Estimation of RTT and Loss Rate of Wide-Area Connections Using MPI Measurements

Raj Kettimuthu

Nagi Rao, Neena Imam (Oak Ridge National Laboratory)

Zhengchun Liu, Raj Kettimuthu, Ian Foster (Argonne National Laboratory)

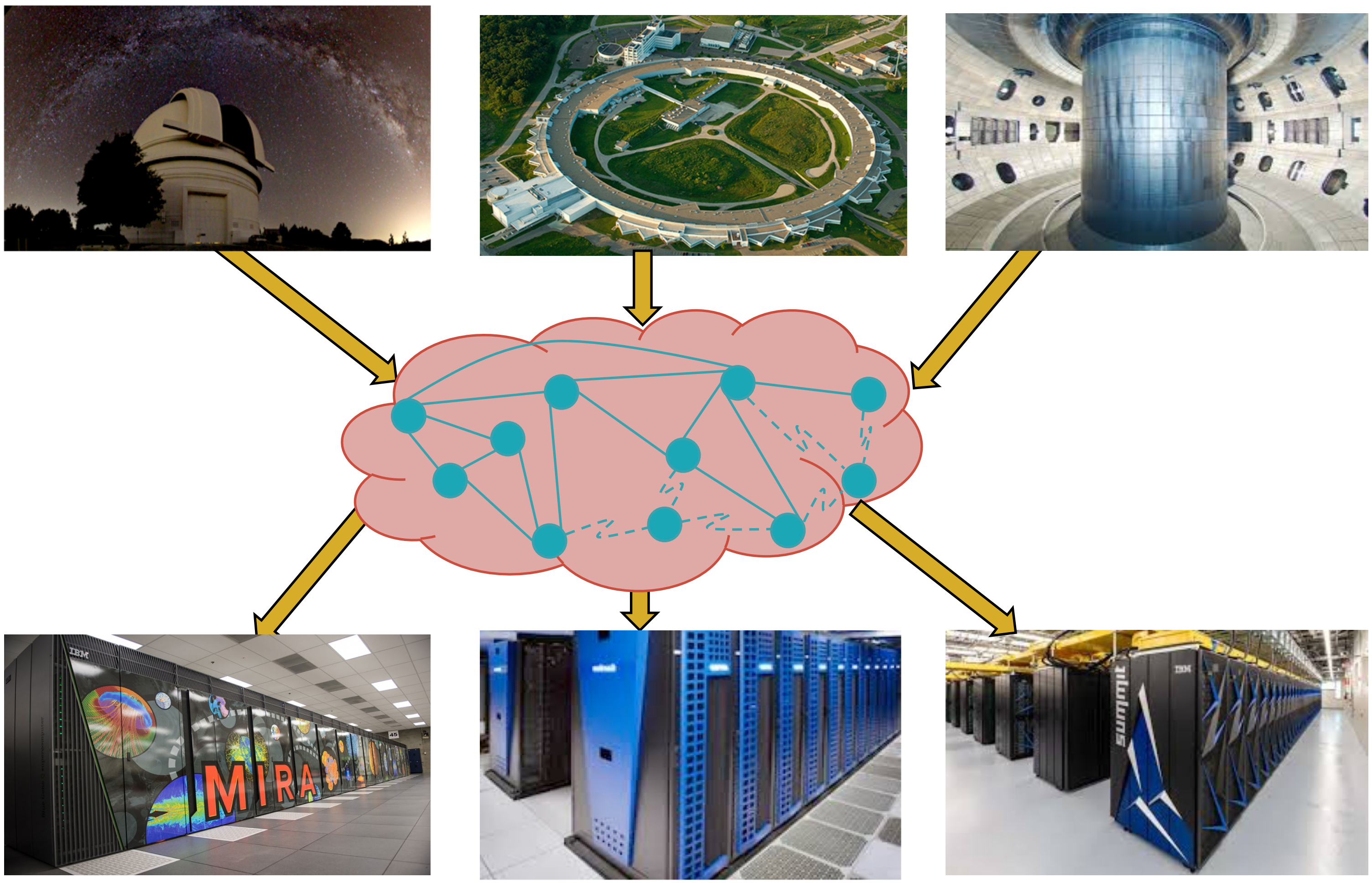
INDIS Workshop at SC2019, Denver, CO, November 17, 2019

Research sponsored by

RAMSES Project, Department of Energy

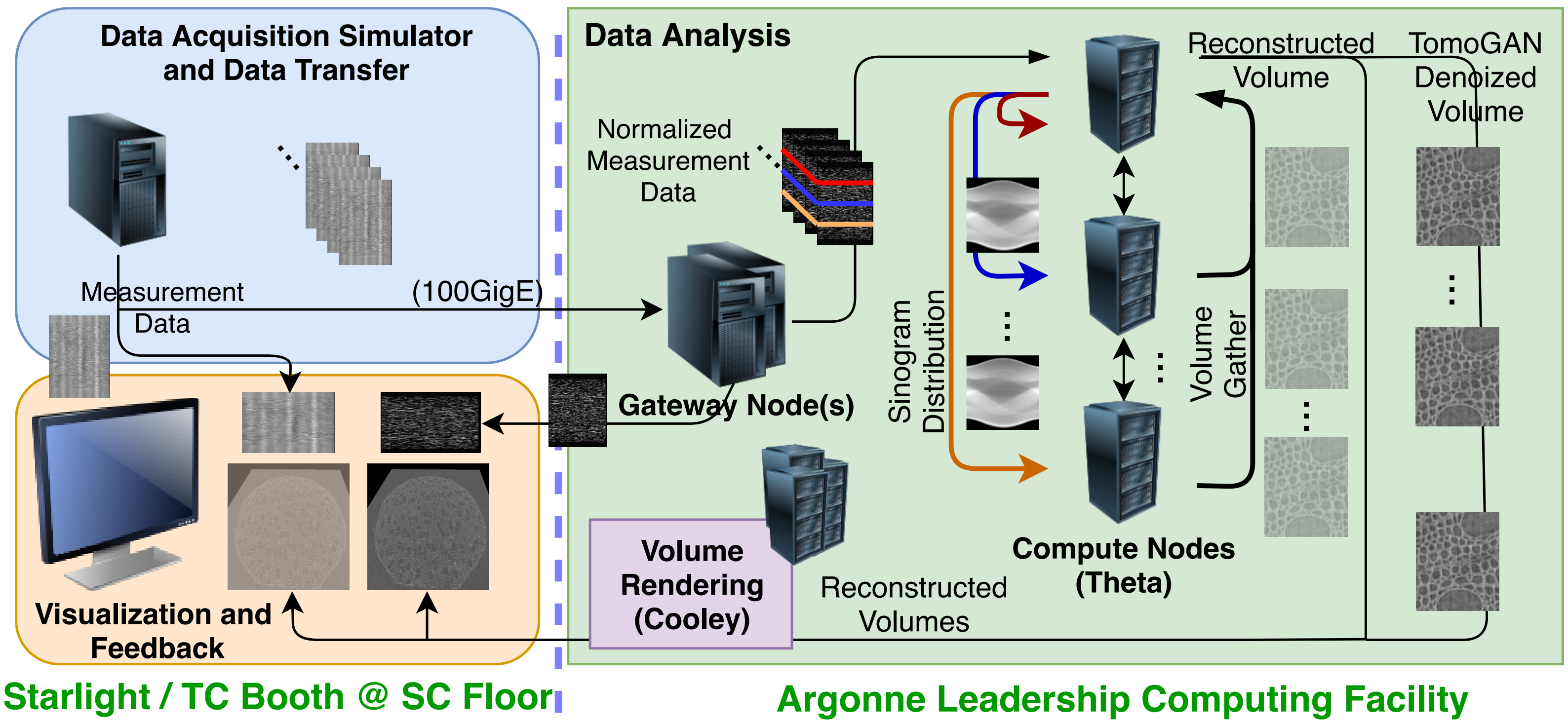
Department of Defense

Distributed computations across geographically dispersed facilities



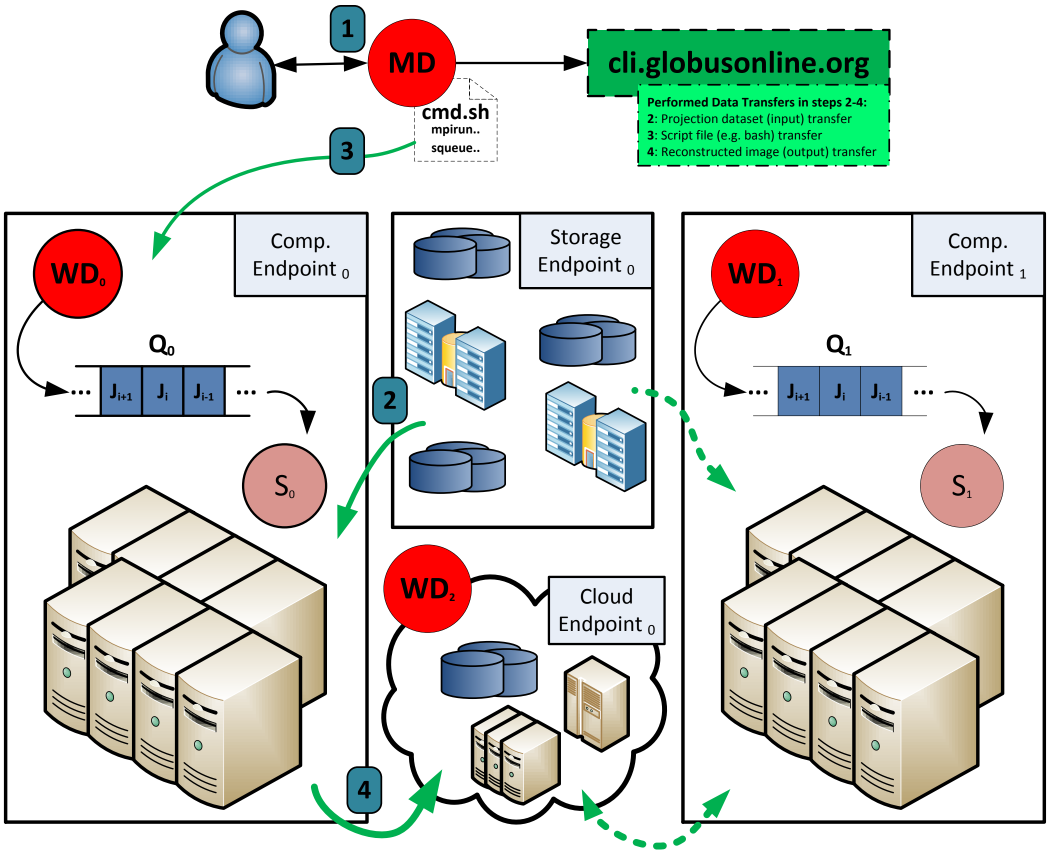
Near real-time analysis

SC19 technology challenge – Near real-time analysis of streaming synchrotron data



Tues, Nov. 19th 1:15pm at SC theater near SCinet booth 1081

Users have allocations on multiple compute resources

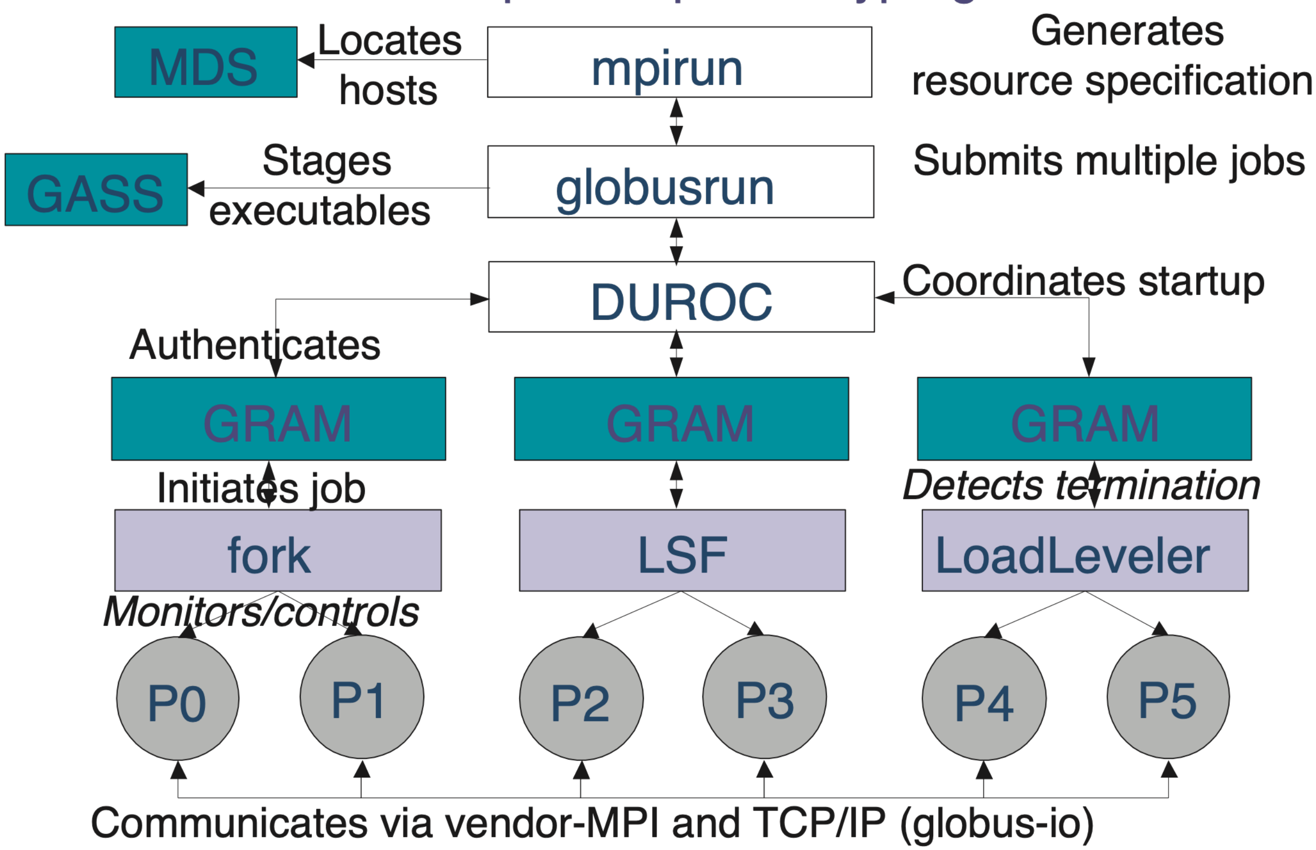


OAK RIDGE National Laboratory

MPICH-G

% grid-proxy-init

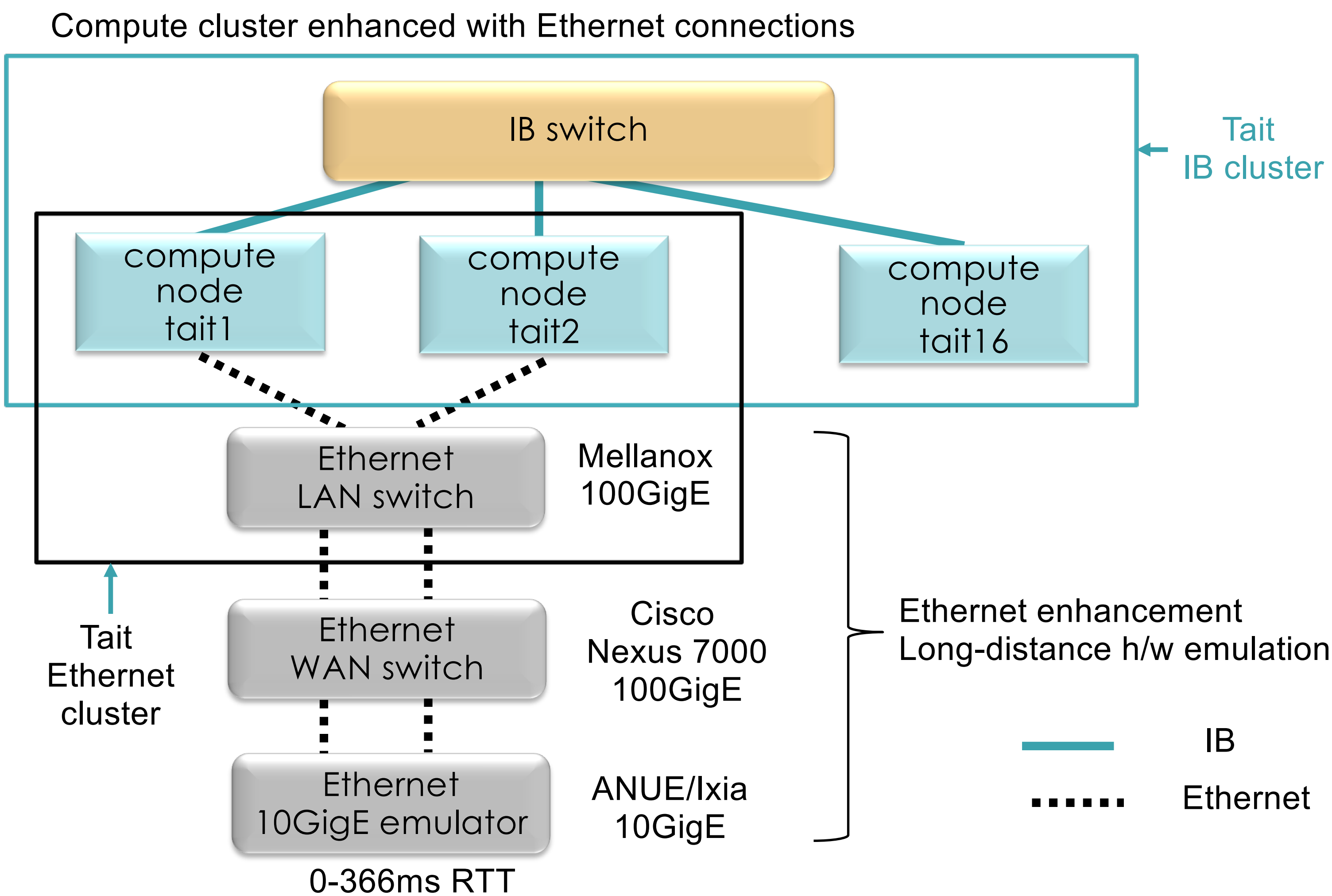
% mpirun -np 256 myprog



Motivation and Goals

- Execution times of MPI applications over WAN depend on connection Round Trip Time (RTT) and loss rate
- Distributed computations need to account for them to be efficient
- Estimate connection RTT & loss rate using MPI measurements
  - Better align distributed computations to network connections
  - Adapt workflows to dynamic network conditions, e.g., unexpected increase in losses
- Study strengths & limitations of ML methods in estimating network-level parameters using application-level measurements
  - Use MPI measurements to estimate connection RTT & loss rate

Testbed



Compute cluster enhanced with Ethernet connections
IB switch
compute node tait1
compute node tait2
compute node tait16
Tait IB cluster
Ethernet LAN switch
Mellanox 100GigE
Ethernet WAN switch
Cisco Nexus 7000 100GigE
Ethernet 10GigE emulator
ANUE/Ixia 10GigE
0-366ms RTT
Tait Ethernet cluster
Ethernet enhancement
Long-distance h/w emulation
IB
Ethernet

Chaotic map diagnosis code

chaotic-map: running rtt: 366ms -- test num: 102

Thu Jun 7 15:20:12 EDT 2018

Number of cores detected=24

Diagnosis completed

Diagnosis Summary:

Core 0: output: 0.932237   : 938210F1

Core 1: output: 0.932237   : 938210F1

Core 2: output: 0.932237   : 938210F1

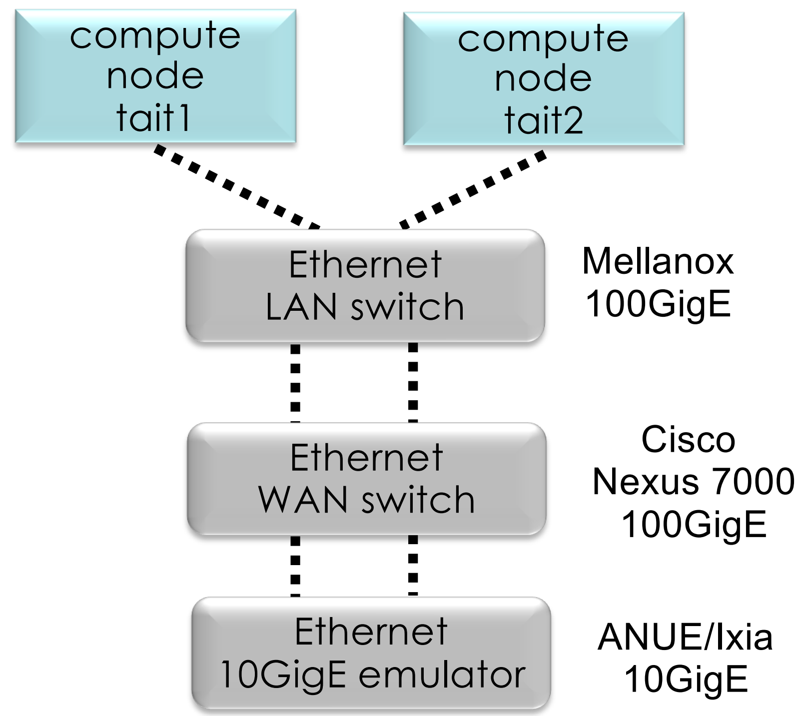
Core 21: output: 0.932237   : 938210F1

Core 22: output: 0.932237   : 938210F1

Core 23: output: 0.932237   : 938210F1

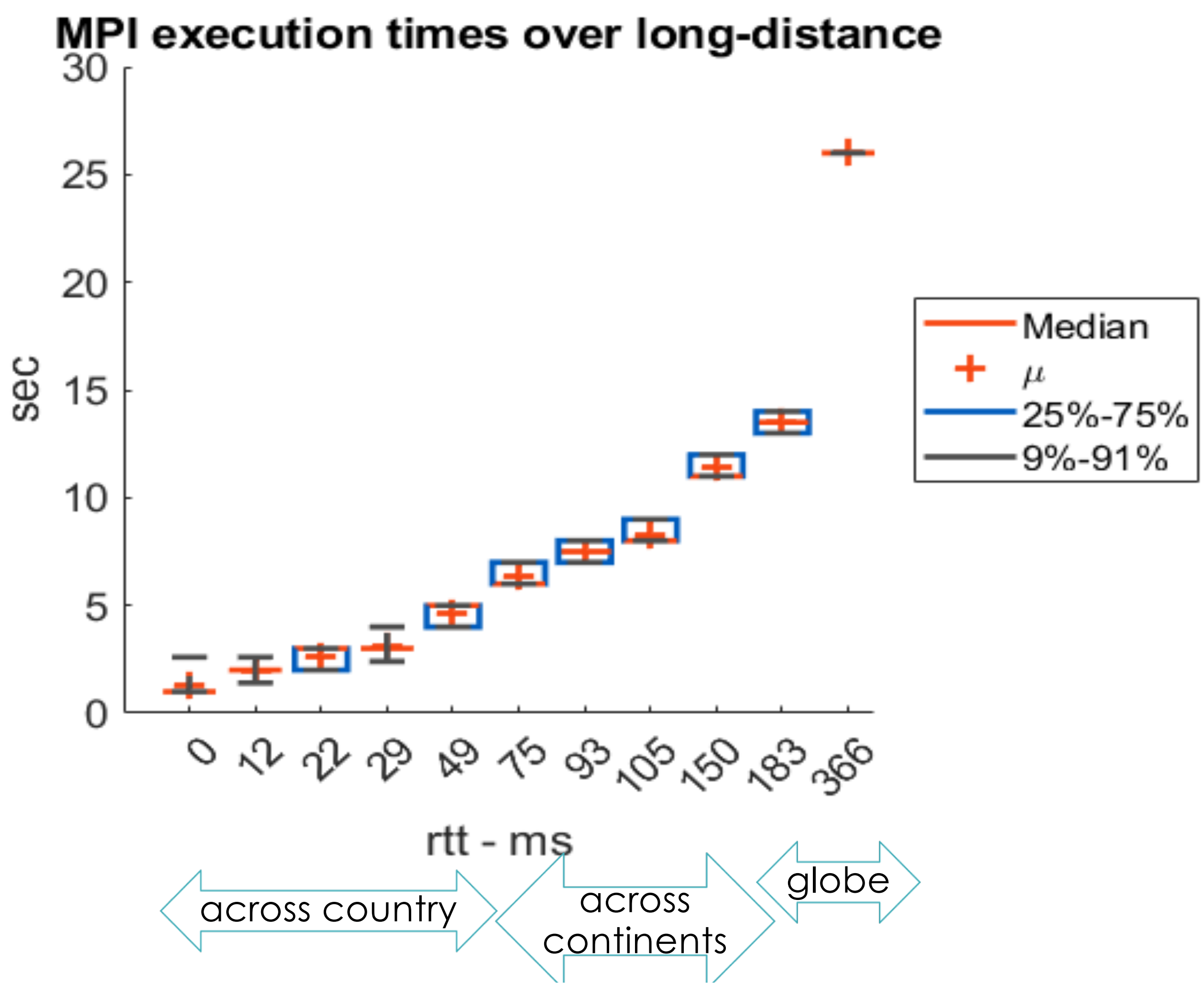
Thu Jun 7 15:20:39 EDT 2018

27 sec execution time for 366ms RTT
MPI demonstrates over round-the-earth distances

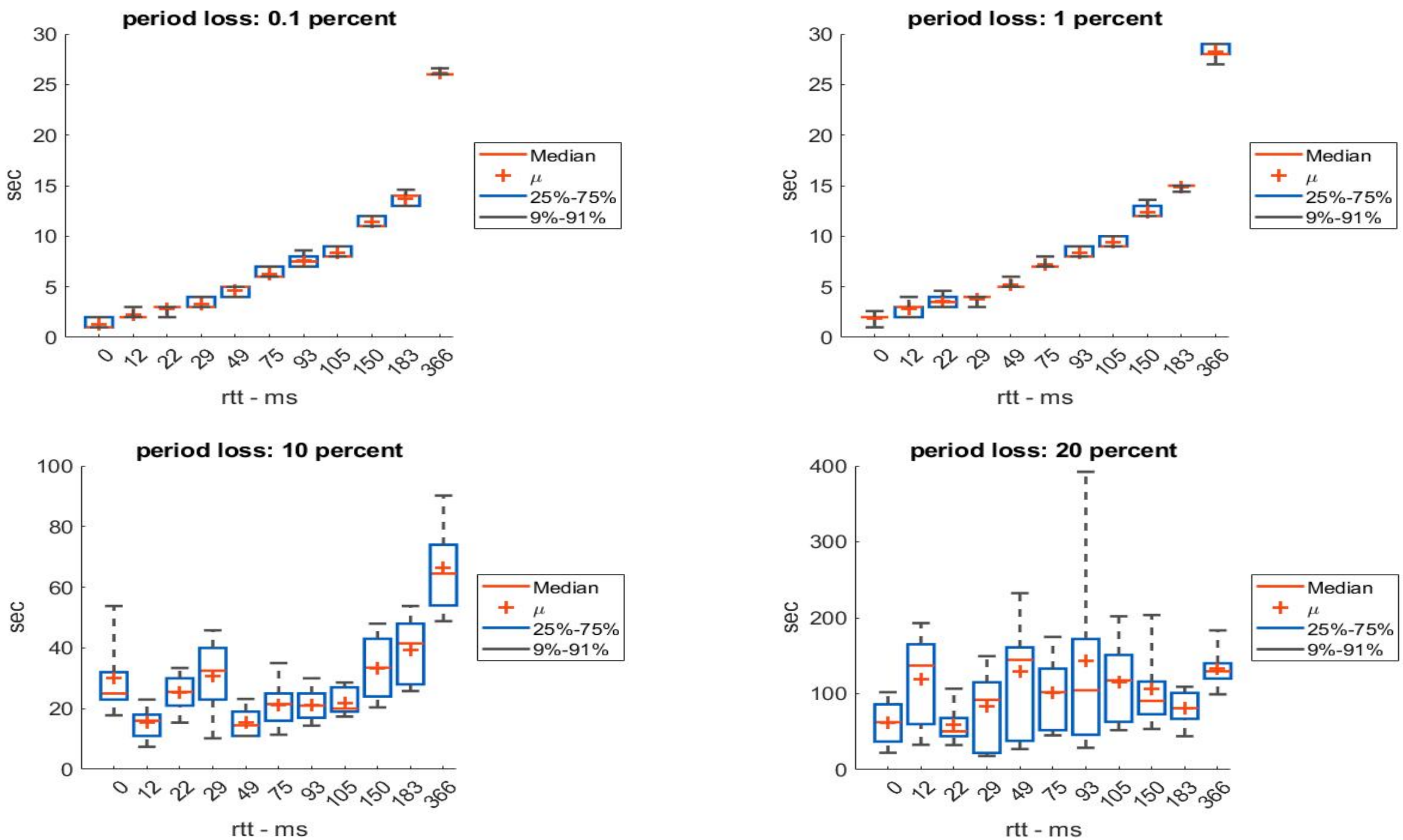


errors during
WAN Ethernet transfers
lead to different outputs

MPI over WAN


MPI execution times over long-distance
30
25
20
15
10
5
0
sec
0
12
22
29
49
75
93
105
150
183
366
rtt - ms
Median
μ
25%-75%
9%-91%
across country
across continents
globe

MPI over WAN at different loss rates

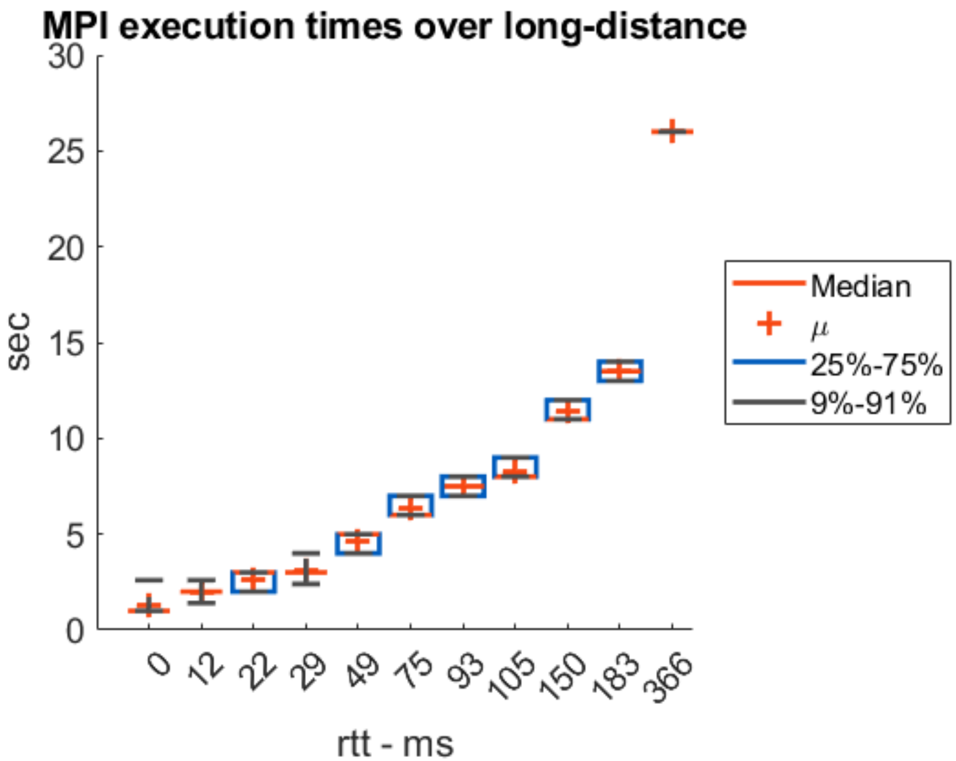


OAK
National

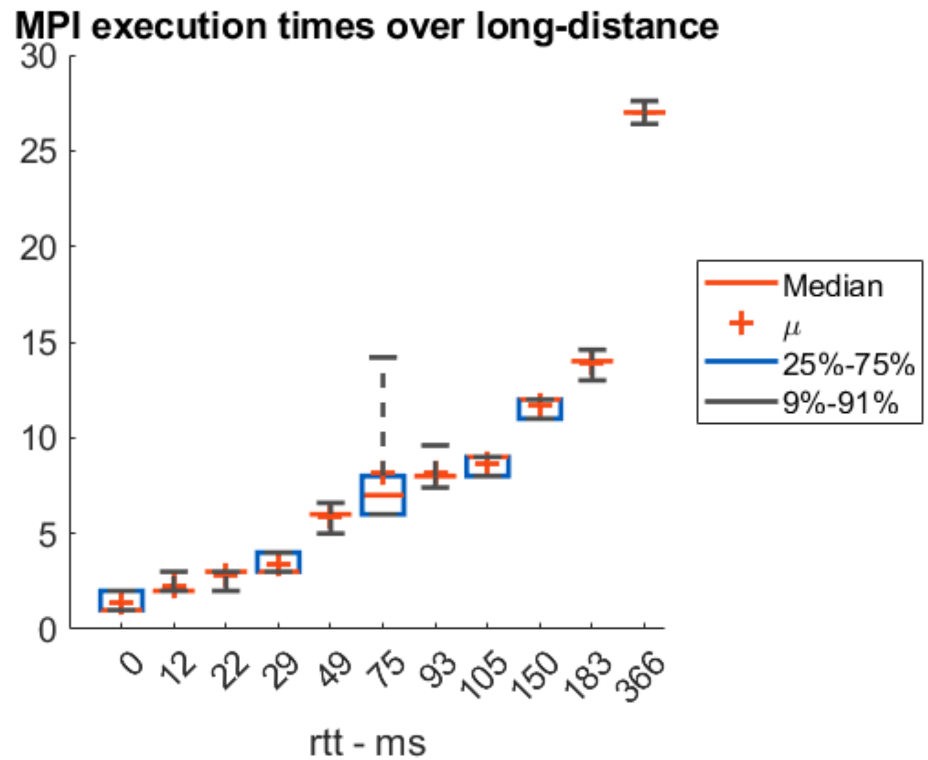
Five machine learning methods

- Linear regression
- Four non-linear estimates
- Two smooth:
  - Support Vector Machine (SVM) and Gaussian Process Regression (GPR)
- Two non-smooth:
  - Ensemble of Tree (EOT) and Regression Trees (RT)

MPI over WAN

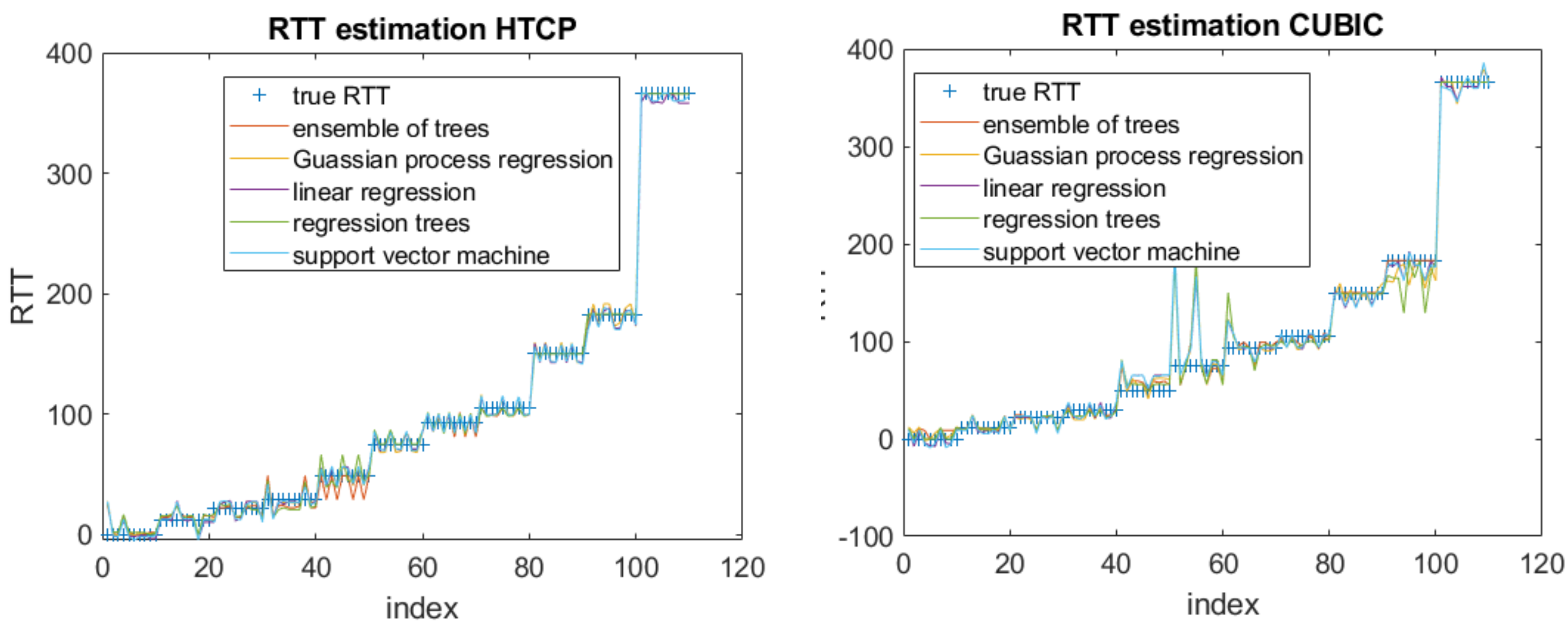


Hamilton TCP



CUBIC TCP

RTT Estimates: No loss Scenarios



method	EOT	GPR	RT	LR	SVM
CUBIC no loss - rms	22.56	18.05	20.71	16.68	16.88
HTCP no loss - rms	8.72	7.32	6.27	7.49	7.52

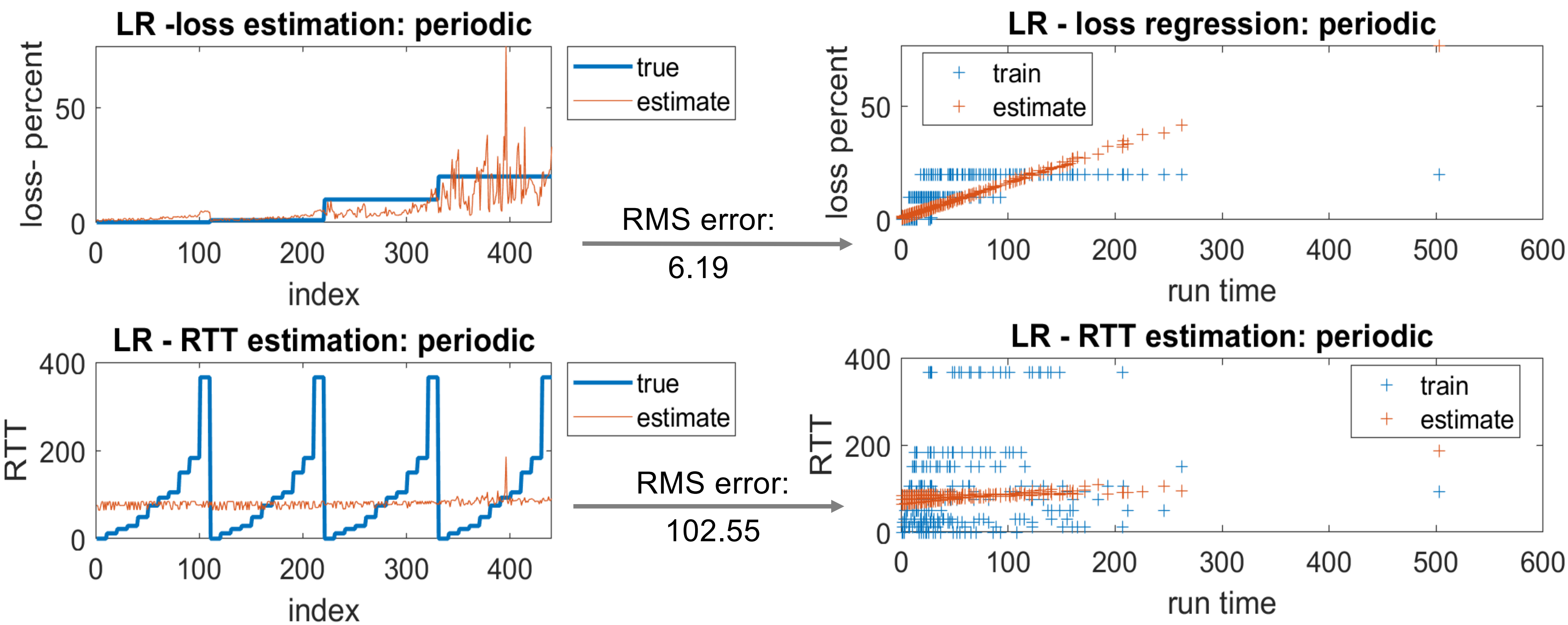
HTCP < 9

CUBIC > 16

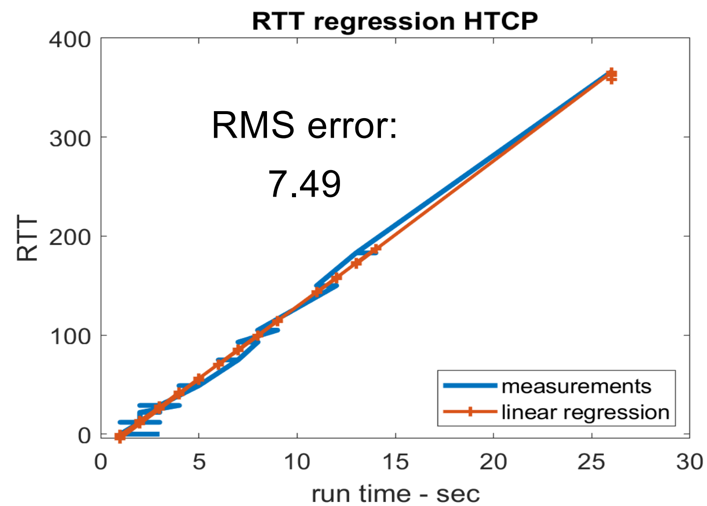
Performance of ML methods is different for CUBIC and HTCP

OAK RIDGE National Laboratory

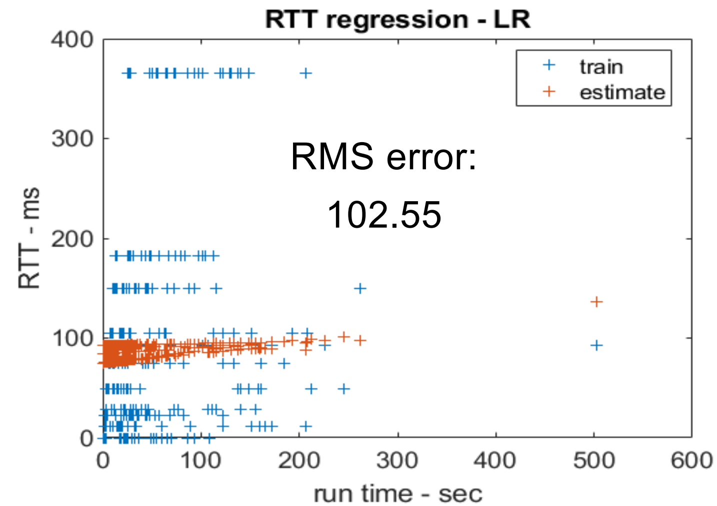
Linear regression: periodic losses

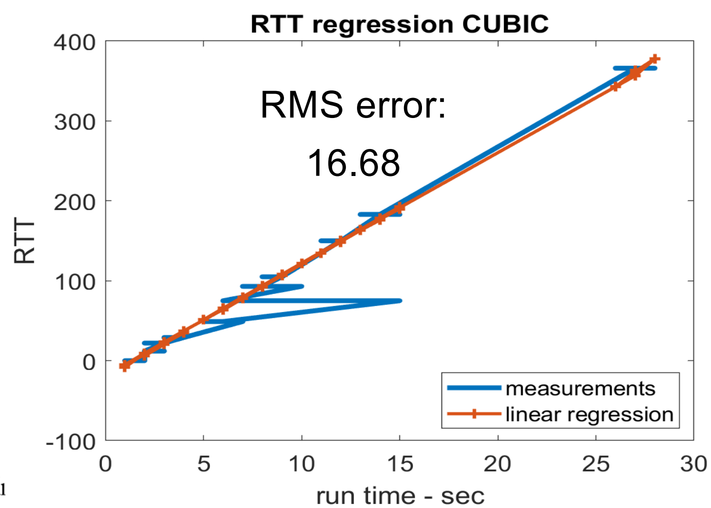


Linear regression: deep dive



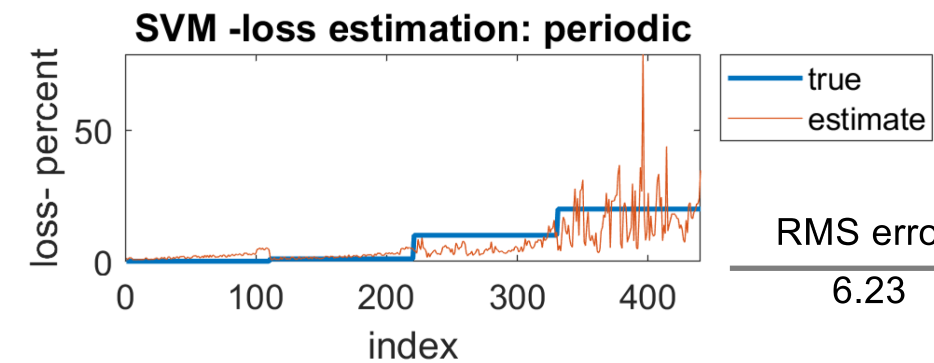
losses



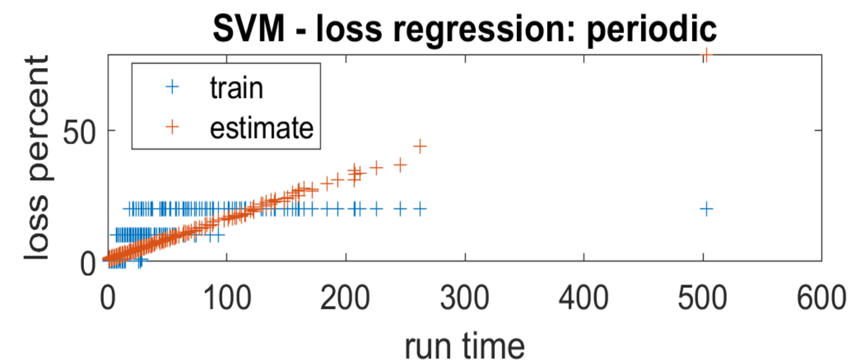


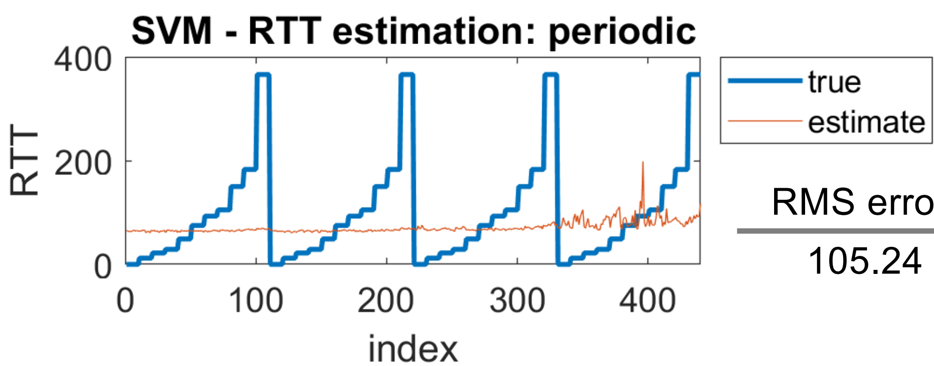
- Under 0.1-20% losses
- Data plot has high scatter
- X-range: from [0,27] to [0,500] outliers
- Linear regression is not good fit due to non-linear scatter

Support vector machines

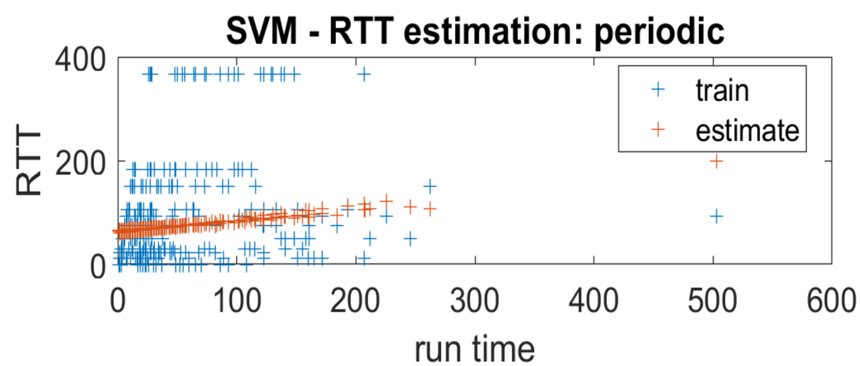


RMS error:
6.23

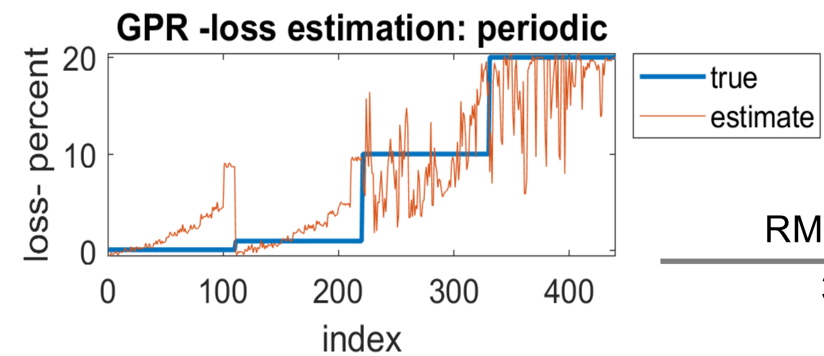




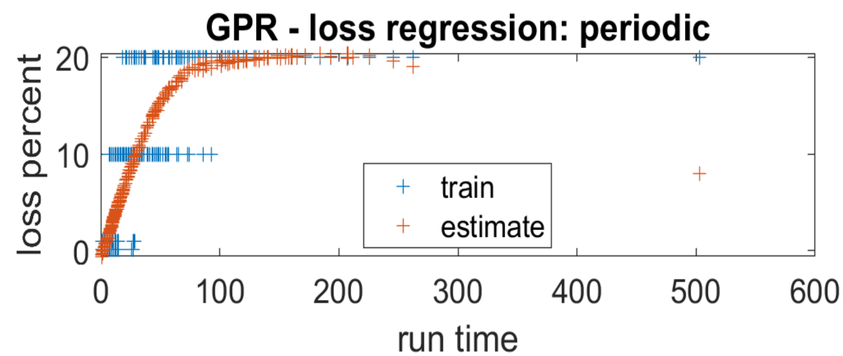
RMS error:
105.24

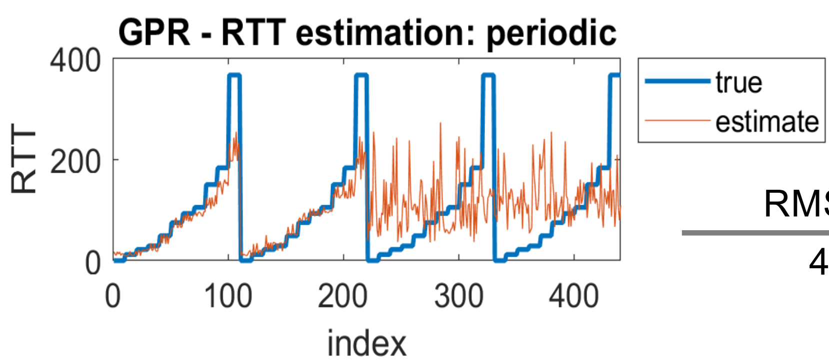


Gaussian process regression

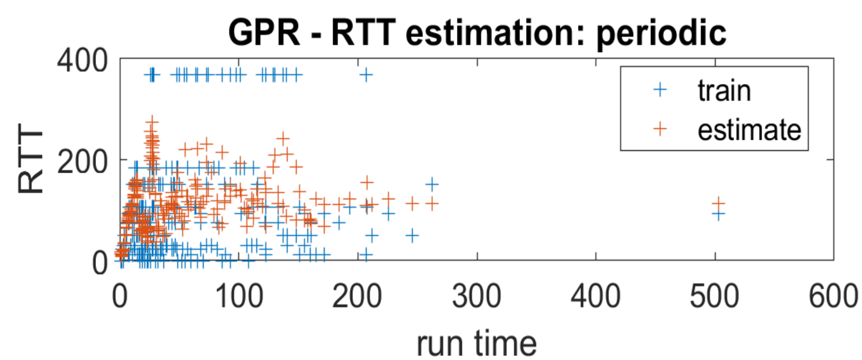


RMS error:
3.92

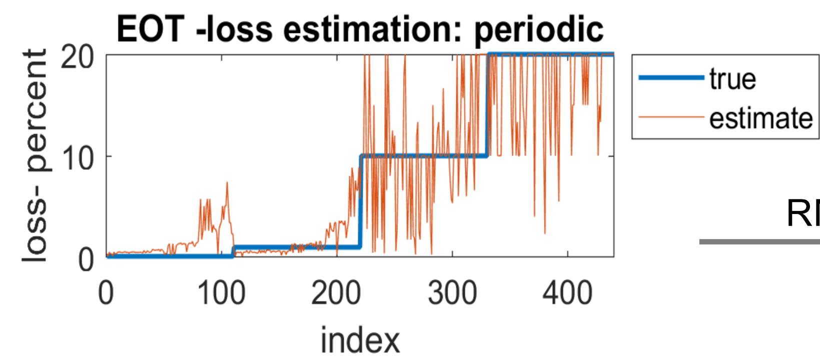




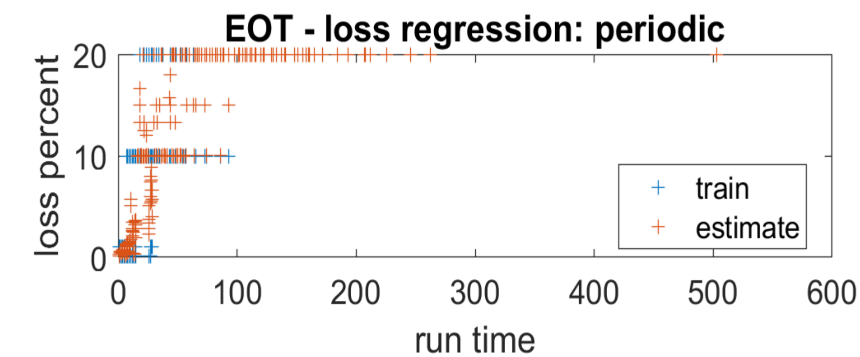
RMS error:
48.76

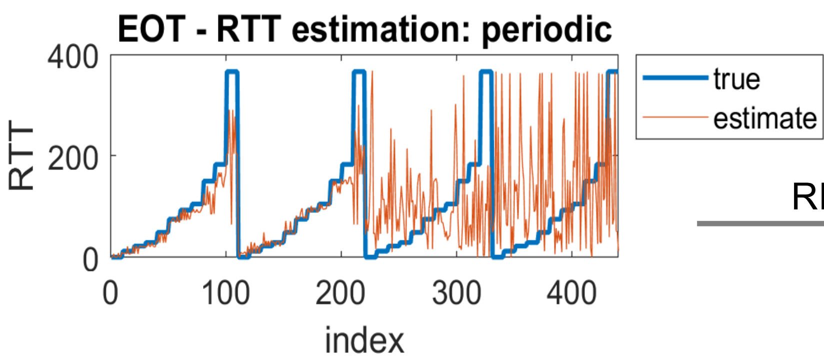


Ensemble of trees

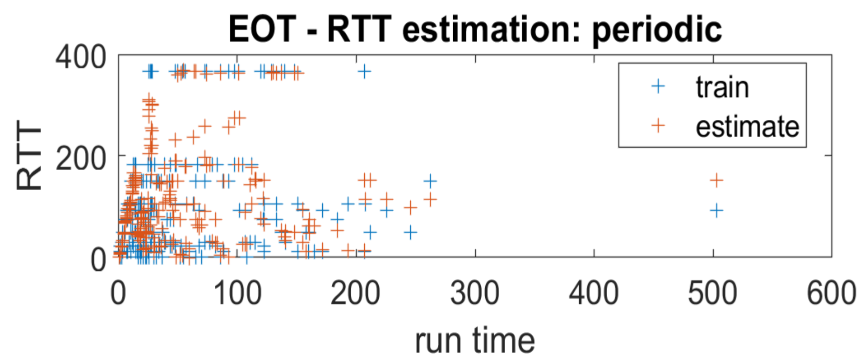


RMS error:
4.14

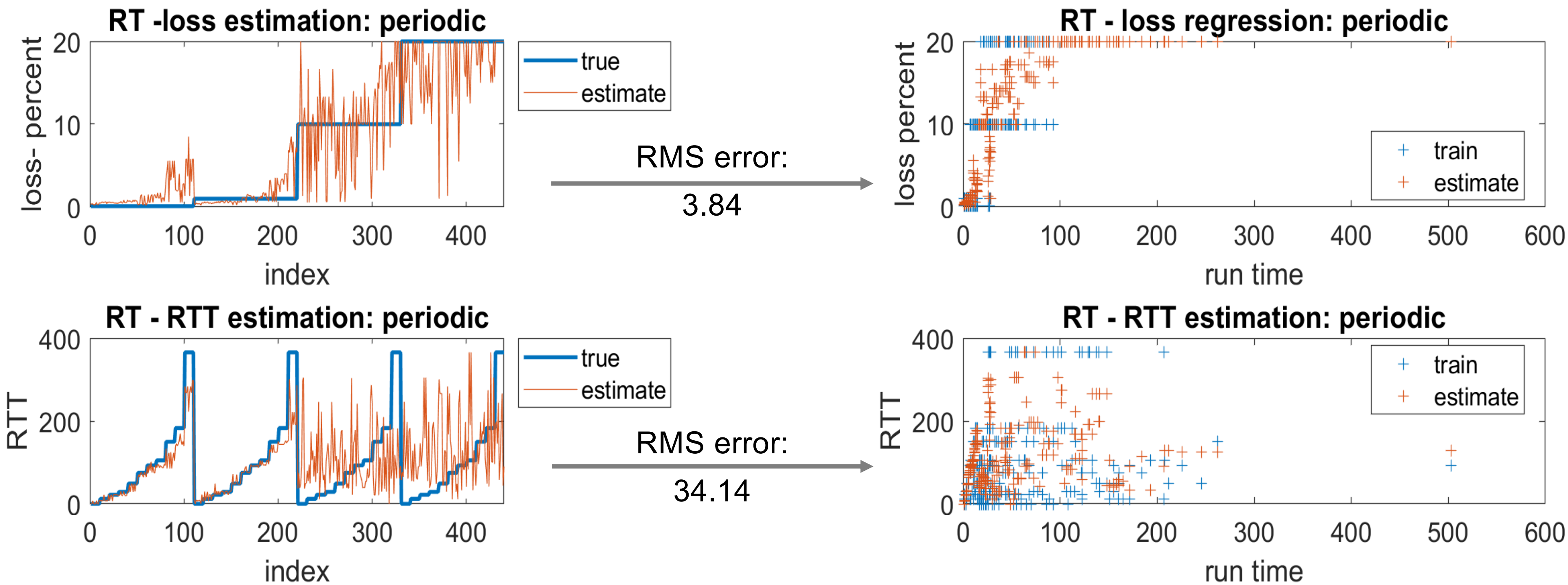




RMS error:
102.55



Regression trees



Root mean square error of ML methods

RTT

method	EOT	GPR	RT	LR	SVM
CUBIC no loss - rms	22.56	18.05	20.71	16.68	16.88
HTCP no loss - rms	8.72	7.32	6.27	7.49	7.52
HTCP periodic loss - rms	52.67	48.76	34.14	102.55	105.24

method	EOT	GPR	RT	LR	SVM
periodic aggregate - rms	52.67	48.76	34.14	102.55	105.24
periodic 0.1, 1% loss - rms	9.19	10.66	8.85	54.13	42.77
periodic 10, 20% loss - rms	84.06	75.39	69.76	135.79	142.42

Loss rate

method	EOT	GPR	LR	RT	SVM
percent - rms	4.14	3.92	6.19	3.84	6.23

Summary

- Studied strengths & limitations of ML methods in estimating network-level parameters using application-level measurements
- Ensemble of Trees (EOT), Gaussian Process Regression (GPR), Linear Regression (LR), Regression Trees(RT), and SVM
  - Accurate estimates at low loss rate (1%) but inaccurate at higher rates
  - Non-linearity better estimates but smoothness has mixed results
- MPI codes on federated supercomputers
  - Mapping based on long-term statistics or dynamic estimation
- Future Directions:
  - Performance under random network losses (uniform, poisson, gaussian)
  - Detailed performance analysis of ML methods: Vapnik's generalization

Acknowledgements

This work is funded by RAMSES Project, Office of Advanced Computing Research, U.S. Department of Energy, performed at Oak Ridge National Laboratory managed by UT-Battelle under Contract No. DE-AC05-00OR22725. This work is also supported by the United States Department of Defense (DoD) and used resources of the DoD Computational Research and Development Programs at Oak Ridge National Laboratory.

MPI – A Distance Scalability Perspective

MPI has been predominantly used in single facility systems with connections over

- Ethernet LAN
- Infiniband (IB) network
- custom interconnects of supercomputers

IB networks deployed in HPC facilities

- subject to 2.5ms timeout - effective to tens of miles

Longer connections: Ethernet

- Transmission Control Protocol (TCP) is a primary transport mechanism utilized by MPI

Motivation

- Recent experimental results show MPI primitives scale to round-the-earth distances
  - Rao et al, Syscon2019
- Execution times of MPI communications operations, hence entire computation, depend on connection RTT and loss rate
  - highly non-linear TCP lead to complicated dynamics of execution times
- **Question addressed here**: How to “invert” the measured execution times to estimate connection RTT and loss rate?

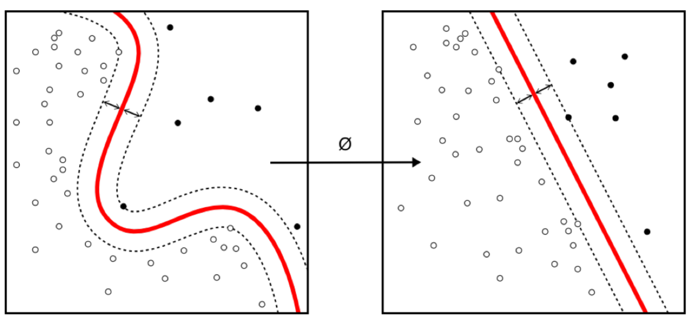
Five Machine Learning Methods

Represent different design principles – no rigorous way to identify universally best method

- Linear regression (LR): baseline estimator
- Four non-linear estimates:
  - two smooth: Support Vector Machine (SVM) and Gaussian Process Regression (GPR)
  - two non-smooth: Ensemble of Tree (EOT) and Regression Trees (RT)

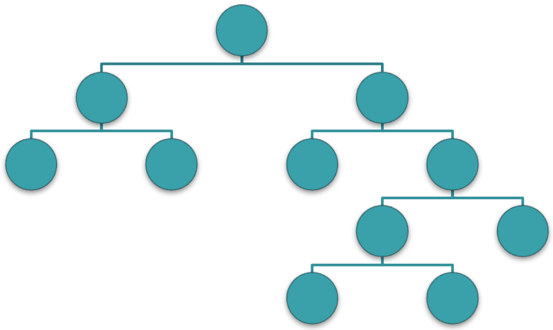
Support Vector Machine and Gaussian Process Regression

- smooth estimator
- non-linear mapping from input space into destination space



Ensemble of Tree and Regression Trees

- non-smooth estimator
- mapping input dataset into collection of trees



Results based on matlab toolbox

OAK RIDGE National Laboratory

Regression Estimation Formulation

R: random variable representing RTT

L: random variable representing loss rate

E: random variable representing execution time of MPI primitive code

- distributed according to $\mathbb{P}_{E,R,L}$
  - quite complex and typically unknown distribution – depends on various properties
    - network connection: RTT, loss rate
    - host system: CPUs, NIC
    - software stack: OS, network and MPI modules
- *RTT-regression function*: expected value of RTT at *E* = *e* given by

$$f^R(e) = \int R d\mathbb{P}_{R,L|e}$$

which is averaged over both *R* and *L*

  - its estimate $\hat{f}_A^R$ using method $A \in \mathcal{A} = \{\text{EOT,GPR,LR,RT,SVM}\}$
- *Loss-regression function*: expected value of loss rate at *E* = *e* given by

$$f^L(e) = \int L d\mathbb{P}_{R,L|e}$$

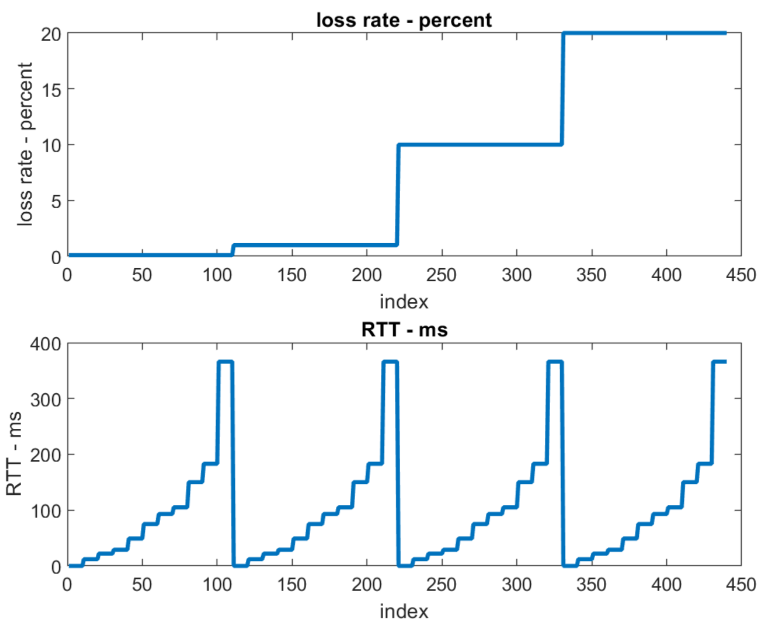
  - Its estimate $\hat{f}_A^L$ using method $A \in \mathcal{A} = \{\text{EOT,GPR,LR,RT,SVM}\}$

Methods:
Ensemble of Tree (EOT)
Gaussian Process Regr. (GPR)
Linear regression (LR)
Regression Trees (RT)
Support Vector Machine (SVM)

Estimates Presented as Composite Plots

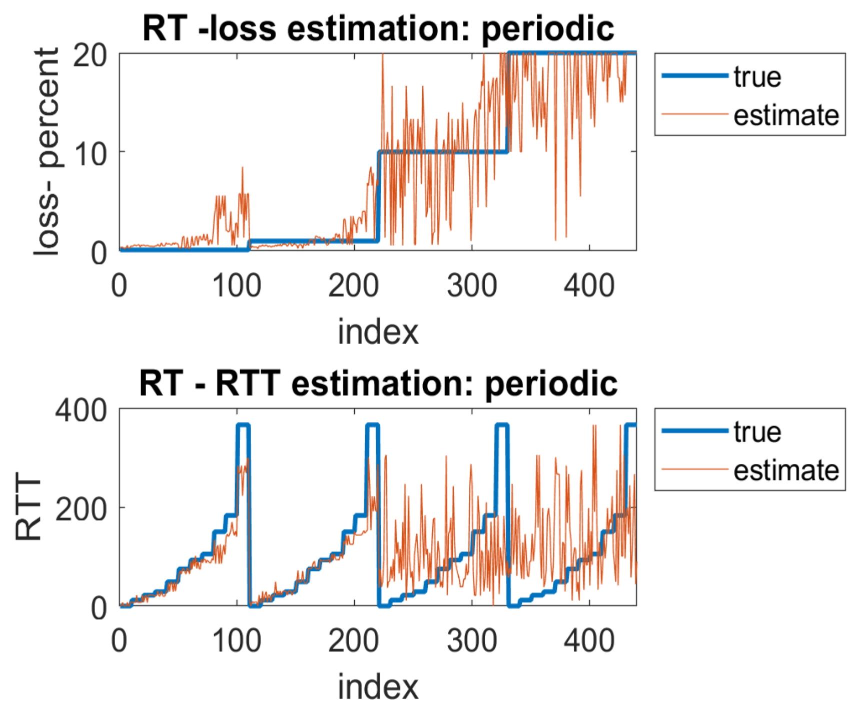
Composite Plot: Provides a snapshot of the estimate at all measurements

- loss rate: increased left to right
- at each loss rate, RTT increased left to right
- at each RTT, measurements are repeated 10 times



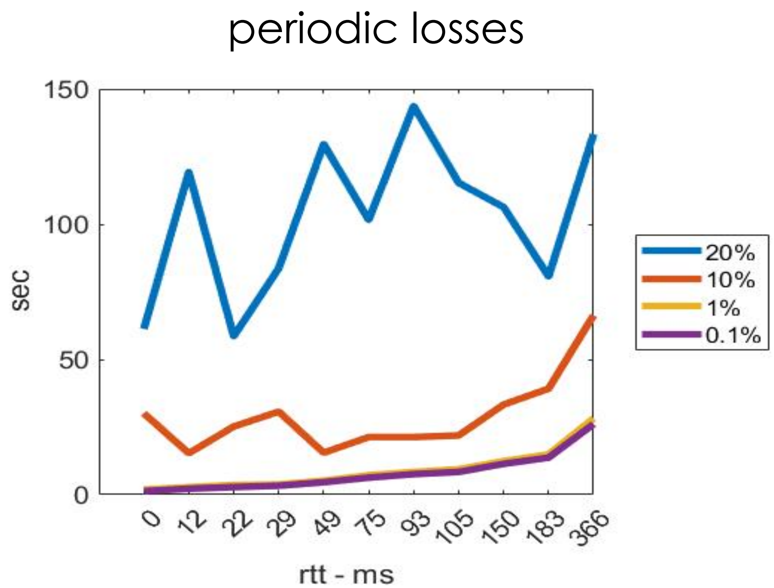
composite plot: RTT

Regression Tree (RT) Estimates



OAK RIDGE
National Laboratory

Measurements Summary: Average Execution times: different loss rates



External packet losses are introduced by ANUE/Ixia emulator

Observations:

- Low loss rates: execution times determined primarily by RTT
- Higher loss rates: execution times dominated by loss rates - result of TCP loss recovery response:
    - higher loss rates: few code executions are not completed due to communications time-out

Results Summary

Study of strengths and limitations of machine learning methods to:

- Estimate connection RTT and loss rate using execution times of MPI Sendrecv operations
- Using measurements over 10Gbps emulated connections with
  - 0-366ms Round-Trip Times (RTT)
    - represent connection lengths ranging from local to round the earth distances
  - Additionally, externally introduced packet losses over these connections

Machine Learning Methods:

- Ensemble of Trees (EOT), Gaussian Process Regression (GPR), Linear Regression (LR), Regression Trees(RT), and Support Vector Machines (SVM)
- Disparate design properties: (a) linear and non-linear, and (b) smooth and non-smooth

Provide useful qualitative insights into strengths and limitations of machine learning methods:

- Low loss rates: execution times determined primarily by RTT and accurately estimated
- Higher loss rates: execution times dominated by complex, non-linear TCP response:
  - loss rates of 10% or higher: show limitations of machine learning methods

Conclusions

Summary: Studied strengths and limitations of machine learning methods for estimating network parameters using application-level measurements:

- Recent results: distance-scalability of MPI over long connections under external packet losses
- In this paper: machine learning methods can be applied in principle to estimate connection RTT and loss rate using execution time measurements of MPI SendReceive operations
- Datasets: MPI_measurements over 10Gbps Ethernet emulated connection with 0-366 RTT under periodic losses up to 20%
- Conclusions: Accurate estimates under low loss rate (1%) but inaccurate at higher rates
  - Non-linearity provides better estimates but smoothness has mixed results: RT and GPR are top two

Implications: MPI codes on federated supercomputers located sites across the globe

- Codes account for latency and loss rates – mapping based on long-term statistics or dynamic estimation

Future Directions:

- Performance under random network losses
  - uniform, Poisson and Gaussian – to be presented at MLN2019
  - more realistic scenarios
- Detailed performance analysis of machine learning methods: Vapnik's generalization