# PREDICTING APPLICATION PERFORMANCE USING SUPERVISED LEARNING ON COMMUNICATION FEATURES

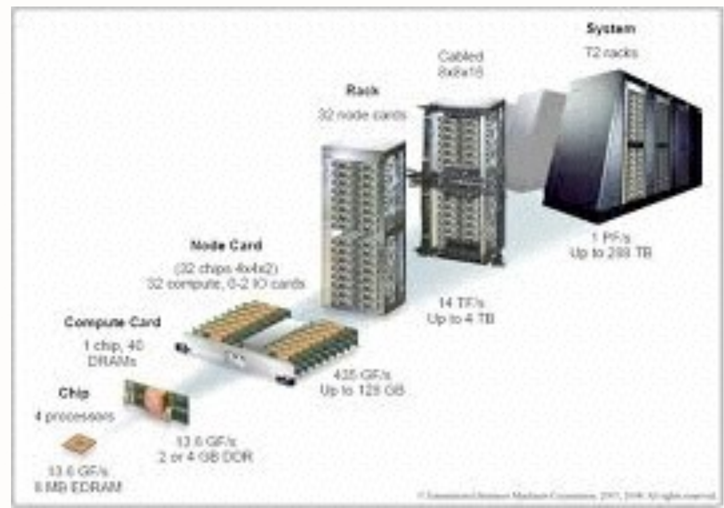
Nikhil Jain*, Abhinav Bhatele†,
Michael P. Robson*, Todd Gamblin†, Laxmikant V. Kale*

*University of Illinois at Urbana-Champaign
†Lawrence Livermore National Laboratory

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work was funded by the Laboratory Directed Research and Development Program at LLNL under project tracking code 13-ERD-055. LLNL IM Release:

# SUPERCOMPUTERS

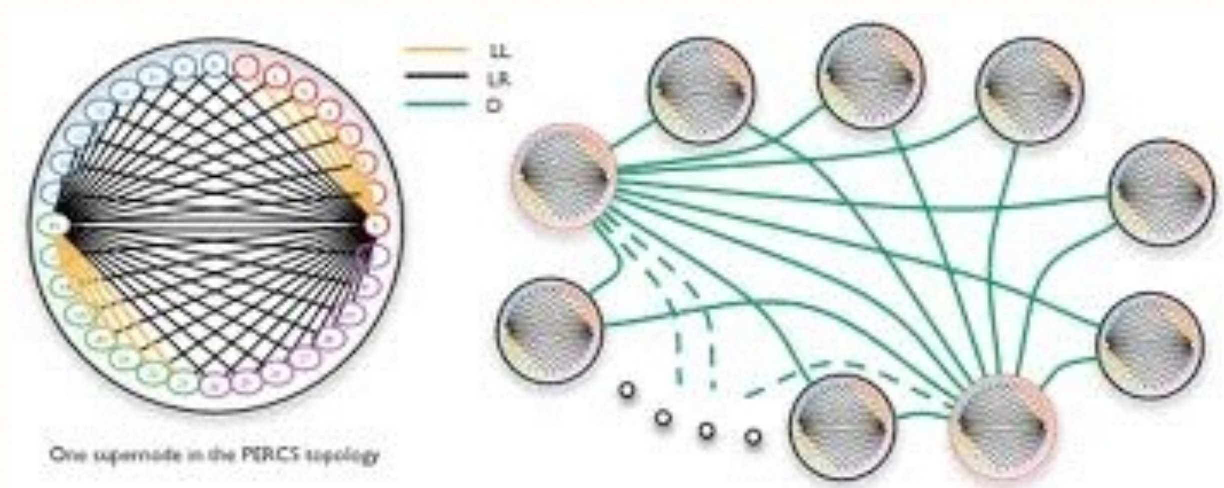

48 GB/s, 1-2 $\mu$s



40 GB/s, 1-3 $\mu$s


150 GB/s, 0.8 $\mu$s

Higher Bandwidth
Lower Latency
Fewer hops

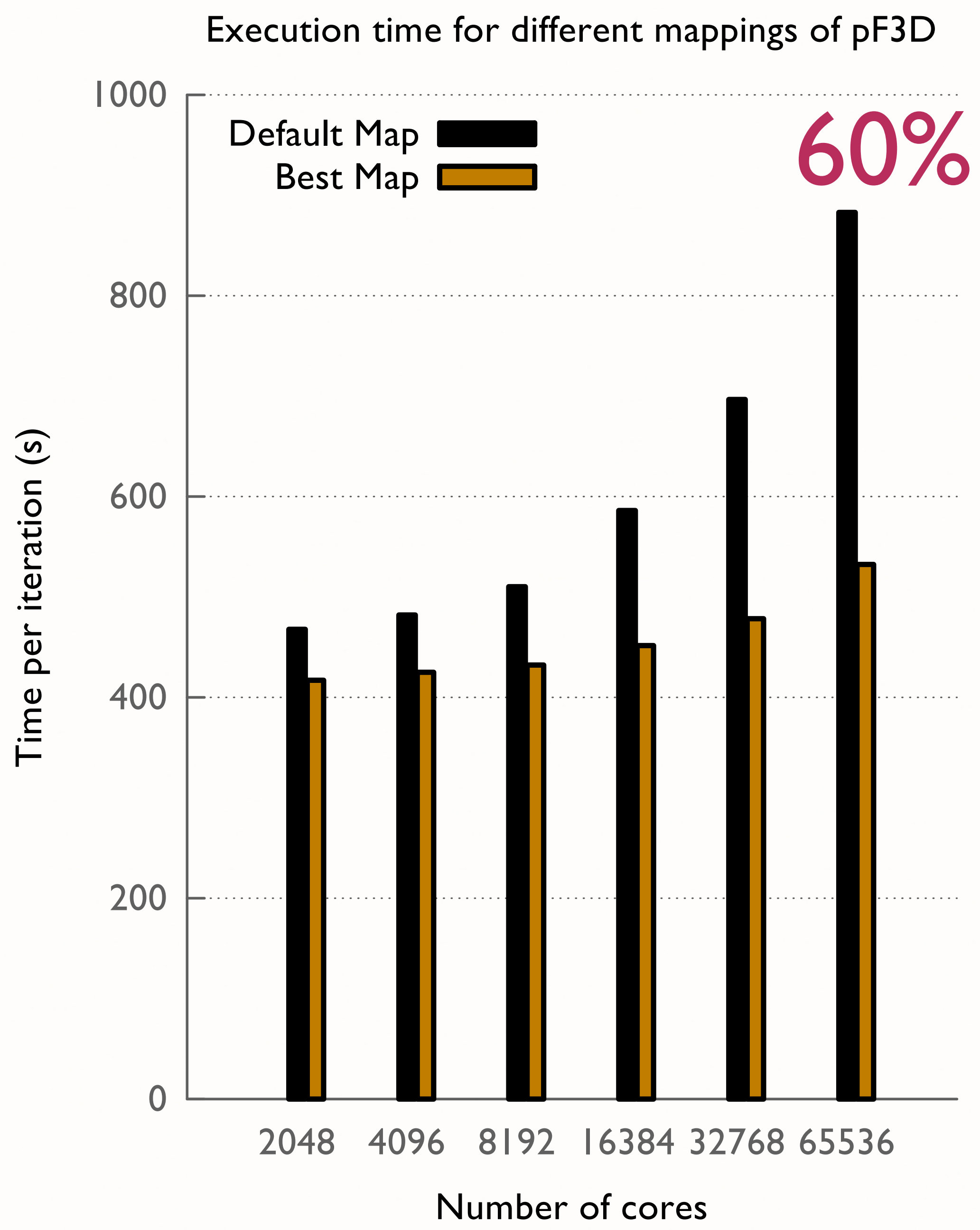



420 GB/s, 1-2 $\mu$s

# WHY STUDY NETWORK PERFORMANCE?

- Peak bandwidth and latency are never obtained in presence of congestion
- High raw bandwidth does not guarantee proportionate observed performance
  - Topology, job interference, I/O
- Find the next generation topology
- Savings are proportionate to core-count

# QUANTIFYING IMPACT




- Mapping via logical operations in Rubik
- What about others mappings?
- How far are we from the best performance?
- Which is the best performing mapping?

A. Bhatele, et al Mapping applications with collectives over sub-communicators on torus networks. In Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC '12. IEEE Computer Society, Nov. 2012 (to appear). LLNL-CONF-556491.

# PERFORMANCE PREDICTION METHODS

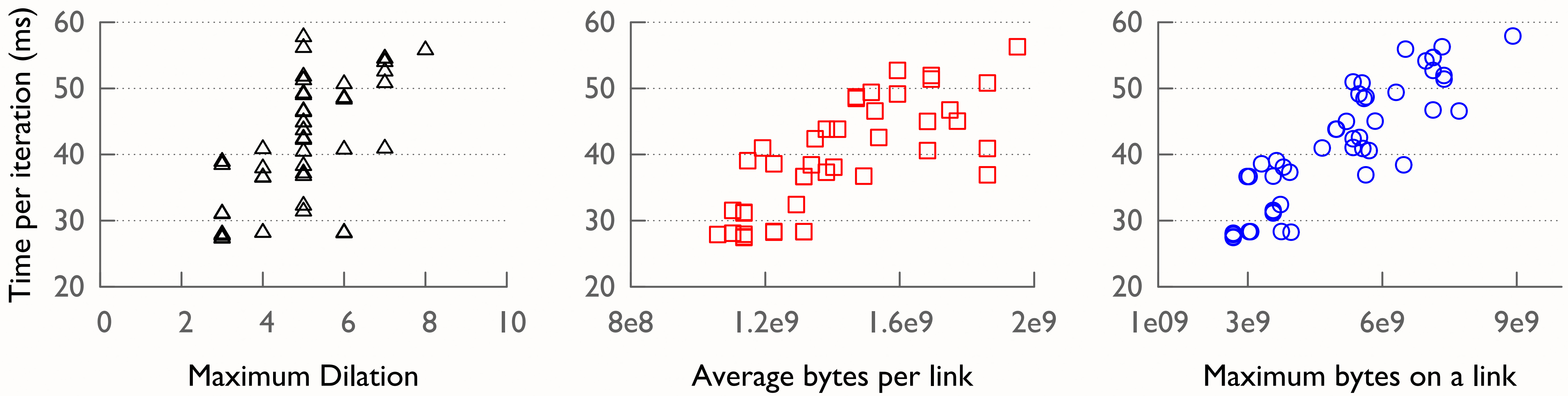
- Theoretically: NP hard
- Simulations: too slow
  - Few days to simulate one use case*
- Real runs: very expensive
  - Application/allocation specific information

| | 2012 | 2013 |
|---|---|---|
| Intrepid | 4.16M | 0.73M |
| Mira | 0.17M | 7.67M |
| Total | 4.33M | 8.40M |

13 million core hours!

*Abhinav Bhatele, Nikhil Jain, William D. Gropp, and Laxmikant V. Kale. 2011b. Avoiding hot-spots on two- level direct networks. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC '11)*. ACM, New York, NY, USA, 76:1–76:11.

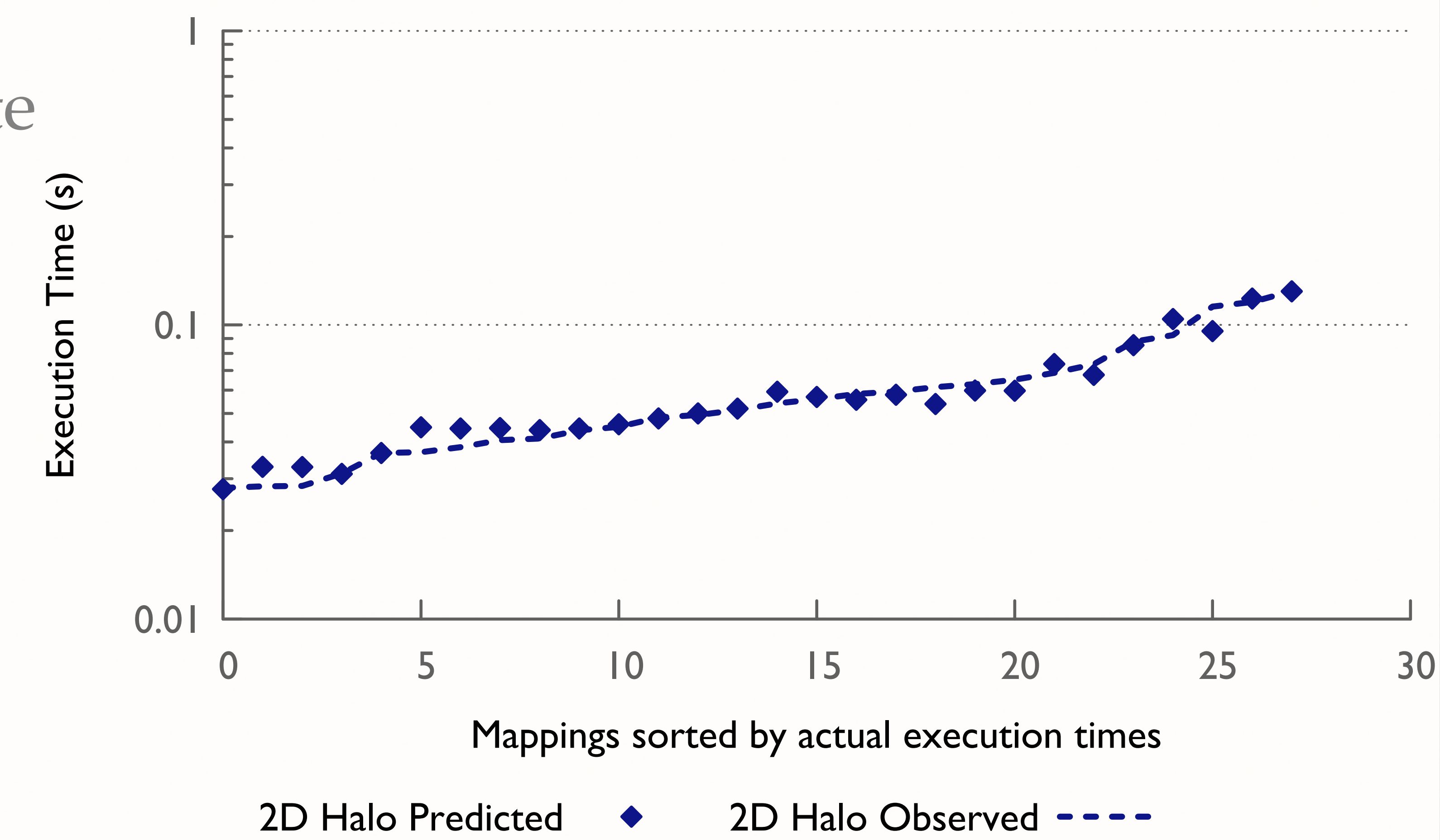
# HEURISTICS
# PRIOR FEATURES



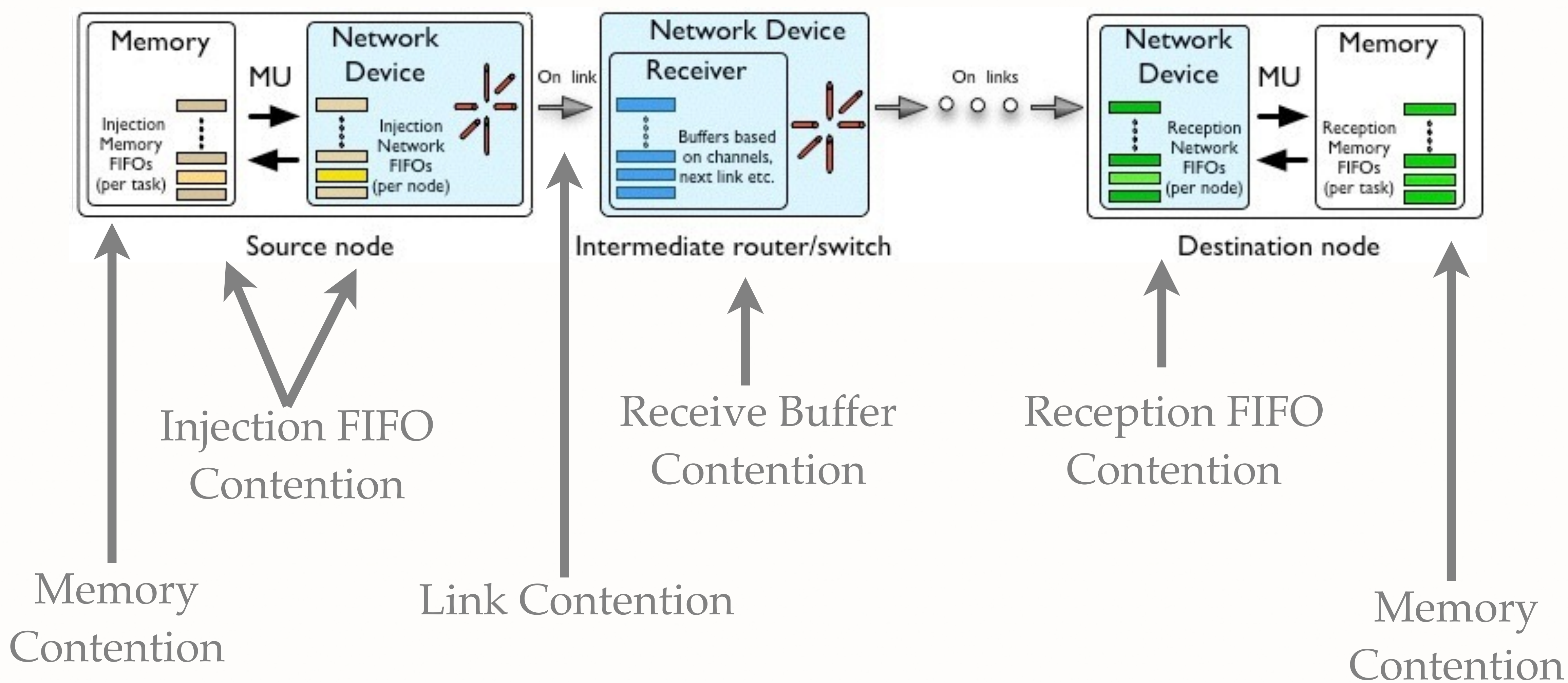

2D-Halo: predicting performance using a linear regression model for prior features

# SUPERVISED LEARNING: OVERVIEW

- Collect/generate data and summarize
- Build models: train performance prediction based on independent features
- Predict and correlate

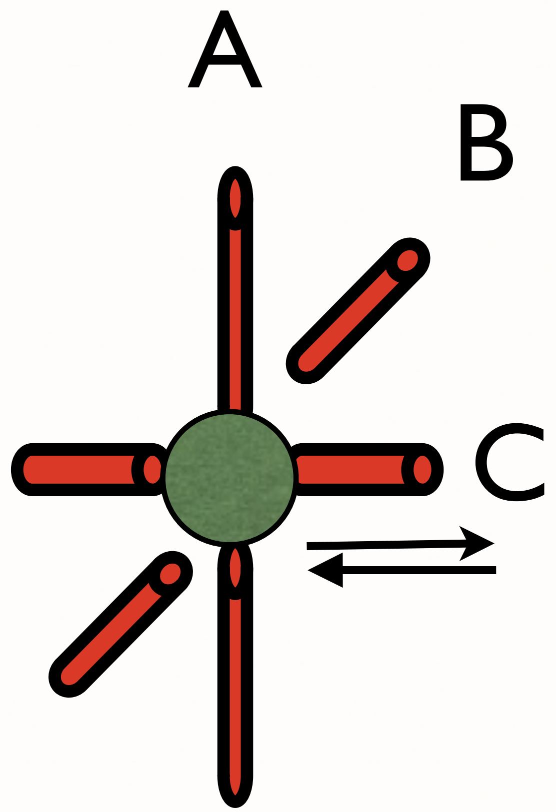


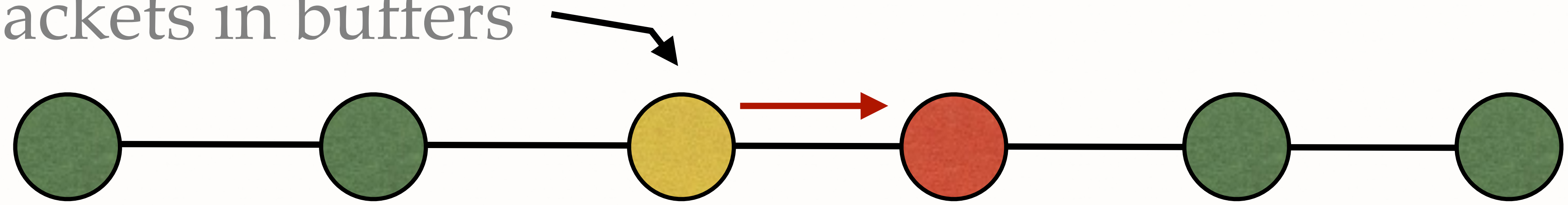
# MESSAGE LIFE CYCLE ON BLUE GENE/Q



# INPUT FROM NETWORK COUNTERS

- A PMPI based BG/Q-Counter collection module
- Packets sent on links in specific directions: A, B, C, D, E
  - deterministic, dynamic
- Packets received on a link
- Packets in buffers



# INPUT FROM SIMULATION

- Simulate the injection mechanism
  - Selection of memory injection FIFO
  - Mapping of memory FIFO to network injection FIFO
- Simulate routing to obtain hops/dilation

# INPUT DATA

| Indicator | Source | Derived from |
| --- | --- | --- |
| Bytes on links | Counters | Sent chunks |
| Buffer length | Counters | #Packets in buffers |
| Delay per link | Counters | #Packets in buffers/ #received packets |
| Dilation | Analytical | Shortest path routing |
| FIFO length | Analytical | Based on PAMI |

# BUILDING MODEL

- Derive features from the raw data on entities, e.g. average bytes on links
- Create a database of derived features and performance; we have used 100 mappings
  - 33% mappings generated randomly
  - 33% using Rubik
  - Rest are based on better performing mappings
- Select two-third entries as training set:
  - Derived features are independent variables
  - Performance is a dependent variable

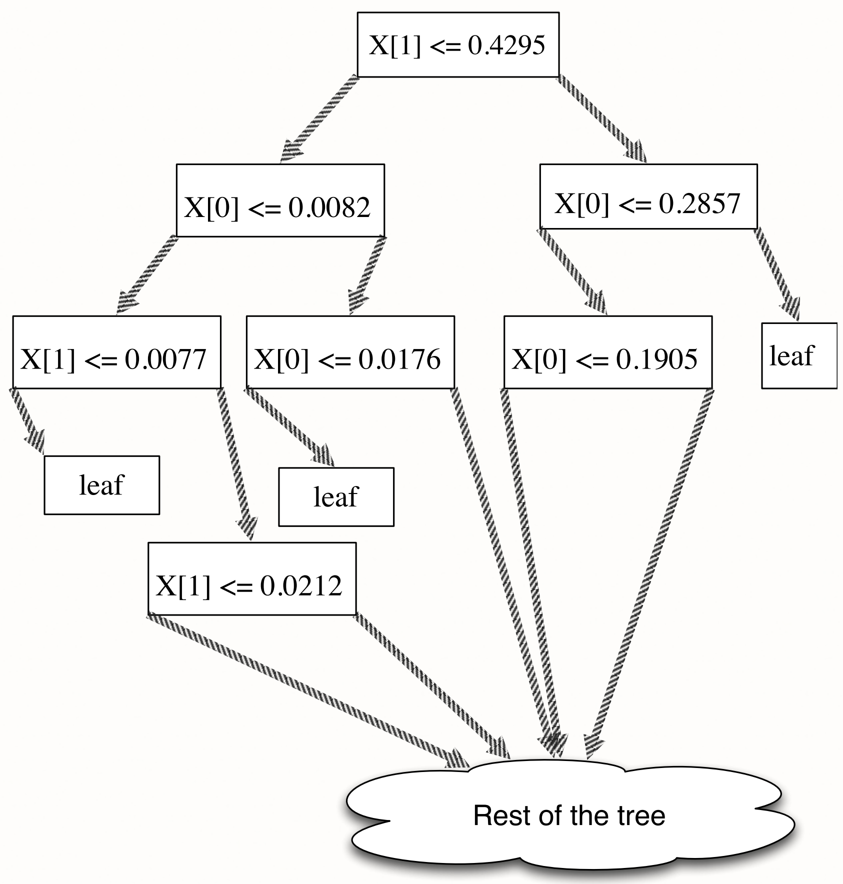
# BUILDING MODEL

- The training set is used to create a model for prediction
- Remaining entries from the database are used as the test set - derived features as input
- Prediction is compared with observed values
- Experimented with a large number of algorithms - linear, bayesian, SVM, near-neighbors, etc.

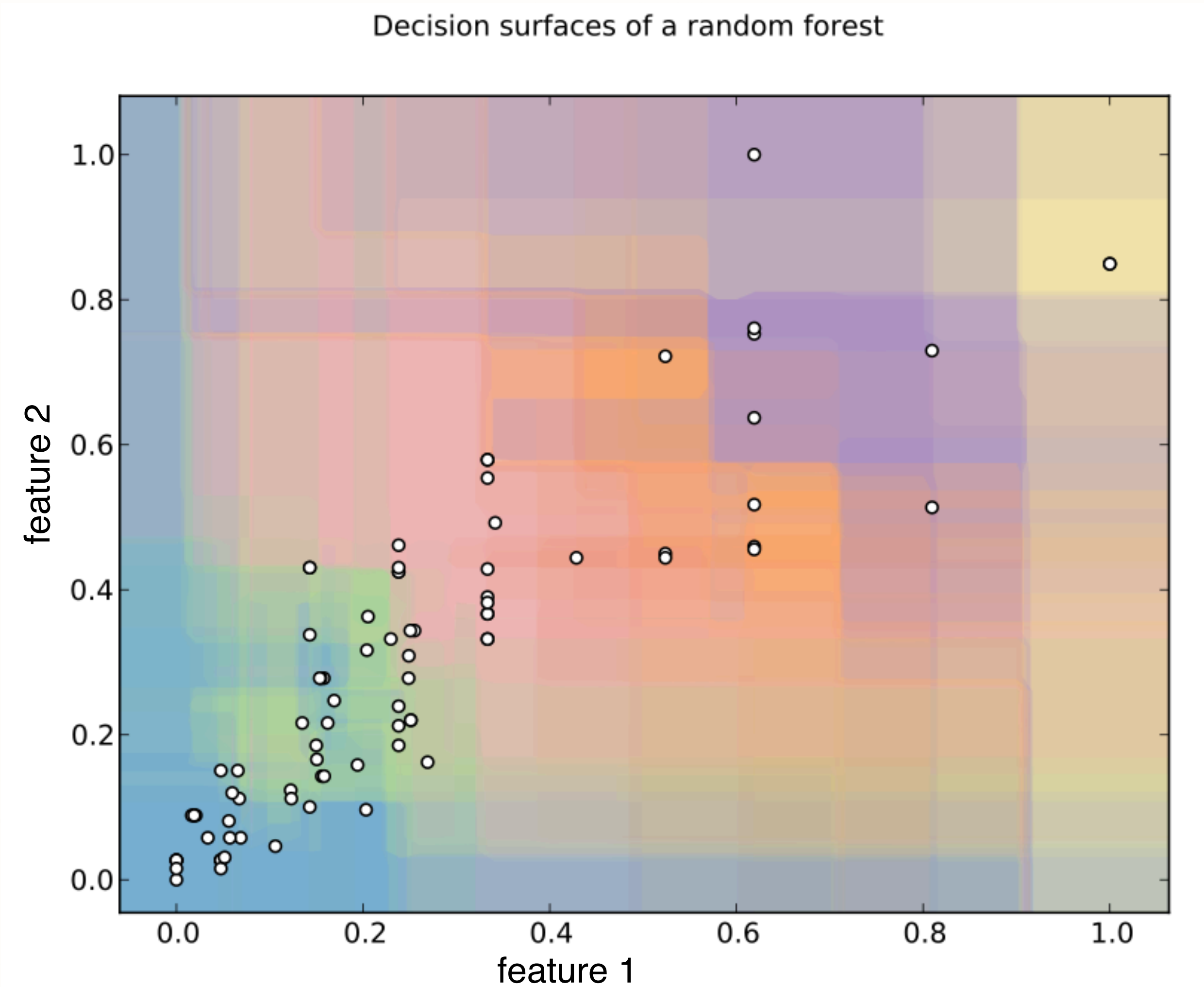
scikit learn http://scikit-learn.org

# LEARNING ALGORITHM

## Decision trees




## Randomized forest of trees




L. Breiman. Random forests. Machine Learning, 45(1):5–32, 2001.

# HOW TO JUDGE A PREDICTION

- Rank Correlation Coefficient (RCC): fraction of the number of pairs of task mappings whose ranks are in the same partial order in predicted and observed performance list

$$concord_{ij} = \begin{cases} 1, & \text{if } x_i >= x_j \text{ \& } y_i >= y_j \\ 1, & \text{if } x_i < x_j \text{ \& } y_i < y_j \\ 0, & \text{otherwise} \end{cases}$$

$$RCC = \Big( \sum_{0<=i<n} \sum_{0<=j<i} concord_{ij} \Big) / \big( \frac{n(n-1)}{2} \big)$$

- Absolute Correlation

$$R^2(y, \hat{y}) = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

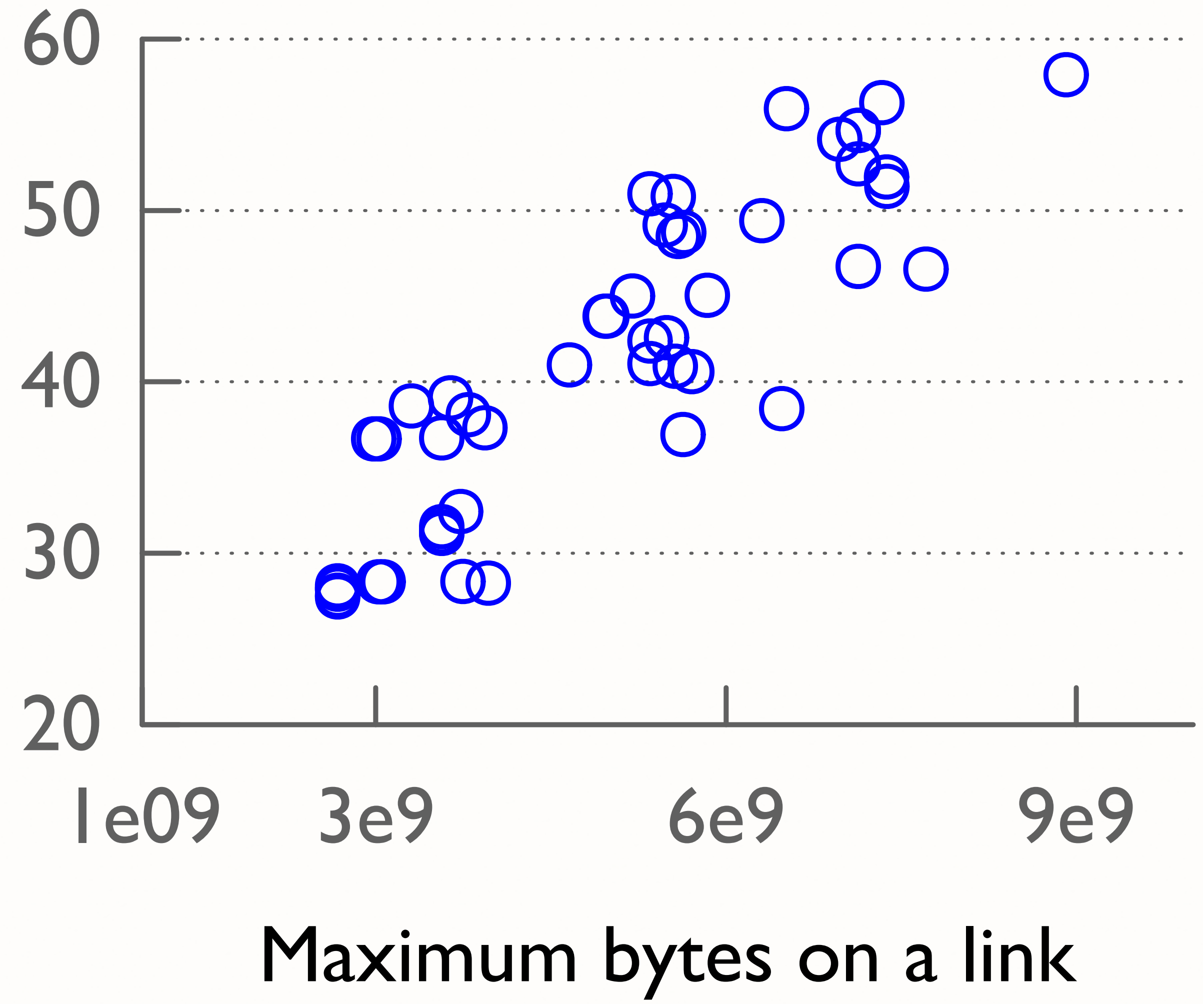
- Higher is better!

# RESULTS: SETUP

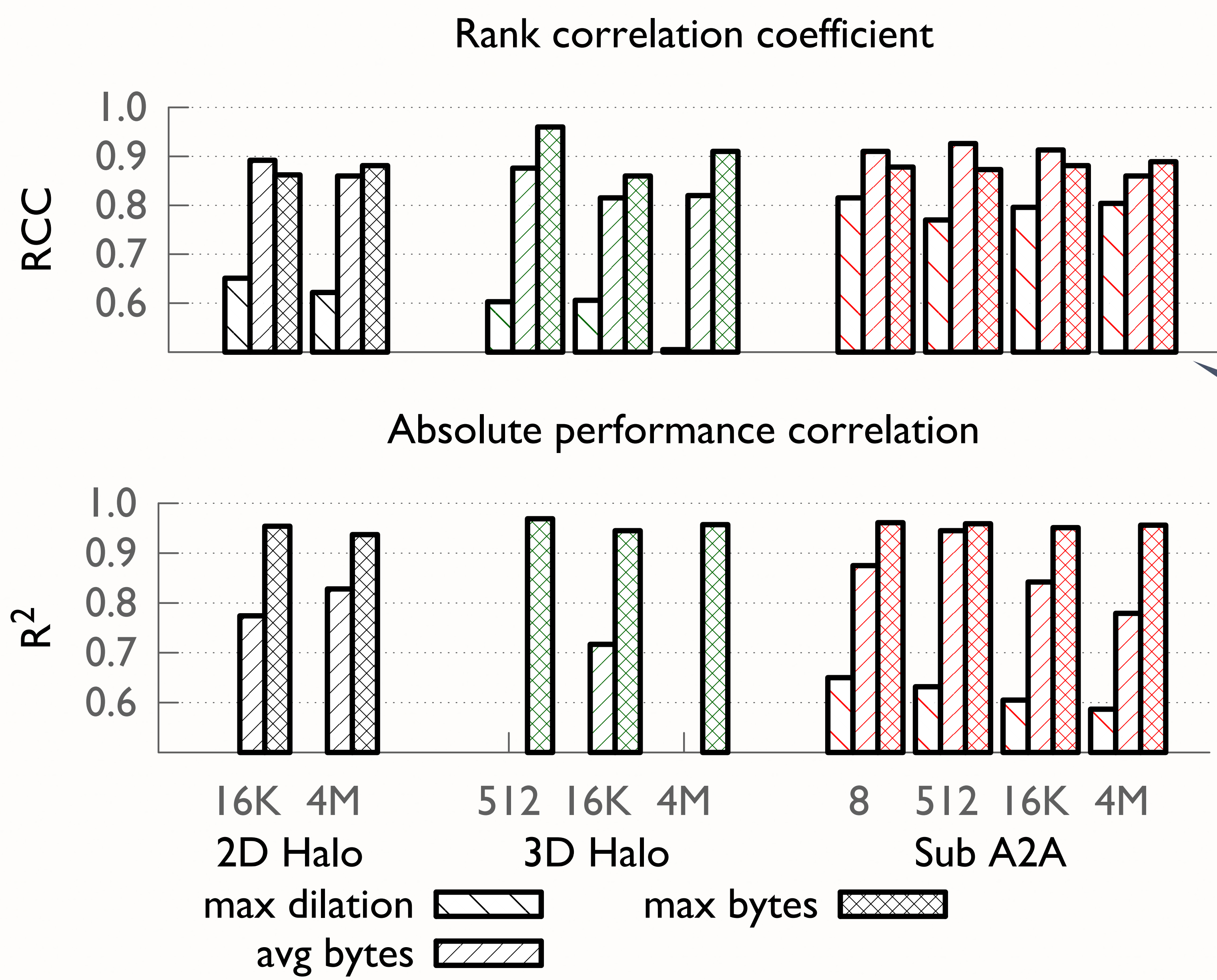
- Three communication kernels
  - Five-point 2D stencil
  - 14-point 3D stencil
  - All-to-all over sub-communicators
- Four message sizes to span MPI and routing protocols

# PRIOR FEATURES

- Entities
  - Bytes on a link
  - Dilation
- Derivation Methods
  - Maximum
  - Average
  - Sum



# RESULTS
# PRIOR FEATURES

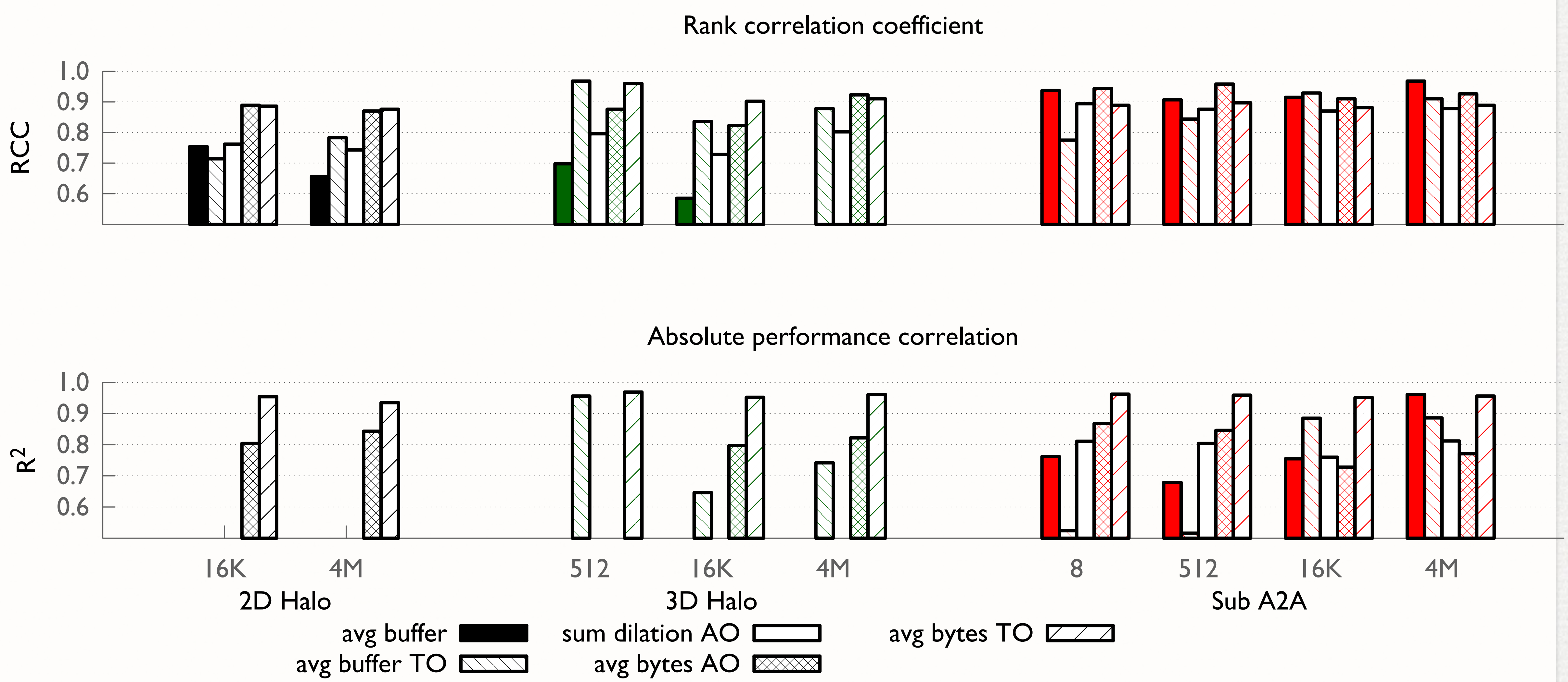



max bytes is good, but incorrect in 10% cases

# NEW FEATURES

- Entities
  - Buffer length (on intermediate nodes)
  - FIFO length (packets in injection FIFO)
  - Delay per link (packets in buffer/packets received)
- Derivation methods
  - Average Outliers (AO)
  - Top Outliers (TO)

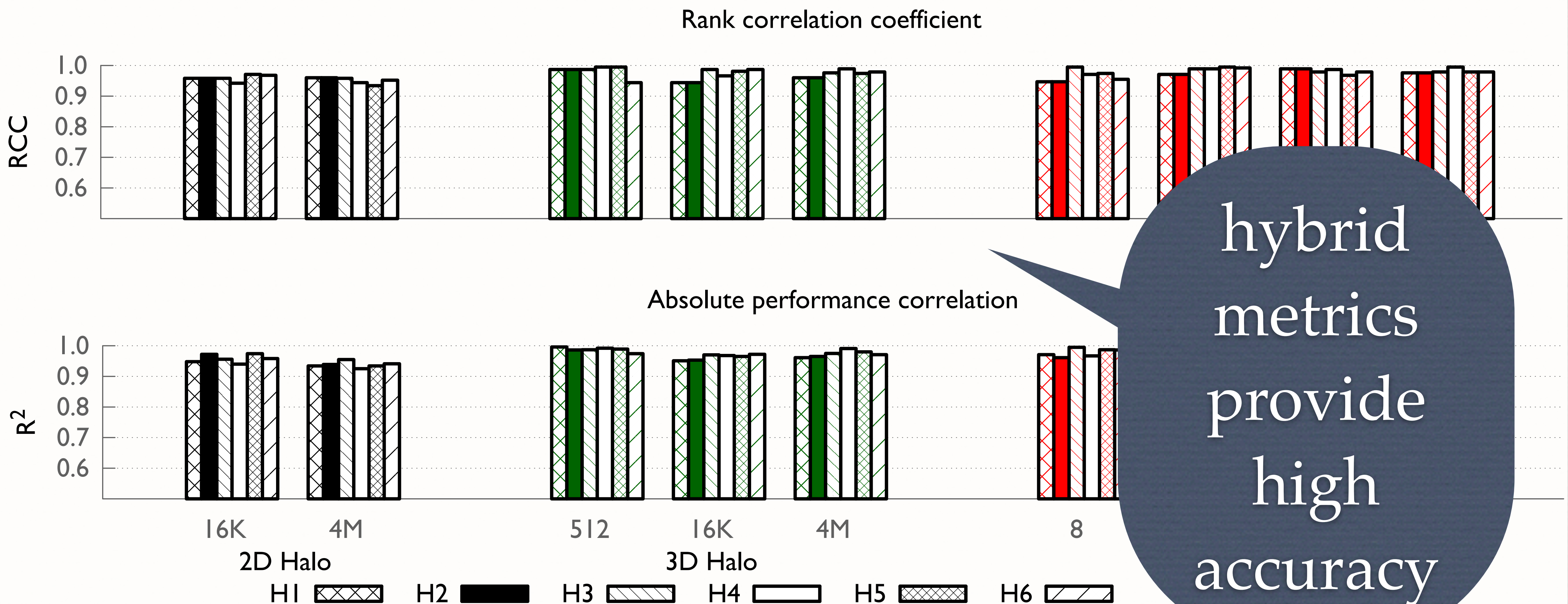
# RESULTS NEW FEATURES



# HYBRID FEATURES

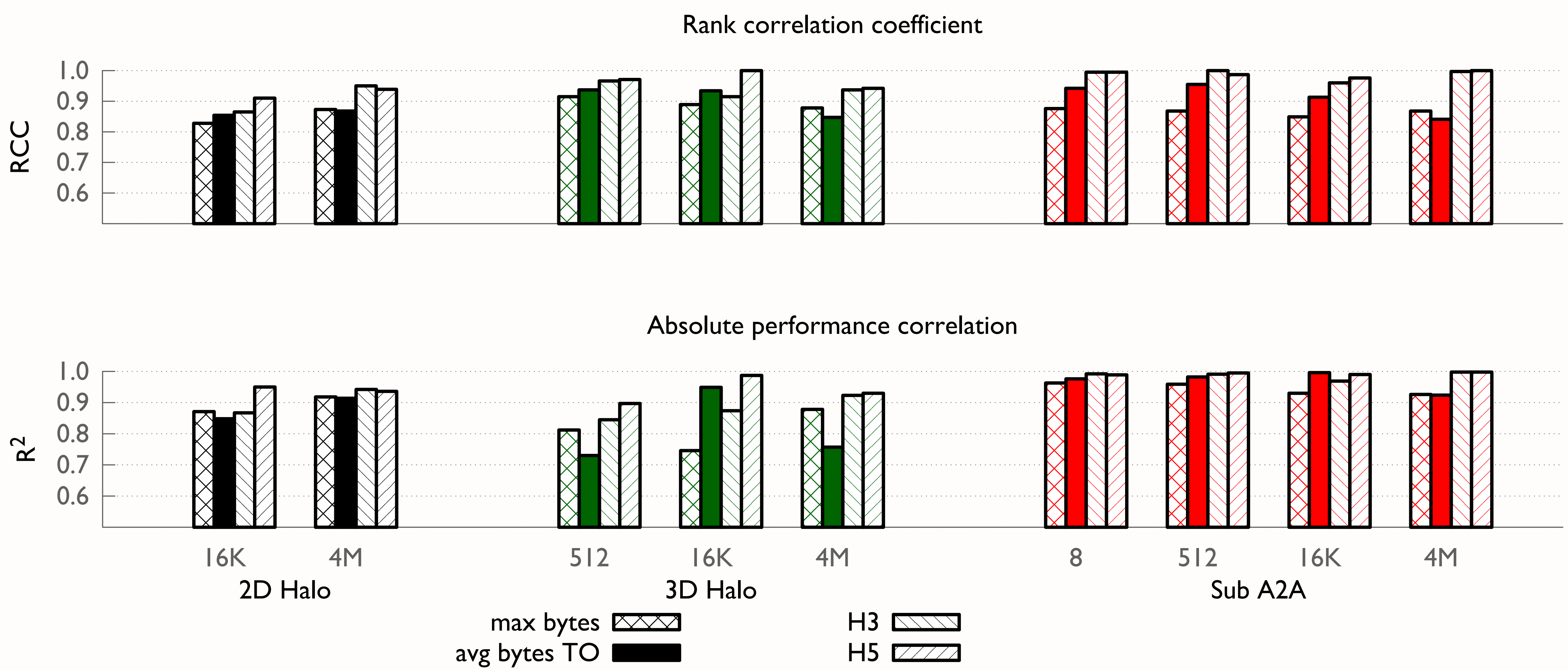
- Combine multiple metrics to complement each other
- Some combinations
  - H1: avg bytes + max bytes + max FIFO
  - H3: avg bytes + max bytes + avg buffer + max FIFO
  - H4: avg bytes + max bytes + avg buffer TO
  - H5: avg bytes TO + avg buffer TO + avg delay AO + sum hops AO + max FIFO

# RESULTS HYBRID FEATURES

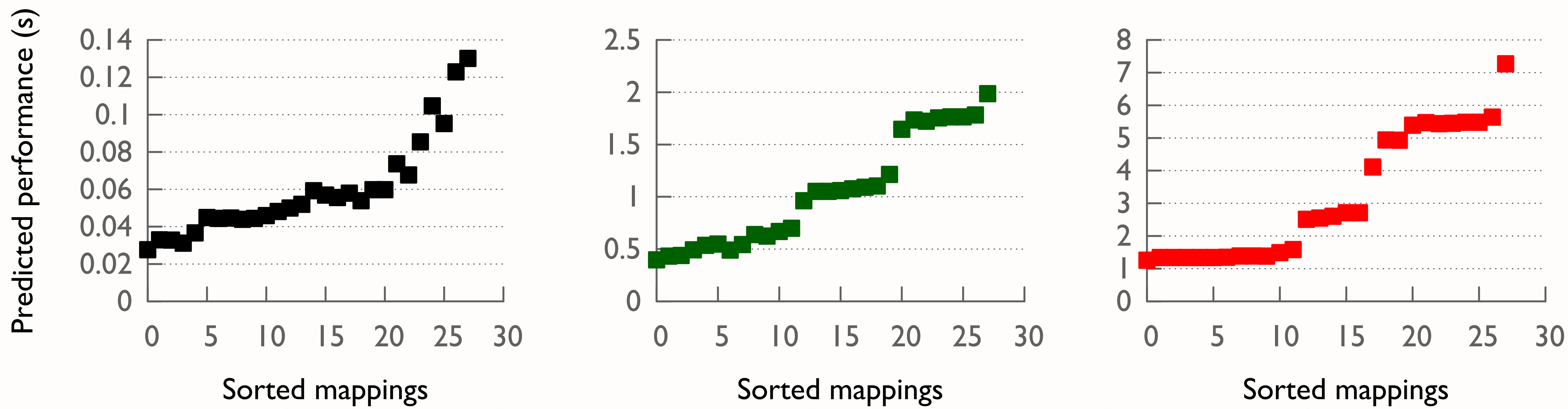



hybrid metrics provide high accuracy

# SUMMARY ON 64K CORES

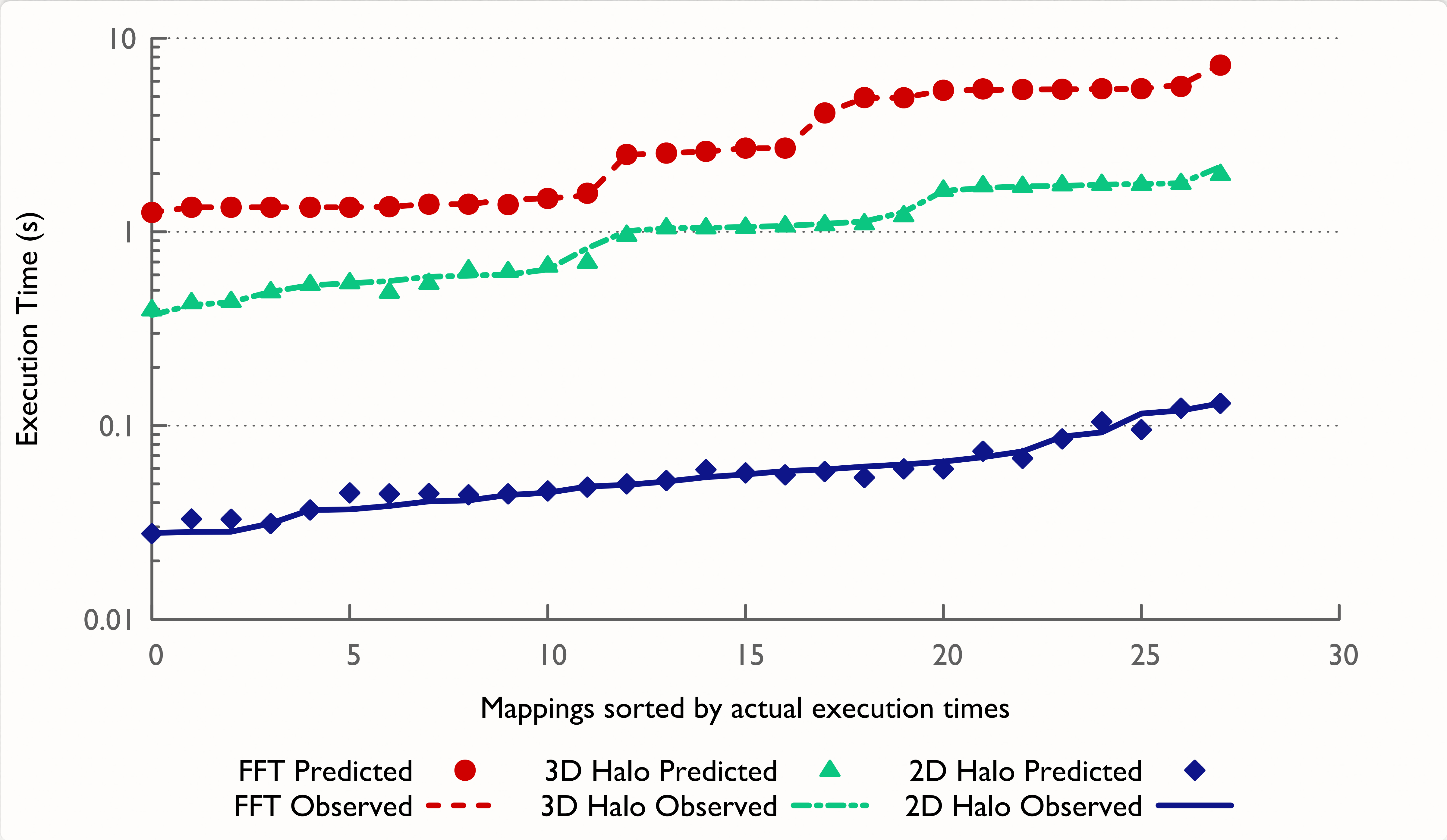


# RESULTS: TREND

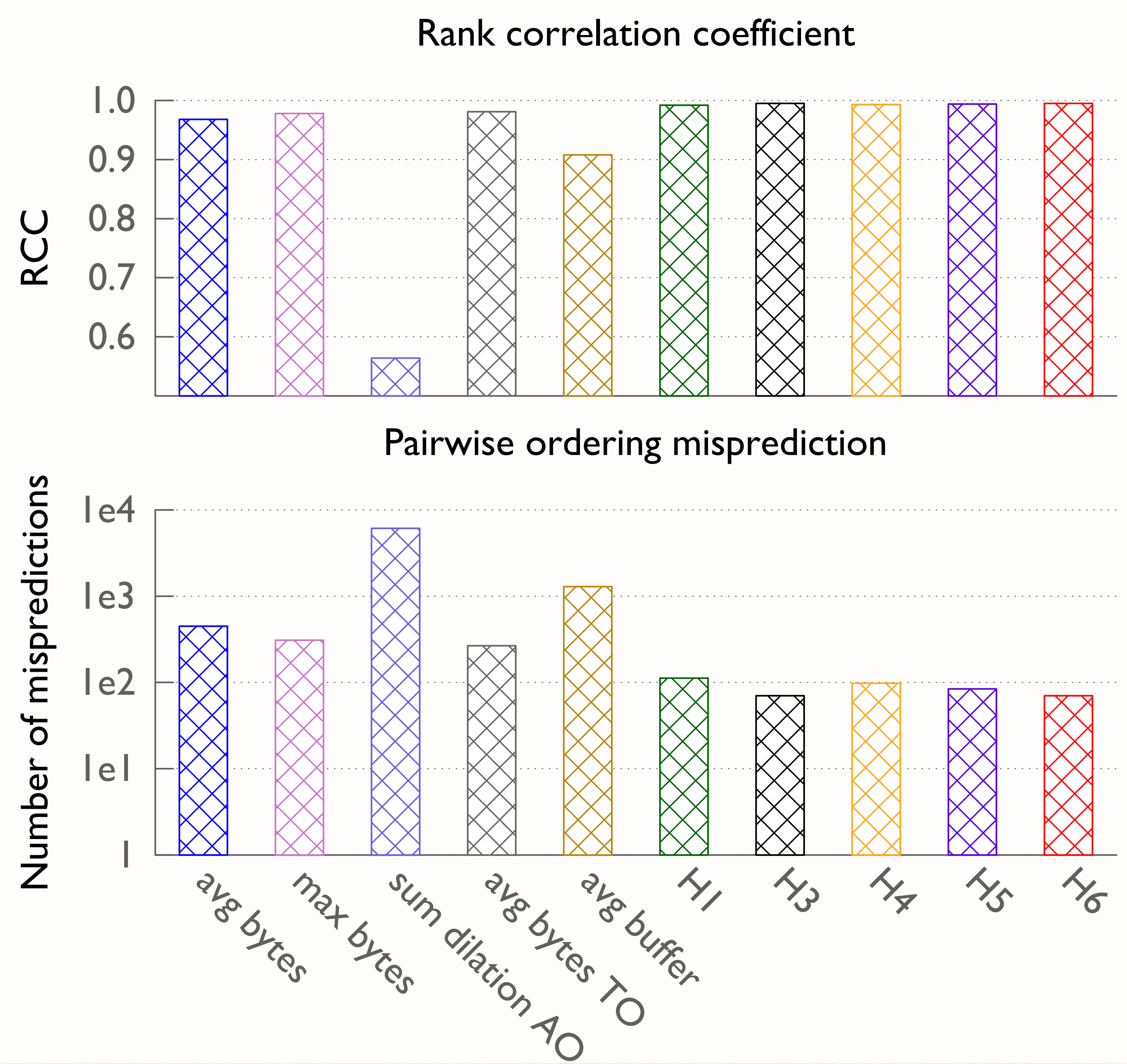



2D Halo 3D Halo Sub A2A

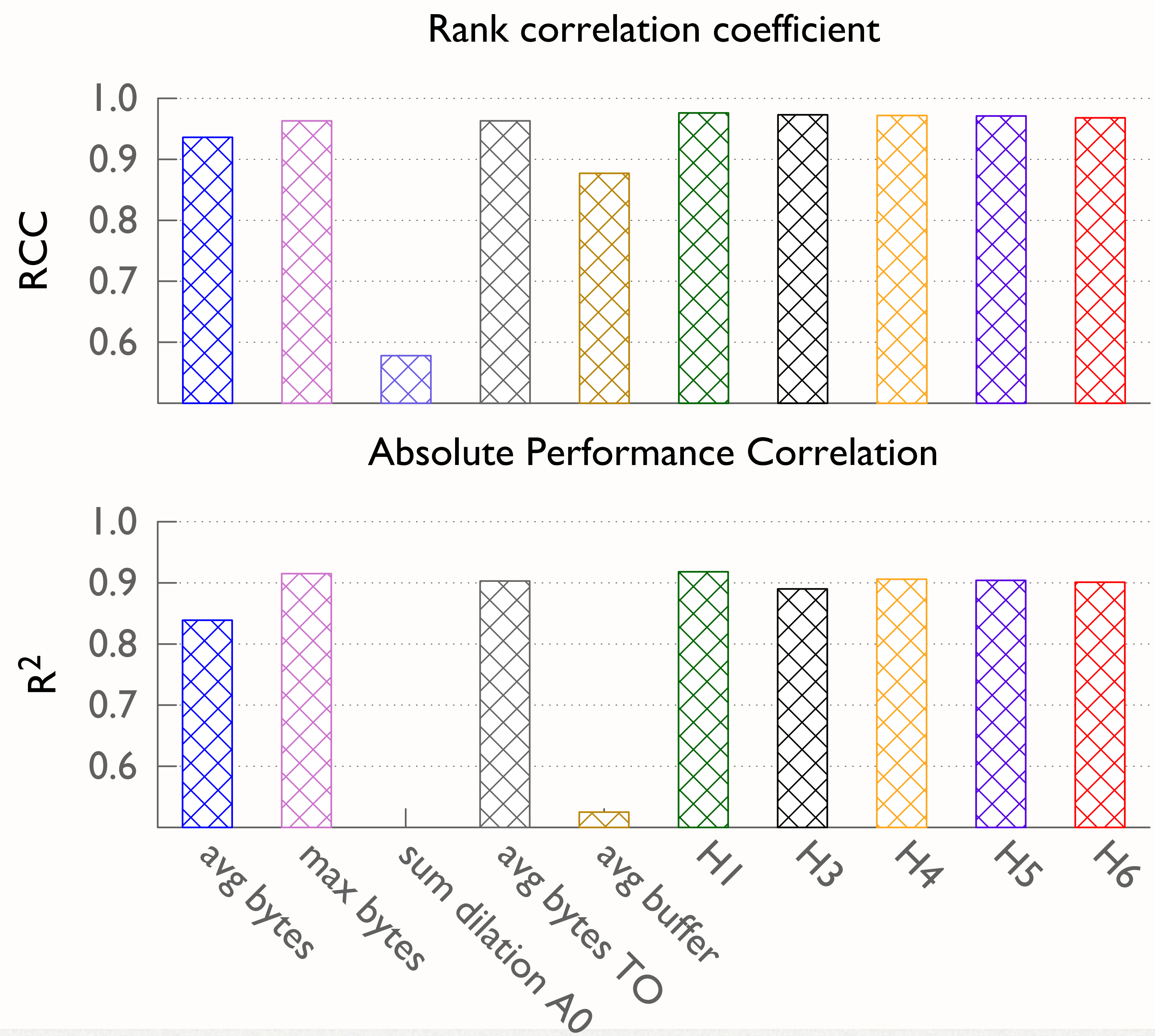
# RESULTS
# ABSOLUTE PERFORMANCE



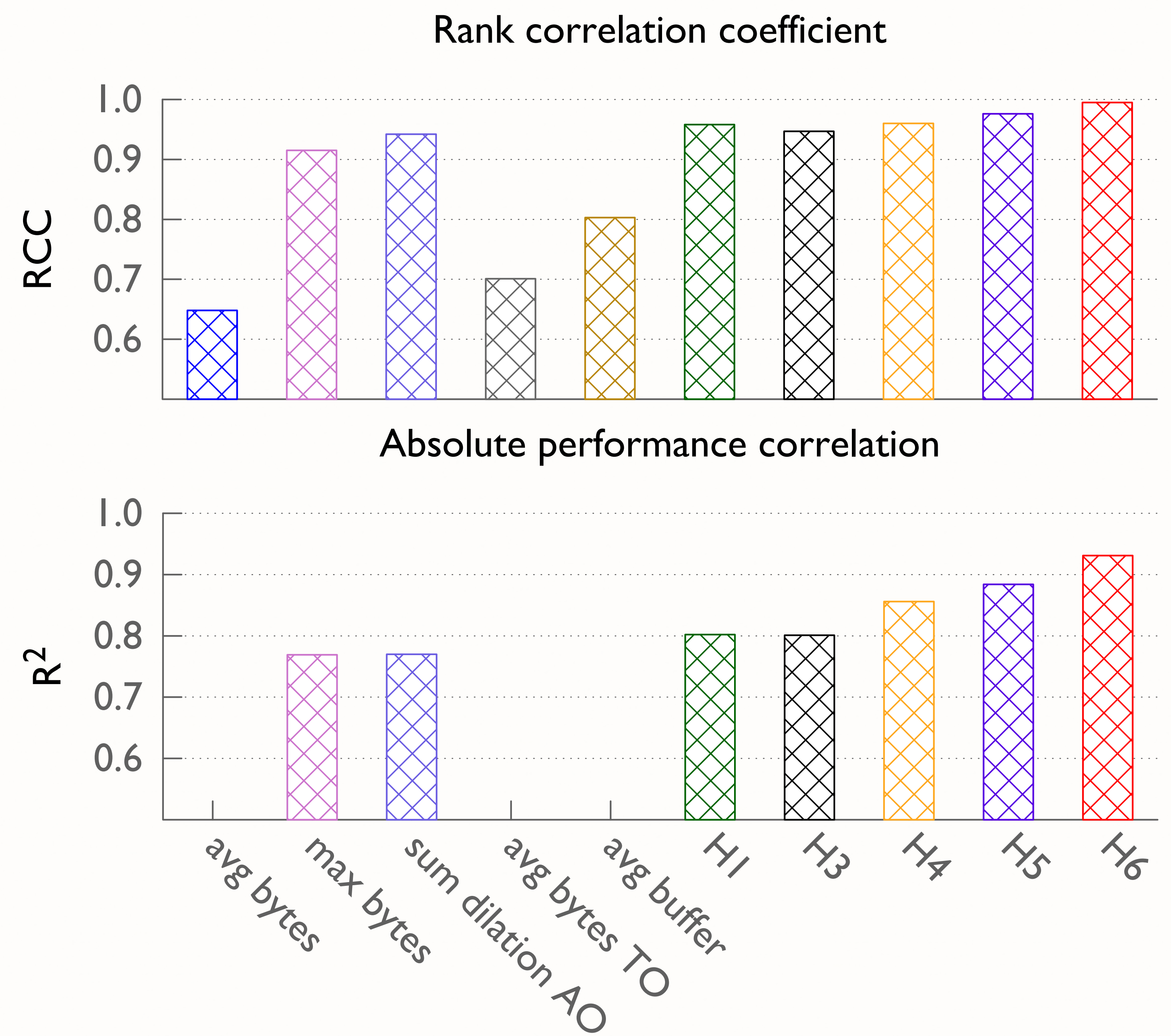
# COMBINING BENCHMARKS



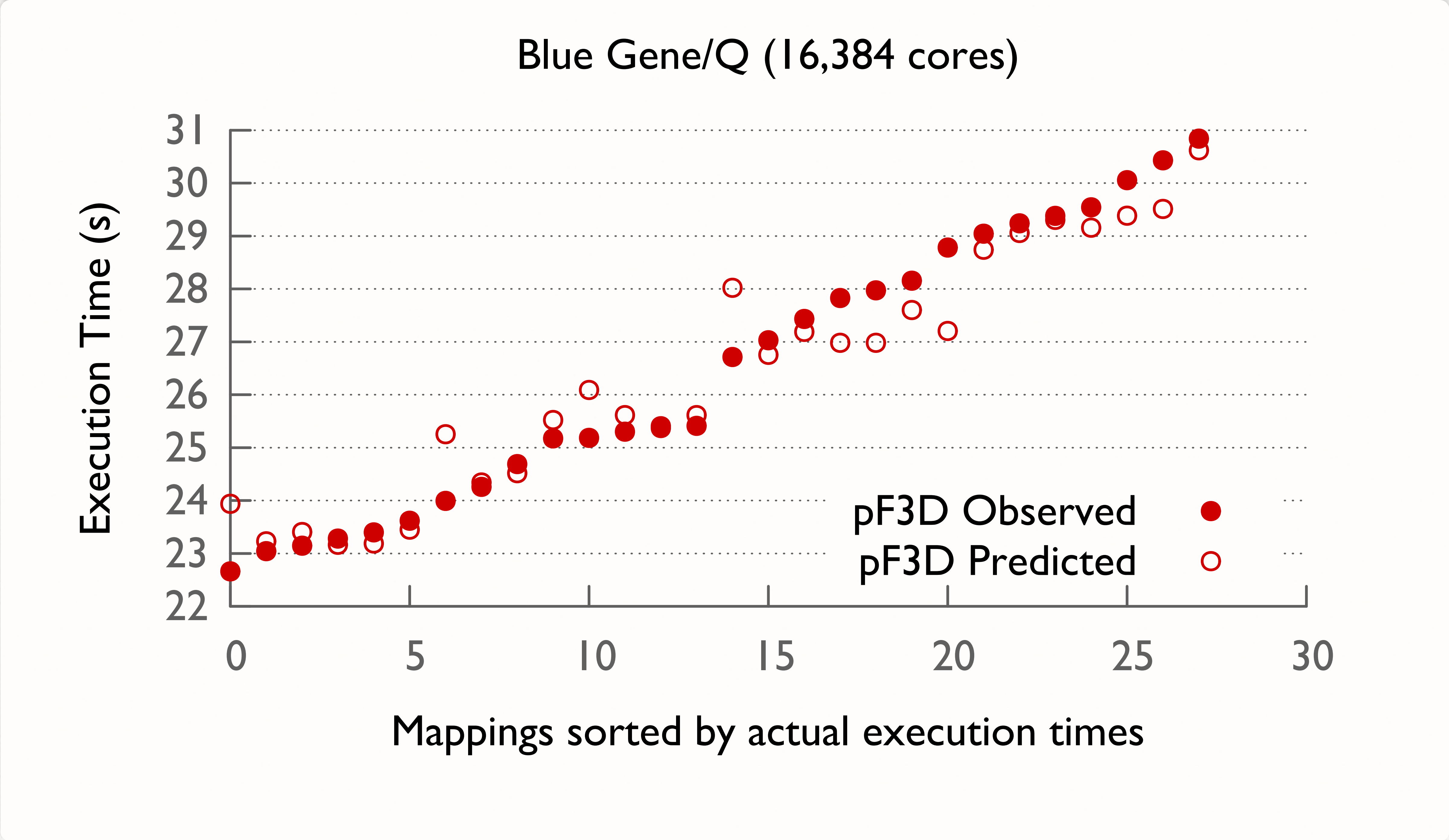
# PREDICTING FOR 64K CORES USING 16K CORES



# RESULTS: PF3D



# RESULTS: PF3D



# SUMMARY

- Communication is not just about peak latency/ bandwidth
- Simultaneous analysis of various aspects of network is important
- Complex models are required for accurate prediction
- There are patterns waiting to be identified!

# FUTURE WORK

- More applications!
- More metrics
- Weighted analysis
- Offline prediction of entities

Questions?