LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-407624

Documentation of high impact visualizations and improvement plans for utilization of VisIt for reactor simulation

H. R.Childs, D. J. Bremer

October 8, 2008

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Documentation of high impact visualizations and improvement plans for utilization of VisIt for reactor simulation[a]

Global Nuclear Energy Partnership

[a] This is report LLNL-TR-407624

Prepared for
U.S. Department of Energy
Reactor Campaign, GNEP
Hank Childs
David Bremer
September 30, 2008
GNEP-REAC-PMO-MI-DV-2008-000173

Reviewed by:

Area Manager (WBS 4)

Andrew Siegel	Date

Campaign Director (WBS 3)

Robert Hill	Date

SUMMARY

The primary goal of this milestone was to enable the visualization and analysis needs of the campaign's simulation codes. This goal was well accomplished. We have extended the VisIt visualization and analysis tool to be suitable for the Nek, UNIC, SAS, and DIABLO code teams. This represented a significant development effort, primarily in terms of tuning the processing of the very large data sets produced by the Nek code. As a result of our development, and of the support we provided, these groups have been able to successfully accomplish their visualization and analysis activities using VisIt.

CONTENTS

ACRONYMS

REACTOR CAMPAIGN DOCUMENTATION OF HIGH IMPACT VISUALIZATIONS AND IMPROVEMENT PLANS FOR THE UTILIZATION OF VISIT FOR REACTOR SIMULATIONS

1. INTRODUCTION

Visualization is an important part of the simulation process. It allows stakeholders to explore simulations and discover phenomena, to confirm assumptions, and to convey findings to a larger audience. Further, visualization software is complex and is an active research area, especially in the area of visualization of very large data sets, such as those produced by the Reactor campaign's Nek code.

To meet the campaign's visualization and analysis needs, we chose to leverage the existing software tool, VisIt. VisIt is an open source, parallel visualization and analysis tool for interactively exploring scientific data. The tool represents approximately fifty man-years worth of effort, much of which was dedicated to techniques for processing large data and also to user interfaces. VisIt originated in the DOE's Advanced Simulation and Computing Initiative (ASCI) program, but is also actively developed by the Office of Science's Scientific Discovery through Advanced Computing (SciDAC) program, as well as by the at large open source community, including university partners.

Our work for this effort consisted of both customizing VisIt to meet Reactor campaign needs and of providing support for stakeholders in the Reactor campaign to ensure they were successful using the tool.

2. DEVELOPMENT ACTIVITIES

Development activities were of three types, each described in their own section:

1. tuning performance for "hero" simulations,
2. adding features to enable specific types of analyses, and
3. adding or modifying file format readers to enable stakeholders to use VisIt.

We comment that the tuning work was a critical component to the success we had with visualizing Nek simulations in the later part of this year and that this work represented the strong majority of our development effort.

2.1 Performance

The Nek code was identified as the code running the most "hero" simulations at this time and the one that required the most tuning for VisIt. This tuning required making many different kinds of changes to VisIt, each described in their own subsection.

2.1.1 Subset Inclusion Lattices

VisIt uses an advanced data structure for graphs, "Subset Inclusion Lattices," to represent subsetting relationships. One relationship represented with this data structure is how the simulation code decomposes the entire mesh into chunks for parallelization. Although VisIt has previously been used to visualize simulations with tens of billions of data points, it had never been faced with such a large number of chunks. Where the largest runs visualized to date by VisIt typically have less than 10,000 chunks, Nek simulations may have up to 360,000 chunks, with more in the future. VisIt handles each "chunk" as its own entity and the overhead for each one of these chunks was leading to noticeable delays. So we massively overhauled VisIt's "Subset Inclusion Lattice" data structure to improve performance for Nek. As a result of this tuning, overhead for handling chunks dropped from 25 seconds to 2 seconds. Further, a related bottleneck was identified that removed a 70 second delay altogether.

2.1.2 Better handling of domain abutments

Nek does not provide VisIt with information regarding how each of its data chunks abut. When determining which facets to render, this causes many internal facets to be incorrectly identified as external and thus to be rendered. To correct this, we implemented a module that determines abutment. This module allowed for 92% of the total facets to be (correctly) discarded, resulting in approximately a 12X increase in rendering performance. Implementation was fairly difficult, because it had to operate in a distributed memory, parallel setting.

2.1.3 I/O improvements

We performed an extensive performance study for visualizing Nek data. An important outcome of this study was that VisIt's style of seeking back and forth through a file was much slower than reading the file in a coherent fashion (beginning to end). An important platform for this work is the parallel linux cluster at Argonne named cosmea, which is used to visualize and analyze simulations run by the campaign. On this machine, the I/O seeking changes dropped the total time to read data dropped from 251 seconds to 63 seconds.[b]

2.1.4 Incorporating meta-data

We extended the Nek reader to provide meta-data that can optimize performance. When visualizing only a cross-section of the data set, everything not on the cross-section does not contribute to the final picture. The meta-data describes where each element lies in three-dimensional space. VisIt can then use this meta-data to read in only the elements that actually intersect the cross-section. This optimization caused performance to go from approximately 300 seconds to approximately 9s. (These numbers represent an aggregate over many timings, as I/O performance varied dramatically.)

2.2 New Features

We made the following changes to support analysis using VisIt:

1. Added support for the "Hexahedron-20" element type, which required modifying algorithms for volume rendering, slicing, isovoluming, contouring, and mesh quality calculations.

---

[b] Note that the target machine for visualizing this data, cosmea, has a poor filesystem that exacerbated this problem. However, this change still reflects the benefits seen by members of this campaign.

2. Added support for the non-traditional connectivity used by UNIC. In this case, each field is node-centered, but is discontinuous from cell-to-cell. This type of data was new to VisIt, and impacts common visualization algorithms, such as external face detection and normals calculations for lighting.
3. Created scripts that extracted slices between pins for comparing cross sections of turbulent flows between different simulation codes.

2.3 File format readers

We made the following file format reader changes.

1. Wrote a reader to support UNIC's HDF5 output
2. Fixed bugs in the Star-CD/CCMIO file format reader
3. Modified the Nek reader to support 2D, ascii, and parallel variants of the Nek format
4. Modified the SAS reader to support new types of data ("assembly type" and "id")

3. SUPPORT ACTIVITIES

We provided many support activities:

1. Ported VisIt to the parallel cosmea machine at Argonne
2. Ported VisIt to the octopus/octagon cluster at Argonne
3. Regular installs of VisIt on Argonne networks
4. Met face-to-face with customers to gather requirements and describe how to best use VisIt
5. Created several high end visualizations for members of the campaign (see 3.1)

3.1 Examples of Visualizations Using VisIt by Campaign Stakeholders

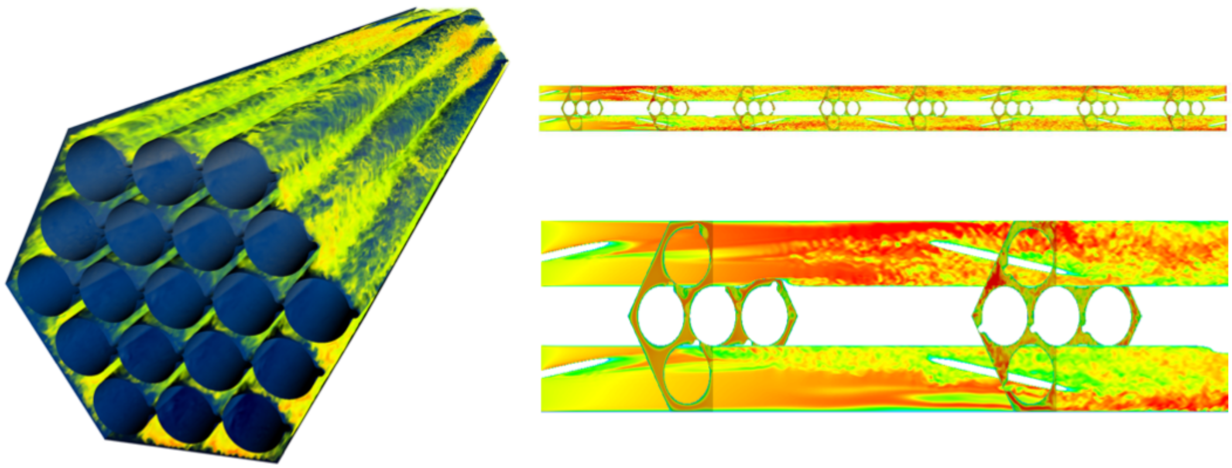

Figure 1: Turbulence in wire-wrapped subassemblies, visualized by axial velocity distributions. Simulation by Paul Fischer, ANL, using the Nek code, visualizations by David Bremer and Paul Fischer, using VisIt. Periodic boundary conditions applied over a single pitch, z in [0,H] are used for the 19-pin case (left) while inflow/outflow conditions at Z=0/3H are used for the 7-pin case (right). Transition to turbulence from a uniform inlet flow occurs at z < H/2 in the 7 pin case (lower right).

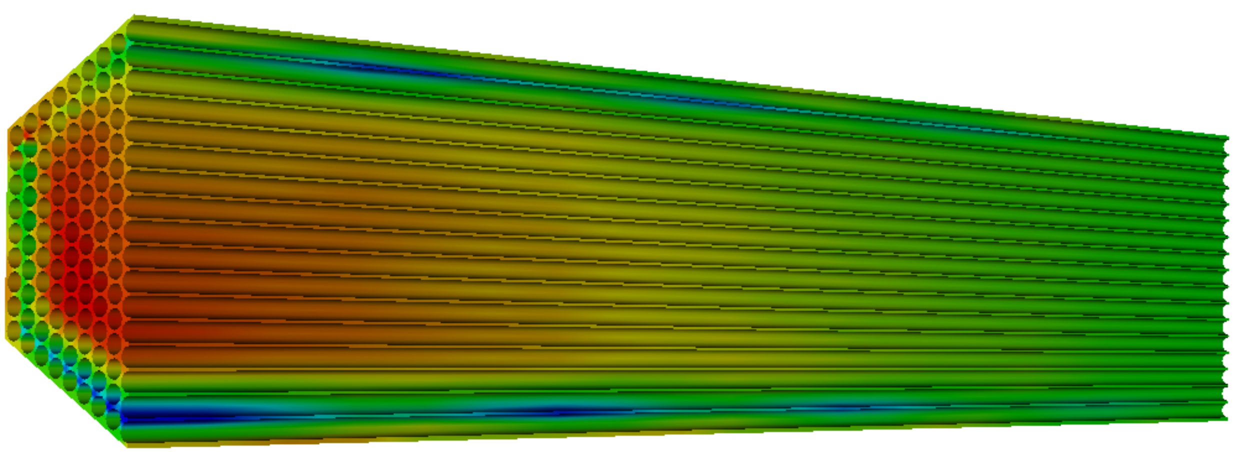

Figure 2: Comparison of differences between sub-channel and RANS coolant temperature distributions in a 217-pin fuel bundle. Simulation by Tom Fanning, ANL, using SAS. Visualization by Tom Fanning using VisIt, with support from our effort, including modifications made to VisIt for Tom.

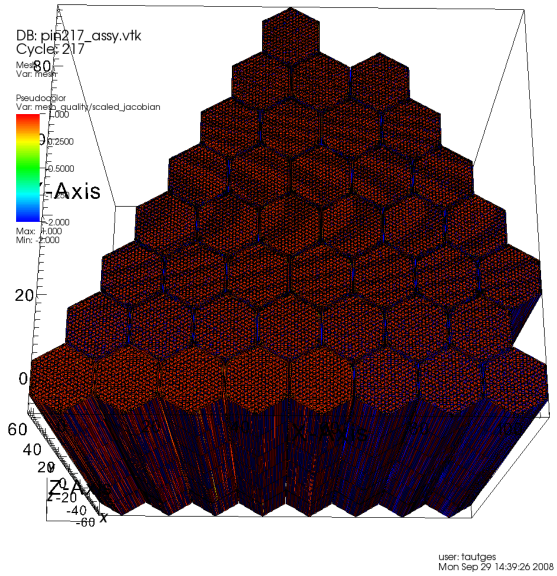

Figure 3: Meshing of a 1/6 core mesh using MOAB, visualized by VisIt. Image by Tim Tautges, ANL.

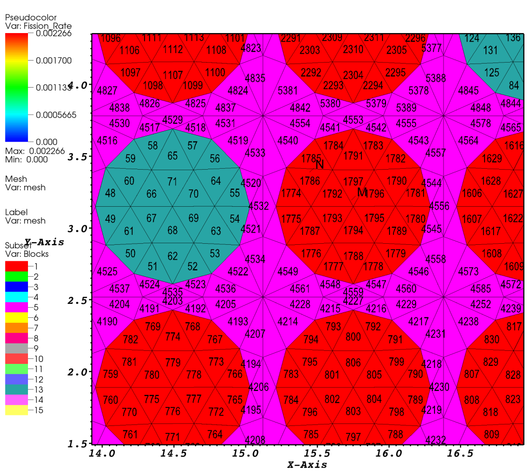

Figure 4: Visualization of UNIC data by VisIt. UNIC is an unstructured neutronics solver developed at ANL and designed to scale to large supercomputers. UNIC's data model required VisIt to be extended and also for a custom file format reader to be written. Image by Hank Childs.

4. RESULTS

The Nek, SAS, UNIC, and DIABLO codes were all able to successfully use VisIt for their visualization needs in the past year. All of the activities described above, both the development activities and the support activities, were a critical component to achieving this success. Also, it is important to note that because this effort contributed to an open source project, the development done, in particular the tuning work, is of benefit to a much larger group, including stakeholders from the ASC and SciDAC programs.

There is much more work to be performed for this effort. The code interoperability effort using MOAB will require additional subsetting capabilities from VisIt. UNIC is producing energy group data and VisIt does not provide adequate techniques for visualizing this data. Many efforts in the campaign are focused on performing comparisons and VisIt is well positioned to play a significant role in that activity. Of course, regular types of support activities (installs, helping users, making images and movies), are ongoing. Also, we note that we planned to begin on many of the development activities in FY08, but were unable to complete them given reduced funding (and manpower) required by GNEP midyear cuts.