A Grid Enabled Version of SOAP for the Aerospace Cluster and CDC Communities

The Satellite Orbital Analysis Program (SOAP) is a widely recognized software and engineering analysis tool authored by Aerospace. Many users in multiple communities use SOAP. SOAP can be computationally demanding, and may require super-computing resources for timely execution. This is particularly critical when used in time-demanding operations, such as concurrent design iterations within the Aerospace Concept Design Center (CDC). However, the use of Aerospace grid computing resources is being hindered by increasing security requirements and the lack of automated job transport mechanisms between the clusters and corporate desktop standard computers. We envision the development of a capability that will allow users of SOAP to task the Fellowship cluster. SOAP includes computation-intensive features such as generation of parametric studies, contouring of global coverage statistics, and determination of orbital close approaches. In partnership with JPL, these features have been adapted for use with distributed computing through the use of Message Passing Interface (MPI). We intend to extend this work to support MS-Windows based clients and to automate the cluster job submission and retrieval process. The results of this project will enable Aerospace users to leverage supercomputing resources with an extensively used tool and will serve as a pathfinder for bringing parallel computing to other desktop client software.

SOAP is being extensively used at The Aerospace Corporation to support a broad spectrum of space programs. A recent application of the software that would benefit from parallel computing was a Defense Science Board study to compare alternative Global Positioning System (GPS) constellation architectures against various measures of performance such as Positional Dilution of Precision (PDOP). These metrics are subjected to a variety of environmental factors, such as constrained receiver line of sight due to the presence of terrain and urban canyons, along with considering the failure potential for one or more satellites.

Although specific modules within SOAP have been successfully adapted for use with MPI and parallel computing, they remain out of reach for most Aerospace users. The problem lies with a lack of mature MPI binding for the MS-Windows platform. In addition, cluster resources have become even more difficult to access due to increasing security requirements. Because of this, the Aerospace cluster can no longer be accessed by using Remote Procedure Call (RPC). It is therefore imperative that the corporation develop alternative methods for submitting jobs to the clusters in real time.

Project
We envision that the end product of this project will work as follows. A SOAP user will set up a problem suitable for batch execution. A panel in SOAP called "Cluster Computing Preference" will have been preconfigured to select the name of the cluster and the number of processors being requested. When the user selects "Submit", a Web browser is spawned and the user provides the cluster username and password. SOAP will then initiate a "Pack and Go" operation that will transport the scenario file and all of its dependencies to the cluster. Once the job is submitted, the user will be able to continue interacting with SOAP and other software. Exiting SOAP will have no effect on the
outcome. A master node on the cluster will launch the scenario and will use MPI to apportion tasks to the child nodes. All of the cluster nodes are "headless", meaning there is no display or user interface. Once the job is completed, the master node assembles the output into a SOAP "Data Table Object" and transmits it back along with an altered scenario file back to the client computer. The user is notified that the job is complete and can click a link to automatically launch SOAP with the completed output displayed in a Data Table Plot View.

Although the pertinent SOAP modules have already been enabled for MPI, the MPI-enabled (server) code base has yet to be adapted to the Aerospace cluster. The first task will be to re-host the server component (the headless node) on the Aerospace cluster. This entails a recompilation of SOAP for BSD UNIX, a new platform for the software. Although the ANSI C compilation should is straightforward, there are a number of libraries (including ASTROLIB) that will have to be either built or located. The next task will be developing and/or deploying the network framework. The process for submitting a job from a client version of SOAP to the grid has not established for the Aerospace cluster. The team presently favors a Web Portal approach, as it would use the ubiquitous browser interface, would not require any client-side middleware, and could be developed as an independent, standalone module. Once we arrive at an initial operating capability, the team can concentrate on stages of successive refinement to offer improved submission, interaction, feedback, and portability. A lessons-learned document will be produced and collaboration on how to extend the methodology to other software projects at Aerospace will be ongoing. Ultimately, we wish to move towards using well-established tools and protocols (such as the Globus Toolkit), though mature bindings for the MSWindows are not generally available for these as of this writing.

**Deliverables**
The final product will be a modified version of SOAP, which will allow the analyst to submit time-consuming operations directly to the Aerospace cluster for offline computation right from their desktop computers. Results will be retrieved and displayed by the modified version of SOAP, greatly reducing the amount of time needed to compute results for many program related studies. We envision a primary user of this capability will be the Aerospace Concept Design Center (CDC), as they could conceivably submit and retrieve MPI jobs within the same interactive session. The advantage of a desktop solution is that it can be developed, tested, and even perhaps trained offline, minimizing risk and the potential for any down time. Such a project will also serve as a pathfinder for the use of supercomputing resources by other MS-Windows based software tools at the corporation. Aerospace customers will be grid enabling legacy applications in the future, so the experience of the process is as important as the actual code.