DOE's Open Source Leadership in Scientific Visualization

2018 Salishan Conference on High-Speed Computing

Thursday April 26, 2018

Cyrus Harrison
Computer Scientist
Weapons Simulation and Computing, LLNL
Abstract

To meet the challenge of digesting and understanding massive data sets from HPC physics simulations, DOE's Advanced Simulation and Computing Initiative invested heavily in scientific visualization. These investments gave rise to two widely used and vibrant open source projects: VisIt and ParaView, each with large user bases and impact beyond DOE. This talk will provide perspectives of the trials and successes of the HPC scientific visualization community and how this community is working together to address the challenges from the evolving HPC landscape. It will also dive into how perceptions and use of open source solutions have evolved over the lifetime of these tools.
Thanks!

Thanks to the following for feedback, blessings, and advice:

- Kathleen Biagas (LLNL)
- Eric Brugger (LLNL)
- Hank Childs (University of Oregon)
- Berk Geveci (Kitware)
- Mark Miller (LLNL)
- Ken Moreland (SNL)
A little about me (to anchor my perspectives) ...

- I work for LLNL’s Weapons Simulation and Computing (WSC) program supporting scientific visualization and data analysis, most visibly on the VisIt Project

- How did I land here?
  - I joined LLNL in 2005 as a computer scientist and the VisIt team in 2007, initially in a unique role: Completely embedded to support data analysis of large scale multi-physics simulations

    A crude way to describe this: “Data analysis first and software development second”

  - Building from this, I grew into more general role on the VisIt team, but I still have the strongest affinity to DOE NNSA mission areas

  - Inherited the VisIt “Architect” role from Hank Childs in 2013
ASCI invested heavily to create scientific visualization capabilities in support of Stockpile Stewardship Science

- **Display Technologies**
  - High Resolution Screens
  - Tiled Displays

- **Visualization Clusters**
  - Leveraging GPUs for rendering
  - Connections to Displays

- **Software**
  - In-house, commercial and open source visualization tools

Images from: “A New World of Seeing”
LLNL Science & Technology Review, October 2000
https://str.llnl.gov/str/Quinn.html

This talk focuses on DOE’s success with two open source visualization tools
ParaView and VisIt are successful open source scientific visualization tools developed with DOE support.

ParaView

https://www.paraview.org

VisIt

https://visit.llnl.gov

These tools have widespread usage in DOE and over a million downloads worldwide.
Program documents and retrospectives hint at how the use of open source visualization tools evolved (1996)

1996 ASCI Program Plan identifies visualization and data analysis gap: “Such tools do not exist today”

Program documents and retrospectives hint at how the use of open source visualization tools evolved (2000)

January 2000 ASCI Program plan for VIEWS outlines investments in visualization tools

https://www.osti.gov/servlets/purl/768266
Program documents and retrospectives hint at how the use of open source visualization tools evolved (2000)

January 2000 ASCI Program plan for VIEWS outlines investments in visualization tools

https://www.osti.gov/servlets/purl/768266
Program documents and retrospectives hint at how the use of open source visualization tools evolved (2005, 2006)
Program documents and retrospectives hint at how the use of open source visualization tools evolved (2009)

Delivering Insight (a LLNL commissioned history of ASCI) highlights investment and success with: Ensight Gold, ParaView and VisIt

https://asc.llnl.gov/asc_history/Delivering_Insight_ASCI.pdf
Program documents and retrospectives hint at how the use of open source visualization tools evolved (2009)

Two other ASCI software tools projects, VisIt and ParaView, also provided important data manipulation capabilities. Both used open-source software as building blocks, such as the Visualization Tool Kit (VTK, from a company named Kitware) to visualize and interact with data. The use of open-source software in these tools was important because it fostered greater engagement with the external community and provided a more robust tool-development effort. Both VisIt and ParaView provide important data manipulation capabilities for ASCI users, and in recognition of its importance, VisIt was awarded an R&D 100 award in 2005.

Delivering Insight (a LLNL commissioned history of ASCI) highlights investment and success with open source in ParaView and VisIt

https://asc.llnl.gov/asc_history/Delivering_Insight_ASCI.pdf
The term “Open Source” was not even part of the nomenclature when ASCI started ...

The term “Open Source” was coined in 1998

https://opensource.org/history
https://opensource.net/20years
DOE ASC(I) and ASCR have explicitly embraced and promoted open source software development since 2002

Benefits outlined in joint 2003 ASCI and ASCR Memo:

- OSS provides HPC sites the opportunity to identify and fix bugs quickly

- The OSS model yields important contributions to the global state of the art, thus providing significant leverage of Government investments

- The OSS model provides a hedge against “change in support” status for software required to execute the missions of these Programs. This yields protection for the investments made in the software

- Access to source code of OSS can enhance cyber security by facilitating rapid identification and repair of security vulnerabilities

https://science.energy.gov/~media/ascr/pdf/research/docs/Doe_lab_developed_software_policy.pdf
Sharing amplifies:

- **Collaboration**
  - Lowers barriers to exchange ideas and code
  - Amortizes development effort
  - Helps establish a developer community

- **Adoption**
  - More users leads to a better understanding of your software
  - Helps establish a user community

- **Competition**
  - Reuse and leverage of existing work, but with access to source as stopgap to remedy issues
  - Brings transparency and avoids “secret sauce” myths
Open source and open development enable sharing

Sharing also can exacerbate:

- Priority Inversion
  - Visibility leads to many competing requests and directions

- Squeaky wheel scenarios
  - Open dialog about bugs and requests can asymmetrically overwhelm the kind hearted

Teams need to invest to grow user and developer communities to scale gracefully!
What are these tools used for?
Visualization and analysis tools are key part of the application ecosystem supporting HPC multi-physics simulations.
Scientific visualization tools support a wide range of use cases

- **Data Exploration**
- **Quantitative Analysis**
- **Visual Debugging**
- **Comparative Analysis**
- **Presentation Graphics**
Scientific visualization tools support a wide range of use cases

- Data Exploration
- Visual Debugging
- Comparative Analysis
- Quantitative Analysis
- Comparative Analysis
- Presentation Graphics

These tools are an important part of the daily workflow used to digest simulation data.
How were these tools created?
VTK was a key substrate that enabled rapid adoption of scientific visualization capabilities

- The software started in support of a book focused on how to do scientific visualization with C++ in 1993

- The success of VTK lead to the founding of Kitware in 1998

VTK was a key substrate that enabled rapid adoption of scientific visualization capabilities

- VTK enabled developers to create and share algorithms that apply to a wide range of meshed-based data sets
- VTK provided infrastructure to help compose these algorithms to support complex user operations
- VTK’s “toolkit” style design allowed ParaView and VisIt teams to share core infrastructure while still providing flexibility that enabled multiple solutions

C++ APIs for mesh-based data structures

Data-flow networks process mesh data
Scientific visualization tools need to scale across the compute nodes of a Supercomputer

Full Dataset
(27 billion total elements)

3072 sub-grids
(each 192x129x256 cells)
ASCI invested to build distributed-memory parallel visualization tools

- VTK did not have distributed-memory parallel capabilities when ASCI started
- ASCI supported research and development addressed this gap

2000: J. Ahrens, et al, outline and demonstrate approaches for parallelism in scientific visualization


ASCI invested to build distributed-memory parallel visualization tools

In ~1999/2000:

- A LLNL team began building a layer on top of VTK to support distributed-memory parallelism and created VisIt

- Kitware and LANL began to expand VTK to support distributed-memory parallelism and created ParaView

March 2000: ASCI directly supported Kitware’s open source development of VTK with a 3 year contract

https://blog.kitware.com/kitware-signs-contract-to-develop-parallel-processing-tools/
Both VisIt and ParaView employ a parallelized client-server architecture.
Domain decomposed meshes enable scalable parallel algorithms

- Simulation meshes are divided into smaller mesh domains
- Domains are partitioned across MPI tasks for processing
- Algorithms coordinate across MPI tasks when non-local info is needed
These tools use Data-flow networks in a Single Program Multiple Data (SPMD) paradigm to achieve distributed-memory parallelism.
VisIt’s “Contract” extension to Data-flow networks demonstrated how to flexibly support optimizations

“A Contract-Based System for Large Data Visualization”, H. Childs et al, IEEE VIS 2005
Distributed-memory compositing is the foundation for parallel rendering

ParaView and VisIt both leverage the open source IceT library (http://icet.sandia.gov) for scalable rendering
Distributed-memory mesh connected components for unstructured meshes is a challenging example

“Data-Parallel Mesh Connected Components Labeling and Analysis”, C. Harrison, H. Childs, K.P. Gaither, EGPV 2011
The VisIt team demonstrated this approach scales to very large data sizes in 2009 and 2010

<table>
<thead>
<tr>
<th>HPC Cluster</th>
<th>Architecture</th>
<th>Problem Size</th>
<th># of Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph (LLNL)</td>
<td>x86_64</td>
<td>20,001² (8 T cells)</td>
<td>12K</td>
</tr>
<tr>
<td>Dawn (LLNL)</td>
<td>BG/P</td>
<td>15,871³ (4 T cells)</td>
<td>64K</td>
</tr>
<tr>
<td>Franklin (NERSC)</td>
<td>Cray XT4</td>
<td>12,596³ (2 T cells)</td>
<td>32K</td>
</tr>
<tr>
<td>JaguarPF (ORNL)</td>
<td>Cray XT5</td>
<td>12,596³ (2 T cells)</td>
<td>32K</td>
</tr>
<tr>
<td>Juno (LLNL)</td>
<td>x86_64</td>
<td>10,000³ (1 T cells)</td>
<td>16K</td>
</tr>
<tr>
<td>Franklin (NERSC)</td>
<td>Cray XT4</td>
<td>10,000³ (1 T cells)</td>
<td>16K</td>
</tr>
<tr>
<td>Ranger (TACC)</td>
<td>Sun</td>
<td>10,000³ (1 T cells)</td>
<td>16K</td>
</tr>
<tr>
<td>Purple (LLNL)</td>
<td>IBM P5</td>
<td>8,000³ (0.5 T cells)</td>
<td>8K</td>
</tr>
</tbody>
</table>

“Extreme Scaling of Production Visualization Software on Diverse Architectures”, H. Childs et al, IEEE CGA, 2010 May-June

Synthetic data allowed testing data sizes well beyond simulation requirements
Open source code and open development helped these tools grow vibrant user and developer communities

- These communities helped drive wide adoption in DOE and beyond
- BSD style (non-copyleft) licenses have helped enable commercialization
- These tools are also used as vehicles to deploy capabilities from visualization research to users
  - Helps address the R&D adoption “valley of death”
DOE’s visualization community is collaborating to create open source tools ready for Exascale simulation data

Addressing node-level parallelism

- VTK-m is an effort to provide a toolkit of visualization algorithms that leverage emerging node-level HPC architectures
- We are also exploring using VTK-m and DIY to share more distributed-memory infrastructure across projects

Addressing I/O gaps with in-situ

- There are several efforts focused on in-situ infrastructure and algorithms

  - [ALPINE](http://alpine.dsscale.org) (ParaView/VisIt)
  - [ParaView Catalyst](http://www.paraview.org/in-situ)
  - [VisIt LibSim](https://visit.llnl.gov)
  - [Sensei in situ](http://www.sensei-insitu.org)
  - [Ascent](https://github.com/Alpine-DAV/ascent)
Looking forward, I see two other high impact opportunities

Simplifying in-memory coupling

- Fluid in-memory exchange between tools opens up many possibilities

- However, in-memory integration is a more intimate vs file-based exchange and developing conventions for sharing data across the ecosystem is hard

Bridging analysis ecosystems

- Data analysis ecosystems are growing explosively, have huge mindshare and powerful capabilities

- There is tremendous opportunity for leverage, but to get traction we need to lower barriers to connect data between ecosystems

These present big data representation and software engineering challenges
Embracing open source software was a key part in establishing DOE’s leadership in HPC scientific visualization

Open source and open development helped:

- Build robust, widely used tools
- Scale user and developer communities
- Unlock funding beyond ASCI, including commercial endeavors
- The broader DOE visualization community collaborate and deploy research