in-situ Visualization

Jean M. Favre[1]

[1]CSCS, Swiss National Supercomputing Center, jfavre@cscs.ch

1 Introduction

The availability of high-performance computing resources enables the exploration of scientific phenomena and engineering design problems with numerical simulations of unprecedent quality and size. Scientific visualization contributes to converting the raw numerical output to mental images, yet it is more and more hindered by the overwhelming data I/O from disks. Similarly, numerical simulations often cannot archive all their results to disk files, because of the sheer sizes involved, and because I/O to disks is the slowest operation overall, imposing a huge bottleneck on the whole process. As we reach exascale computing, the conventional post-processing model *compute-store-analyze* can no longer cope with this data deluge. At that scale, it would take a visualization platform almost as powerful as the supercomputing running the simulation to read and analyse the data.

In-situ visualization is a process whereby visualization computations are done simultaneously with the simulation. It is attractive, because all relevant data about the simulation is available in memory; it can greatly contribute to the overall efficiency by limiting the data archived to disk files; it is a way of achieving run-time monitoring or even steering of a simulation. A couple of open-source solutions for data analysis tools enabling *in-situ* visualization are now emerging.

However, some refactoring of the simulation code is most likely required to enable live monitoring. In this talk, we wish to describe the generic parallel execution model proposed by VisIt[1], the leading *in-situ* visualization software. Simulation codes need to make room for a control interface to accept connections and receive commands from the remote visualization program and to provide a callback mechanism by which to return or share data. Beyond command and data transfers, both the simulation and the visualization must be able to continue their own processing independently. We will describe how the restructured execution models will enable this in a totally application-independent manner, for all common grid types.

References

[1] VisIt, *A Parallel Visualization Software*, https://wci.llnl.gov/codes/visit/home.html