ParaView Catalyst

Scalable *In Situ* Analysis and Visualization

http://catalyst.paraview.org/

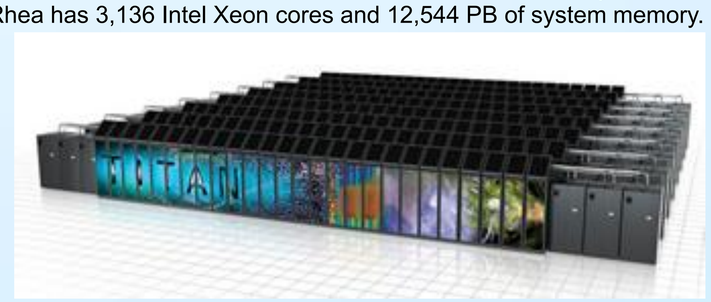
Ken Moreland[1], Jim Ahrens[2], Andy Bauer[3], Berk Geveci[3]
[1]Sandia National Laboratories, [2]Los Alamos National Laboratory, [3]Kitware, Inc.

Overview

To avoid data bottlenecks in big data simulations, an *in situ* toolset, ParaView Catalyst, integrates core data processing and visualization with simulations to enable scalable data analysis. I/O is one of the most pressing challenges with large-scale simulations, and it is already common for simulations to discard most of what they compute in order to minimize time spent on I/O. Since storing data is no longer viable for many simulation applications, data analysis and visualization must now be performed *in situ* with the simulation to ensure that it is running smoothly and to fully understand the results that the simulation produces.

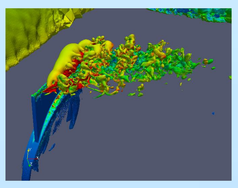
Disparity Between Computational Resources

Typically there is about 2 orders of magnitude difference of computational resources between the leading edge HPC hardware and corresponding visualization hardware. At OLCF, Titan has 299,008 Opteron cores, 18,688 NVIDIA Tesla K20 GPU accelerator cores and 710 TB of system memory. Rhea has 3,136 Intel Xeon cores and 12,544 PB of system memory.

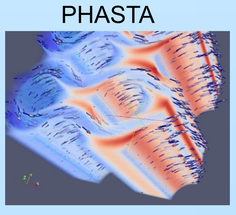


Simulation Codes Instrumented with Catalyst

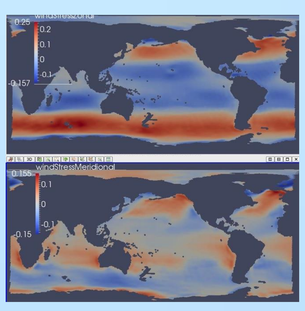
DoE	DoD	Other
Albany	AdH	BEC (William & Mary)
Alegra	GEMS	LESLIE (Georgia Tech)
CTH	Helios	PHASTA (UC Boulder)
Hydra	SM/MURF	Code Saturne (EDF)
Mantevo/miniFE-2.0		UH3D (SciberQuest, Inc.)
MPAS-O		
NPIC		
Sierra		
VPIC		
XRAGE		



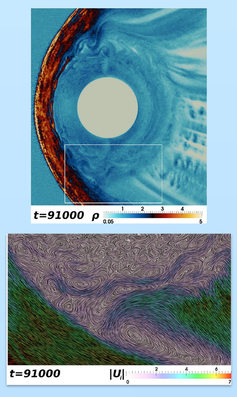
PHASTA



Code Saturne



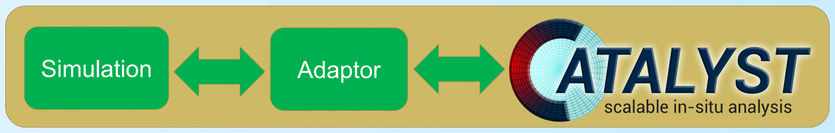
MPAS-O



UH3D

Architecture

ParaView Catalyst uses VTK's classes to represent grid and field information. Due to the variety of grids and field information that simulation codes provide, it is difficult to create a simple and efficient interface for automatically generating this information in VTK data structures inside of Catalyst. To this end, we use an adaptor design pattern to create VTK data structures from the simulation data structures. This results in low impact on the simulation code.



With this design there is the potential to waste memory since two separate data structures are used to store the same data. This can be alleviated by having VTK data structures directly use the simulation code data structure's memory.

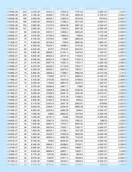
Workflow

The workflow for generating ParaView Catalyst output includes generating Catalyst pipelines to be executed during certain points in the simulation. The simplest way to do this is by generating Catalyst pipeline Python scripts in the ParaView GUI. Besides this, Catalyst C++ or Python pipelines can be hard-coded into the adaptor. In this case, the Catalyst pipelines typically are provided run-time parameters to give a bit more control over the generated output.

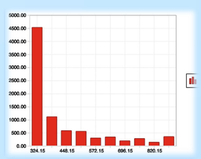
$$\rho \frac{D\vec{v}}{Dt} = -\nabla p + \mu \nabla^2 \vec{v}$$

Simulation + Catalyst

Statistics

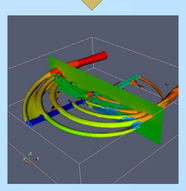


Series Data

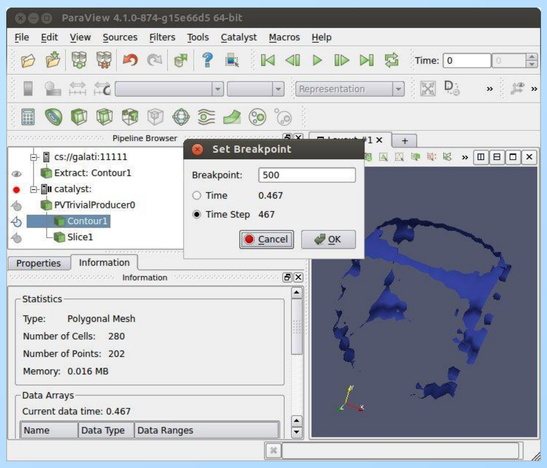


Data Extracts

Screen Shots

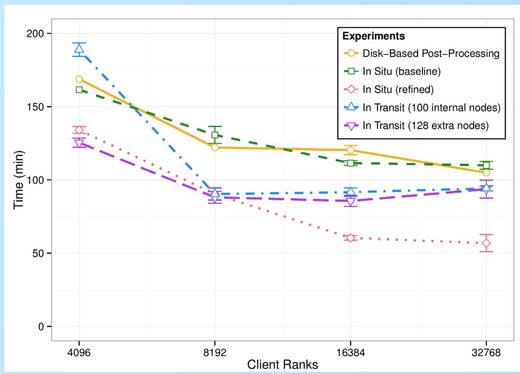


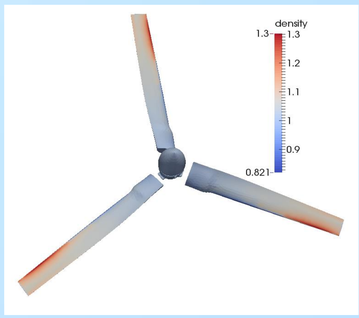
The ParaView client can connect to a Catalyst enabled simulation run to check on the status as well as change the Catalyst pipelines. For at scale runs the updates occur without stopping the simulation. For debugging runs the simulation can be stopped at specified points to check the simulation state.



Objective

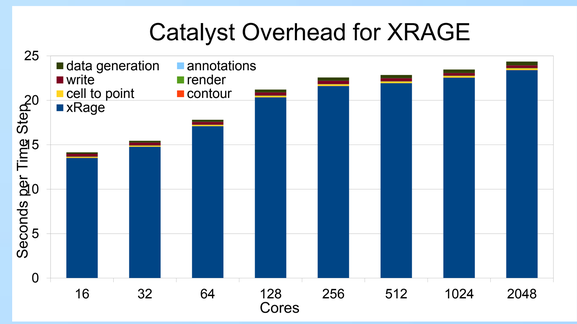
The main objective for using ParaView Catalyst with a simulation run is to decrease the amount of time it takes to gain insight into a given problem. This includes all pre-processing, simulation run and post-processing time required in the workflow. The graph below shows that using *in situ* analysis can significantly reduce this time to insight.





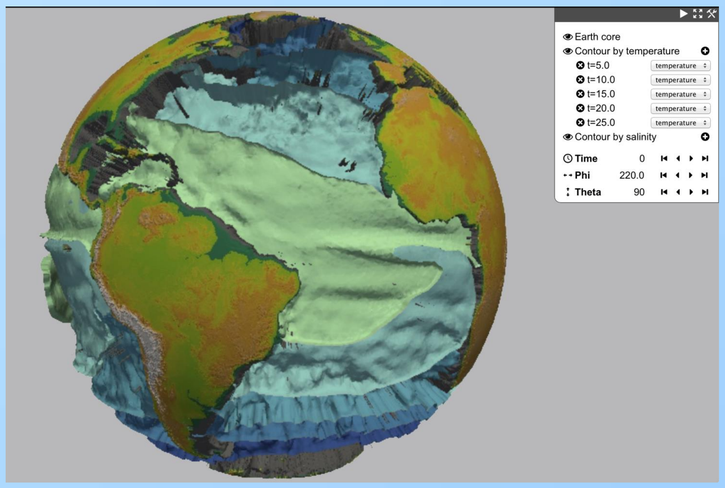
The keys to this time reduction include reducing the amount of file I/O and computing on appropriate hardware with scalable algorithms. For the Helios simulation shown on the right, for a single time step it took 448 MB to store the full simulation data, 2.8 MB to store the surface extract of the blades and 71 KB to store the image shown.

Because of the reduced I/O and scalable algorithms, the overhead cost of using ParaView Catalyst is kept low.



ParaView Cinema

ParaView Cinema can simplify accessing Catalyst output. This is done by having Catalyst automatically generate a variety of images at each time step through exploring parameter spaces of camera angles, filter settings, etc. Then the images are viewed interactively through a web-browser



SDAV Scalable Data Management, Analysis, and Visualization

U.S. DEPARTMENT OF ENERGY | Office of Science