Networking Simulation Clusters with Visualization Clusters for Real-Time Data Analysis

Thomas Marrinan, Joseph A. Insley, Silvio Rizzi, and Michael E. Papka

Motivation

Basic Job Queue Paradigm

Computational Simulation
- Submit job
- Wait in queue
- Simulation runs to completion
- Results saved to disk

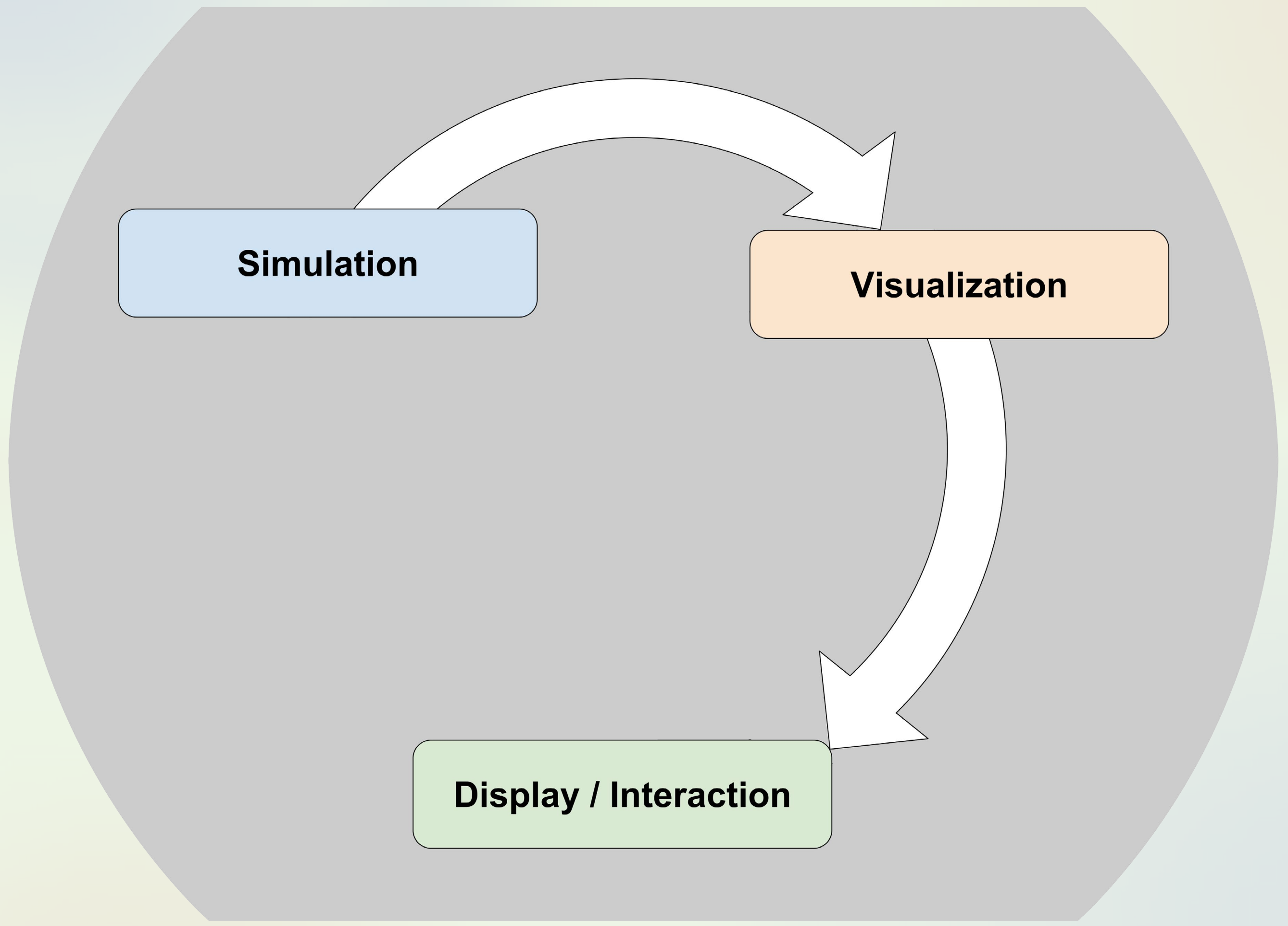
Post-processing of Data
- Read data from disk
- Visualize data
- Save resulting images / video to disk

Viewing Results
- Read images / video from disk
- View and explore the results

Real-Time Paradigm

- Concurrent visualization and display of a running simualtion
- Data streamed between simulation and visualization resources in real-time

*Real-time streaming **eliminates costly reads to / writes from disk**, enables a **higher sampling rate** for analysis, and can **reduce the time-to-discovery**!*



Results

Parallel Streaming

- Data distributed in *M* simulation ranks
- *N* visualization ranks each connect to a simulation rank
- Data streamed in parallel (*M*-to-*N*)

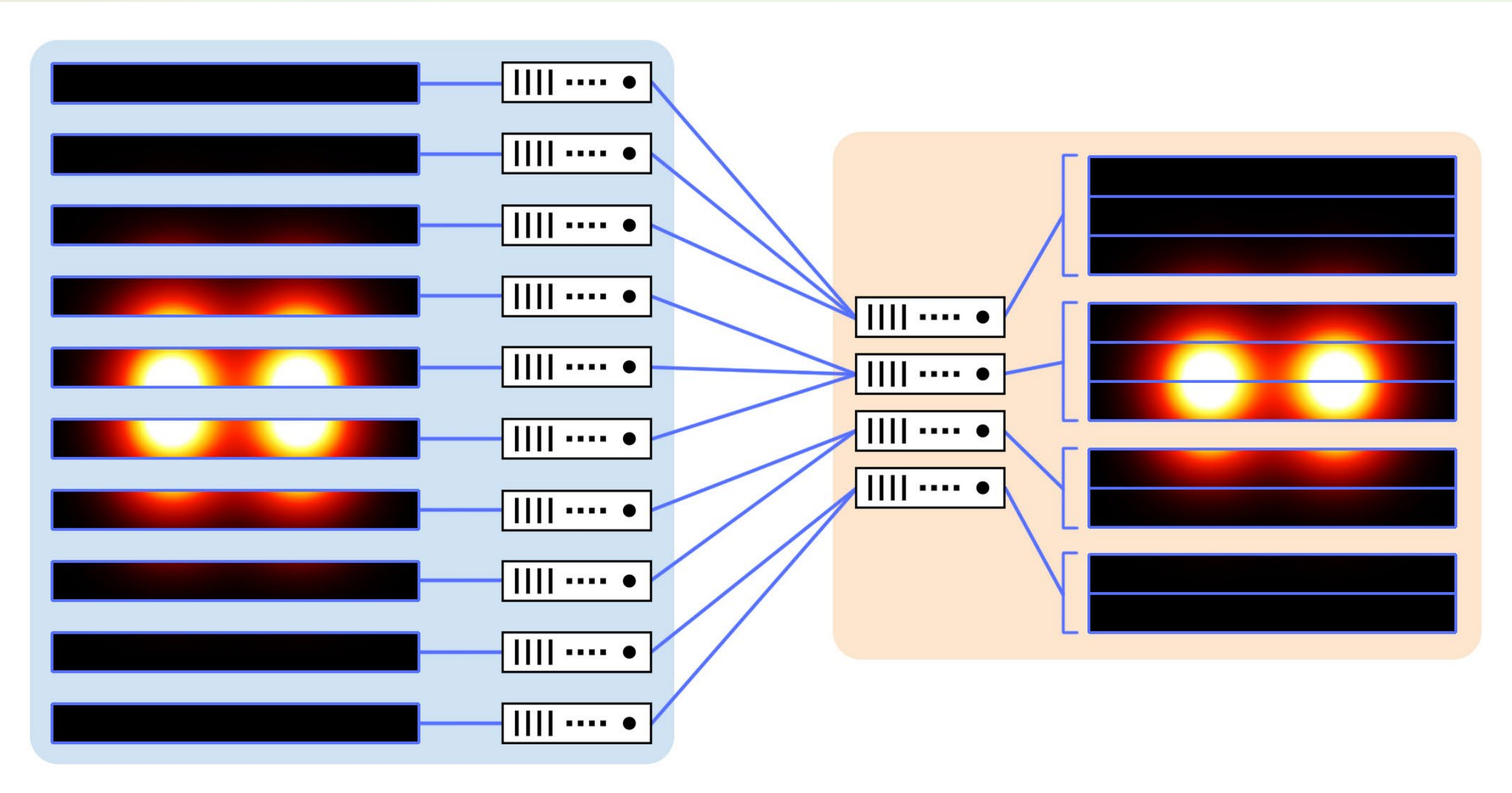
Data Redistribution

- Visualization declares how it expects data
- Incoming data is redistributed to fit the expected layout
- Works regardless of layout in simualtion

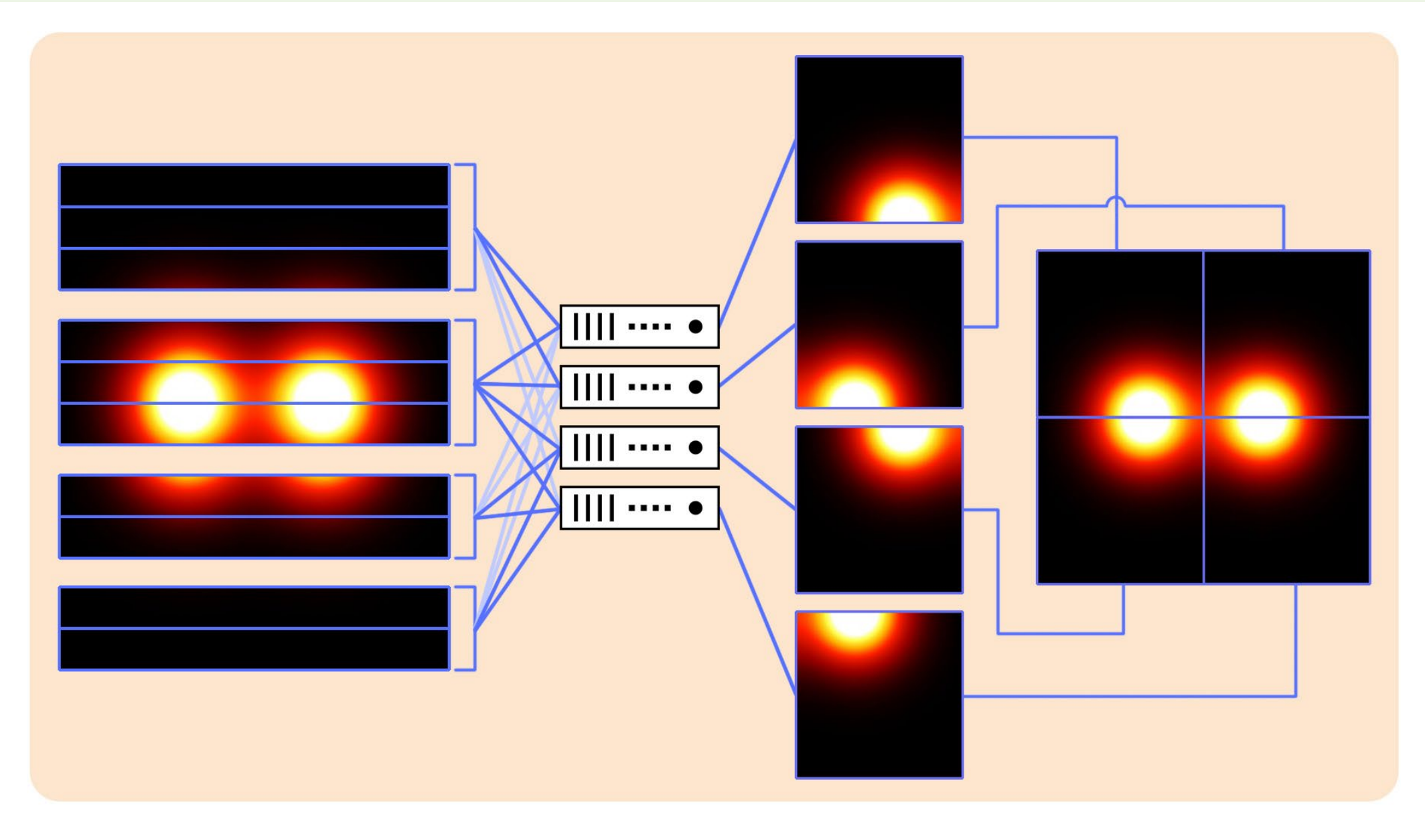
Use Case: 2D Heat Diffusion

Grid Dimensions	Raw Data Size	Processed Data Size
2048x2048	3.2 GB	19.9 MB
4096x4096	12.8 GB	61.0 MB
8192x8192	51.2 GB	217.8 MB
16384x16384	204.8 GB	830.9 MB

Raw data saved directly from 4-byte float array. Processed data saved as JPEG images rendered from visualization application.



Parallel data streaming of 2D heat diffusion data. This illustration shows 10 simulations ranks streaming data to 4 analysis ranks.



Data redistribution by the visualization application. Regardless of incoming data layout, data is automatically reorganized to fit the layout specified by the visualization application.

Impact

Wide-Ranging Applicability

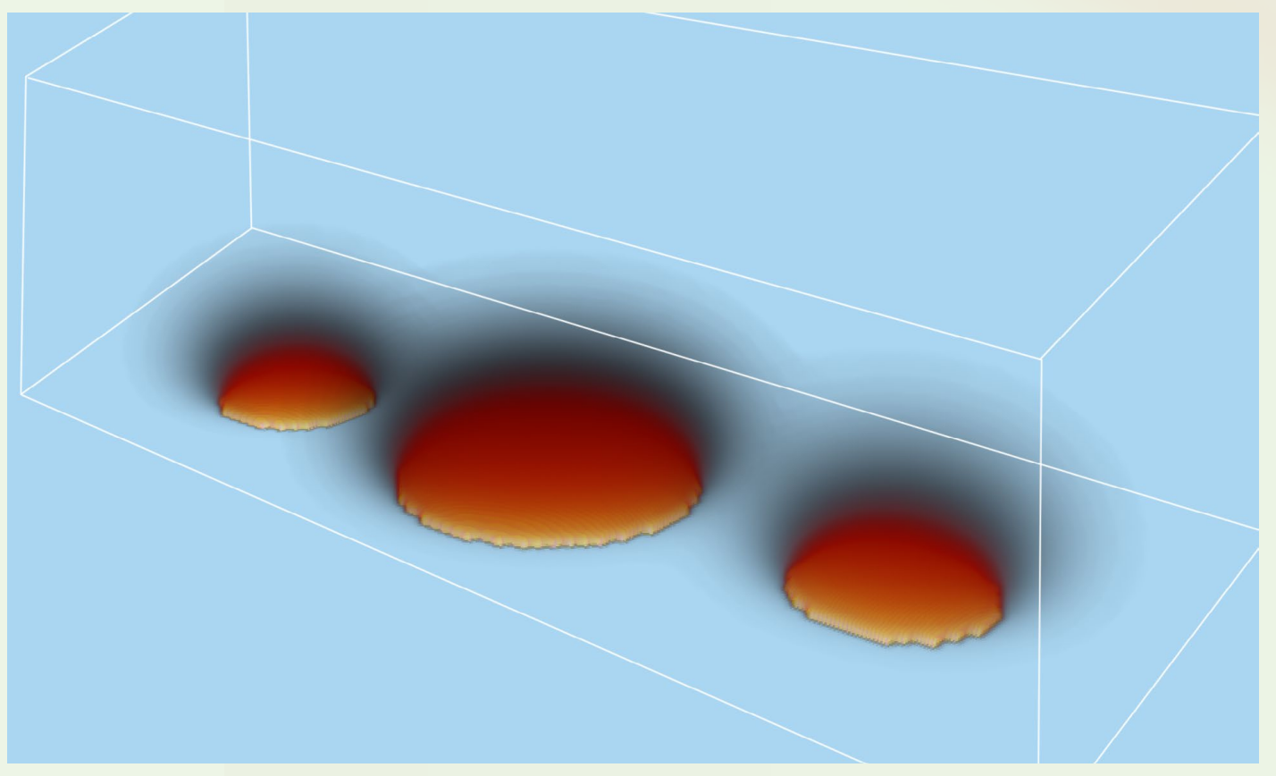
Computational Simulation and Visualization
- Improves sampling rate without increasing storage needs
- More fully utilizes compute resources (leverages high-bandwidth networks and GPUs)
- Visualization can be run and viewed on demand
- Necessary for exascale computing

Example Applications
- 2D regular grid simulations → texture mapped rendering
- 3D regular grid simulations → direct volume rendering
- Particle-based simulations → particle rendering

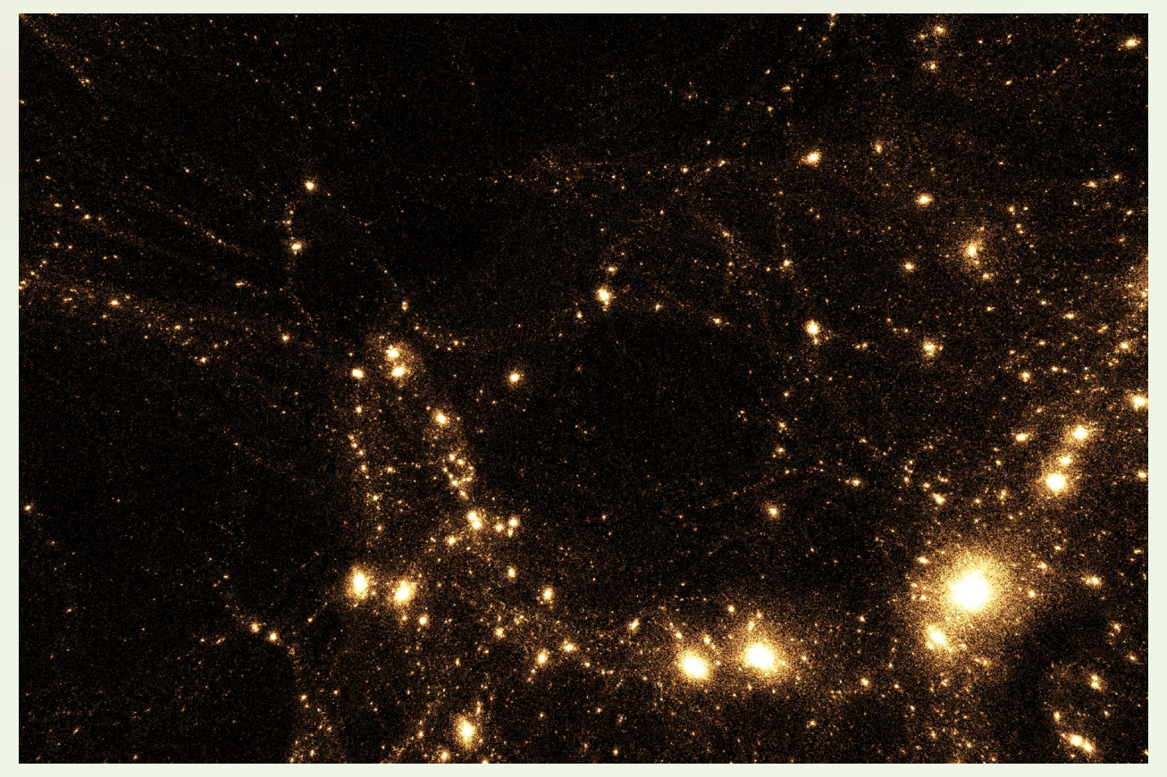
Benefits

Reusable Visualization Applications
- Generalizable (examples on left)
- Handle data from a variety of different simulations

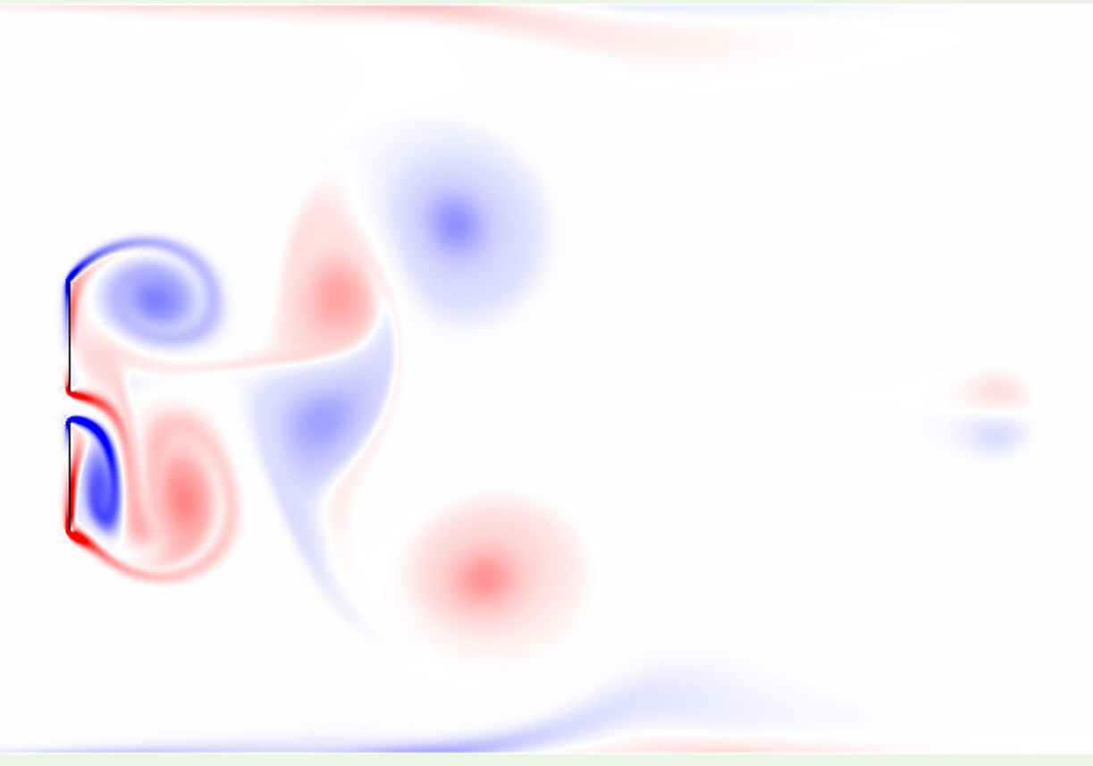
Little Overhead for Simulations
- Data redistribution done in the visualization application
- Limits run-time costs to the simulation



3D heat diffusion.



Dark matter particle physics.



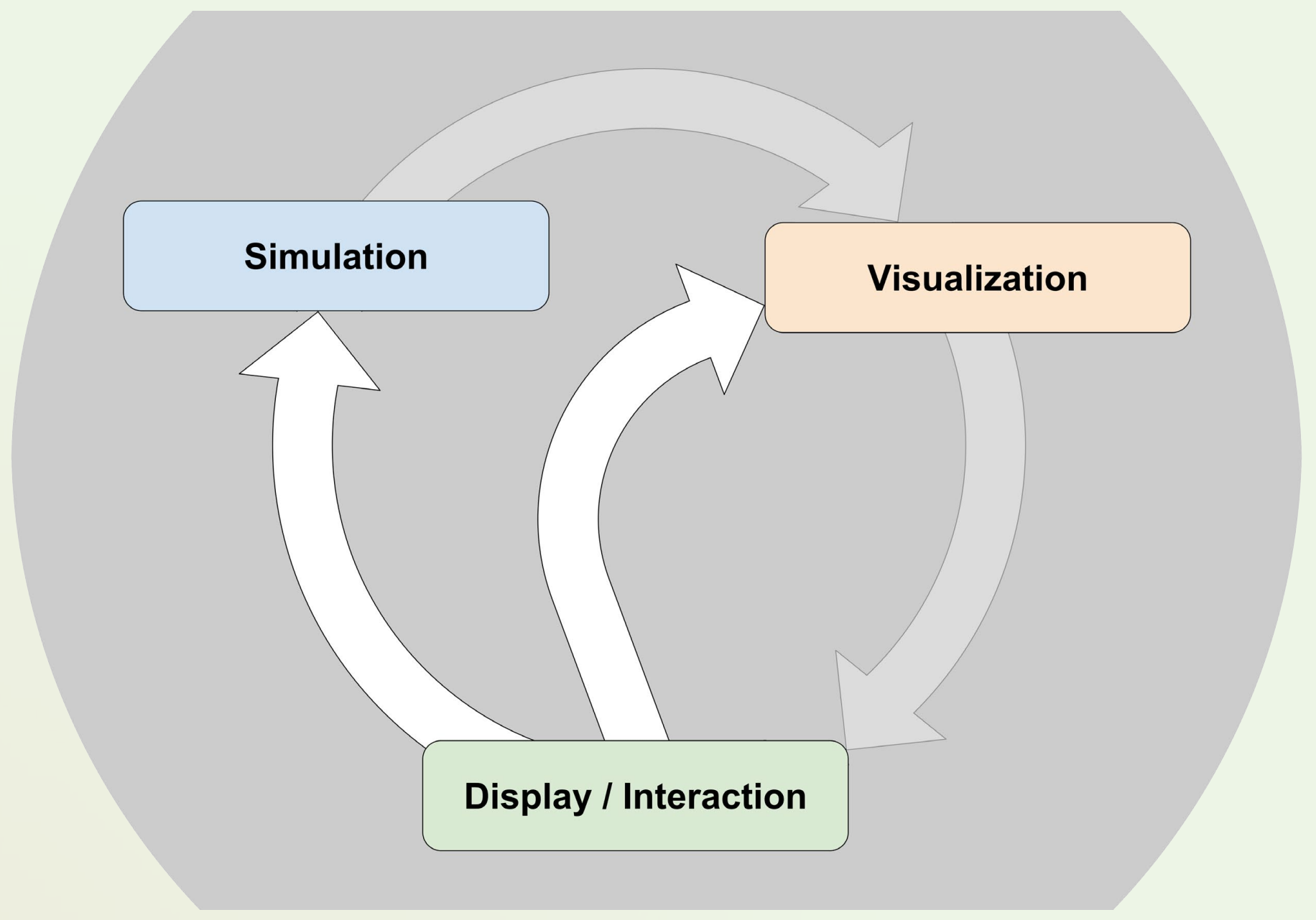
2D computational fluid dynamics.

Future Directions

Computational Steering

Feedback Loop for Simulation and Visualization
- User interaction to modify simulation or visualization
- Checkpointing system (allow for 'undo')
- Make HPC Applications look and feel like a desktop app

Example Uses
- Focus on area of interest
- Tweak parameters of simulation or visualization
- Explore 'what if' scenarios



Acknowledgements

This work is supported by the Department of Energy contract DE-AC02-06CH11357. We also thank Argonne National Laboratory's Leadership Computing Facility, where we evaluated our work.

Contact

Thomas Marrinan
Postdoctoral Appointee
Computing, Environment, and Life Sciences
Argonne National Laboratory

tmarrinan@anl.gov
(630) 252-1689