

# Modeling Resource-Coupled Computations

Mark Hereld

Computation Institute

Mathematics and Computer Science

Argonne Leadership Computing Facility

Argonne National Laboratory

University of Chicago



# Roadmap

- **issues and ideas**
- **models and measurements**
- **implications and work in progress**





# Issue

- Given increasingly massive (and complex) datasets...
- how to connect them to computational and display resources that support visualization and analysis?
- holistic approaches to allocating simulation, analysis, visualization, display, storage, and network resources
- create and exploit ways to optimally couple these resources in real time



## Common sense

- Analysis engines must be co-located with simulation engines
- ...or even, analysis code must be co-located with simulation code, i.e **in situ**
- Display resources must be integrated locally with HPC resources
- In general, wide-area applications will become impossible...
  
- But, maybe the situation isn't so dire.



---

## ideas

- Ideas
- Models
- Measurements
- Consequences
- Future

ideas



# Mitigation

- More efficient I/O practices
  - Many (most) inefficiencies in R/W rates amenable to better practices by application developer
  - In addition to improvements in performance of I/O libraries
- Better data management
  - Better data layout
- Better brute force compression methods
  - Uncertainty aware; domain aware
- Leveraging limitations at the destination
  - Pixel real estate
  - Perceptual limitations (and features)



# Coupled Resources

- **remote visualization:** couple data and large computational resources to remote display hardware
- **in situ** analysis and visualization: merge simulation and analysis code on single machine
- **co-analysis:** couple simulation on supercomputer to live analysis on visualization and analysis platform



---

## models

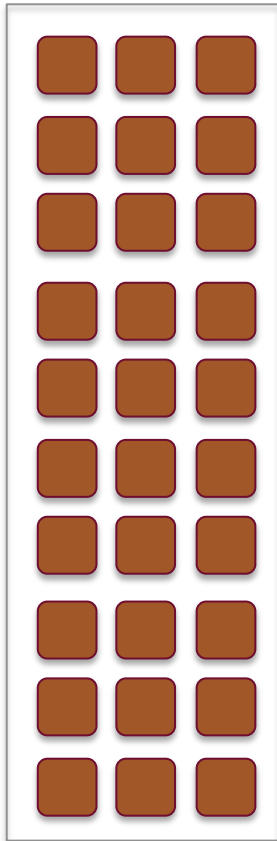
- Ideas
- Models
- Measurements
- Consequences
- Future

# models



# ALCF Network Architecture

40K BGP Compute Nodes



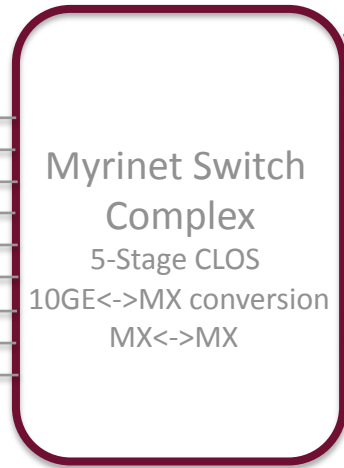
Tree  
4.3Tbps

640 BGP I/O Nodes



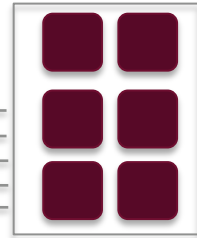
10GE  
  
640 x 10G  
= 6.4 Tbps

Tbps – Terabits/sec

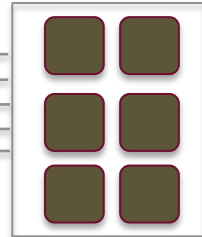


10G MX  
  
100 x 10G  
= 1 Tbps  
  
10G MX  
  
128 x 10G  
= 1.28 Tbps

Eureka  
100 Nodes



128 File-Server  
Nodes



Theoretical Max Bandwidth from **I/O Nodes to Eureka** (Memory to Memory) = 1 Tbps

Bi-directional = 2 Tbps

Theoretical Max Bandwidth from **I/O Nodes to FileServer** (Memory to Memory) = 1.28 Tbps

Bi-directional = 2.56 Tbps

Theoretical Max Bandwidth from **Eureka to FileServer** (Memory to Memory) = 1 Tbps

Bi-directional = 2 Tbps

# Data Analytics Resource: Eureka

- Data analytics and visualization cluster at ALCF
- (2) head nodes, (100) compute nodes
  - (2) Nvidia Quadro FX5600 graphics cards
  - (2) XEON E5405 2.00 GHz quad core processors
  - 32 GB RAM: (8) 4 rank, 4GB DIMMS
  - (1) Myricom 10G CX4 NIC
  - (2) 250GB local disks; (1) system, (1) minimal scratch
  - 32 GFlops per server



# Application

- FLASH
  - Multi-physics code: Gravitation, nuclear chemistry, MHD
  - Laboratory to Universe
- Multiple (~20) simulations
  - 8km resolution, 10K to 100K blocks each ( $16 * 16 * 16$ ) voxel
  - 2 Racks (8K cores) of the ANL's Intrepid (BGP)
  - typical simulation is 10 runs each 12 hours
- O(hour) per checkpoint cycle
  - 66% time spent simulating
  - 33% time spent non-overlapping I/O









## measurements

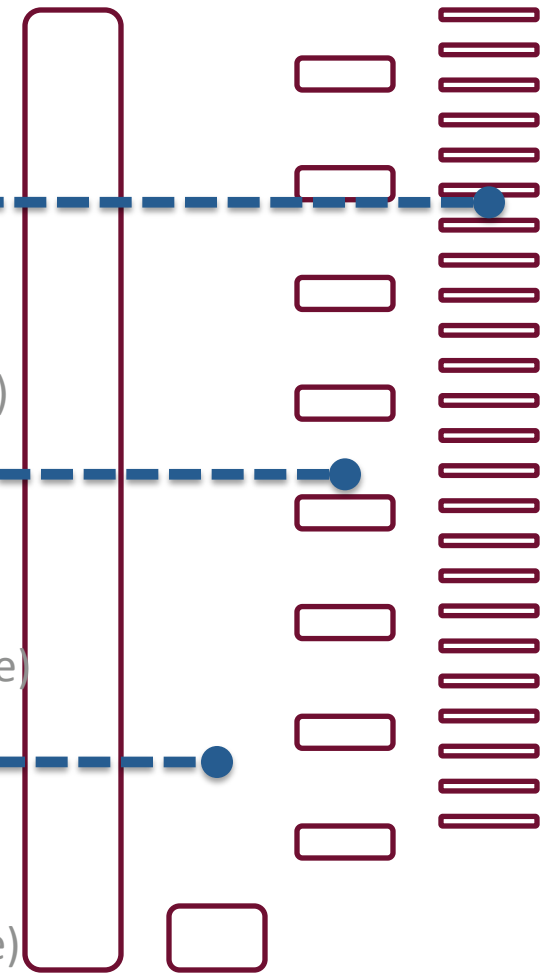
- Ideas
- Models
- Measurements
- Consequences
- Future

# measurements



# Flash IO for 1 run (12 hours)

- Total Run time = 41557 secs 
  - IO time during run = 14325 sec (34% of the time)
  - Circa March 2009
- Particle Data: 
  - 417 Files (0.1GB each) = 41.7 GB
  - Time spent writing = 9047 secs ( 22% of the run time)
- Plot files: 
  - 104 files (2.5GB each) ;Total = 260GB
  - Time spent in writing = 3897 secs ( 9% of the run time)
- Checkpoint files: 
  - 10 files (8 GB each) ;Total = 80GB
  - Time spent in writing = 1144 secs ( 3% of the run time)



# FLASH Supernova Explosion Project

- multiple (~20) simulations
  - 8km resolution
  - 10K to 100K blocks each (16 \* 16 \* 16) voxel
  - 2 Racks (8K cores) of the ANL's Intrepid (BGP)
  - typical simulation is 10 runs each 12 hours
  - Circa November 2009

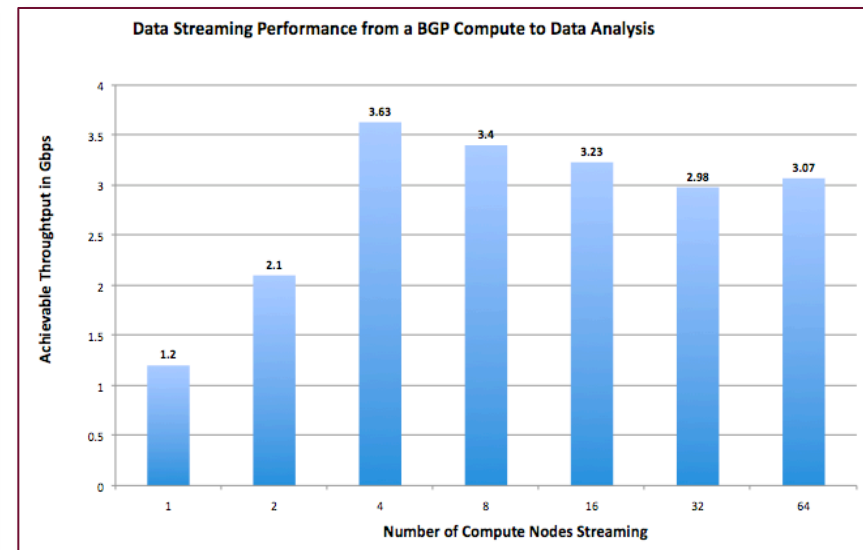
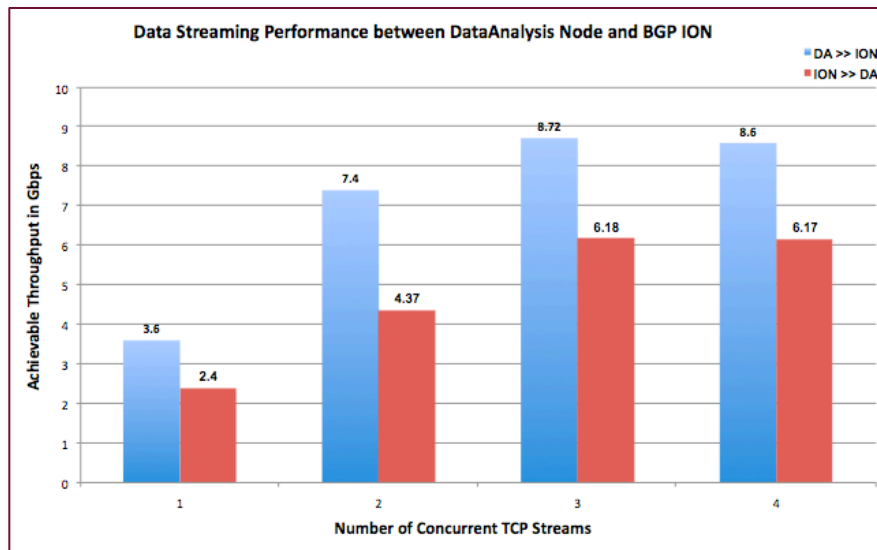
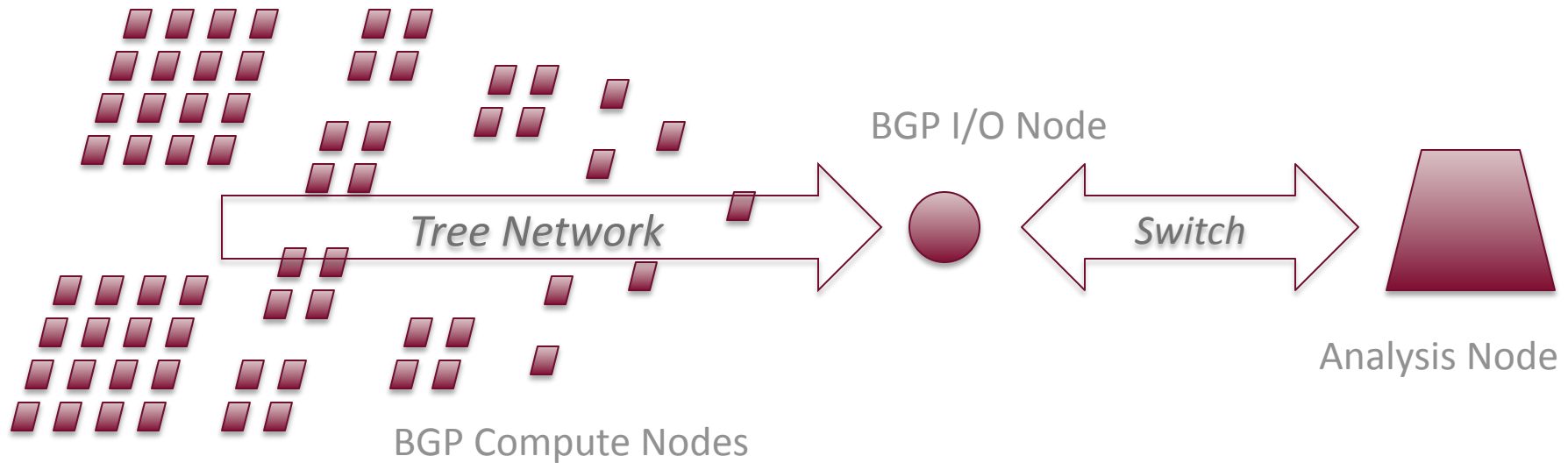
• =====

File Type	File Size	#files / Run	#files / Sim	Data Size
Particle	~ 131 MB	~ 500	5000	500 GB
Plot	~ 13 GB	40-90	800	10 TB
Checkpoint	~ 42 GB	5-10	100	4.2 TB

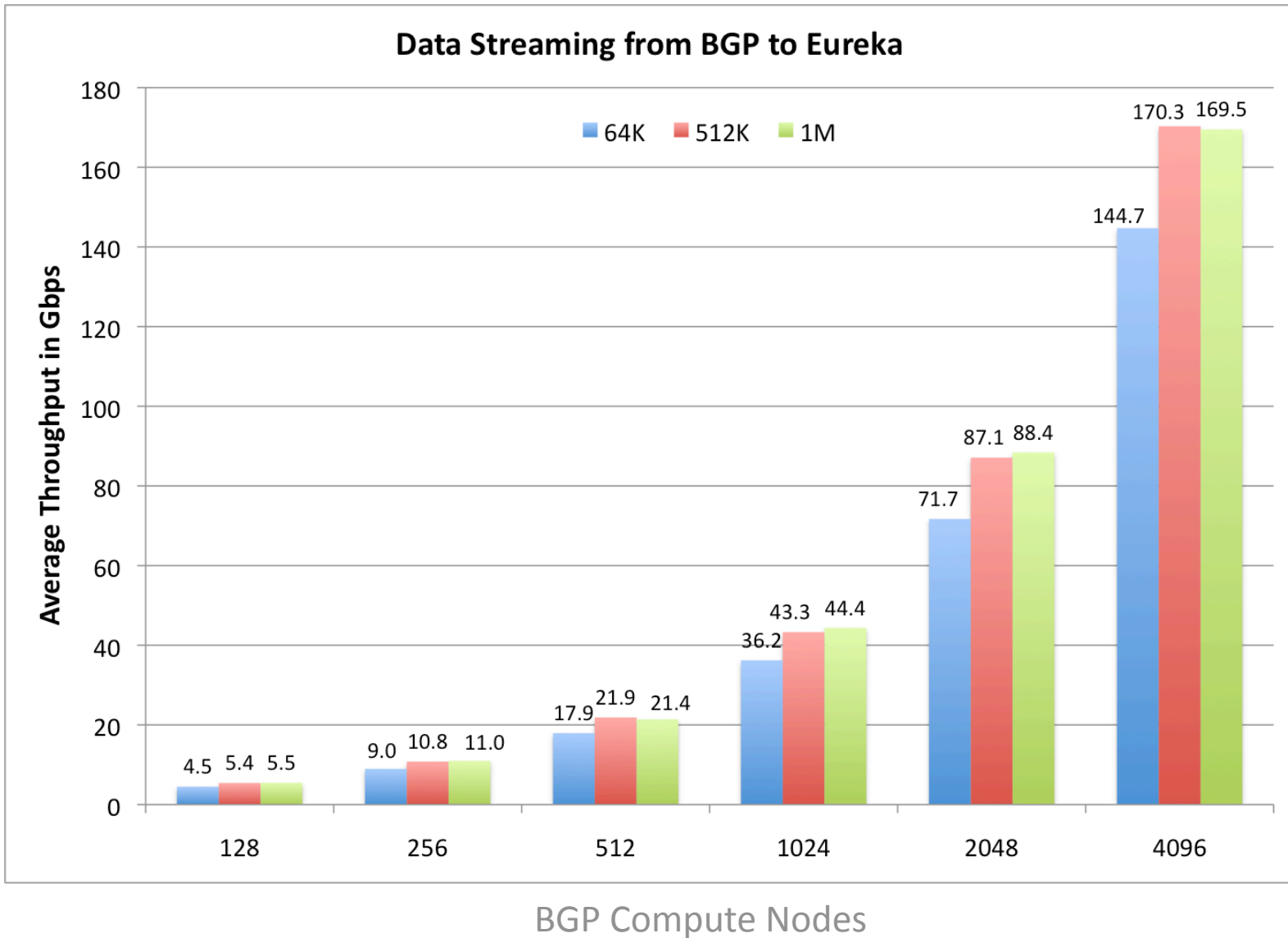
• =====



# Internal Network Experiments



# Toward middleware to facilitate co-analysis





## consequences

- Ideas
- Models
- Measurements
- Consequences
- Future

# consequences



# Map Intrepid I/O to Eureka

- Speed up the application
  - Offload data organization and disk writes
- Free co-analysis
  - Produce several high resolution movies
  - Data compression
  - Multi-time step caching for window analysis
- Eureka is an accelerator and co-analysis engine at only 1-2% cost of Intrepid



---

## future

- Ideas
- Models
- Measurements
- Consequences
- Future

# future

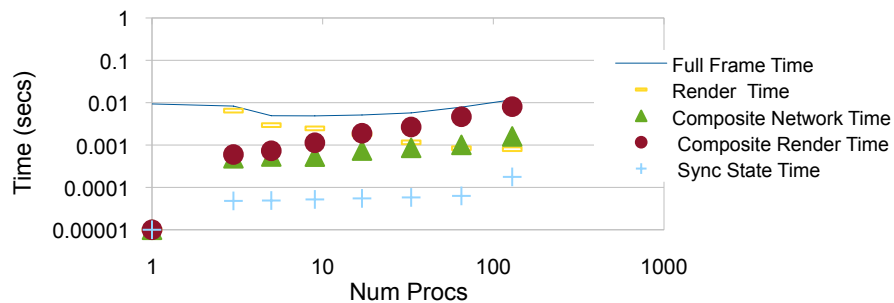


# Works in Progress

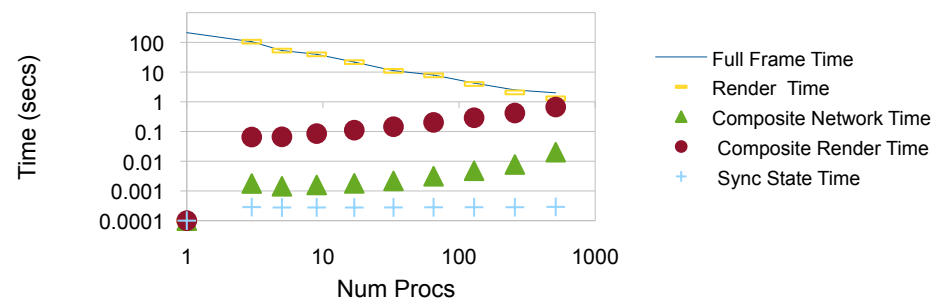
- Footprints
  - System level use pattern data collection
  - Booting up a mini-consortium of resource monitoring enthusiasts
- in situ
  - Papka parallel software rendering
  - Tom Peterka and Rob Ross scaling software rendering algorithms
  - HW-SW rendering comparison experiments
- Co-analysis
  - StarGate experiments
  - Intrepid <> Eureka communication experiments
  - FLASH test
- Remote Visualization
  - Pixel shipping experiments and frameworks



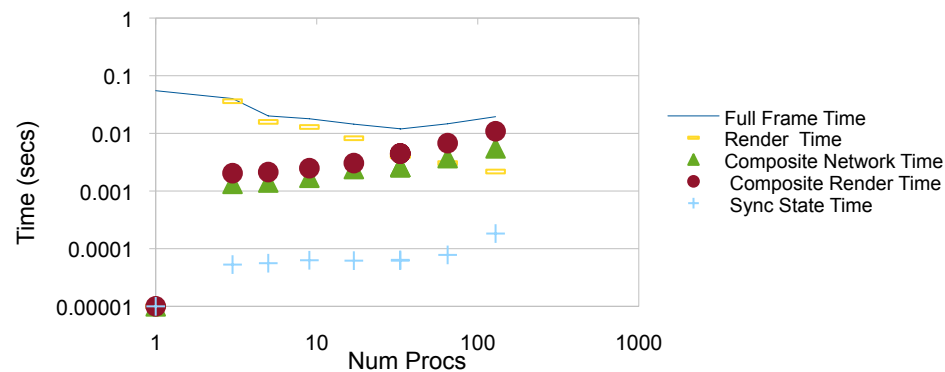
### Eureka Rendering Times 256x256x256



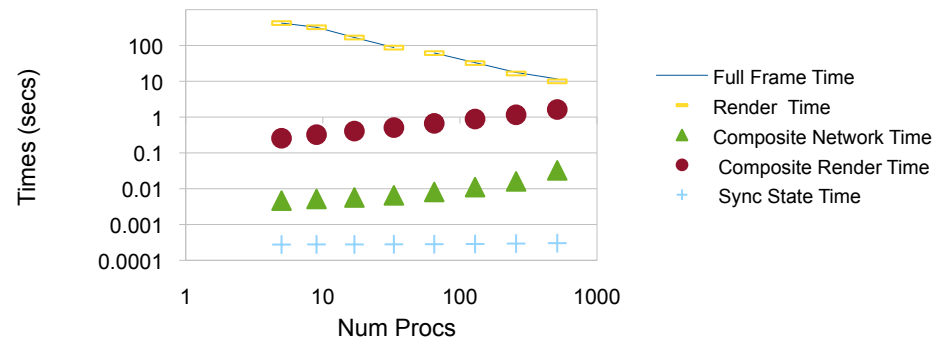
### Surveyor Rendering Times 256x256x256



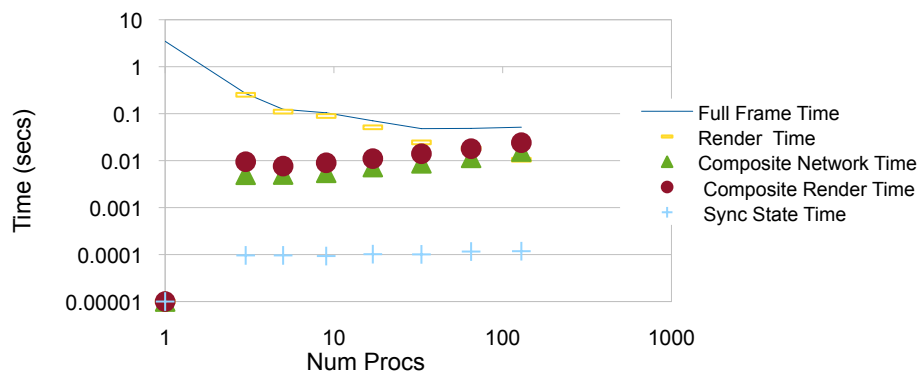
### Eureka Rendering Times 512x512x512



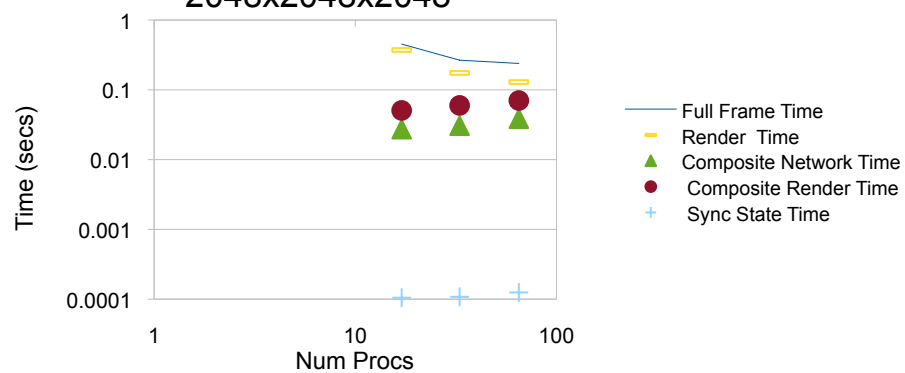
### Surveyor Rendering Times 512x512x512



### Eureka Rendering Times 1024x1024x1024



### Eureka Rendering Times 2048x2048x2048



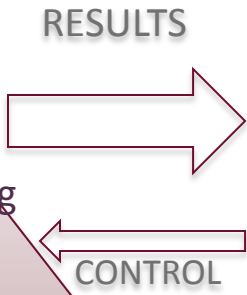
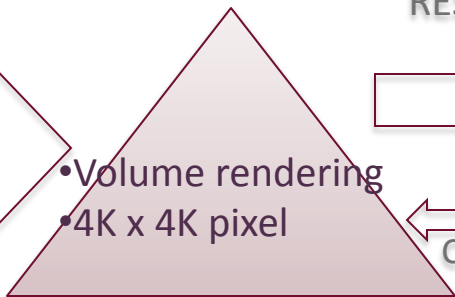
# Wide Area Experiments

## Simulation

- 4K uniform grid cube
- Single variable, float
- 257 GB per time step
- 577 time steps
- 150 TB total



## Visualization



## Interactive Display



DETAILS AND DEMO IN SDSU BOOTH



# Summary

- Discussion of the **issues** with illuminating example
  - Presumed impending Doom outlined
- Discussion of the **ideas** with examples
  - Resource-coupled computations
    - **In situ** couples simulation and analysis in real time on shared compute
    - **Remote vis** couples compute and data resources to remote display clients
    - **Co-analysis** couples two compute resources in real time
- Discussion of the **work in progress** with status
  - Suite of experiments underway to characterize system components
  - Strawman use cases in place provide challenging and exciting goals
  - Stunning results and paradigm shifts forthcoming



# Acknowledgements

- Venkat Vishwanath
- Michael Papka
- Eric Olson
- Joe Insley
- Tom Uram
- Tom Peterka
- Rob Ross
- Rick Stevens
- Rick Wagner, UCSD
- Michael Norman, UCSD
- Robert Harkess (UCSD)
- Narayan Desai
- David Ressler
- William Scullin
- Loren Wilson
- Linda Winkler
- ESNET2



---

end

- Questions?

end

