Center for Interoperable Technologies for Advanced Petascale Simulations: Year 1 Progress Report for Subcontract B563994

Carl Ollivier-Gooch
Advanced Numerical Simulation Laboratory
The University of British Columbia

October 12, 2007

This report describes work undertaken as part of the SciDAC-2 Center for Interoperable Technologies for Advanced Petascale Simulations (ITAPS). UBC's role in this Center focuses on the infrastructure development side of things rather than on the applications side, and we have participated actively in the Center's progress in this area over the past year.

Interfaces

There have been three significant developments in the ITAPS interfaces in the past year.

1. The ITAPS team has made a philosophical shift in our approach to language interoperability. In the past, our interfaces have been designed to use the Scientific Interface Description Language (SIDL) and its Babel preprocessor to address language interoperability issues. However, we have found that this approach can lead to large overhead, especially for fine-grained access to data through the interfaces. To reduce overhead, we have switched to defining the interfaces primarily using a C applications programming interface (API), with SIDL/Babel as an additional, optional layer above this. Tim Tautges (now at Argonne National Laboratory) provided the initial C version of the API, including an implementation based on MOAB (the Mesh-Oriented datABase). I developed the second implementation of the C API, and provided significant feedback on the syntactic details of that API.

2. In response to requests from several applications, the ITAPS team has developed an interface to support non-linear mesh entity shapes. In the short term, this interface will be used specifically to support high-order finite element simulations for a SLAC application. The interface has been designed, however, to support a broad range of entity shape types and therefore ultimately a variety of different approaches to high-order accurate simulation. I was central in helping develop an approach to this problem that is general, efficient for common cases, and does not adversely affect applications that are not using high-order methods.
3. Much of the two most recent face-to-face meetings of the ITAPS team have been devoted to determining the requirements of applications that use parallel mesh services and developing a parallel data model compatible with those requirements. At this point, we have agreed on and prioritized a list of requirements and outlined a data model. Current work is aimed at defining an interface that will support those requirements. I have been actively involved in these discussions and am a member of the subgroup working on interface definition.

In another interface-related development at UBC over the past year, a summer student under my direction worked to improve our current testing framework for the iMesh interface to reduce or eliminate dependencies between functions to be tested. The current testing framework — which is based on testing the entire interface, and therefore is most useful for complete implementations — is not helpful for testing individual functions, because many of the tests rely on comparing information returned by different functions with related functionality. Removing these inter-function testing dependencies will be very helpful to programmers wishing to use iMesh-based services in existing applications. Such an application can provide a service access to its internal data by implementing the subset of iMesh functions the service actually uses, and, with the modular testing framework under development, test the correctness of their implementation before linking to the service, where testing would be less direct.

Component/Integrated Services

In this area, we have begun work with Lori Diachin (LLNL) to use Mesquite in combination with the iMesh-based mesh swapping tool developed at UBC. At this point, Lori has successfully used the two tools to improve the same mesh within the same program. We will continue to work closely with her and with the rest of the Mesquite team to ensure compatibility between the tools in terms of quality measure

definitions and the use of sets and tags to define sub-meshes that are candidates for improvement.

Other Developments

In two other ITAPS-related developments, a current MASc student is completing work on an anistropic mesh adaptation technique which will become the basis of an ITAPS mesh service in coming years. Also, I have recruited a new PhD student who will eventually implement much of the ITAPS parallel mesh interface and various parallel mesh modification services built on it. Both of these students have, thus far, been supported by Canadian government funding that enabled my participation in the SciDAC-1 predecessor to ITAPS. With that funding now expired, both of these students will be supported directly from ITAPS funding.