Pangeo Benchmarking Analysis: Object Storage vs. POSIX File System

Haiying Xu, Kevin Paul, Anderson Banihirwe

NCAR

Oct 07, 2020

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Introduction

- Motivation
- Introduction to Pangeo
- Varied testing conditions
- Benchmark setup
- Performance results
- Discussion
- Future work

Motivation

- Become a standard tool to benchmark Pangeo stack
  - Make the metric a standard to compare among different systems
- Compare the read/write throughput of Zarr vs. NetCDF
- Show the performance and scalability of object storage

Pangeo

- Pangeo
  - A community of geoscientists and software developers promoting open, reproducible, and scalable science
  - Core of software stack: Dask, Xarray, and Jupyter lab
  - Dask
    - Parallel computation and out-of-core memory capability
  - Xarray
    - Array-oriented data with labeled metadata such as dimension, coordinates and attributes
  - Jupyter lab
    - Web-based interactive environment to the Pangeo platform

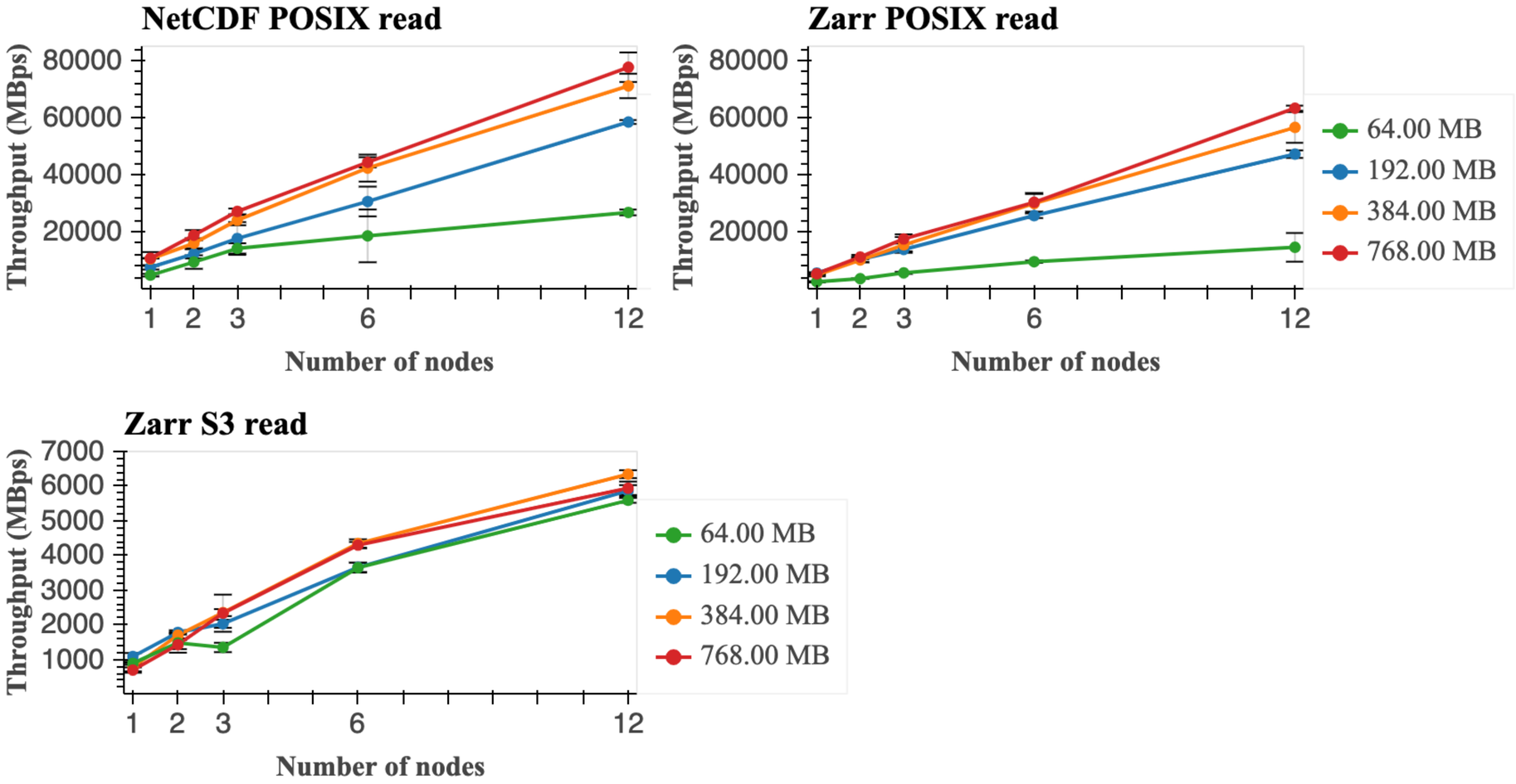
Varied Testing Conditions

- Object storage vs. POSIX storage
  - Object storage - ActiveScale from Quantum at 8 GBps transfer rate (multiple stream)
  - POSIX storage - DDN storage at 200 GBps transfer rate
- IO format: NetCDF vs. Zarr
- Read vs. write
  - The NetCDF API with Dask does not allow direct write to object storage yet
- Cluster size
  - Node count: 1, 2, 3, 6, 12
- Chunk size
  - 64MB, 192MB, 384MB and 768MB

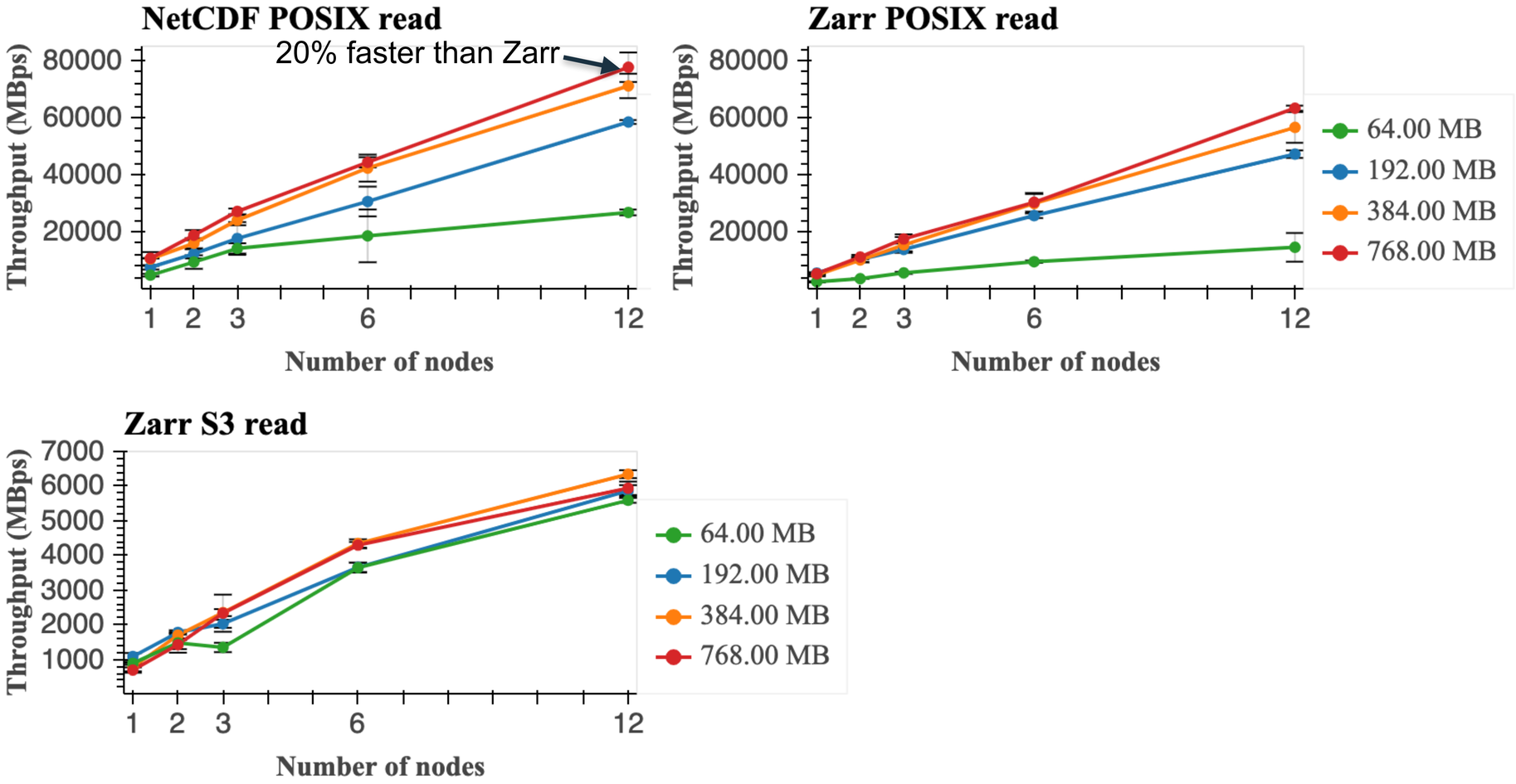
Benchmark Setup

- A xarray dataarray with 3 dimensions (time, lon, lat), with randomly generated data
- Dask cluster
  - Nodes, workers, memory usage
  - Cheyenne supercomputer at NCAR:
    - Intel Xeon processor cores in 4,032 dual-socket nodes (36 cores/node)
- Weak scaling analysis
  - Measure read and write throughput for a **fixed** dataset size **per processor** as the node count varies
  - Look like scaling a CESM simulation from low resolution with a few nodes to high resolution with many nodes
- Strong scaling analysis
  - Measure read and write throughput for a **fixed total** dataset size (460GB) as the node amount varies
  - Look like scaling a CESM simulation with a fixed resolution from low number of nodes to high number of nodes

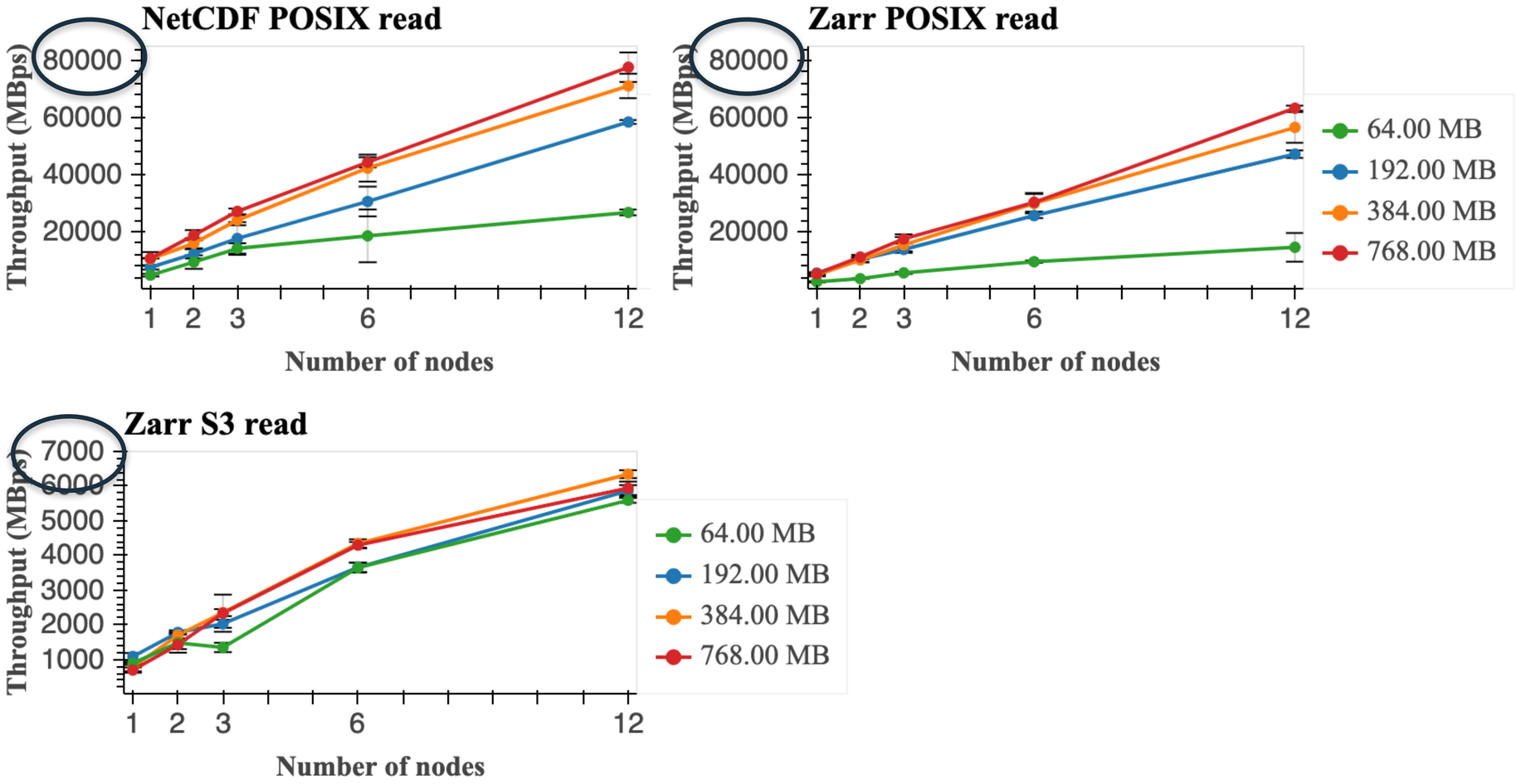
Weak Scaling Read



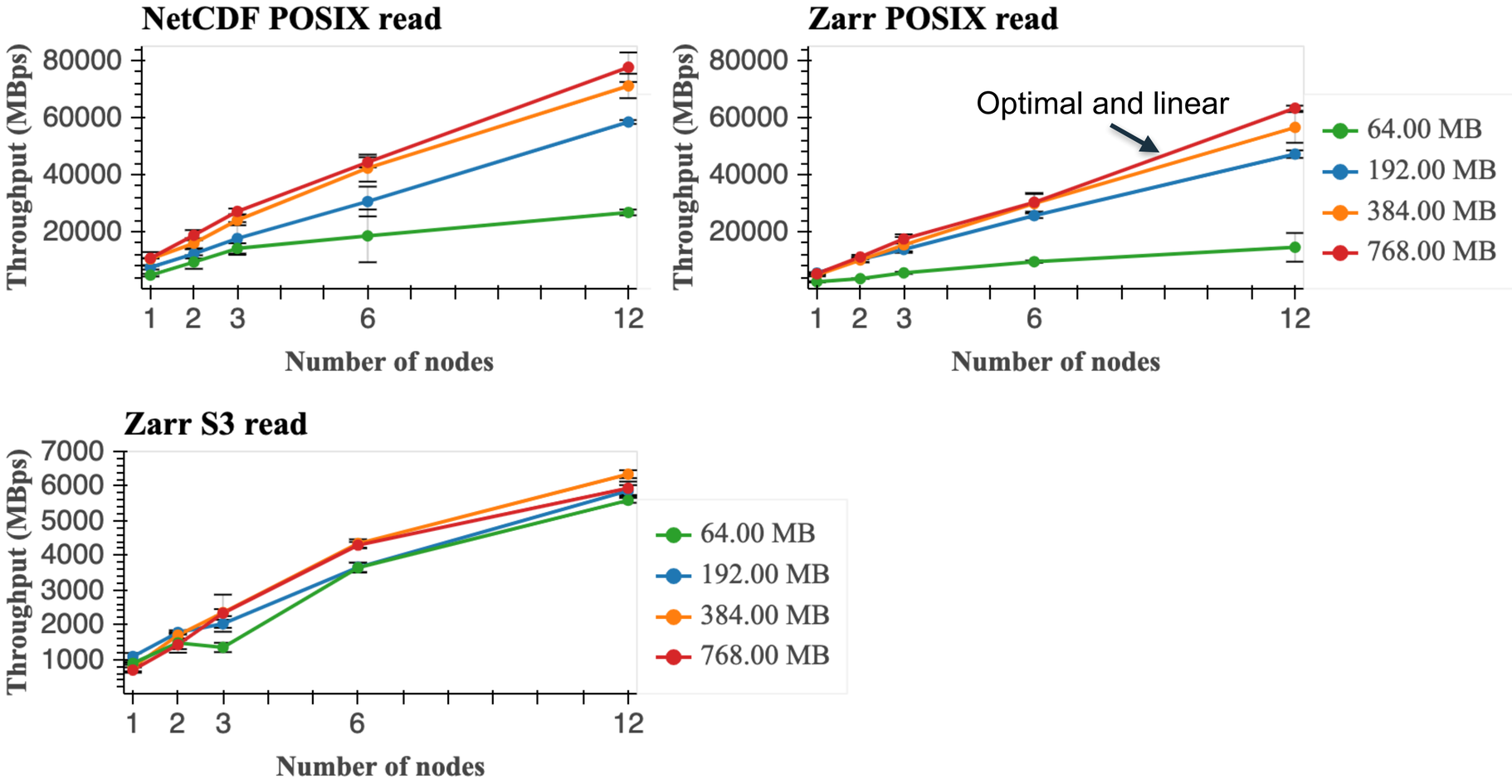
Weak Scaling Read



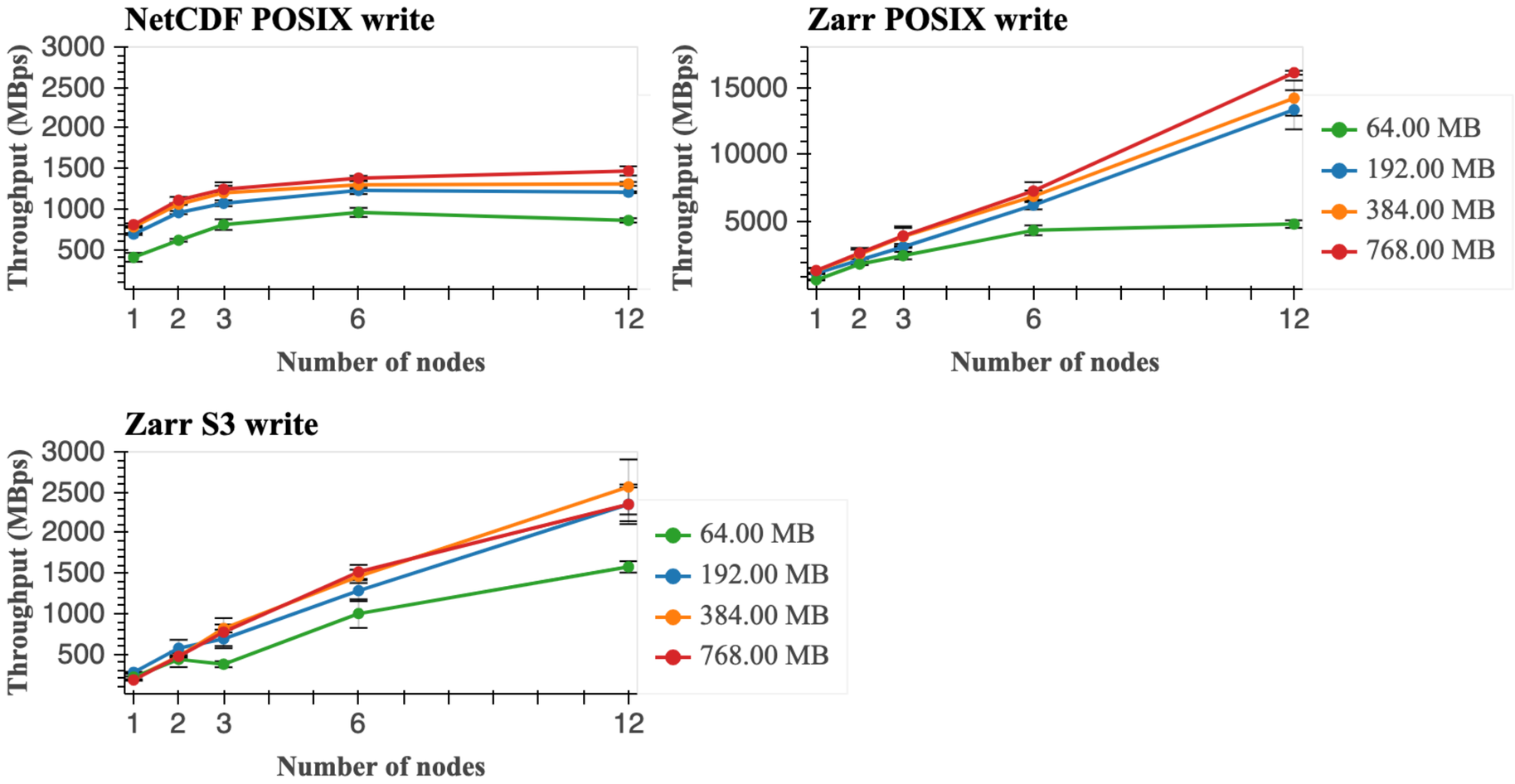
Weak Scaling Read



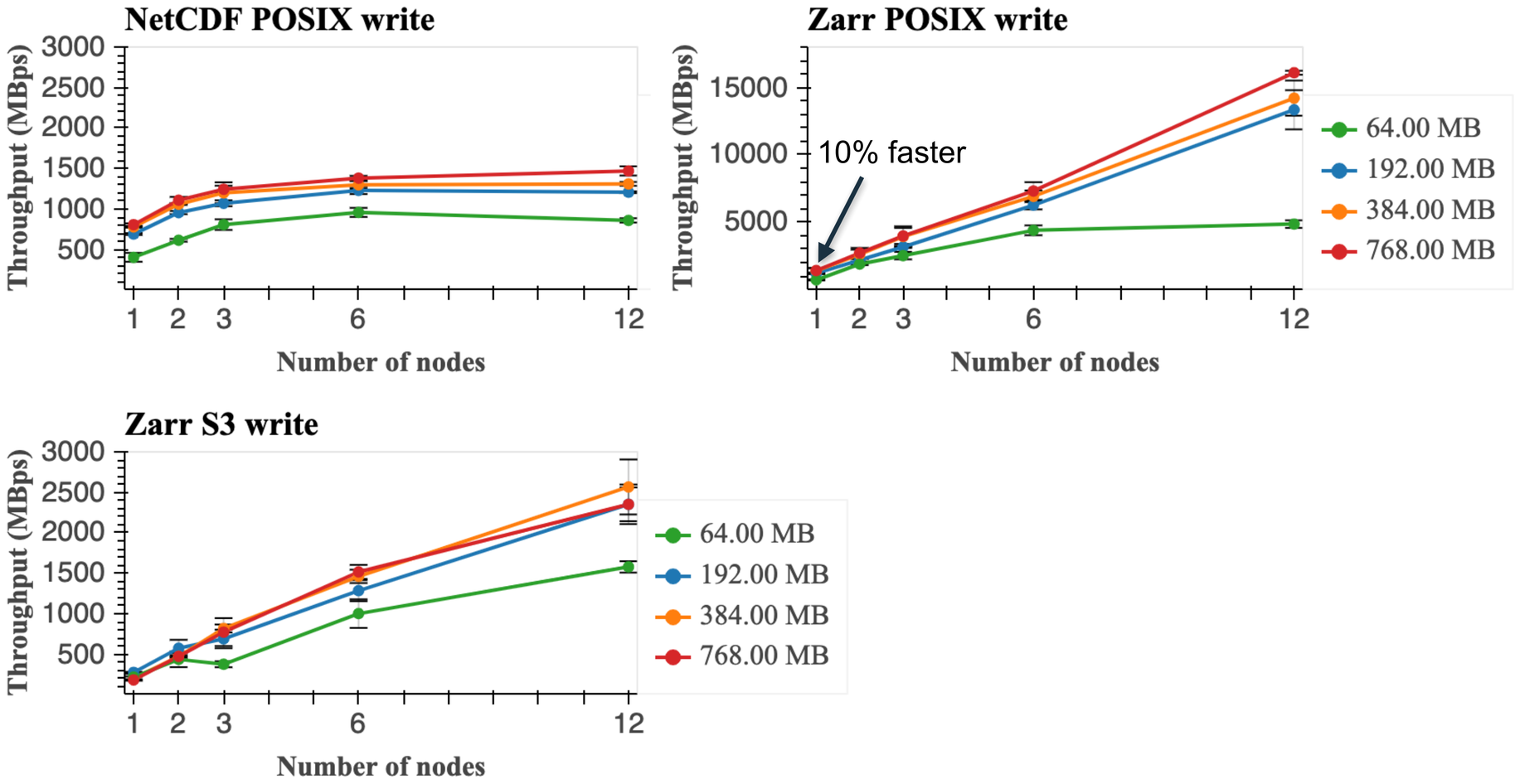
Weak Scaling Read



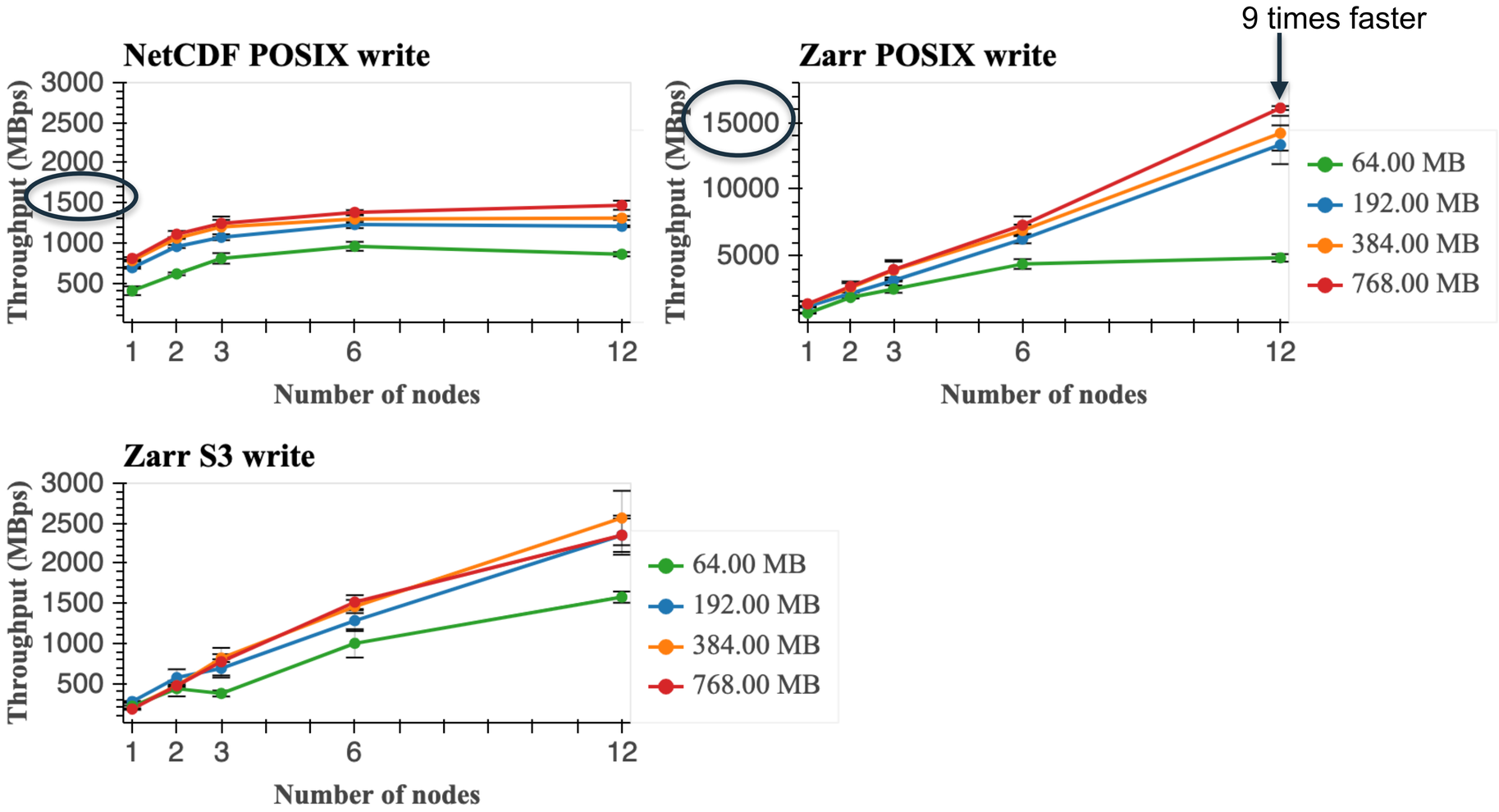
Weak Scaling Write



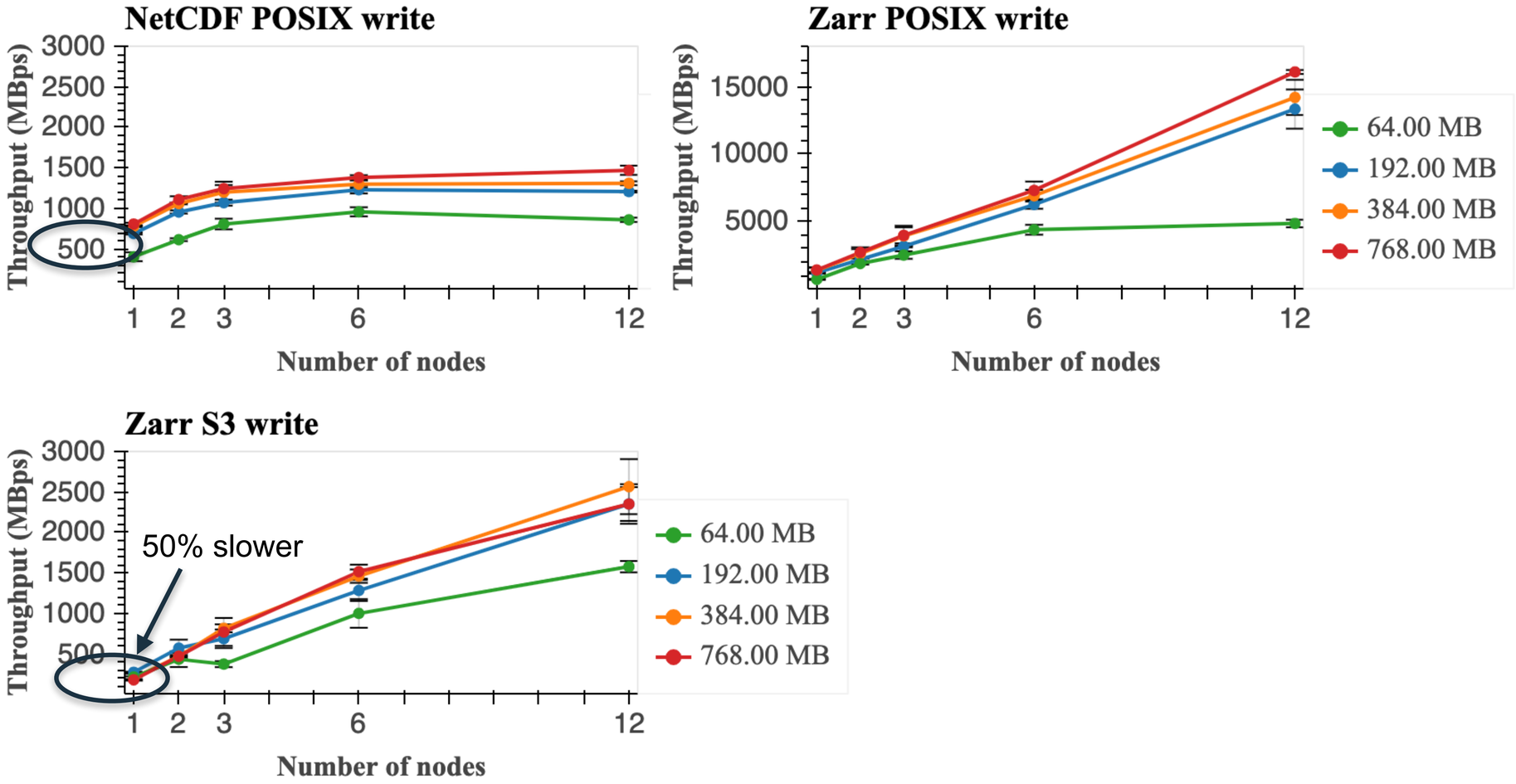
Weak Scaling Write



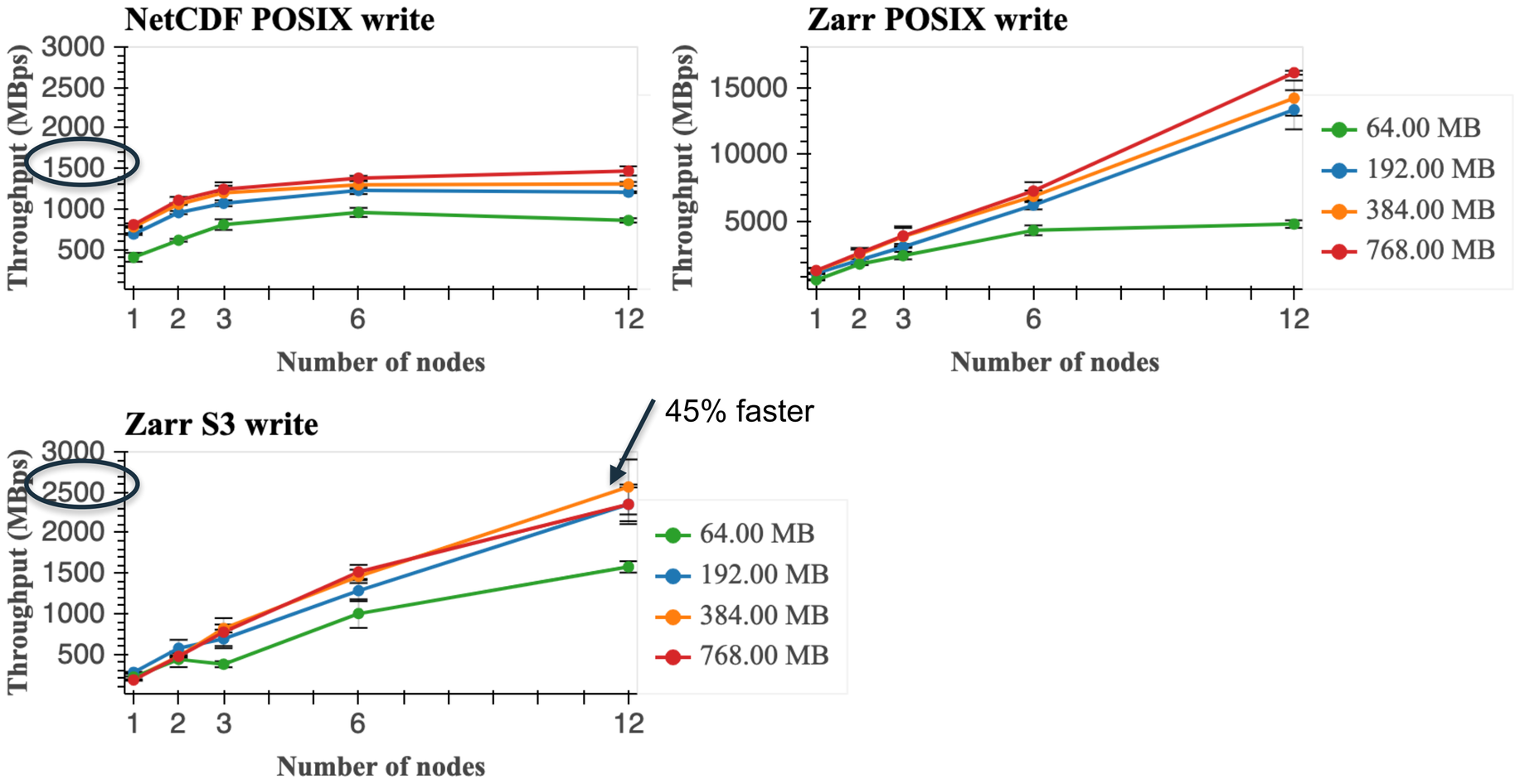
Weak Scaling Write



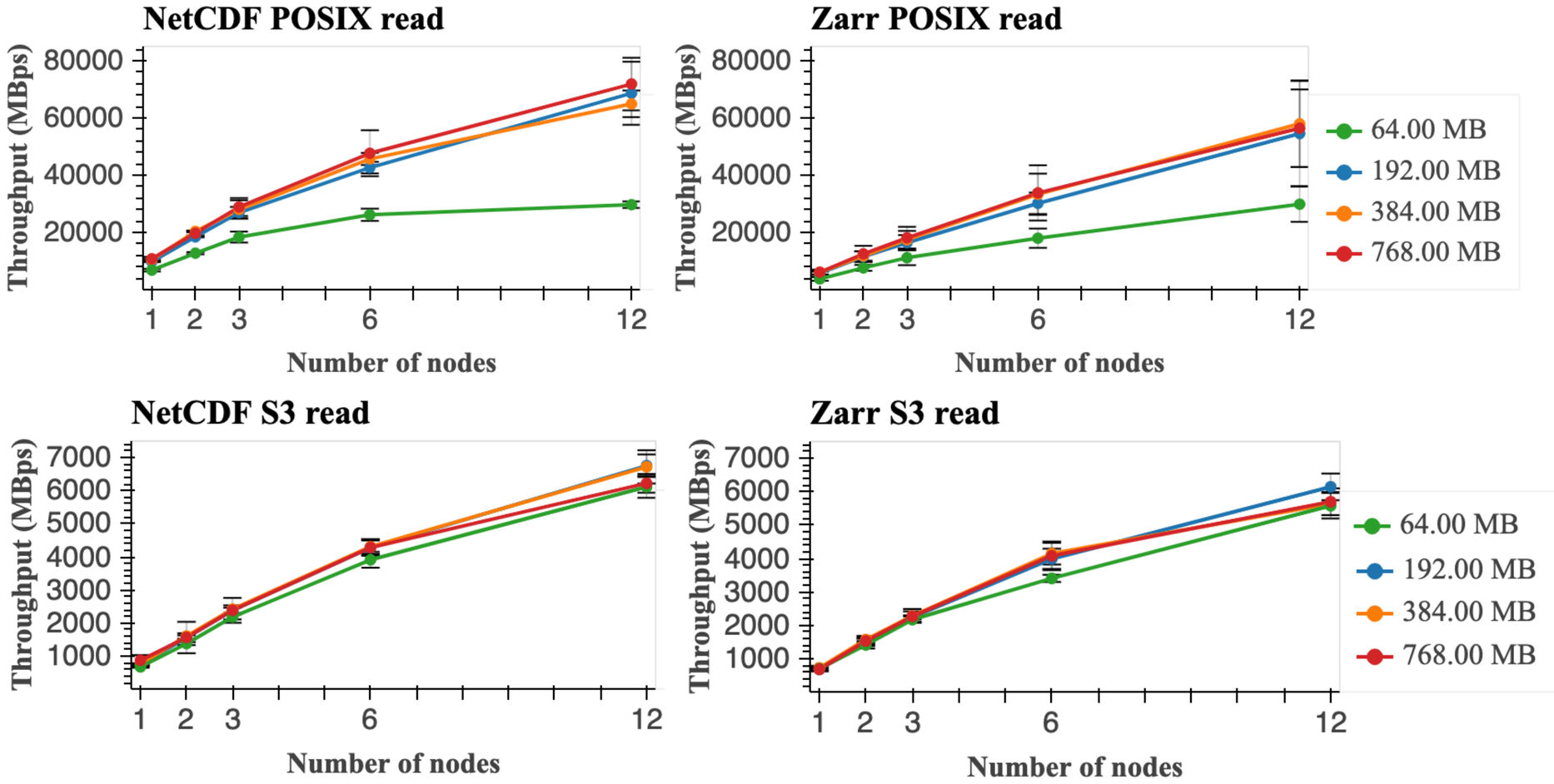
Weak Scaling Write



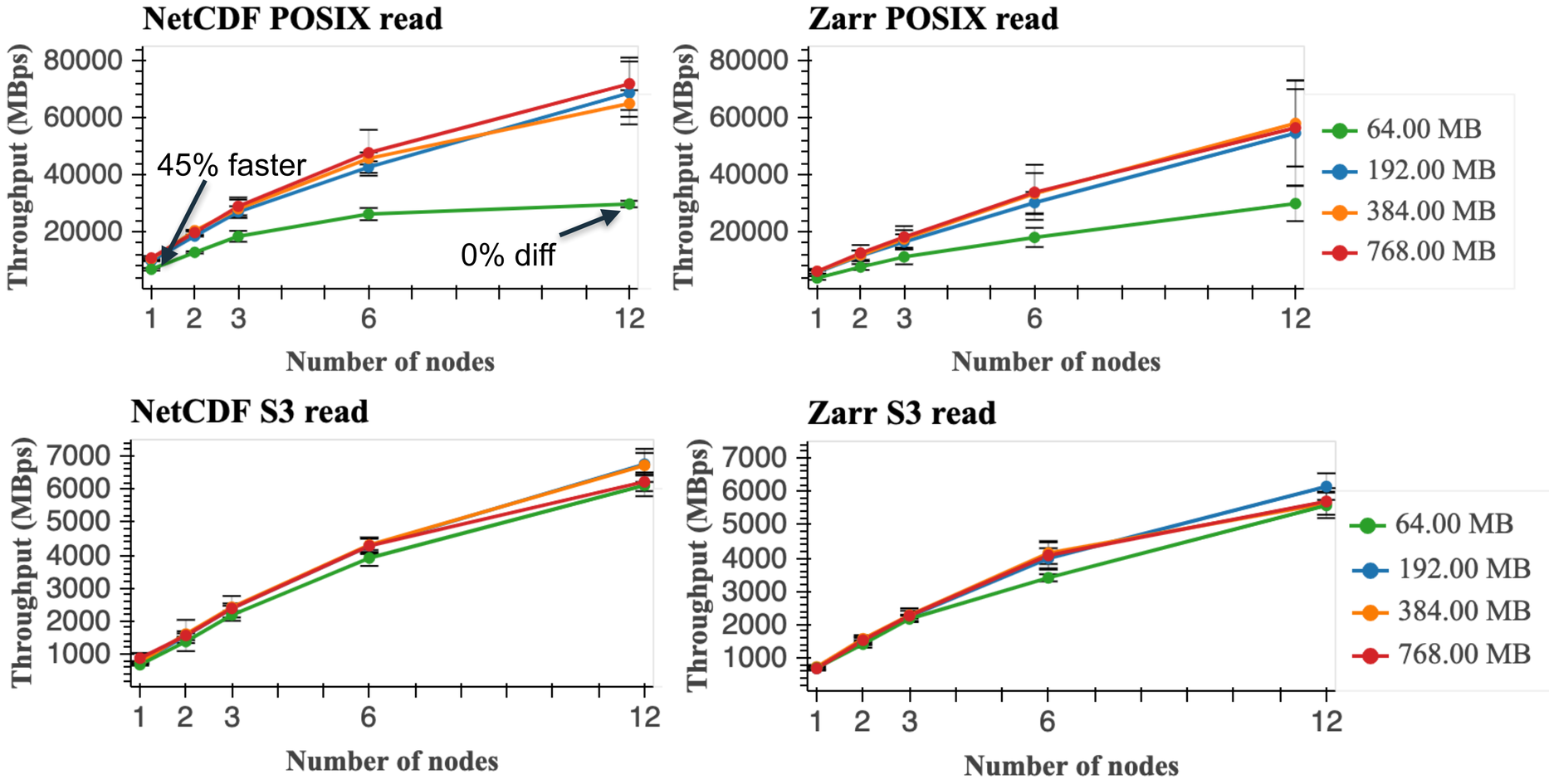
Weak Scaling Write



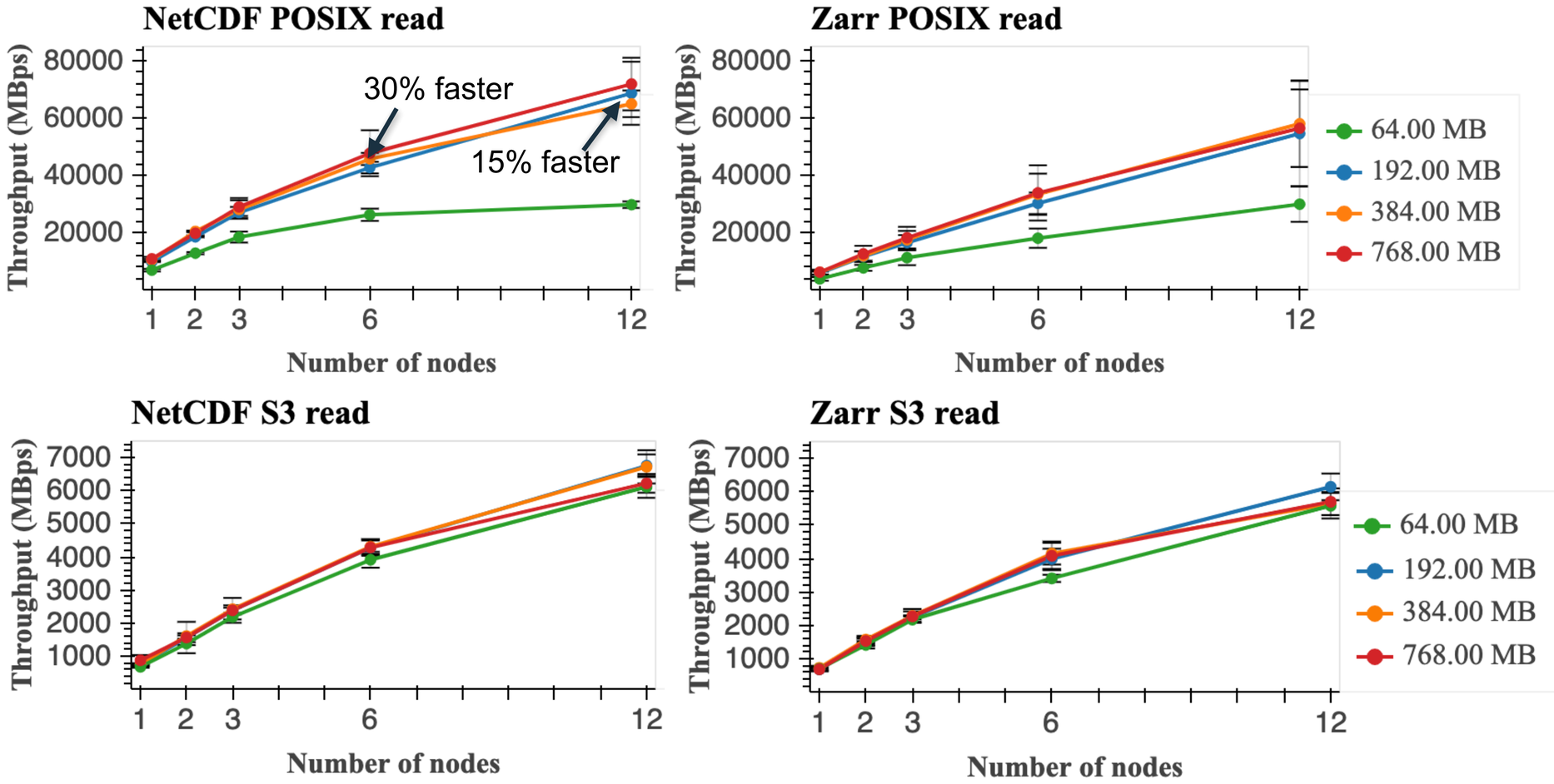
Strong Scaling Read



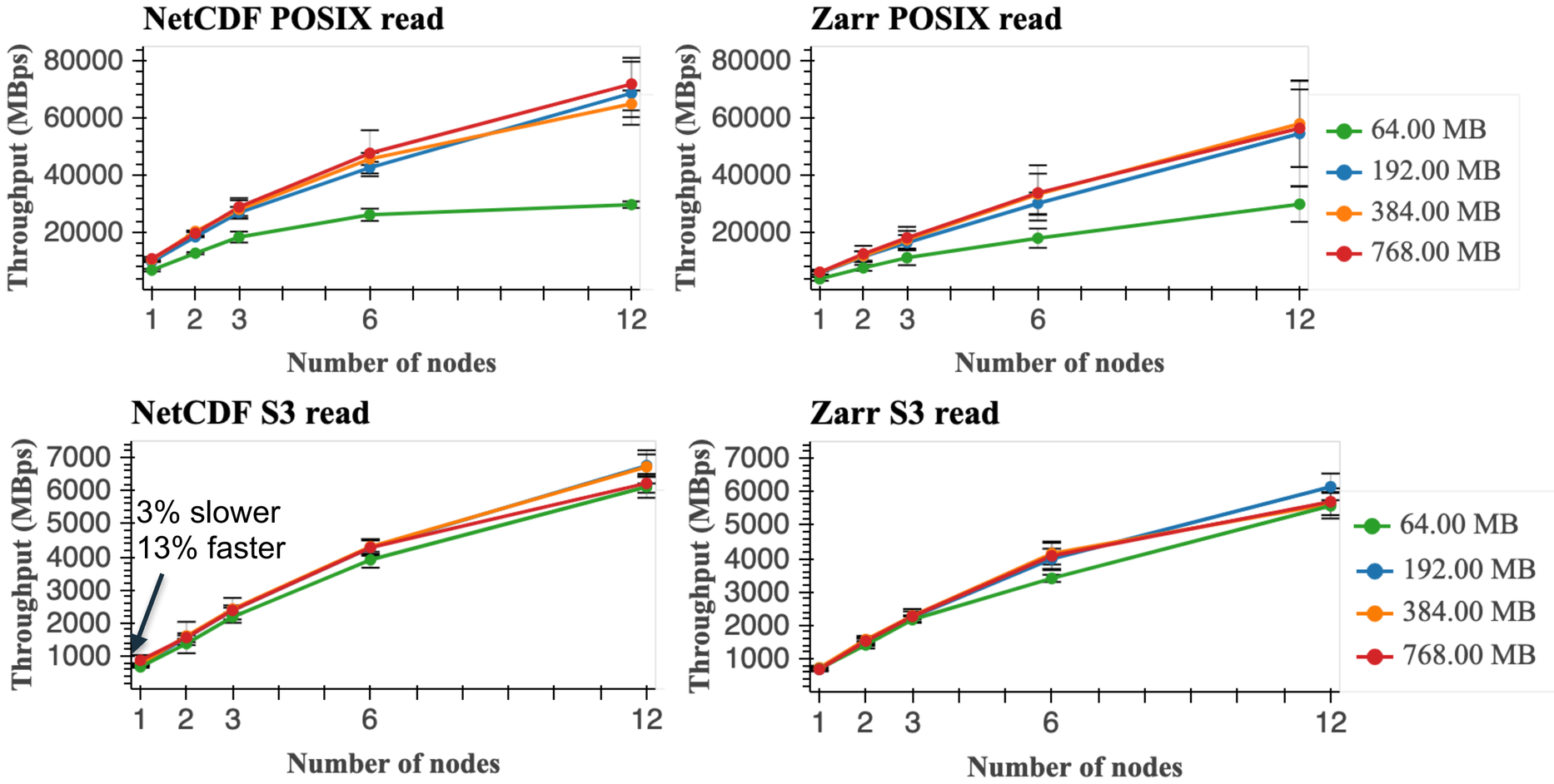
Strong Scaling Read



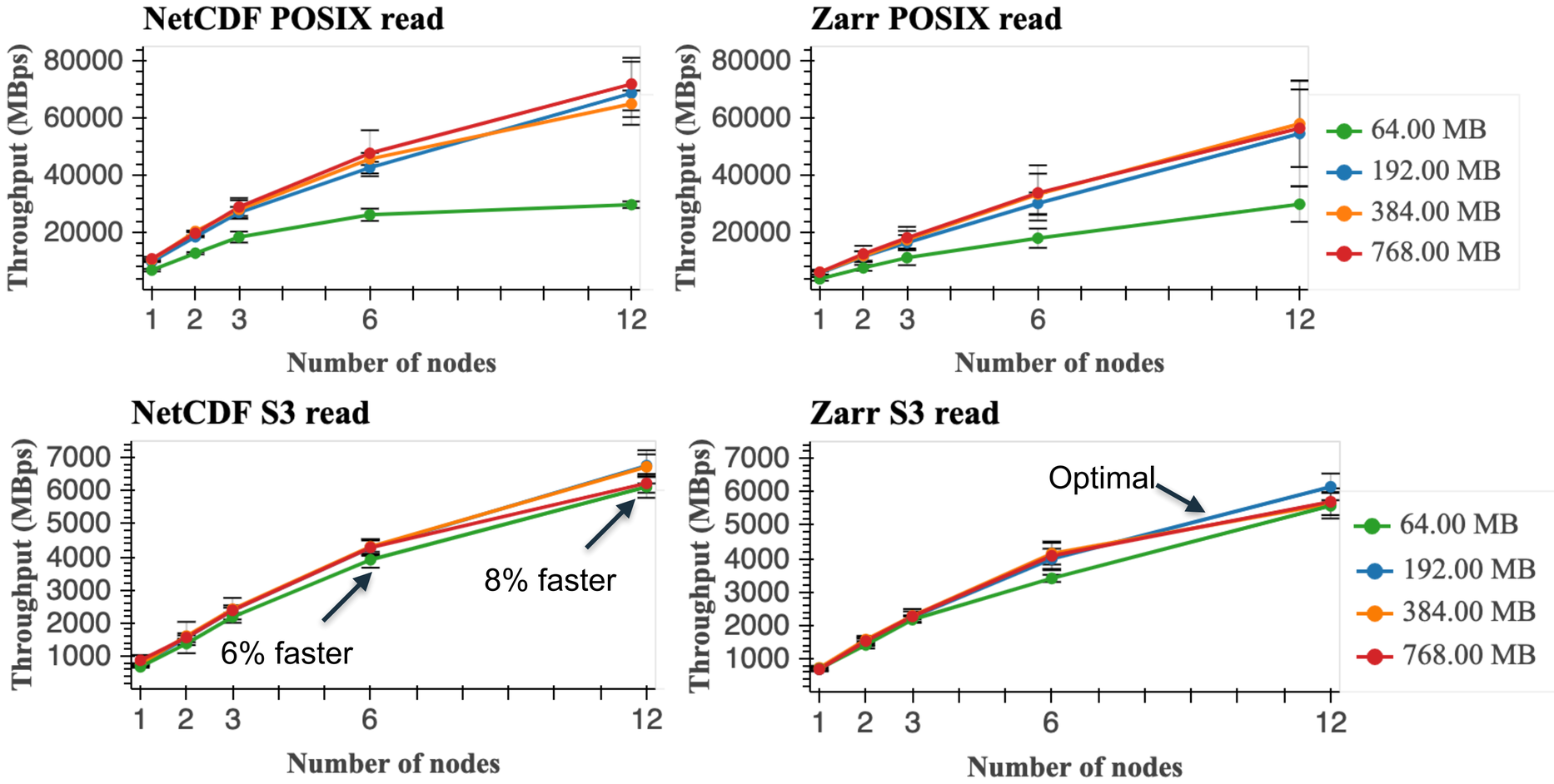
Strong Scaling Read



Strong Scaling Read



Strong Scaling Read



Discussion

- Object storage
  - Zarr read throughput same as NetCDF
- POSIX file system
  - NetCDF format reads a little faster
  - Zarr scales better
- Zarr format is beneficial geoscience
  - Lossy compression with faster write throughput
  - Flexible storage API
- Optimization on Zarr
  - skip_instance_cache
  - use_listing_cache

Future Work

- Enable asynchronous mode in Dask
- Containerize the benchmarking tool with Docker (for cloud) or Singularity (for HPC)
- Compare write performance against PnetCDF
- Benchmark on high throughput scalable object storage
  - AWS or Google cloud
  - Benchmark with cost in mind