

AUTOMATED CFD PARAMETER STUDIES ON DISTRIBUTED PARALLEL COMPUTERS

Stuart E. Rogers, Michael Aftosmis, and Shishir Pandya

NASA Ames Research Center
Moffett Field, California 94035

Edward Tejnil and Jasim Ahmad

Eloret
Moffett Field, California 94035

Introduction

As Computational Fluid Dynamics (CFD) technologies and software matures, and as computational-resource costs continue to drop, it becomes possible to use CFD methods to run large parameter studies. Computational requirements for inviscid Euler CFD solvers for a complete aerospace vehicle are on the order of 5 to 20 hours on a high-end PC or workstation, depending on the solver and the complexity of the vehicle. Computational requirements for viscous CFD solvers are typically 10 to 20 times that of an Euler solution. By taking advantage of parallel computing platforms, it becomes possible to reduce the wall-clock time per solution to less than an hour. If enough parallel computing platforms are available to a user, it becomes possible to perform a large parameter study. With this comes the possibility of using high-end CFD analysis in a number of unique ways, including trade studies and in building stability and control databases.

A number of difficulties arise when attempting to run a large parameter study. Often a user must spend time performing a number of mundane tasks for each CFD job. These include pre-processing input files, logging into the compute system and copying the input files there, executing and monitoring the job, post-processing, and archiving the solver output. This can become a full-time job, and rather tedious, when running more than a few jobs. The use of simple scripts to perform most of these tasks will help when running a few dozen jobs. But when one wants to run thousands of job, a more sophisticated processes is required. This is particularly true when the compute resources are spread out over a number of different heterogeneous compute platforms at more than one location.

NASA Information Power Grid (IPG) effort. This is currently a major focus under the NASA Computing, Information and Communication Technologies (CICT) Program, under the Computing, Networking, and Information Systems (CNIS) Project. The IPG provides software for security and user authentication, and and job submission.

The objective of the current work is to build a prototype software system which will automated the process of running CFD jobs on IPG resources. This system should remove the need for user monitoring and intervention of every single CFD job. It should enable the use of many different computers to populate a massive run matrix in the shortest time possible. Such a software system has been developed, and is known as the AeroDB script system.

The approach taken for the development of AeroDB was to build several discrete modules. These include a database, a job-launcher module, a run-manager module to monitor each individual job, and a web-based user portal for monitoring of the progress of the parameter study. The details of the design of AeroDB are presented in the following section. The following section provides the results of a parameter study which was performed using AeroDB for the analysis of a reusable launch vehicle (RLV). The paper concludes with a section on the lessons learned in this effort, and ideas for future work in this area.

AeroDB Design

The AeroDB scripts consisted of a database service on a MySql server, a Job Launcher, a Remote Execution script, a job submission script, and a web portal. A flowchart of the AeroDB design is shown in Fig. 1. The database was used as the communication hub for all the scripts and the user web interface; it was used to store all status information and job attributes. The Job Launcher script submitted jobs for execution using the globusrun command; it chose the appropriate compute resource for each job based on information from the Grid Common Services (GCS) Broker. The Remote Execution script was executed for each remote compute job; it pre-staged and pre-processed the input files, executed and monitored the flow solver, and post-processed and post-staged the output files, while reporting each step to the database. The job-submission scripts were used to enter information about new jobs into the database. The web portal provided an interface for the user to see the status of each job in the database, and provided a mechanism for re-running and restarting jobs. The AeroDB effort also included the development and use of a Run Manager library which was used by the CFD flow solvers to monitor convergence and remaining execution time. This provided a mechanism for the flow-solver to automatically run until the converged solution was obtained.

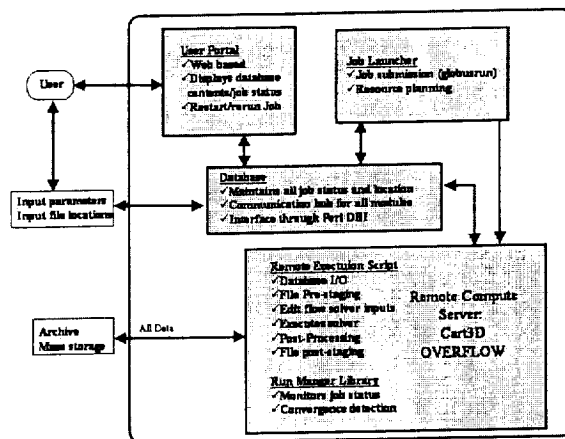


Fig. 1 AeroDB flowchart.

Results

The AeroDB scripts were used to execute and monitor cases for a large parameter study for a Liquid Glide-Back Booster (LGBB) vehicle. Jobs to execute both the Overflow and Cart3D flow solvers were launched starting at 9:00 AM on September 10th. Within 72 hours over 1000 Cart3D cases had completed and over 100 Overflow cases had completed. We continued to execute cases for seven days. At the end of this time, 2863 Cart3D cases had completed, and 211 Overflow cases had completed successfully. These cases utilized 13 different compute resources at four different locations. Many of these cases required multiple globusrun job submissions in order to obtain enough computing time. Table 1 shows the approximate number of globusrun job submissions sent to each compute resource, and the approximate number of CPU hours used on each host. Table 1 also lists the other compute resources used and their function. No special-priority queues were used on any of the computers.

Location	Host	Hardware	# of Jobs	CPU Hrs
ARC	chapman.nas.nasa.gov	SGI O3K	3489	25485
	lomax.nas.nasa.gov	SGI O3K	1074	15678
	steger.nas.nasa.gov	SGI O2K	477	8017
	hopper.nas.nasa.gov	SGI O2K	411	4702
	evelyn.nas.nasa.gov	SGI O2K	61	262
	simak.nas.nasa.gov	Sun Ultra	136	234
	sharp.as.nren.nasa.gov	SGI O2K	126	1014
GRC	aeroshark.as.nren.nasa.gov	Linux PC	70	976
	modi4.ncsa.uiuc.edu	SGI O2K	99	483
ISI	jupiter.isi.edu	SGI O2K	21	212
Total			5964	57065

Conclusion

The final version of this paper will include more details of the design of the AeroDB system, of the results and effectiveness of the current approach for automating a CFD parameter study, and of suggestions for future improvements.