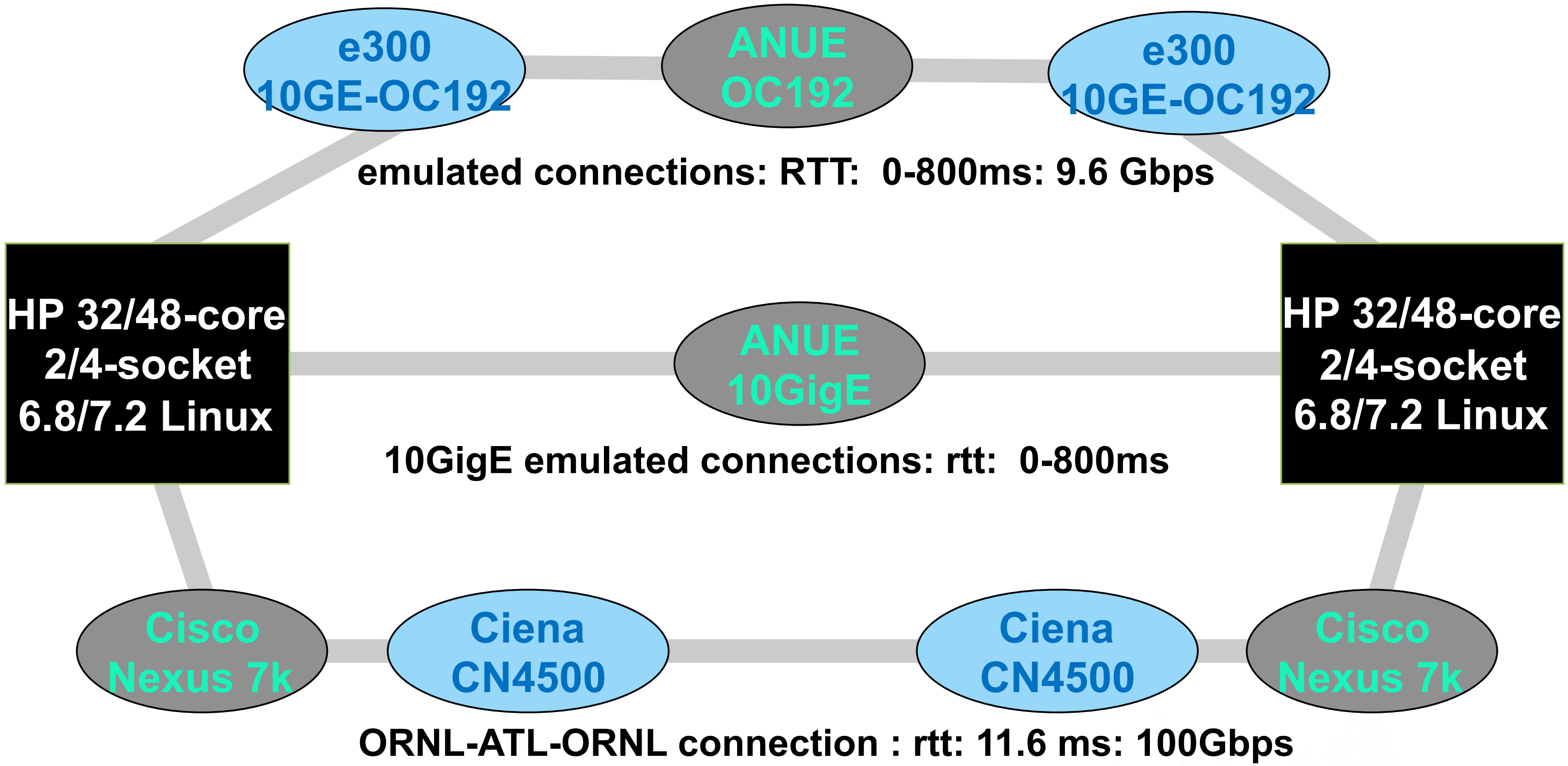
TCP Profiles: memory to memory transfer

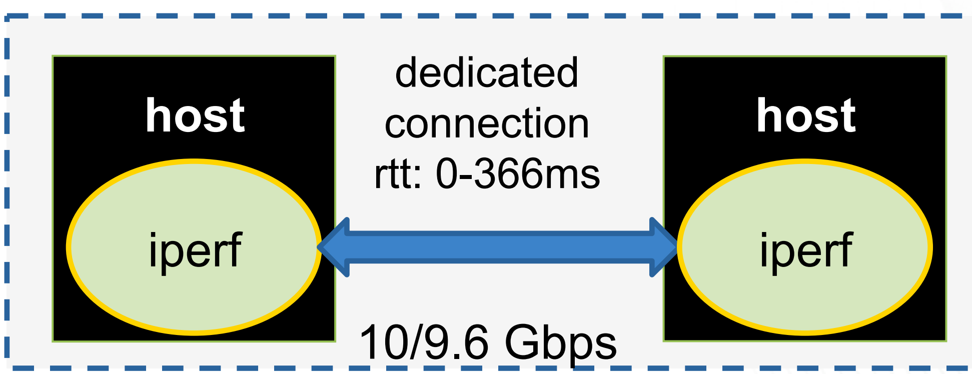
10Gbps dedicated connections: bohr03 and bohr04
- CUBIC congestion control module- default under Linux
- TCP buffers tuned for 200ms rtt: 1-10 parallel streams



RTT: cross-country (0-100ms), cross-continents (100-200ms), across globe(366ms)

ORNL Network Testbed (since 2004)





TCP Throughput Profiles

- Most common TCP throughput profile
  - convex function of rtt
  - example, Mathis et al (1997)

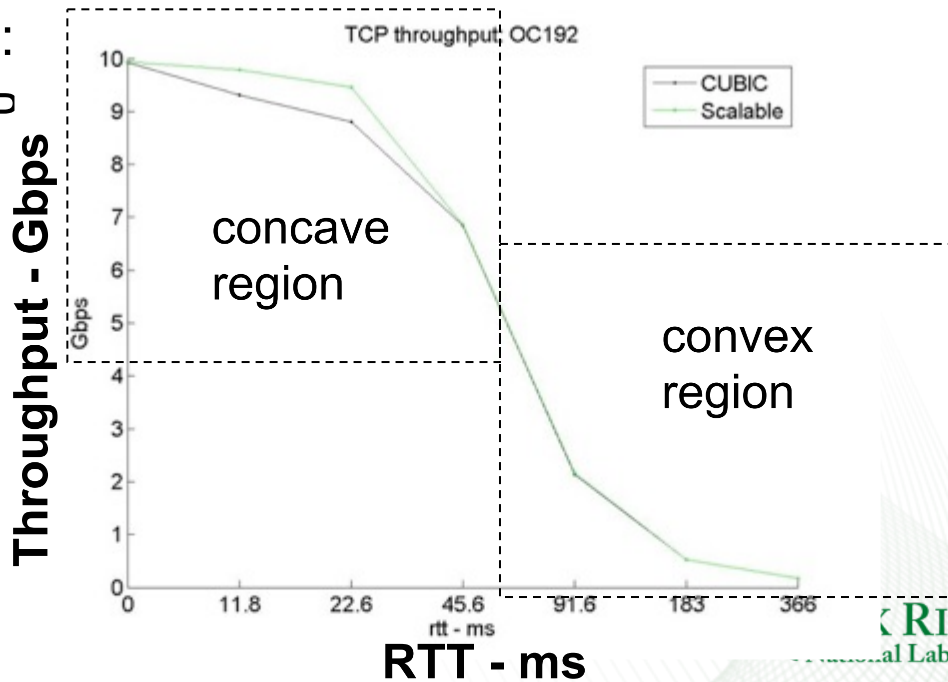
Throughput at rtt τ loss-rate p

$$T_M(\tau) = \frac{MSS * k}{\tau\sqrt{p}}$$

- Observed Dual-mode profiles:
  - CUBIC, STCP, HTCP, HS-TCP

Smaller RTT
- Concave region

Larger RTT
- Convex region



Desired Features of Concave Region

- Concave regions is very desirable
  - throughput does not decay as fast
  - rate of decrease slows down as rtt

Function $\mathrm{T}(\tau), \tau \in I$ is concave iff derivative $\frac{d\mathrm{T}}{d\tau}$ is non-increasing

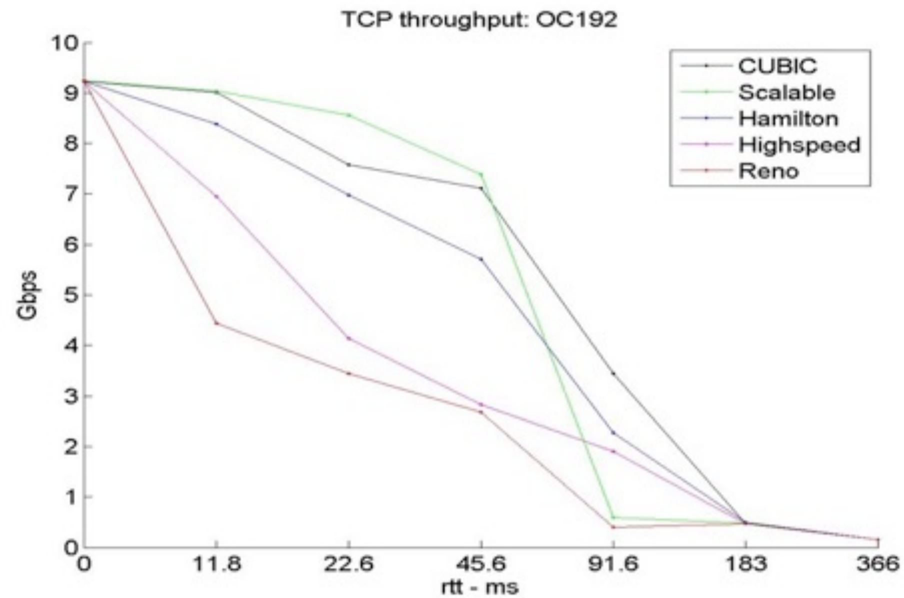
not satisfied by Mathis model:

$$\frac{d\mathrm{T}_M}{d\tau} = -\frac{C}{\tau^2\sqrt{p}}$$

- Measurements: throughput profiles for rtt: 0-366ms
  - Concavity: small rtt region

    Only some TCP versions:
    - CUBIC,
    - Hamilton TCP
    - Scalable TCP

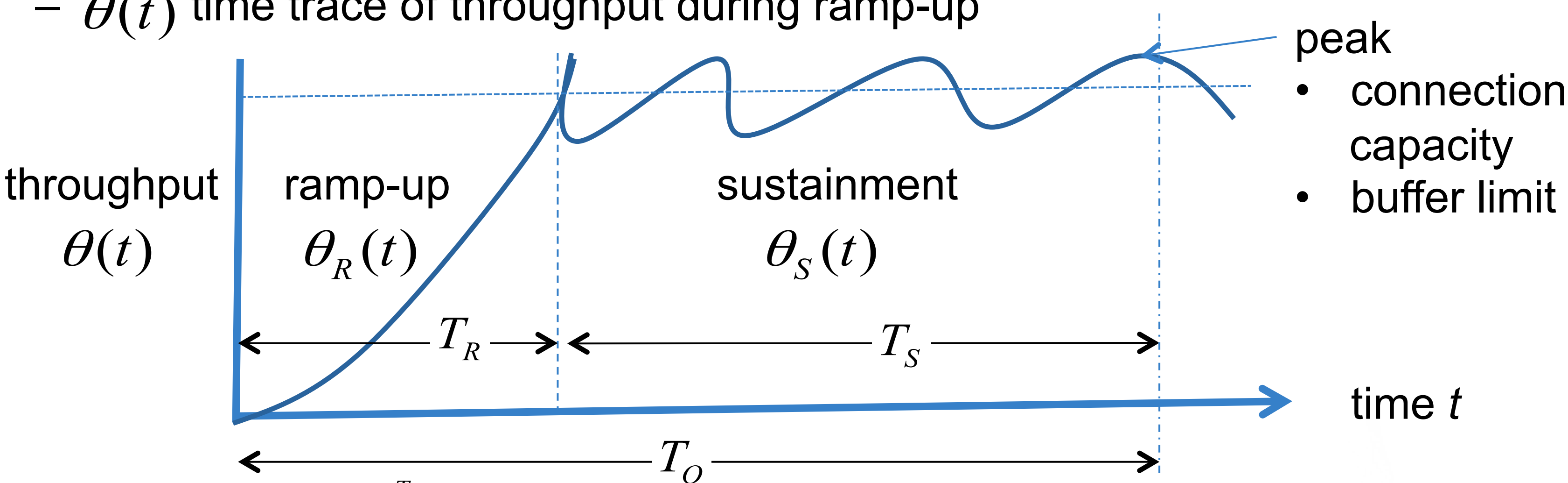
    Not for some TCP versions:
    - Reno
  - These are loadable linux modules



OAK RIDGE National Laboratory

Generic Transport Model

- Ramp-up Phase
  - throughput increases from initial value to around capacity e.g. TCP slow start, UDT ramp-up
  - $\theta(t)$ time trace of throughput during ramp-up

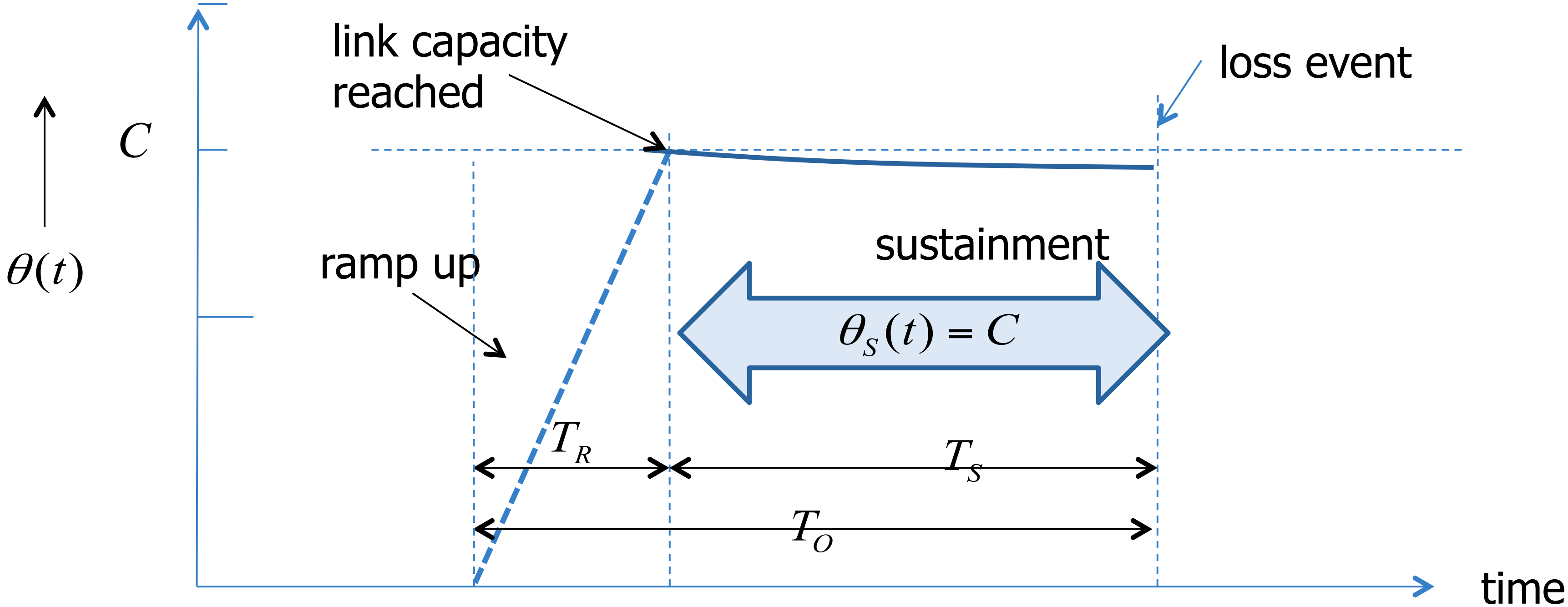


$$\bar{\theta}_R = \frac{1}{T_R}\int_0^{T_R} \theta(t)dt \qquad \bar{\theta}_S = \frac{1}{T_S}\int_{T_R}^{T_O} \theta(t)dt$$

- Sustainment Phase
  - Throughput is maintained around a peak value
    - TCP congestion avoidance, UDT stable peak
  - $\theta(t)$ time trace of throughput during sustainment

$$\Theta_O(\tau) = \frac{1}{T_O}\int_0^{T_O} \theta(\tau,t)dt$$

Boundary Case



Average Throughput: Monotonicity

$$\Theta_O = \bar{\theta}_R \frac{T_R}{T_O} + \bar{\theta}_S \left[\frac{T_O - T_R}{T_O}\right]$$

$$= \bar{\theta}_R f_R + \bar{\theta}_S \left[1 - f_R\right]$$

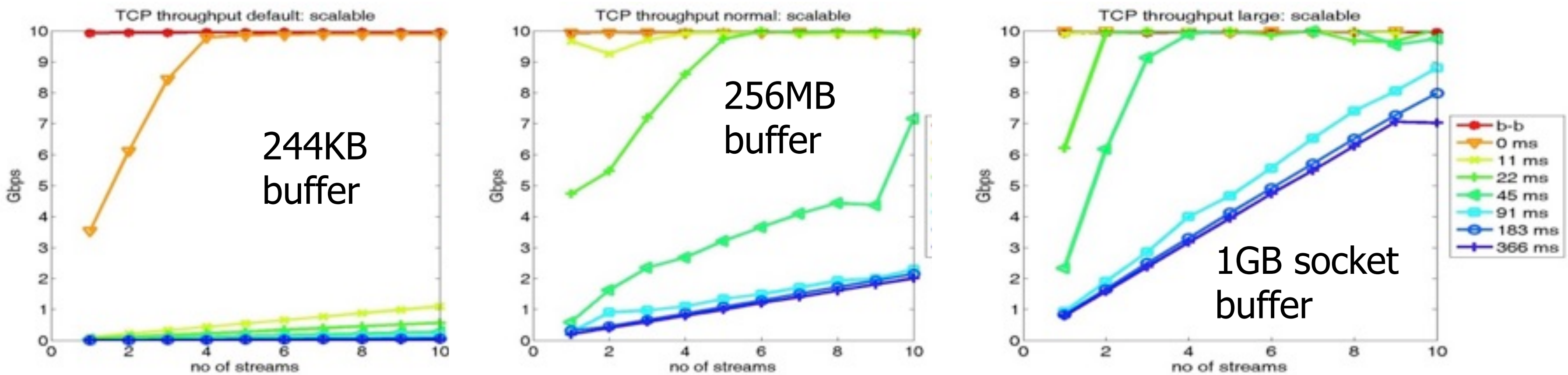
$$= \underbrace{\bar{\theta}_S - f_R \underbrace{\left[\bar{\theta}_S - \bar{\theta}_R\right]}_{+}}_{\text{decreases with } \tau}$$

increases with τ

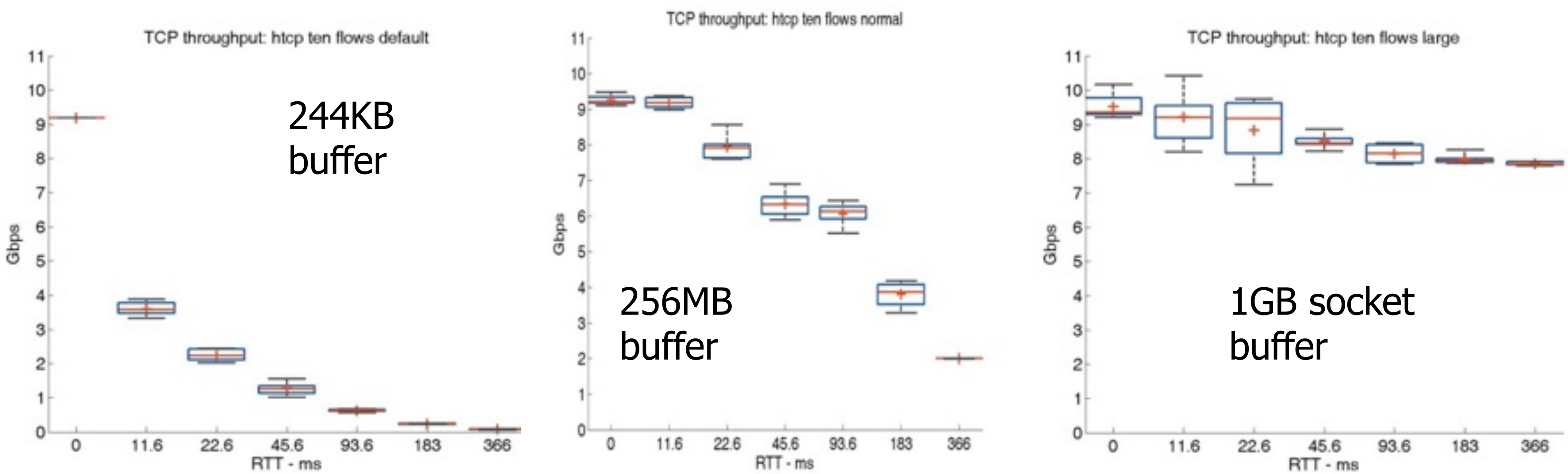
Multiple TCP Streams and Large Buffers

Both provide higher throughput and expanded concave region

- **Increase in average throughput: STCP over 10GigE**



- **Expanded Concave Region: Hamilton TCP: 10 flows over OC192**



Concavity: Faster than Slow Start – Multiple TCP flows

Faster than Slow Start:

More increases than slow start: $n_k = \tau^{\in_\tau} \log C \qquad \in_\tau > 0, \tau > 1$

$$T_R = \tau n_k = \tau^{1+\in_\tau} \log C$$

data sent: $1 + 2 + \cdots + 2^{n_k} = 2^{n_k+1} - 1 = 2^{1+\tau^{\in_\tau}} C - 1$

$$\bar{\theta}_R \approx \frac{2^{1+\tau^{n_k}} C}{\tau^{1+\in_\tau} \log C}$$

Average Throughput:

$$\Theta_O(\tau) = \frac{2^{1+\tau^{\in_\tau}} C}{T_O} + C\left[\frac{T_O - \tau^{1+\in_\tau} \log C}{T_O}\right]$$

$$\frac{d\Theta_O}{d\tau} = \underbrace{-\frac{(1+\in_\tau)\tau^{\in_\tau} C \log C}{T_O}}_{\text{decreasing function of } \tau}$$

implies concavity of $\Theta_O(\tau)$

Monotonicity Conditions: Not always decreasing in τ

Average Throughput:

$$\Theta_O(\tau) = \bar{\theta}_R(\tau)\frac{T_R(\tau)}{T_O} + \bar{\theta}_S(\tau)\left[\frac{T_O - T_R(\tau)}{T_O}\right]$$

$$= \quad \bar{\theta}_S(\tau) + f_R(\tau)\left[\bar{\theta}_R(\tau) - \bar{\theta}_S(\tau)\right]$$

Effective sustained phase: $\bar{\theta}_S(\tau) > \bar{\theta}_R(\tau)$

$\Theta_O(\tau)$ monotonically decreases in τ

$\bar{\theta}_S(\tau) \Downarrow \qquad f_R(\tau) \Uparrow$

Ineffective sustained phase: $\bar{\theta}_S(\tau) < \bar{\theta}_R(\tau)$ $\longleftarrow$ may lead to:

- lower throughput
- convex region

$\Theta_O(\tau)$ may increase in τ

If $f_R(\tau)$ decreases "faster" than $\bar{\theta}_S(\tau)$

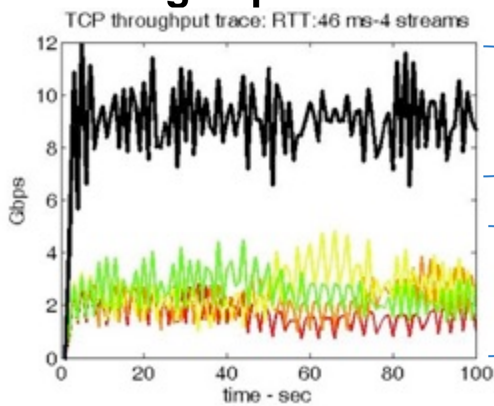
Some UDT measurements show this behavior

OAK RIDGE National Laboratory

Poincare Map

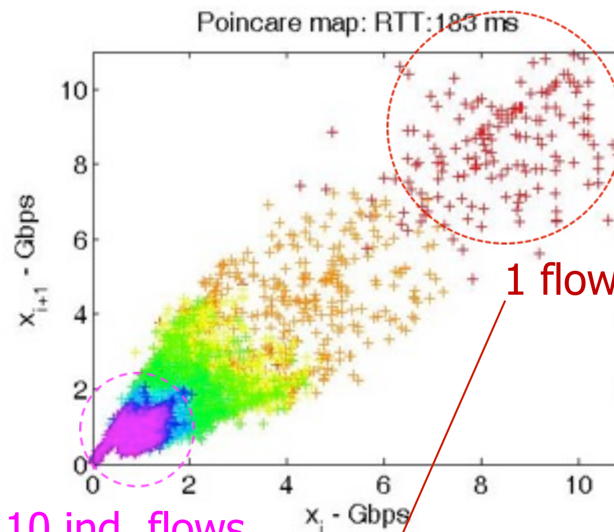
Well-Known tool for analyzing time series – used in chaos theory

- **Poincare map** $M : \Re^d \rightarrow \Re^d$
  - **Time series:** $X_0, X_1, \ldots, X_i, X_{i+1}, \ldots$
  - **generated as** $X_{i+1} = M(X_i)$ $X_i = M^i(X_0)$
- **Effect of Poincare map:**
  - **range specifies achievable throughput**



combined throughput

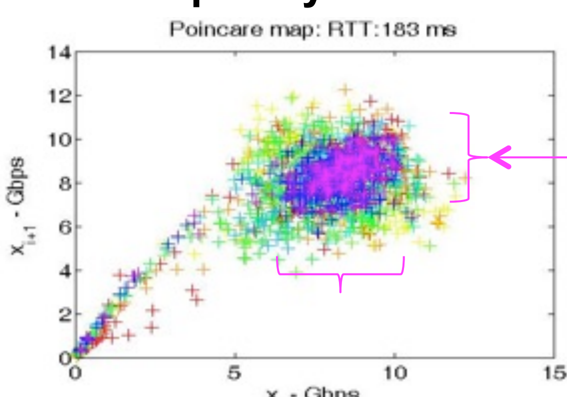
4 individual flows CUBIC 46ms



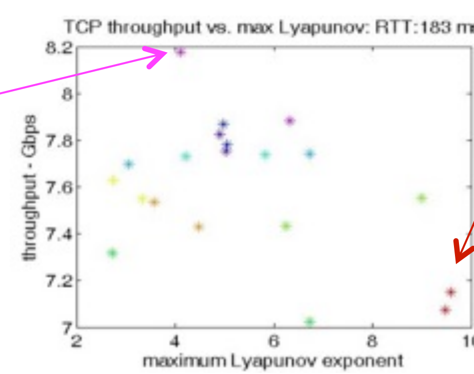
1 flow

10 ind. flows

  - **complexity indicates rich dynamics – lower throughput and narrow concave**



sum of 10 flows



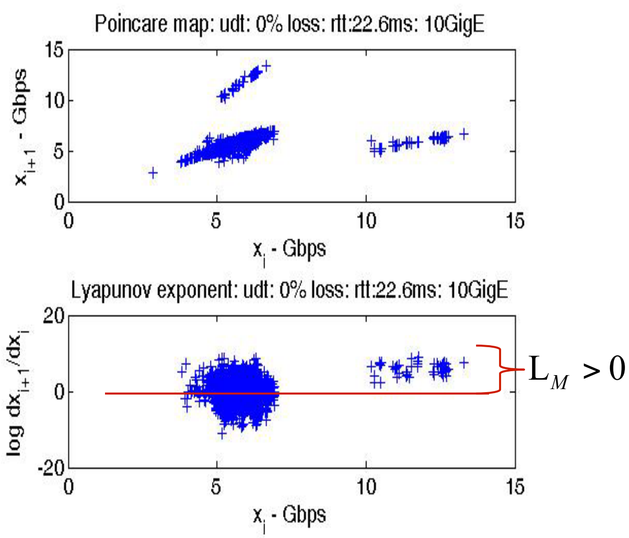
large Lyapunov exponent
- low throughput
  next slide

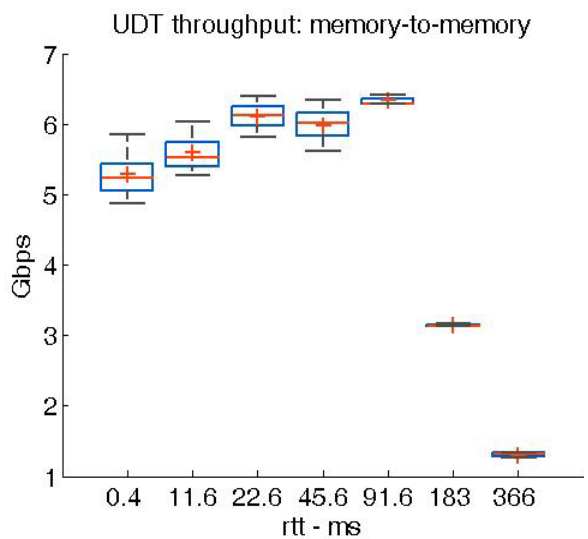
Lyapunov Exponent: Stability and Concavity

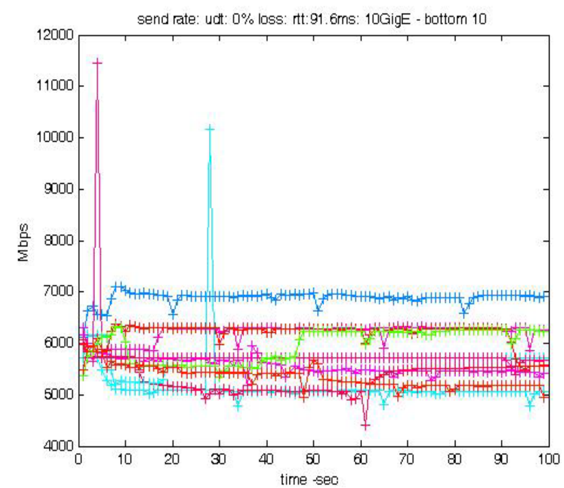
- **Log derivative of Poincare map**

$$\mathrm{L}_M = \ln\left|\frac{dM}{dX}\right|$$

- **Provides critical insights into dynamics**
  - **Stable trajectories:** $\mathrm{L}_M < 0$
  - **Chaotic trajectories:** $\mathrm{L}_M > 0$
    - **indicate exponentially diverging trajectories with small state variations**
    - **larger exponents indicate large deviations**
  - **protocols are operating at peak at rtt**
    - **stability implies average close to peak - implies concavity**
    - **positive exponents imply lowered throughput – trajectories can only go down**
      - » **then, weak sustainment implies convexity**







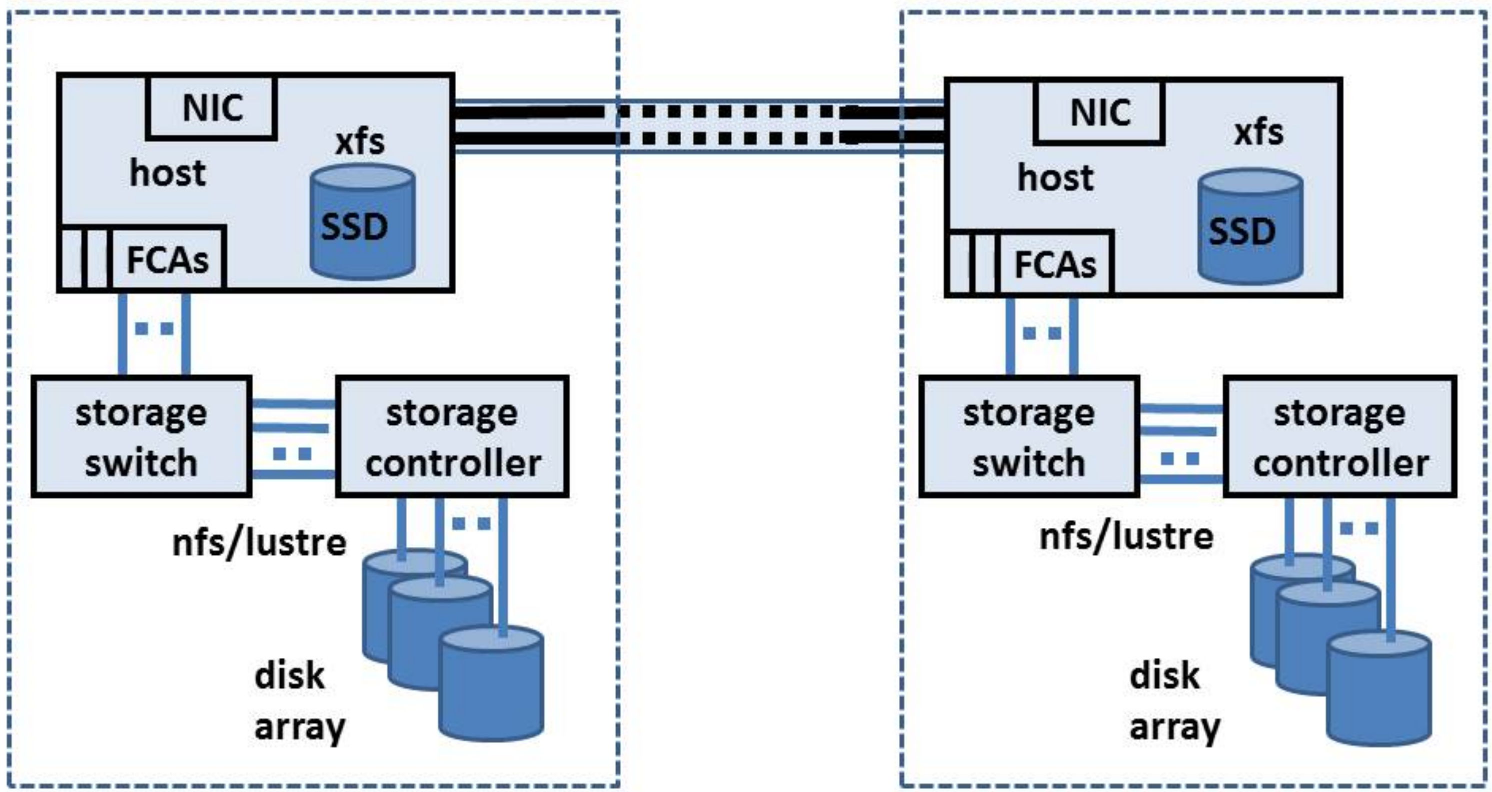
XDD: host-to-host file transfer tool

- XDD started as a file and storage benchmarking toolkit
  - storage tuning options
- Added a network implementation, python frontend, multi-host coordination, and NUMA tuning options
  - multiple NIC's from a single process for "better" storage access
- xddprof: sweep relevant tuning parameters
  - identify storage parameters to align with network performance profiles

xddmcp: Composing host-to-host flows

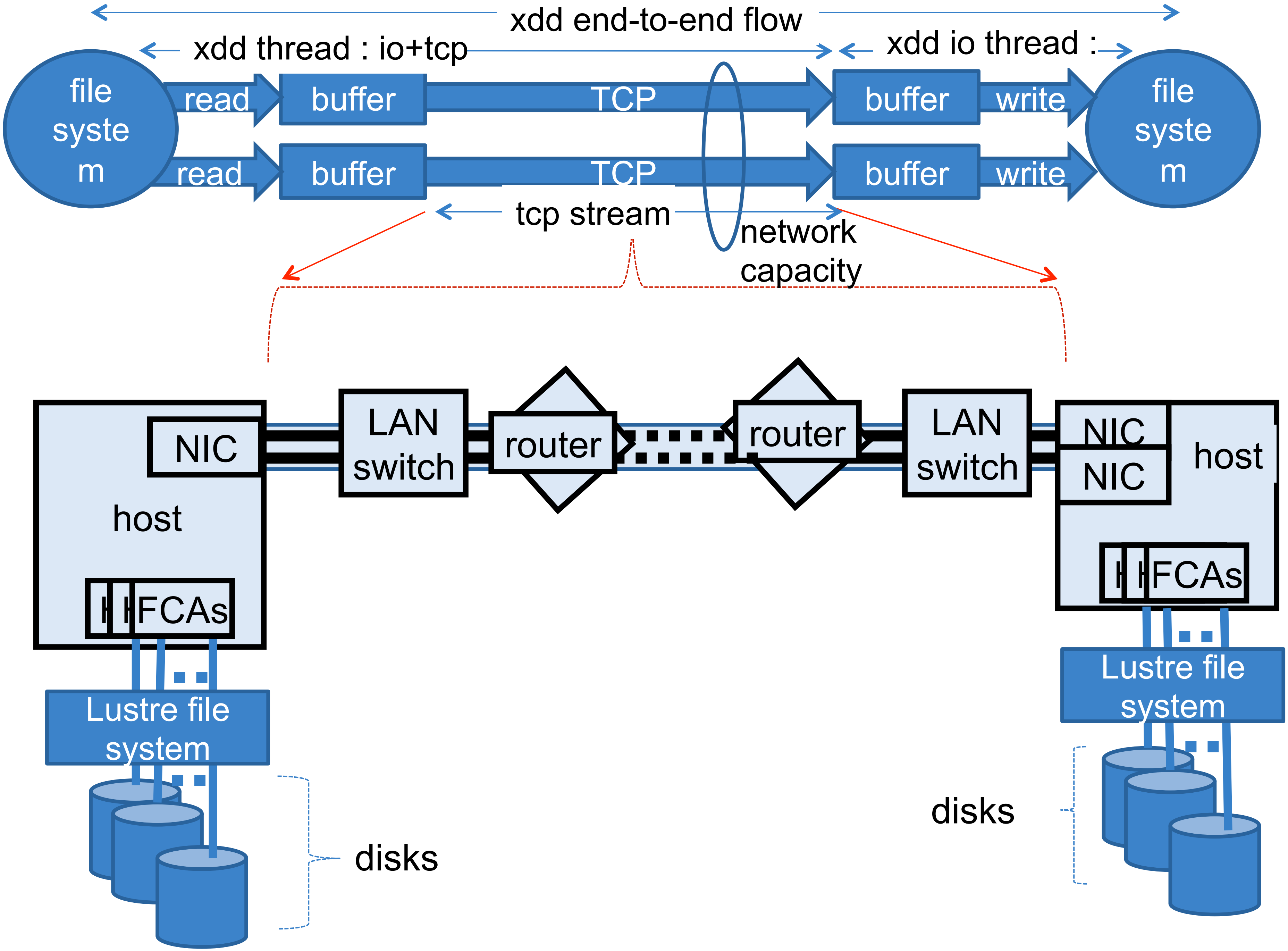
- Network and storage tuning optimizations aren't always complementary
  - network may prefer high number of streams
  - storage may prefer lower thread counts
- Leverage profiling information to understand performance
- Identify *compatible* network and storage parameters

ORNL Testbed: nfs, xfs and lustre file systems



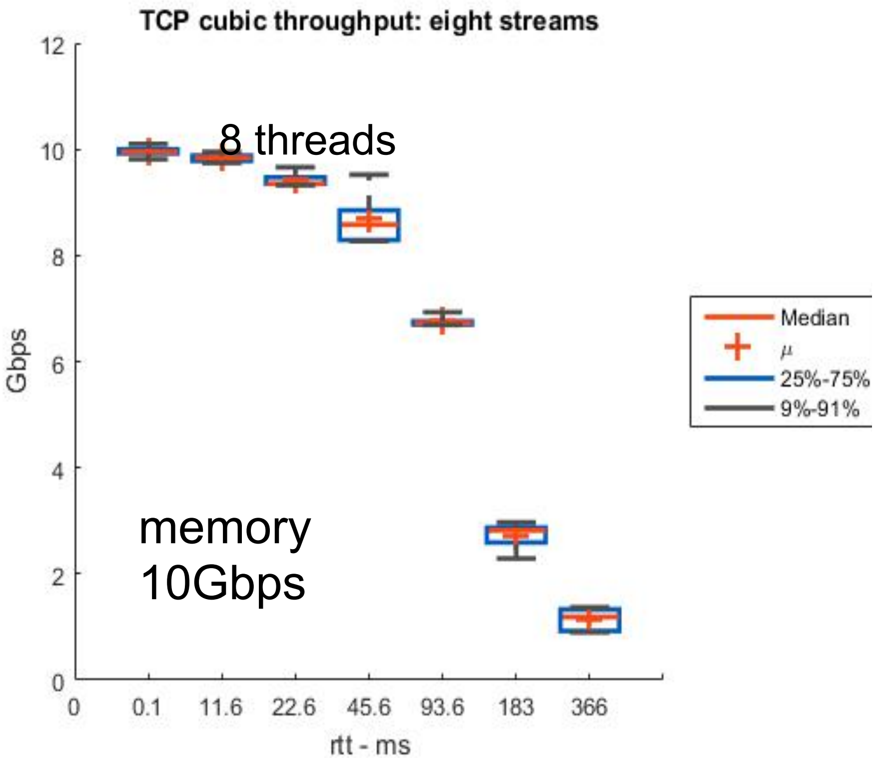
Peak IO rates: xddprof on hosts
- **nfs: ~2Gbps**
- **xfs: ~40 Gbps**
- **lustre: ~32 Gbps**

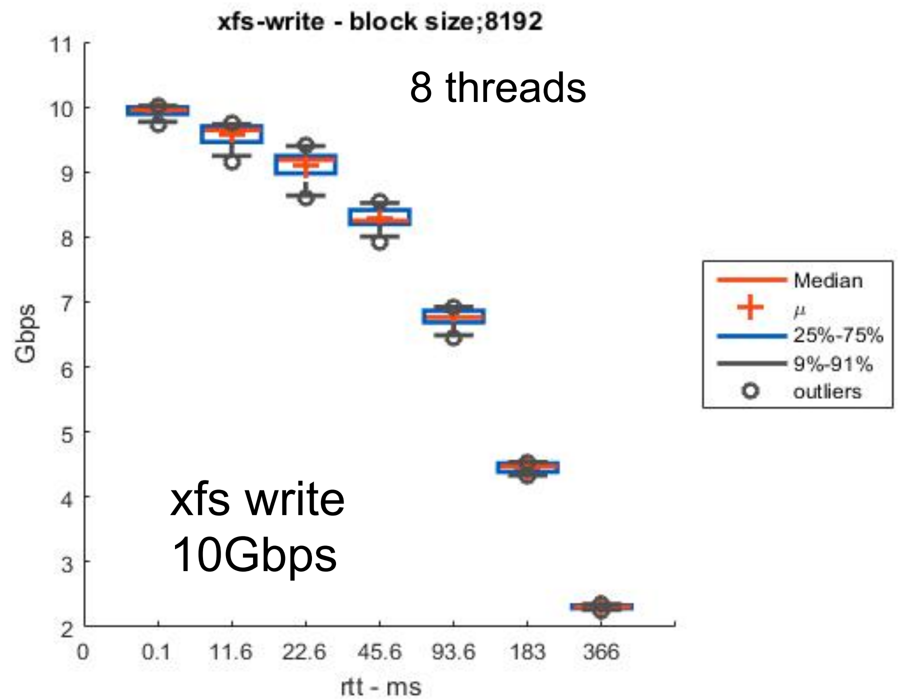
Peak n/w throughput: iperf
- **TCP: > 9Gbps**
- **UDP/T: > 8Gbps**
- **for 0ms rtt**


xdd end-to-end flow
xdd thread : io+tcp
xdd io thread :
file
syste
m
read
buffer
TCP
buffer
write
file
syste
m
read
buffer
TCP
buffer
write
tcp stream
network
capacity
NIC
host
LAN
switch
router
router
LAN
switch
NIC
NIC
host
FCAs
FCAs
Lustre file
system
Lustre file
system
disks
disks

TCP CUBIC and xfs

- **xddmcp host-to-host file transfers: peak: 10Gbps**



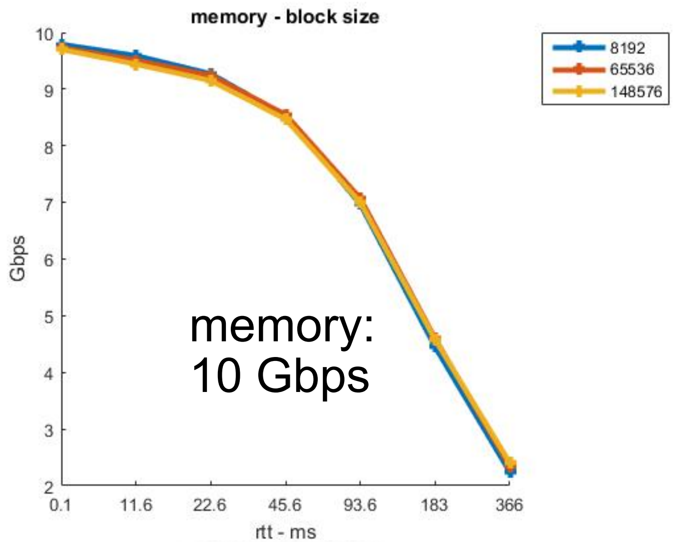


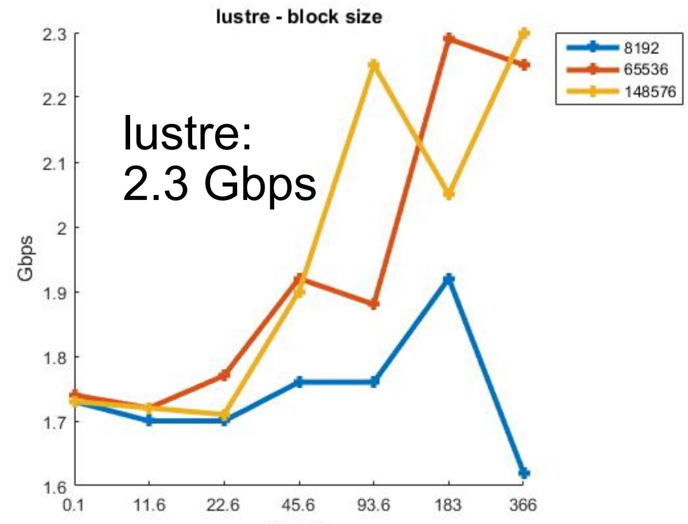
xdd file IO throughput is close to TCP throughput

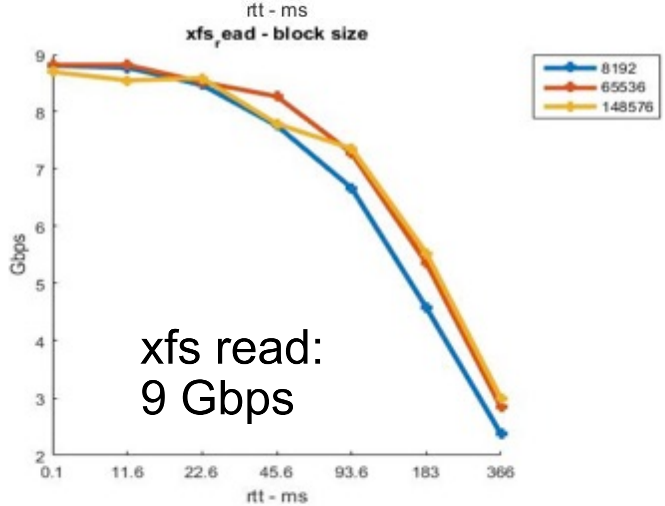
- 8 IO threads and 8 TCP parallel streams
- Impedance mismatch is quite small

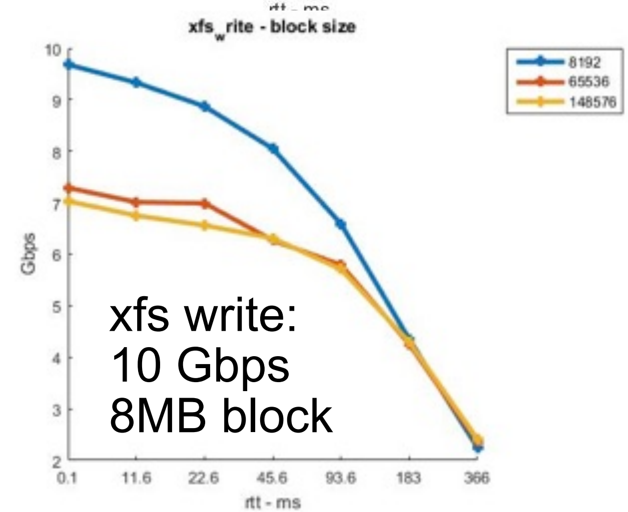
Average Throughput: lustre, xfs

8 streams: lustre throughput is lower compared to 1 stream



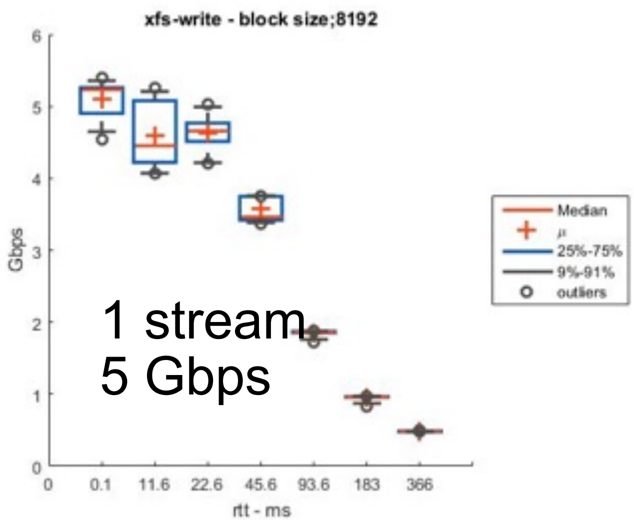


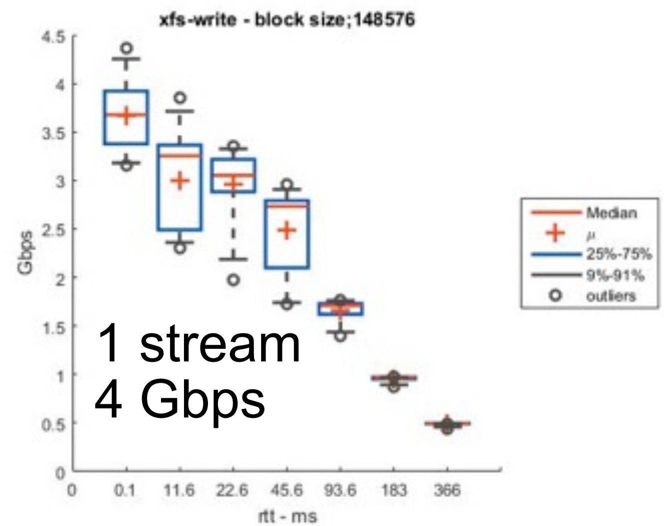


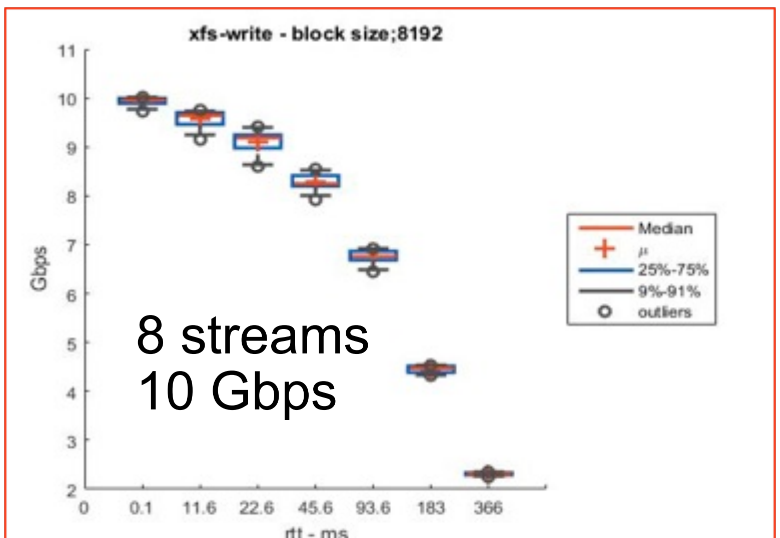


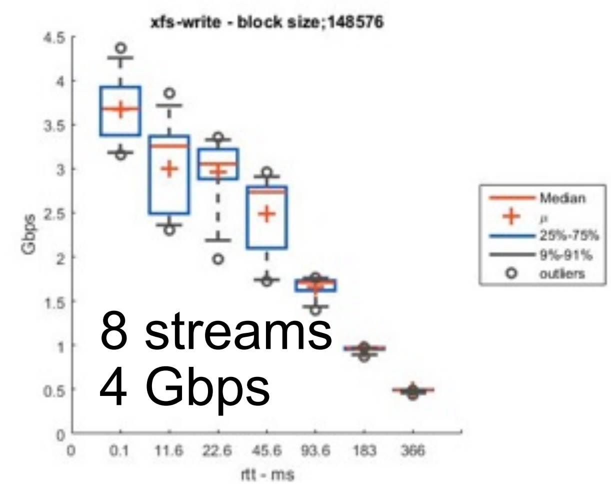
xfs: write

- **8 streams, 8MB blocks: 10Gbps**







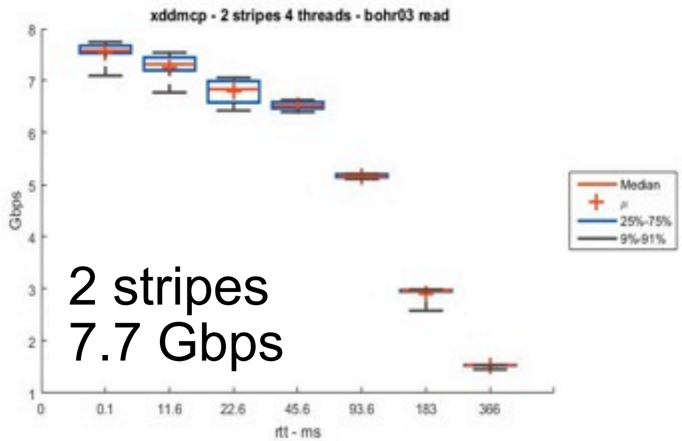


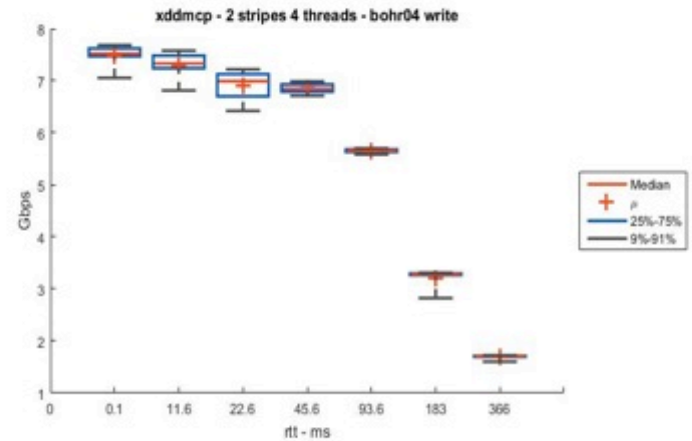
Smaller 8MB blocks for higher throughput

Oak Ridge National Laboratory

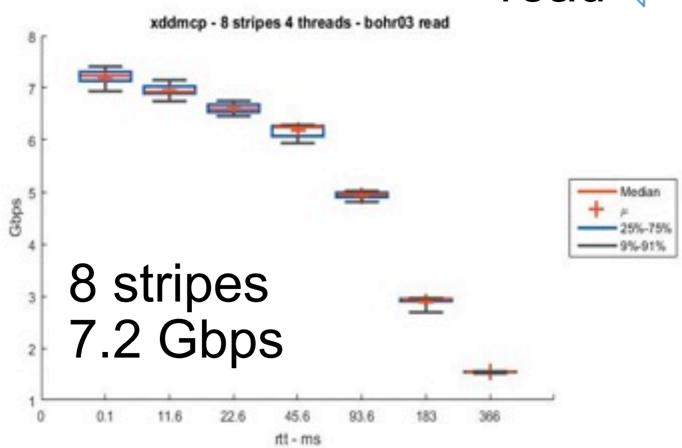
Best Case: Lustre default IO

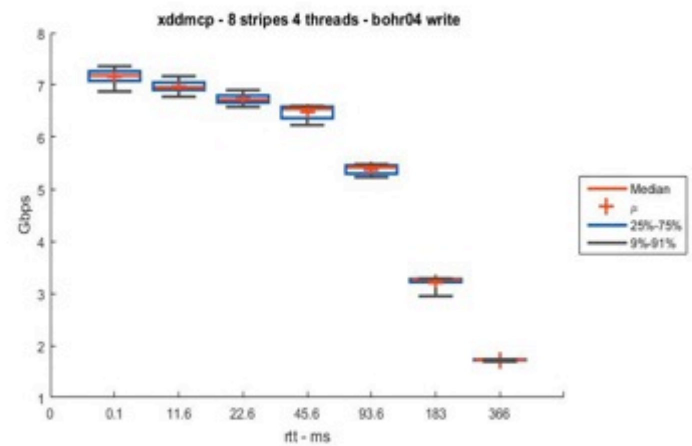
4 streams – 2 stripes: 7.5 Gbps





read ⟺ write

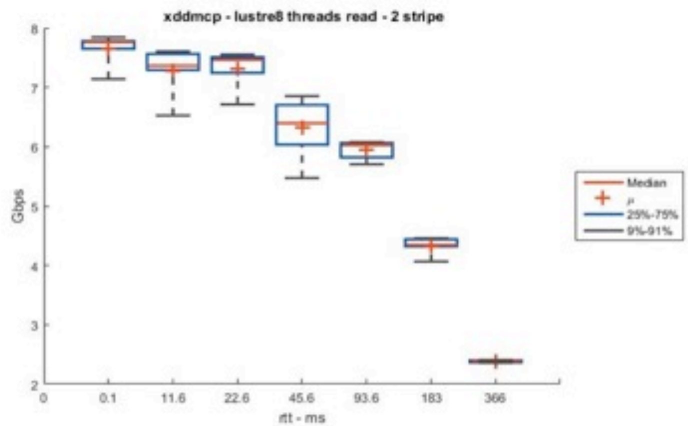


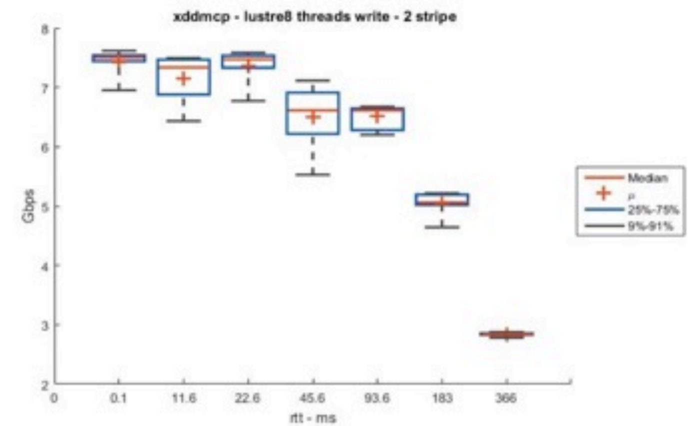


- **2 stripes provide higher throughput**

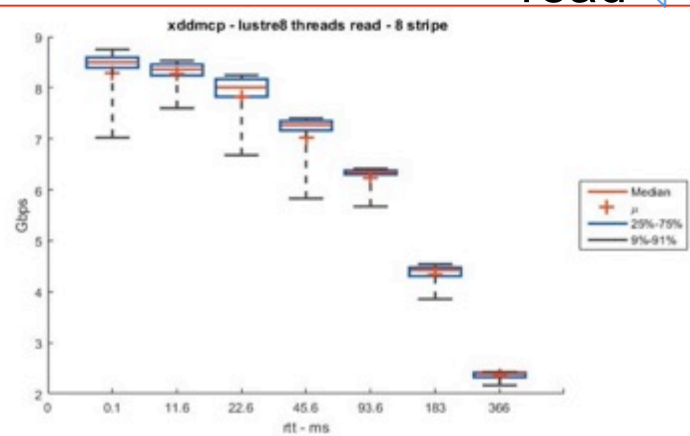
Best Case with direct IO

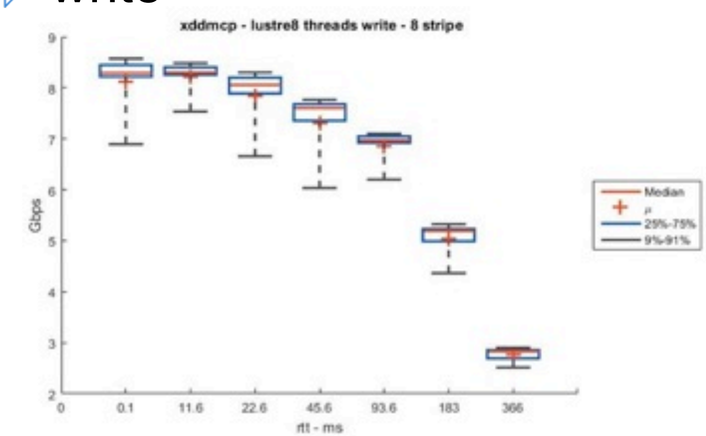
- **8 threads – 8 stripes : 8.5Gbps**





read ⟺ write





8 stripes provide higher throughput by 1Gbps over default IO best case

OAK RIDGE National Laboratory

d-w Method

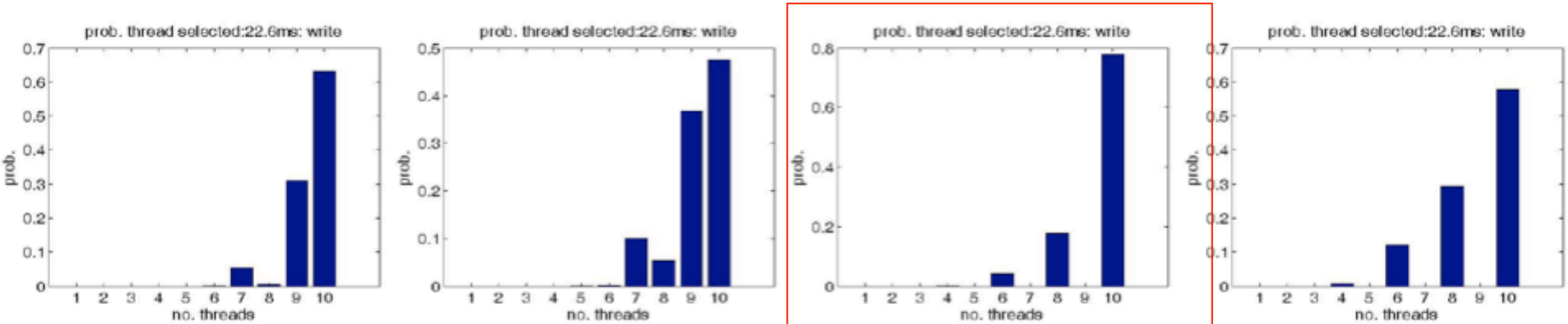
- Joint file I/O and network transport parameters for peak rate
  - Deriving them from full-profile takes months of measurements
  - Statistical variations are significant – gradient estimate is noisy
- Developed a *Depth-Width* or - method
  - exploits overall unimodality: throughput vs. number of streams
  - stochastic gradient search method: using repeated measurements over window
- Result: Identified peak configurations:
  - 97% of peak transfer rate for XFS and Lustre
  - probing 12% of parameter space – days vs. months for full profile

d-w algorithm

- <u>initialization</u>: start with largest number of flows
- <u>repeat until halting criterion</u>:
  - jump over a w-sized window for new probing configuration (different number of flows)
  - compute maximum throughput of d collected measurements at current configuration
- <u>halting criterion</u>:
  - maximum throughput decreases for two consecutive iterations

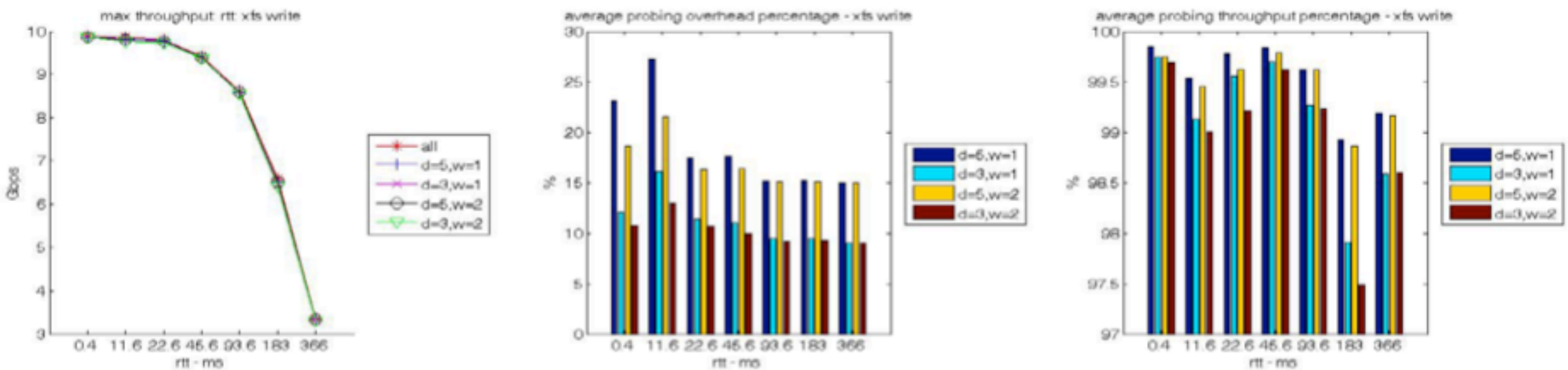
d-w Probing XFS Write

Confidence Estimate: fraction of 700 configurations conducive to d-w method



(a) $d = 5,\ w = 1$ (b) $d = 3,\ w = 1$ (c) $d = 5,\ w = 2$ (d) $d = 3,\ w = 2$

Confidence estimates of each configuration being selected with XFS file write transfer for $RTT = 22.6$ ms



(a) maximum throughput (b) probing overhead (c) probed vs. actual peak throughput

Profile of d-w probing with XFS file write transfer

Conclusions

Contributions

- Collected extensive transport measurements over dedicated connections
  - Multiple TCP variants and UDT: new insights
    - concave region of throughput profile
    - rich dynamics: complex Poincare map and positive Lyapunov exponents
- Developed throughput model
  - simple enough not to require detailed protocol parameters
  - still explains the basic qualitative
    - concavity analysis
    - Poincare Maps and Lyapunov exponents: link dynamics to profiles
- Applied for fine tuning XDD file transfers

Future Directions

- Detailed analytical models to explain concavity
- Automatic parameter optimization methods
- Differential methods:
  - align analytical models with measurements
  - capture difference and apply corrections

Thank you
Questions?