

# **Spatio-Temporal Data Processing and Visualization in Parallel Using UV-CDAT and ParaView**

**AMS 2013**

Presented By

**Aashish Chaudhary**

R & D Engineer, Kitware Inc.

# Overview

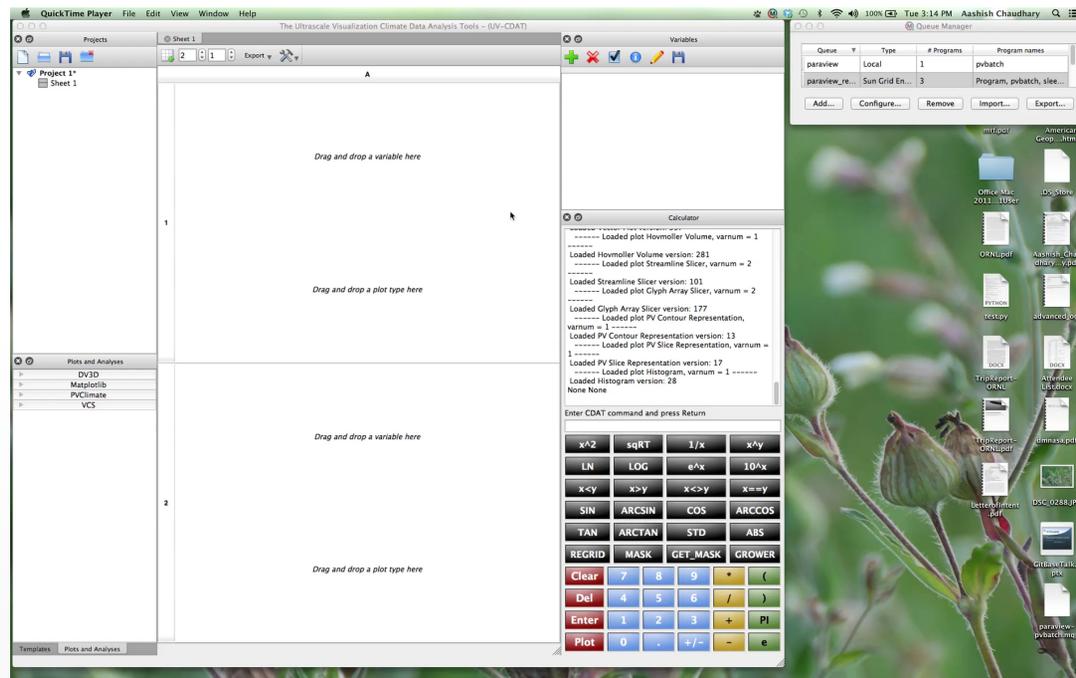
- Spatio-Temporal parallelism with ParaView in **UV-CDAT**
  - Demo movie
  - Demo details and workflow
  - Technical Details
    - Description
    - ParaView integration within **UV-CDAT**

# Overview

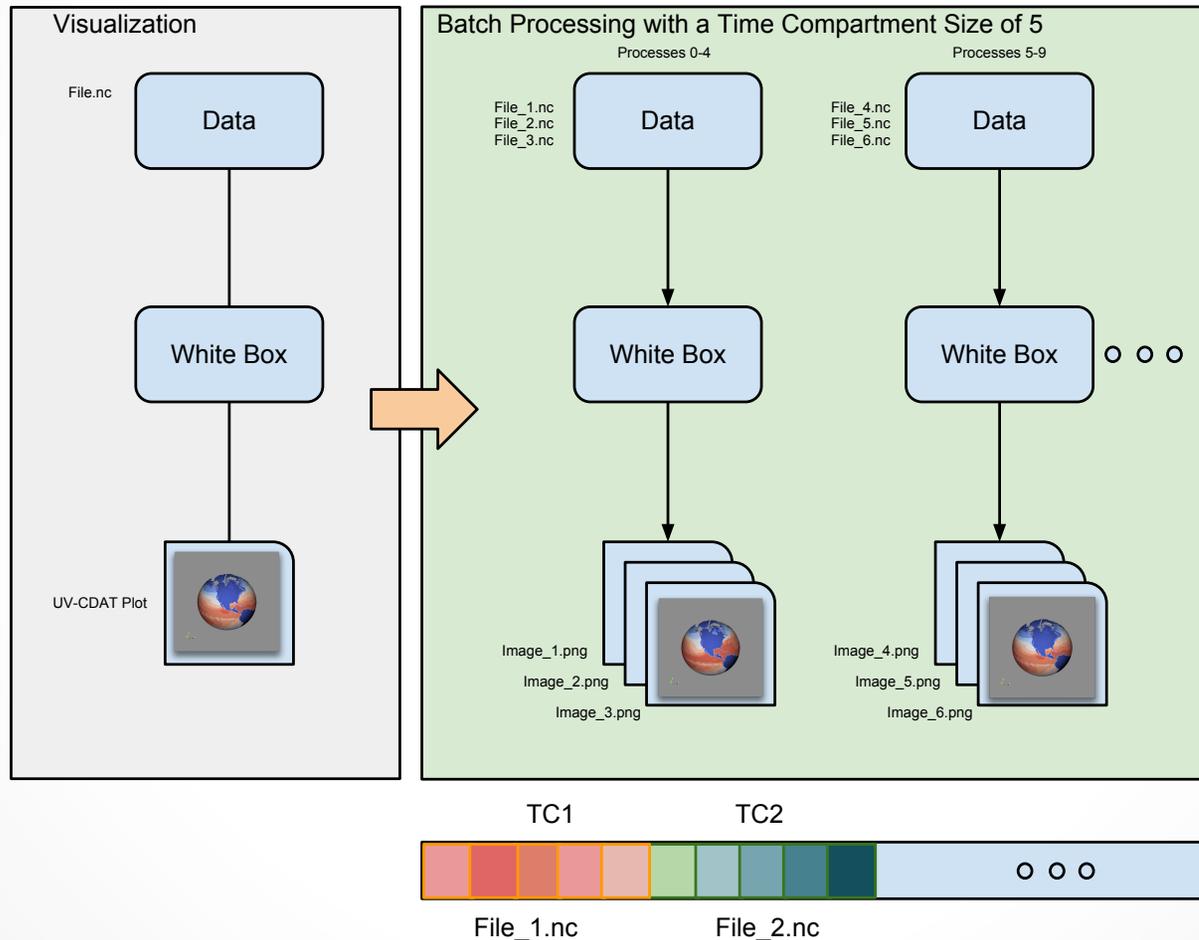
- **ParaView**
  - Introduction
  - Sources
  - Filters
  - Visualization
  - Client-Server model
  - Python API
  - MoleQueue

# Demo

- Implements **UV-CDAT** use case I; High spatial resolution, parallel, image sequence production



# Spatio-Temporal Parallelism



Spatio-Temporal Data Processing and Visualization in Parallel

# Demo - Workflow

- User creates a visualization
- User then selects
  - Input / Output location
  - Input dataset
  - Queue
- Users submits the job
- MoleQueue notifies the user when the job finishes
- User analyze the output

# Performance Metrics

Compartment Size	Number of Processes	Time (seconds)
1	46	1090 ~ 18 mins
1	92	785
1	184	Did Not Complete
2	184	454
2	368	Did Not Complete
4	368	307
<b>8</b>	<b>368</b>	<b>304 ~ 5 mins</b>
16	368	345

- As measured on Jaguar supercomputer
- 363 files (each file is one timestamp)
- Using 23 nodes
- Each timestamp is about ~1.4 GB
- Each node has 32GB
- Each node has 16 cores and two processors

# Demo - Tools

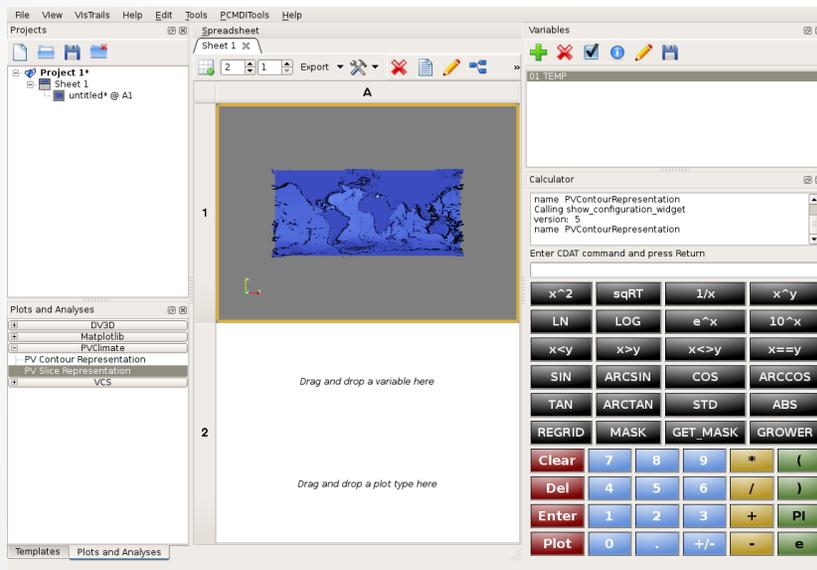
- Users **pvserver** (ParaView server)
- Uses **pvbatch**
  - Python interpreter
  - Command line executable specialized for batch processing
- Uses **MoleQueue**

# ParaView – Integration

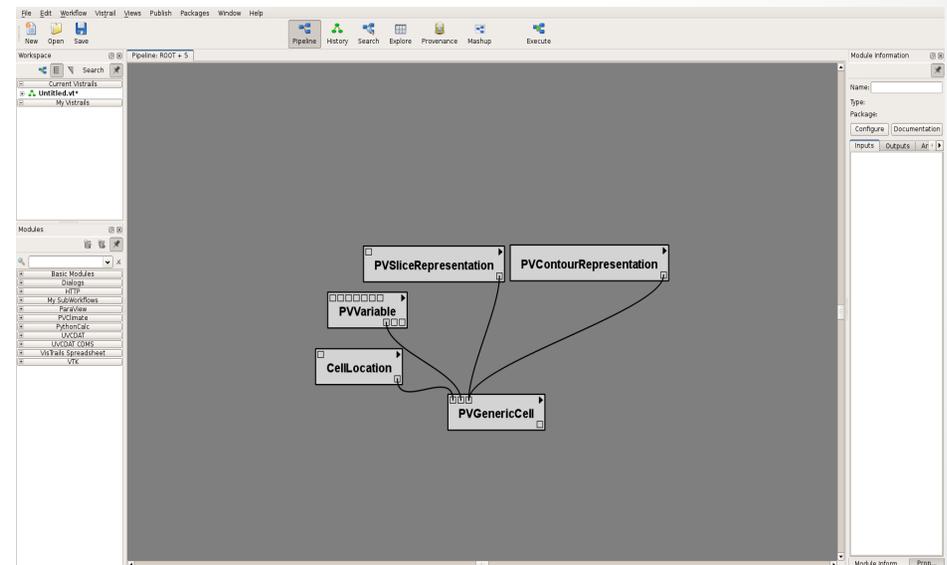
- Tight coupling
  - **ParaView** within **VisTrails** workflow
  - **Provenance**
  - Custom interface for Climate Scientists

# ParaView – Integration

- ParaView workflow



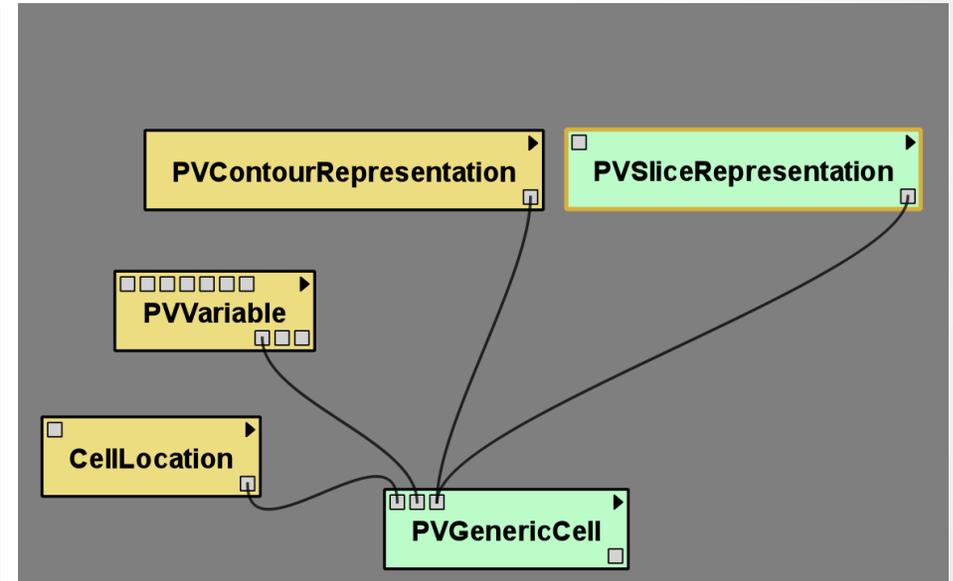
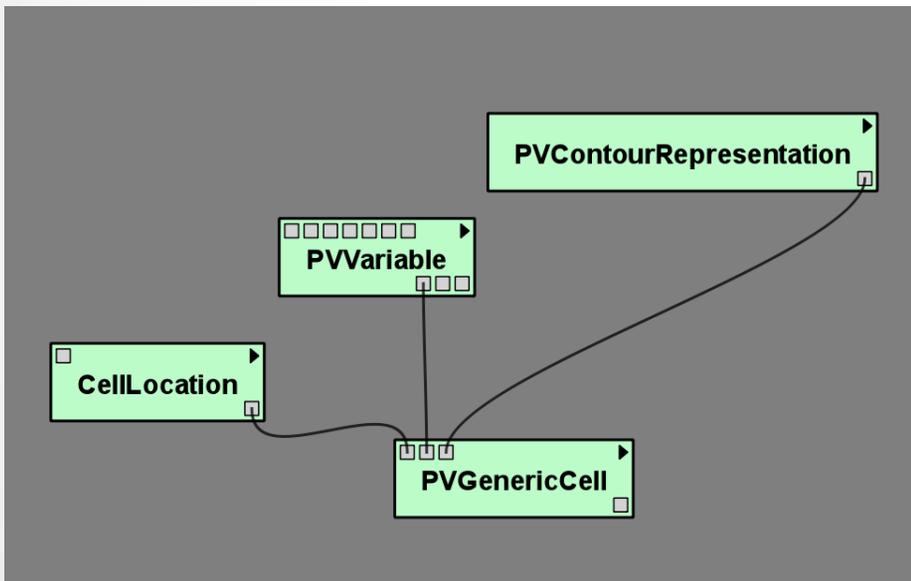
Visualization



Workflow

# ParaView - Integration

- Provenance



# ParaView - Integration

- Supports CDMS variable
- **Custom** representations
  - Easy to create representations
  - Common base class
- ParaView pipeline helper
  - Builds plot pipeline
  - Creates instances of ParaView VisTrails modules

# ParaView - Integration

- PVGenericCell
  - Contains view and can handle multiple input representations
- New readers
  - Unstructured POP reader
  - MOC reader
- New filters
  - Project sphere filter

# ParaView - Introduction

- An application and framework for the analysis and visualization of massive scientific datasets
- **Provides**
  - **Application** – You don't have to write code to analyze data
  - **Architecture** – Provides a framework to easily extend ParaView and is scalable



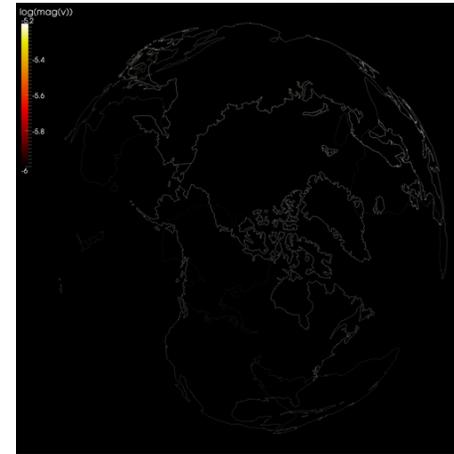
Sandia  
National  
Laboratories



# ParaView

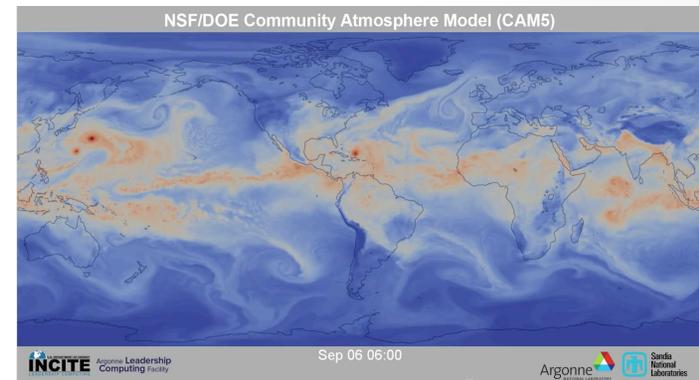
- Global seismic wave propagation simulation

(Courtesy: Visualization at the Texas Advanced Computing Center, The University of Texas at Austin by Greg Abram)



- Total perceptible water

(Courtesy: Argonne National Lab, Sandia National Lab)



# ParaView - Community

- **Active community**

Subscribers	Total	August 2012 Traffic
VTK users	3654	528
VTK developers	504	188
ParaView users	1098	296

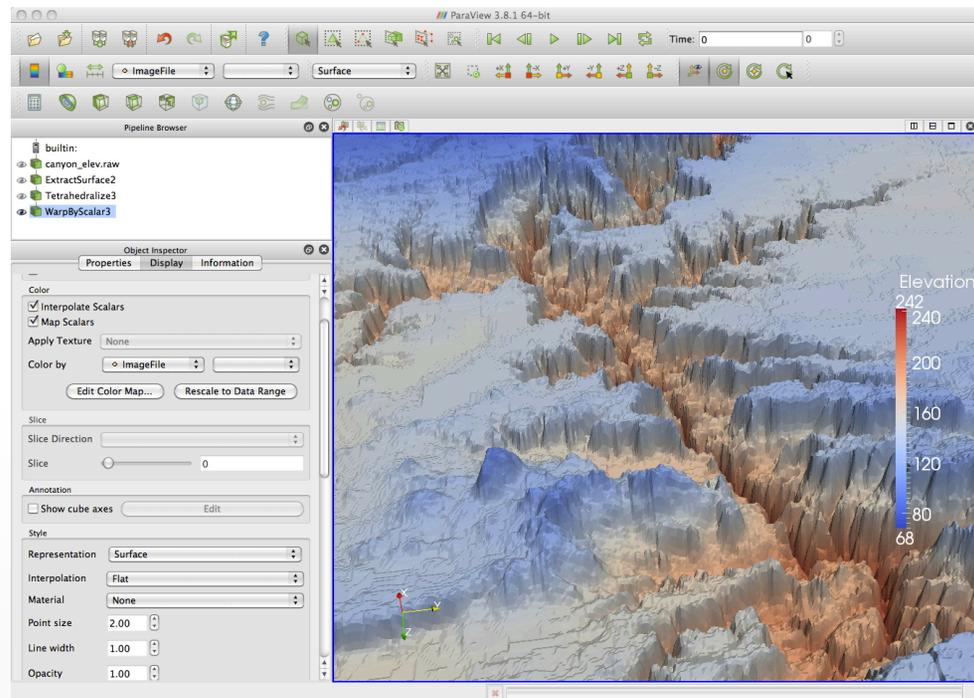
Active Developers	Count
VTK	32
ParaView	11

# Data Ingestion

- **Over 100 file formats supported**
  - Handles structured (uniform rectilinear, non-uniform rectilinear, and curvilinear grids), polygonal, unstructured, tabular, graph, multi-block, AMR and time varying data

# ParaView - Pipeline

- User builds a pipeline for data processing and visualization
- Example pipeline



# ParaView - Sources

- **Readers**

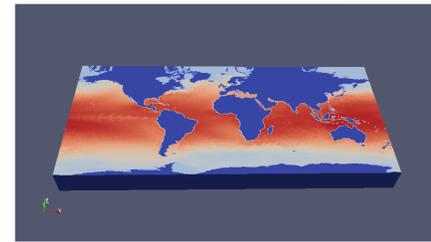
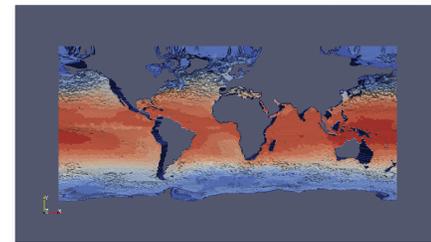
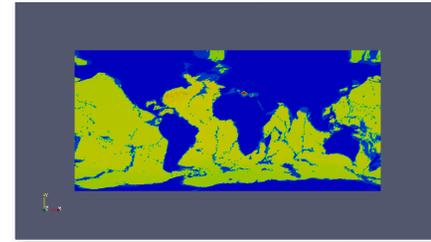
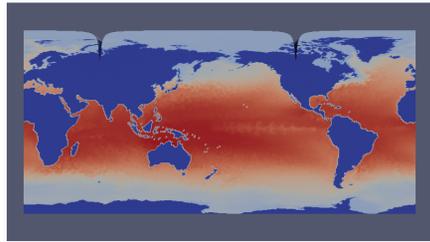
- NetCDF POP reader
- POP unstructured reader
- POP rectilinear reader

- **Generators**

- Cone source
- Sphere source
- Wavelet source

# ParaView - Filters

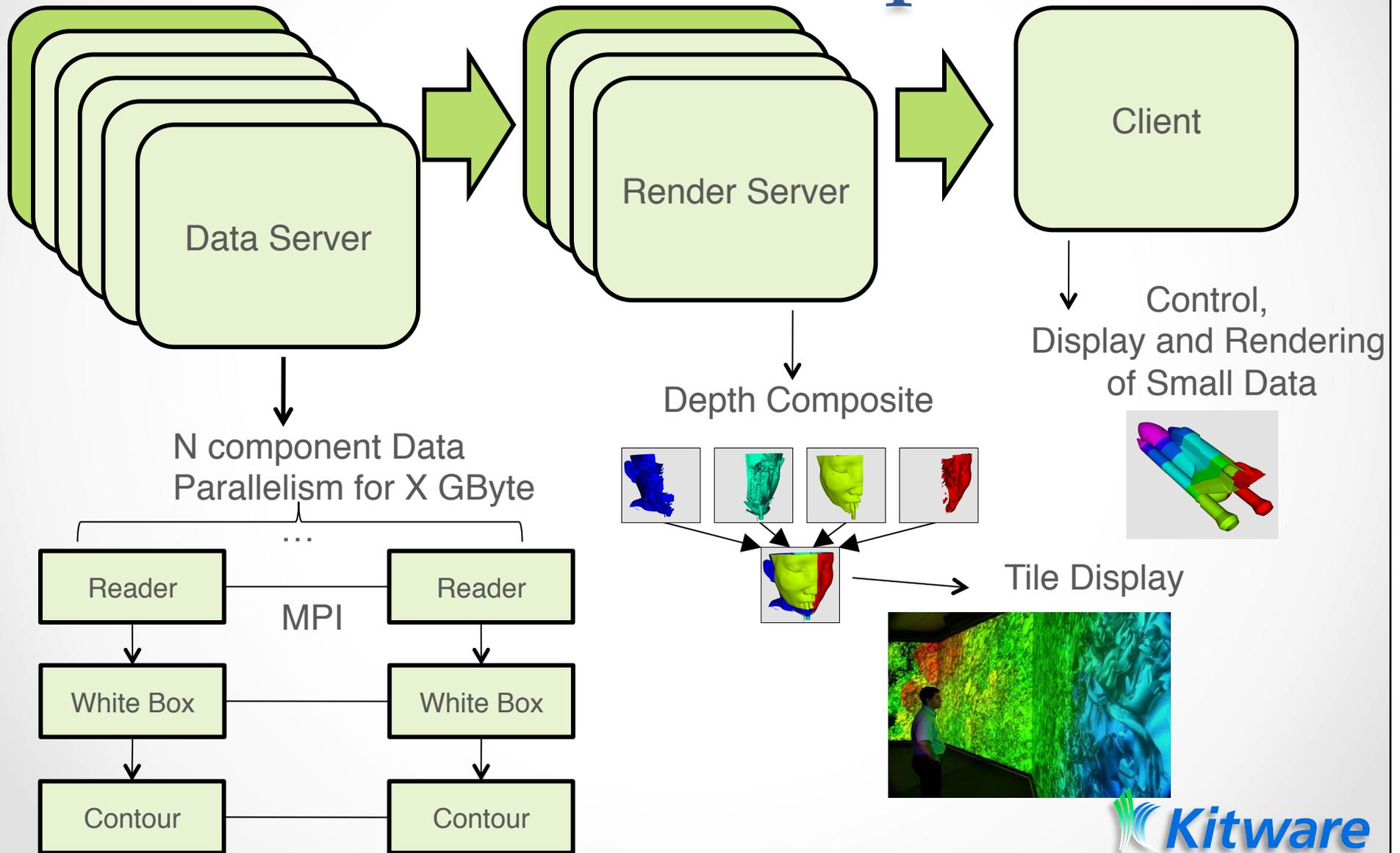
- Slice
- Contour
- Clip
- Project Sphere



# ParaView - Visualization

- **Standalone**
  - For smaller datasets
- **Parallel**
  - For large datasets

# ParaView - Components



# ParaView – Client Server

- **Data server**
- **Render server**
- Allows ParaView clients to run on variety of platforms
  - Mobile phones
  - Supercomputers

# ParaView - Python API

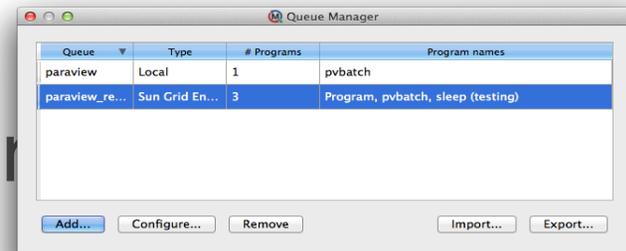
- **Control over the entire pipeline, not just inside one filter**
- Within or without GUI
  - In GUI (Tools->Python Shell)
    - Script and GUI state are Synched
    - tab completion and help browsing
  - Outside (pvpython, pvbatch, or standard python shell with paths)

# ParaView - Python API

- Supports batch processing
- Syntax is fully described in online wiki, but trace is a best way to learn it
- Used in UV-CDAT

# MoleQueue

- The MoleQueue application provides a graphical, standardized interface that bridges desktop applications with high-performance computing (HPC) resources.
- Support for Sun Grid Engine, Portable Batch System-base queuing systems and the local workstation.
- C++ and Python client libraries



# Team

- Berk Geveci (PI)
- Aashish Chaudhary
- Andrew Bauer
- Chris Harris
- Dave DeMarle

