PortHadoop-R: The Merging of HPC and Big Data

Kun Feng

Illinois Institute of Technology

Outline

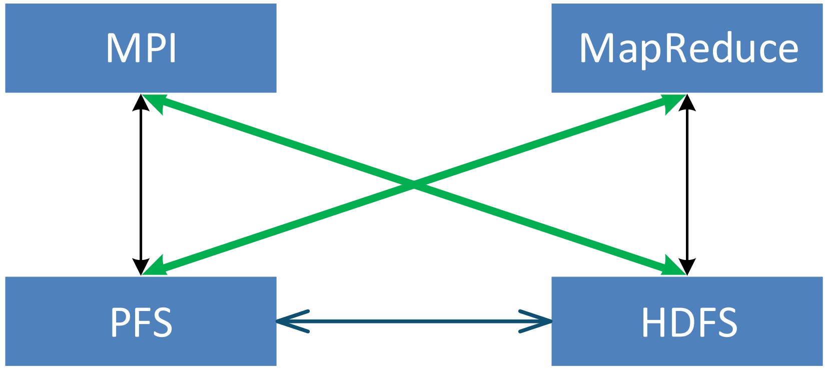
- Introduction
- Background
- Design
- Evaluation
- Conclusion

Introduction

- The merging of HPC and big data analytics is inevitable.
    - High Performance Computing (HPC) is becoming data intensive.
    - Big data applications are requiring more and more computing power.
    - Two ecosystems are designed for different applications and with different design principles.
- Two ecosystems will co-exist.
    - Neither can have all the merits of the other.
    - The best we can have is a hybrid system which can provide the functionalities and merits of both ecosystems.

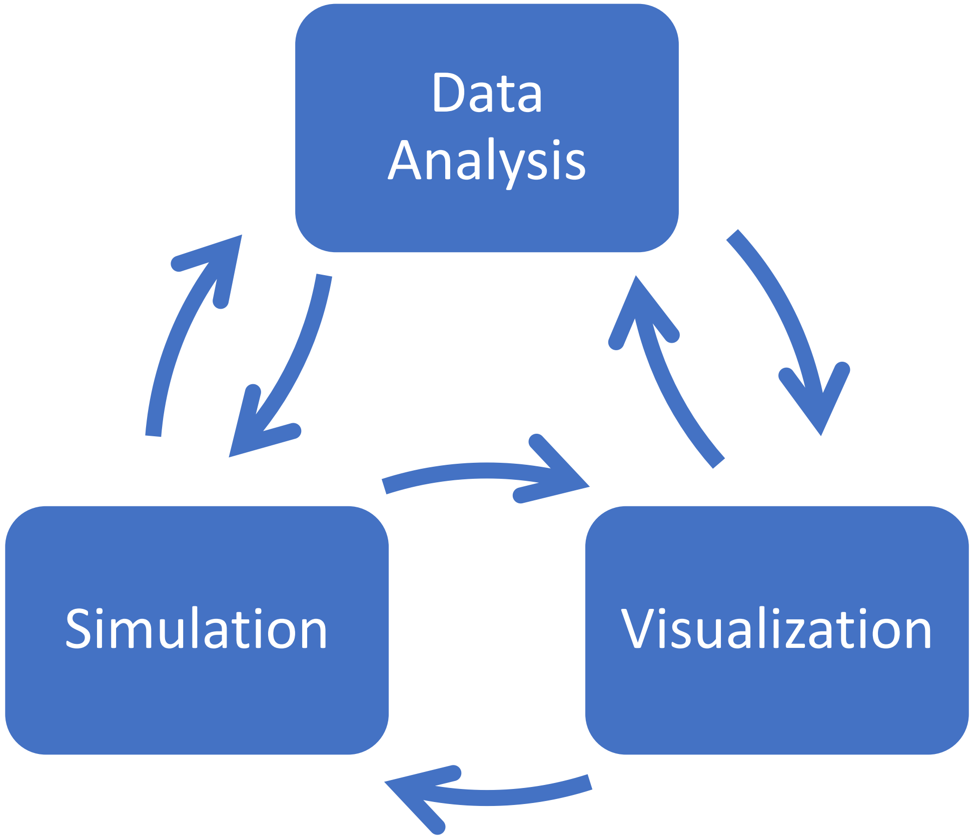
Background -- Motivation

- Native data access exists
  - HPC applications read from/write to PFS (POSIX I/O and MPI-IO)
  - Big Data applications read from/write to HDFS (HDFS API)
- Data dependency in-between
  - Big data applications analyze data generated by simulations
  - Analysis results guide simulation runs
  - Use explicit **serial** copy
  - Redundant data store in HDFS



Background -- Use case

- Iterative process
  - Simulation generates large data
  - Analyze and visualize iteratively
  - Analysis results help steering simulations
- Simulation
  - MPI-based
  - HPC platform
  - Use PFS as data storage
- Visualization
  - Coarse visual analysis
  - Quick identification
  - Animated images
- Data analysis
  - Statistical analysis
  - SQL query
  - Use HDFS as data storage



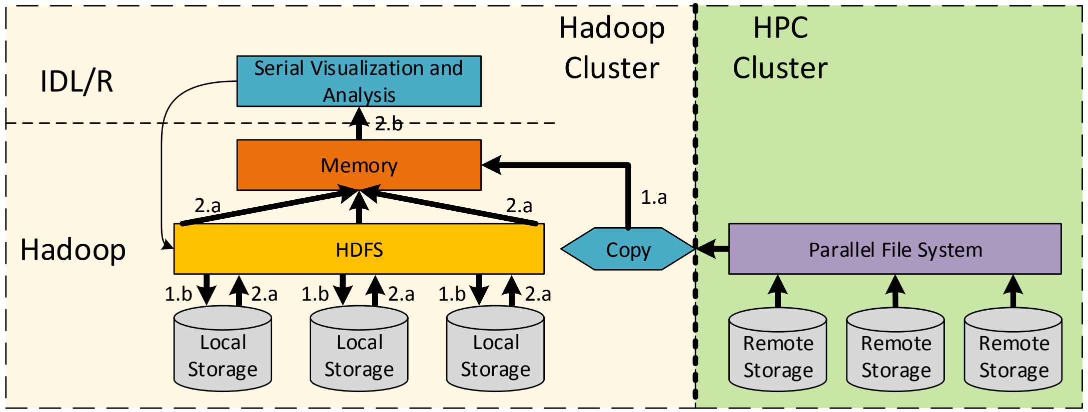
Our solution

- We propose PortHadoop-R to support the merging at the data access level
  - Allow reading data directly from PFS to the memory of Hadoop nodes
  - Integrate the data transfer with R data analysis and visualization
  - Optimized to utilize the merits of PFS and MapReduce to achieve concurrent data transfer and latency hiding
  - Tested with real data of NASA climate modeling applications

Design -- Architecture

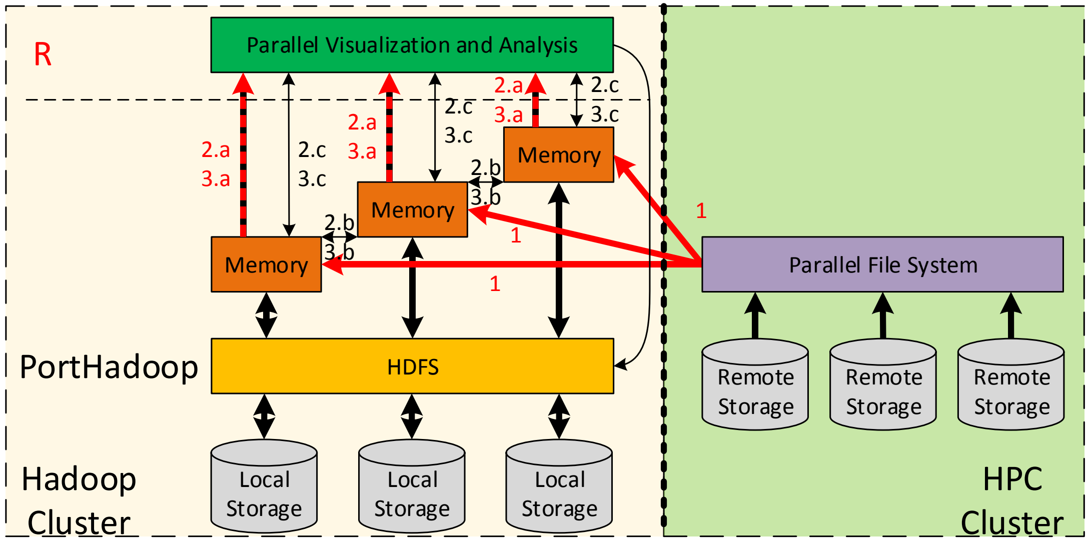
Conventional solution

- Serial data copy
- Fat node design
- Serial visualization and analysis



PortHadoop-R

- Copy free design
- Parallel data access
- Parallel visualization and analysis



New Data on PFS

Existing Data on HDFS

Result Data

1. Data Access 2. Visualization 3. Data Analysis

Design -- PortHadoop (cont.)

- Virtual block
  - Create HDFS data block virtually
  - Map HDFS block to file segment in PFS (filename, offset, size)
  - Stored in NameNode
- On-demand data access
  - MPI-based PFS file reader
  - Triggered only when the block is accessed
  - Minimal data transfer
- Portability
  - MPI-IO guarantees the PFS compatibility

Design -- Parallel Image Plotting

- Multiple concurrent image plotting tasks
  - Key is defined based on dimension of the dataset (e.g. along time or altitude)
  - Each map task is assigned to plot one image
  - Parallelism of the Hadoop cluster is utilized
  - The target file segment info can be obtained from virtual block
  - Multiple read requests are issued to read data from PFS concurrently
  - PFS bandwidth is fully utilized as well

Design -- Parallel Data Analysis

- Data analysis can be carried out alongside image plotting
  - Statistical information can be extracted (e.g. min, max and avg)
  - SQL queries can be executed upon the data (e.g. select top 10 records)
  - Results are packed with plotted images

Design -- Animation

- Animated image sequence is a very good way to analyze the HPC data
  - Images and analysis results are shuffled according to the key
  - Reduce tasks combine images together to make animation
  - Based on the *convert* command in ImageMagick
  - Not parallelizable
  - Animations are saved in HDFS

Experiment Setup

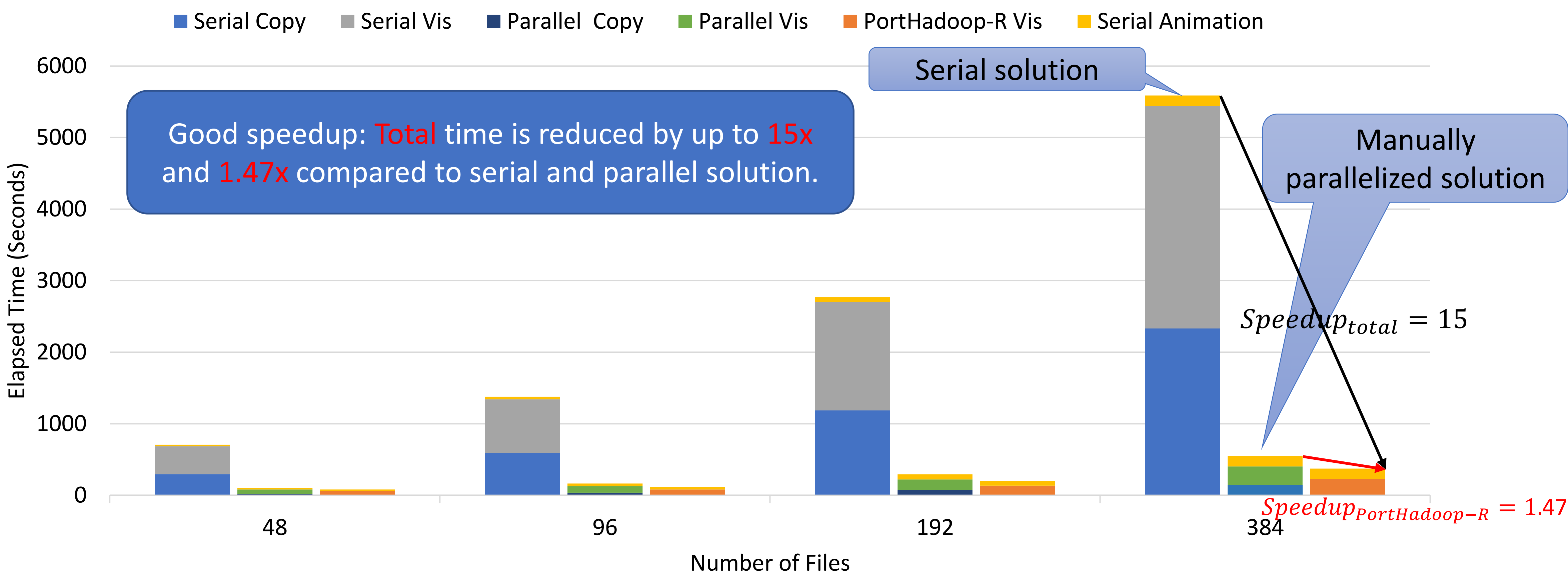
- Hardware
  - Chameleon at TACC
  - Baremetal
  - 4, 8 and 16 Hadoop nodes, 8 OrangeFS nodes
- Software
  - CentOS 7.3
  - Java 1.7.0_79
  - PortHadoop-R (modified from Hadoop-2.5.0-CDH5.3.3)

Experiment Setup (cont.)

- Configuration
  - Input data: 48, 96, 192 and 384 single-level data files from a real NU-WRF 1250x1250x50 run (up to 23.14GB). Each file is ~61.7MB.
  - All data are read from 8-node OrangeFS. Each Map task visualizes one file and creates one image. A SQL query is also applied in Map task for Anlys workload. A Reduce task merges all images into an animation.
  - The elapsed time, measured on the master node.
- Workload

Name	Image Plotting	Animation	Analysis
Vis-only	Yes	Yes	No
Img-only	Yes	No	No
Anlys	Yes	Yes	Yes

PortHadoop-R VS Conventional Methods: Performance (Vis-only)



Serial Copy: one copy command issued on one Hadoop node

Parallel Copy: 4 copy commands issued on each Hadoop node (32 in total)

Serial Vis: read all data into one fat node, visualize

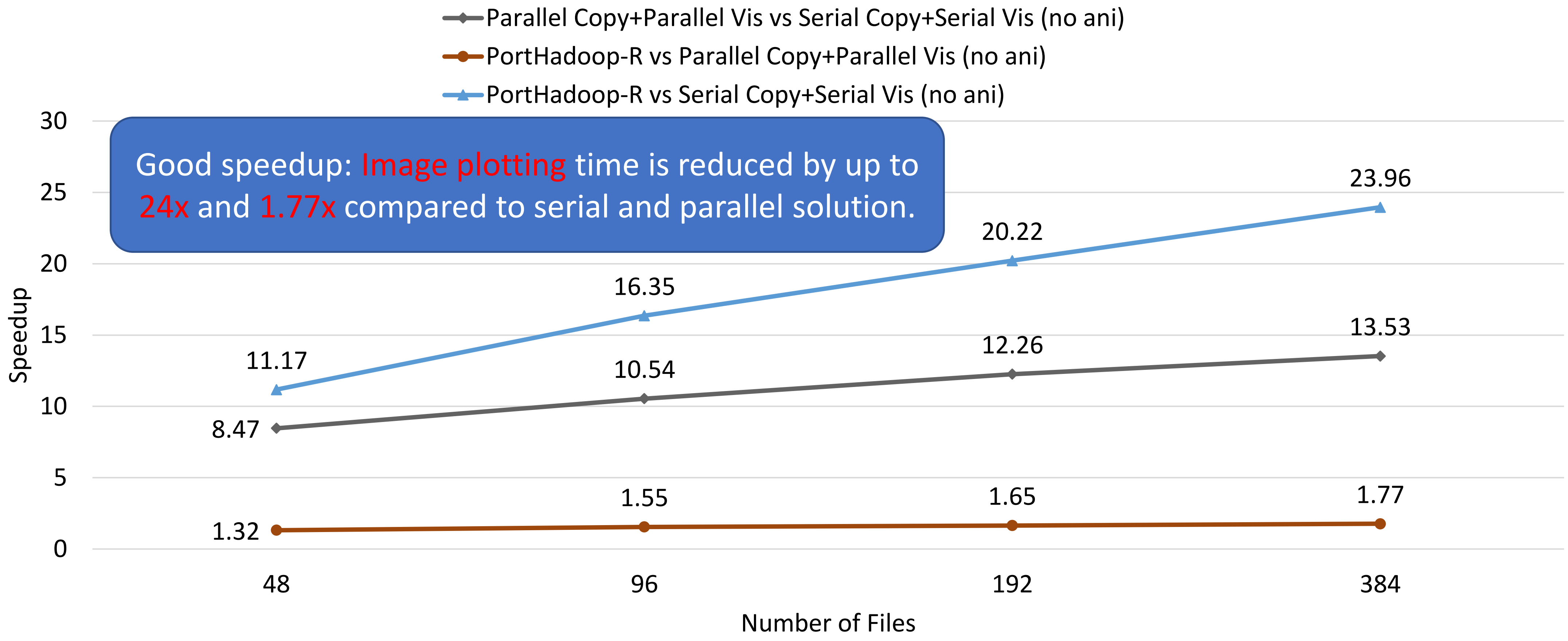
Parallel Vis: visualize all files on all Hadoop nodes

ILLINOIS INSTITUTE OF TECHNOLOGY

PortHadoop-R Vis: visualize all files into images

Serial Animation: merge all images into one animation

PortHadoop-R VS Conventional Methods: Speedup (Img-only)



Serial Copy: one copy command issued on one Hadoop node

Parallel Copy: 4 copy commands issued on each Hadoop node (32 in total)

Serial Vis: read all data into one fat node, visualize

Parallel Vis: visualize all files on all Hadoop nodes

ILLINOIS INSTITUTE OF TECHNOLOGY

PortHadoop-R: visualize all files into images

PortHadoop-R Scale-out Performance (Vis-only)

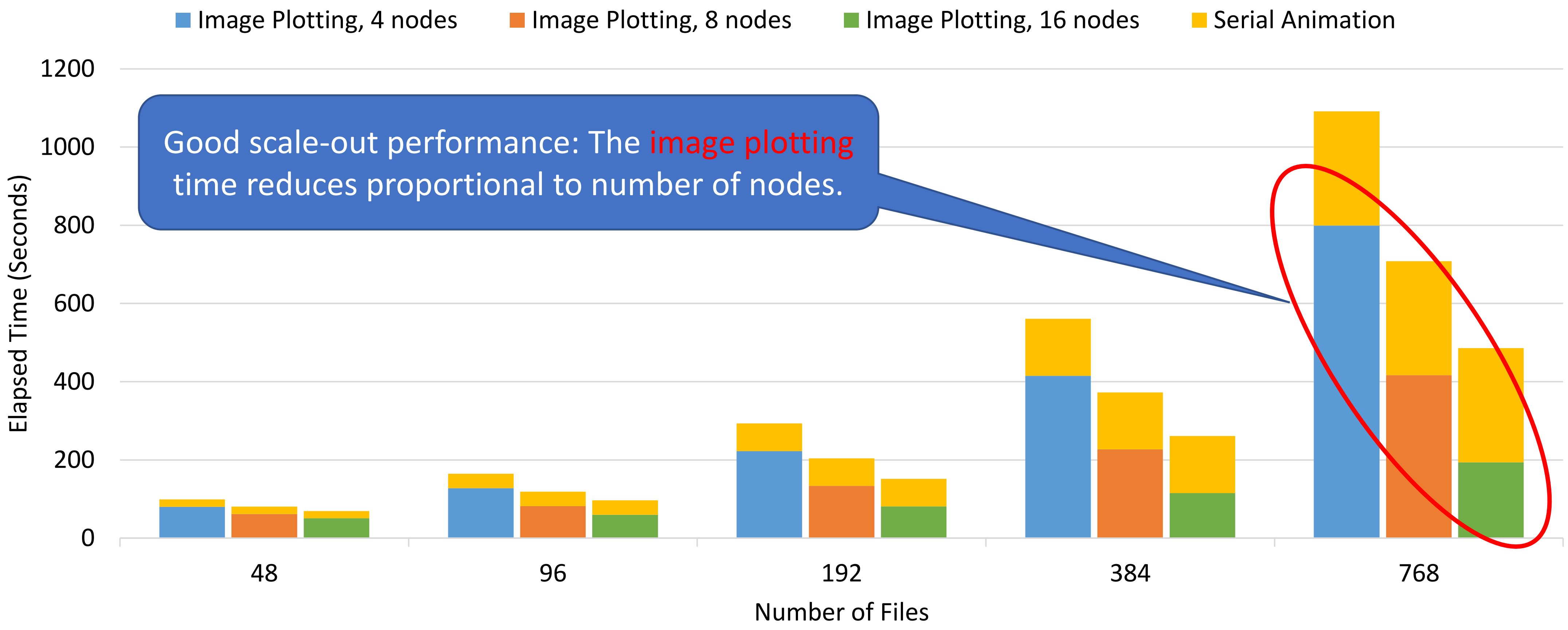


Image Plotting: visualize all files into images

Serial Animation: merge all images into one animation

PortHadoop-R Scale-up Performance (Vis-only)

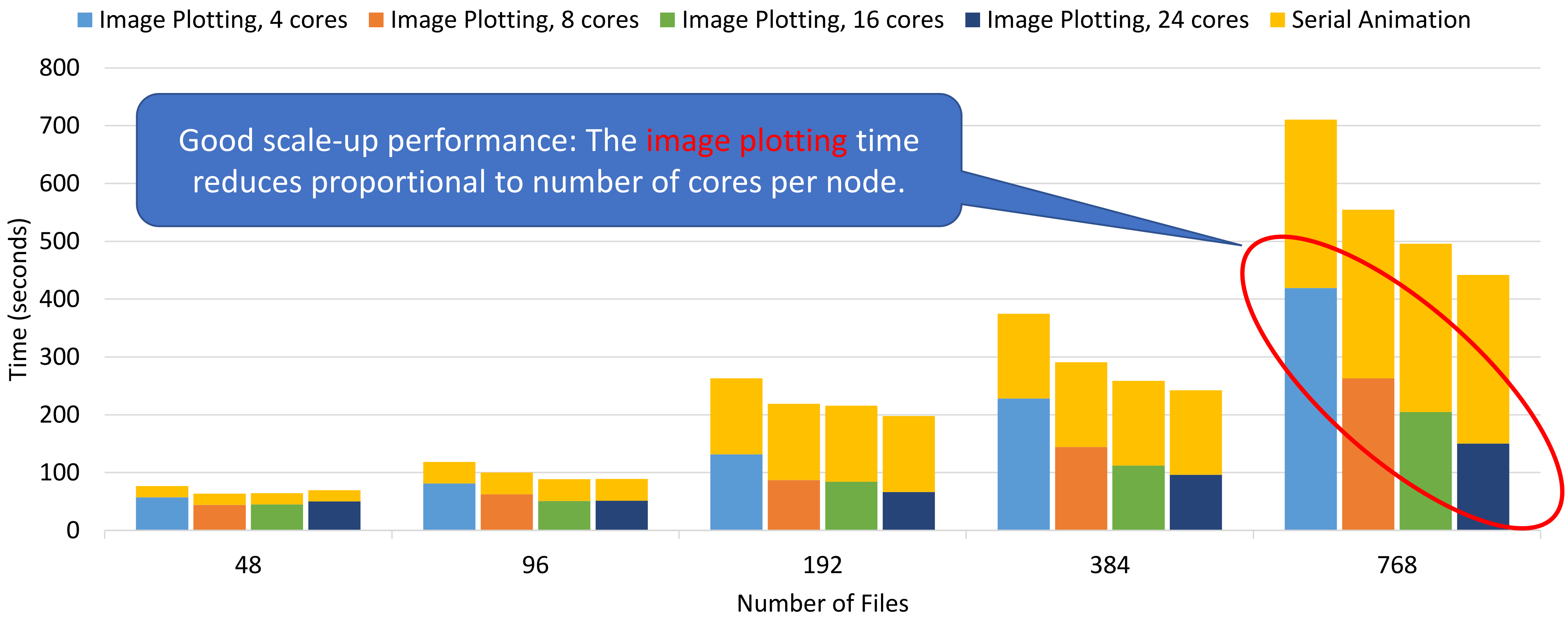


Image Plotting: visualize all files into images

Serial Animation: merge all images into one animation

Conclusion

- PortHadoop-R supports the merging of HPC and big data in data access level
- Leverage merits of both systems
- Great performance (speedup, scalability)
  - Speedup in visualization
    - 15x and 24x: compared with serial solution for Vis-only and Img-only
    - 1.47x and 1.77x: compared with manually parallelized solution for Vis-only and Img-only
  - Good scalability (scale-out and scale-up)
- Applicable to other frameworks

Chameleon toolkit

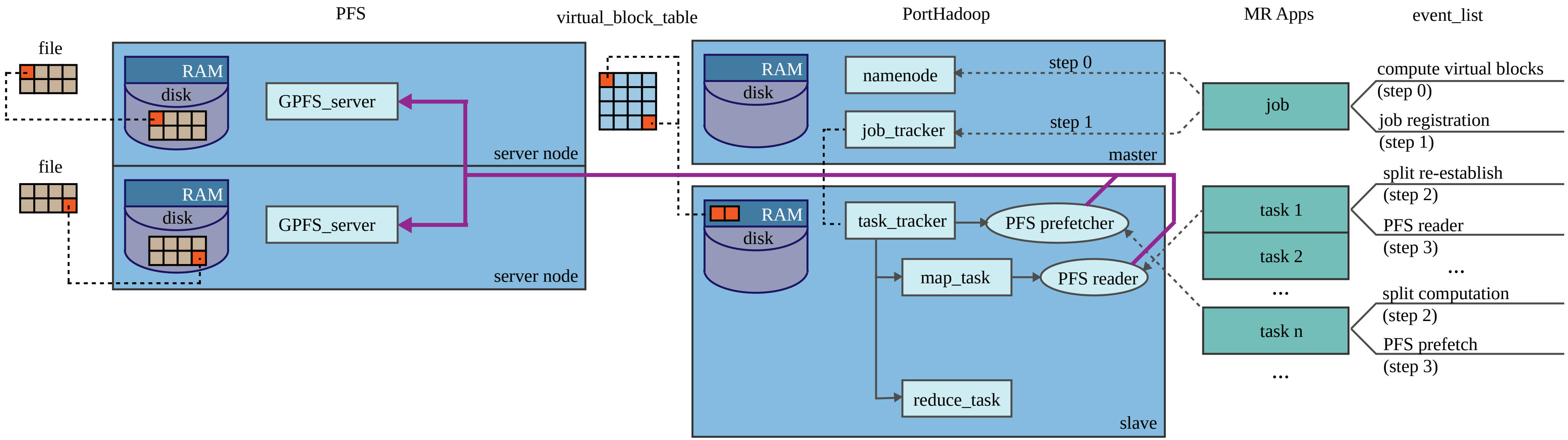
- A small tool created to facilitate our experiments on Chameleon
  - Forked from the official cc-snapshot repo
  - Helper scripts to wire up and authenticate instances
  - Helper commands to keep files on multiple nodes synchronized
  - https://github.com/fkengun/Chameleon-toolkit

Thank You

Q&A

ILLINOIS INSTITUTE OF TECHNOLOGY

Design -- PortHadoop (backup)



Related work (backup)

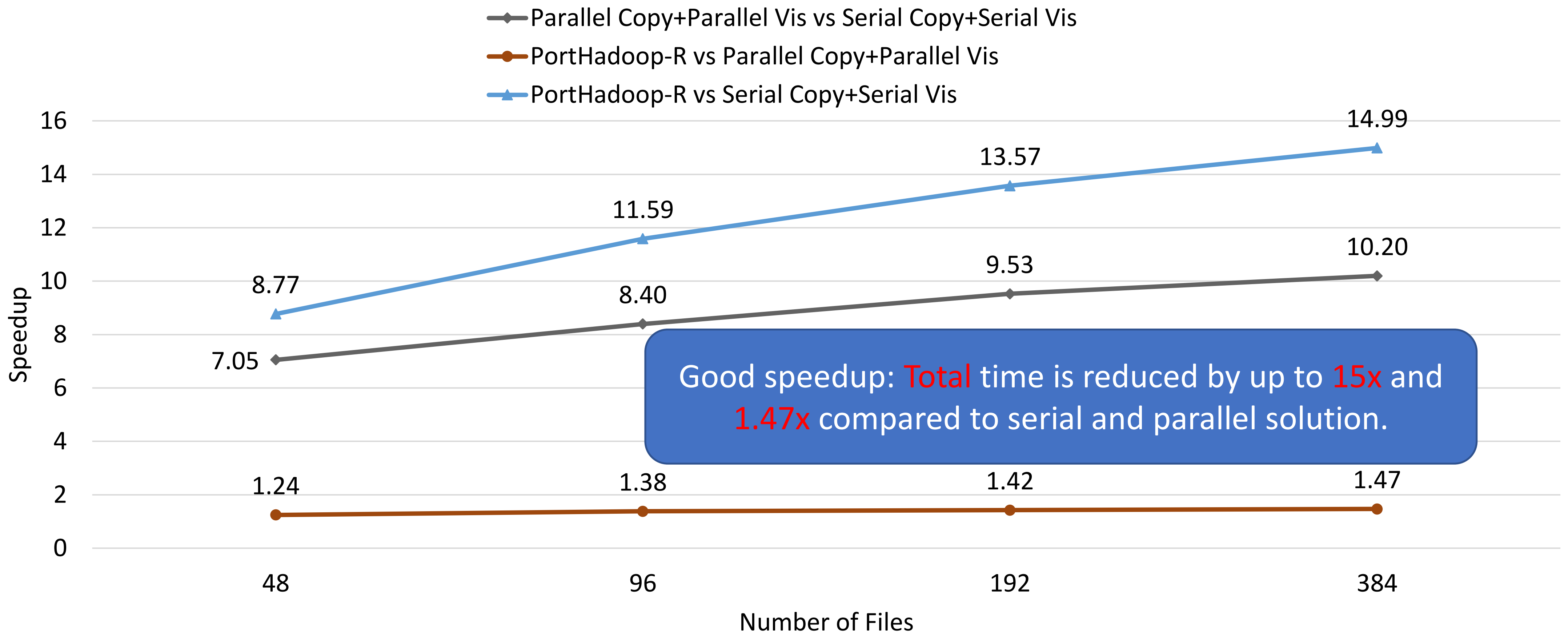
- In-situ analysis in HPC
  - E.g. VisIt, ParaView
  - Compete with simulations for resources
- Big data frameworks
  - E.g. Hadoop
  - Require explicit serial data copy to load data
- Improved big data frameworks
  - E.g. SciHadoop
  - Only support processing data on HDFS

Design -- R Interface (backup)

- Leverage existing R packages
  - Easy to use
  - Rich capabilities in analysis and graphics
  - Rely on RHadoop for distributed computation

Package	Version	Description
R	3.2.2	
rhdfs	1.0.8	HDFS connector
rmr2	3.3.1	R interface for Hadoop MapReduce
Cairo	1.5-9	Graphics library to plot image
plot3D	1.1	Provides 3D plotting functions
sqldf	0.4.10	Run SQL upon dataframe in R

PortHadoop-R VS Conventional Methods: Speedup (Vis-only) (backup)



Serial Copy: one copy command issued on one Hadoop node
Parallel Copy: 4 copy commands issued on each Hadoop node (32 in total)

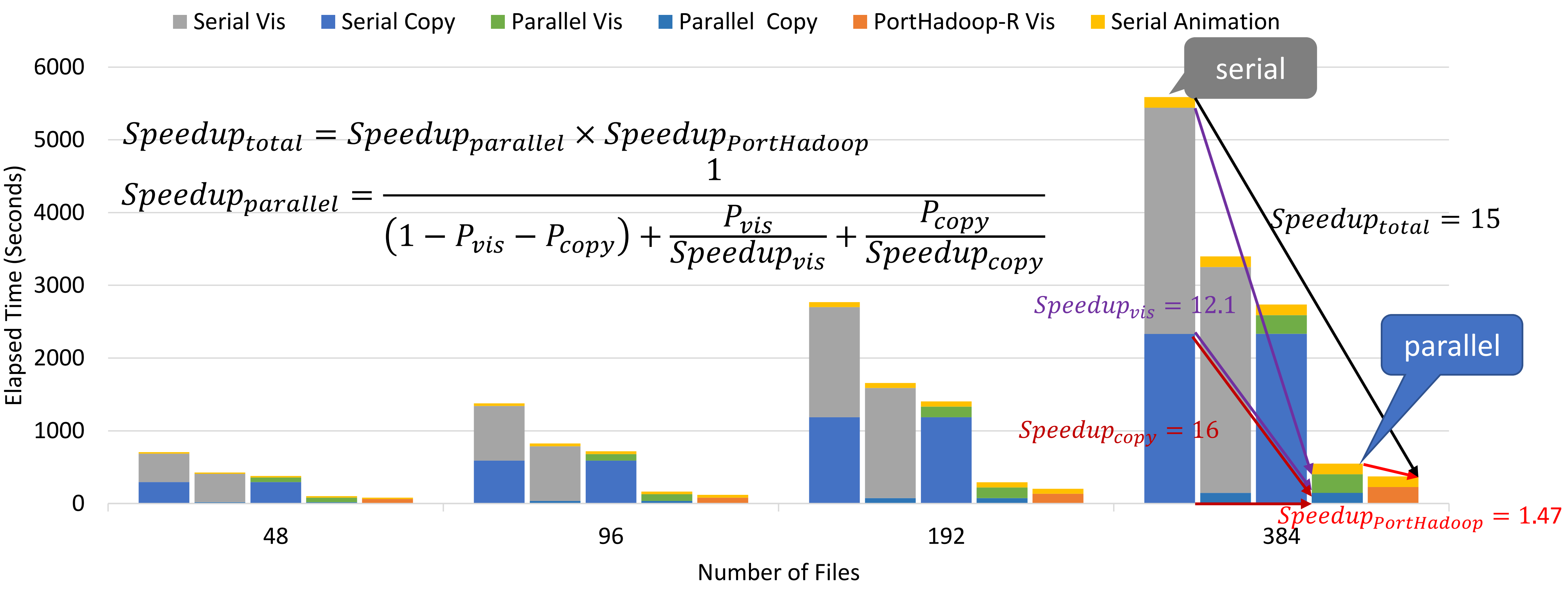
Serial Vis: read all data into one fat node, visualize and animate
Parallel Vis: visualize all files on all Hadoop nodes and generate animation

PortHadoop-R: visualize all files into images and animate

PortHadoop-R VS Conventional Methods: Performance (backup)

$$Speedup_{total} = Speedup_{parallel} \times Speedup_{PortHadoop}$$

$$Speedup_{parallel} = \frac{1}{\left(1 - P_{vis} - P_{copy}\right) + \frac{P_{vis}}{Speedup_{vis}} + \frac{P_{copy}}{Speedup_{copy}}}$$



Serial Copy: one copy command issued on one Hadoop node

Parallel Copy: 4 copy commands issued on each Hadoop node (32 in total)

Serial Vis: read all data into one fat node, visualize and animate

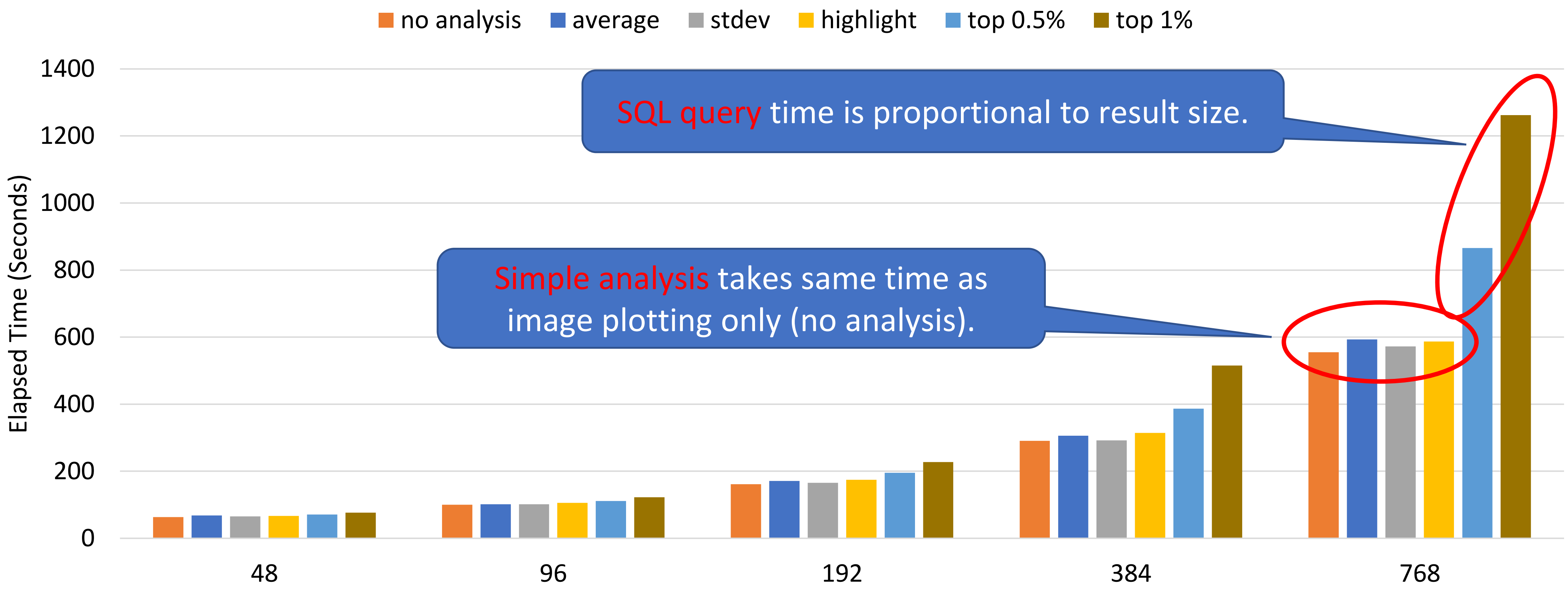
Parallel Vis: visualize all files on all Hadoop nodes and animate on the master node

PortHadoop-R Vis: visualize all files into images

Serial Animation: merge all images into one animation

ILLINOIS INSTITUTE OF TECHNOLOGY

PortHadoop-R Data Analysis Performance (Anlys) (backup)



no analysis: image plotting and animation only
average: plot image and calculate average of the data set
stdev: plot image and calculate the standard deviation of the data set

highlight: plot image and highlight the top 1% area
top 0.5%: plot image and subset the top 0.5% data of the data set
top 1%: plot image and subset the top 1% data of the data set

ILLINOIS INSTITUTE OF TECHNOLOGY