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**
  o Demo movie
  o Demo details and workflow
  o Technical Details
    • Description
    • ParaView integration within **UV-CDAT**
Overview

• ParaView
  o Introduction
  o Sources
  o Filters
  o Visualization
  o Client-Server model
  o Python API
  o 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

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

- 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
  o **Application** – You don’t have to write code to analyze data
  o **Architecture** – Provides a framework to easily extend ParaView and is scalable
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**

<table>
<thead>
<tr>
<th>Subscribers</th>
<th>Total</th>
<th>August 2012 Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTK users</td>
<td>3654</td>
<td>528</td>
</tr>
<tr>
<td>VTK developers</td>
<td>504</td>
<td>188</td>
</tr>
<tr>
<td>ParaView users</td>
<td>1098</td>
<td>296</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Active Developers</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTK</td>
<td>32</td>
</tr>
<tr>
<td>ParaView</td>
<td>11</td>
</tr>
</tbody>
</table>
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
  o NetCDF POP reader
  o POP unstructured reader
  o POP rectilinear reader

• Generators
  o Cone source
  o Sphere source
  o Wavelet source
ParaView - Filters

- Slice
- Contour
- Clip
- Project Sphere
ParaView - Visualization

- **Standalone**
  - For smaller datasets

- **Parallel**
  - For large datasets
ParaView - Components

Data Server

Render Server

Client

N component Data Parallelism for X GByte

Depth Composite

Control, Display and Rendering of Small Data

Tile Display

Reader

Reader

MP1

White Box

White Box

Contour

Contour
ParaView – Client Server

• Data server
• Render server
• Allows ParaView clients to run on variety of platforms
  o Mobile phones
  o 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