For optimizing the fit of the model to the experimental data, CAMM will use the Dakota toolkit [1] which has many options for algorithms for optimization and sensitivity analysis. This software was written to be a flexible, extensible interface between simulation codes and analysis methods from least squares to genetic algorithms. To change the optimization algorithm or sensitivity analysis, the parameters in the Dakota text input file are simply changed. No changes need to be made to the simulation workflow. This is designed to run in parallel and request the fitness from as many simultaneous simulations as appropriate for the optimization method. For example, methods requiring numerical derivatives will request simultaneous small steps for each parameter and genetic algorithms will calculate the fitness of each child in parallel. The Dakota software keeps a binary restart file of all parameters evaluated which can be used to restart the optimization in case of workstation failure or software crash. Dakota also uses this data to never evaluate the same parameters twice. It just returns the previous fitness if the algorithm requests a repeated evaluation.

For the workflow that calculates the fitness of the simulation results, CAMM will use the Kepler software [2]. The workflow is developed using a graphical user interface, but the run-time engine can execute the workflow saved in an XML file from the command line or the graphical interface. This software has a parallel director and over 350 actors for determining file existence, reading files, doing Boolean switches, executing web services, and running simulations on remote machines. The software just needs one remote job ssh connection for the complete optimization and has the options for using a password, ssh key, or grid certificate.

The system we are using to test the optimization and workflow is an eight core front-end workstation that runs the Dakota and Kepler software and a 192 core cluster called Chadwick that runs the simulation software. Both Dakota and Kepler will be running simultaneously on the front-end and the Kepler workflow will copy results back from Chadwick to be used in the Dakota optimization.

The first simulation that has been successfully run using this test system is a simple three parameter optimization of test data where the simulation is remotely submitted to Chadwick using Kepler actors and the resulting fitness is written to a file that is copied back to the workstation. Presently we are using an ssh key for the connection between the workstation and Chadwick for our tests, but we have a grid certickificate ready for testing when the globus software is installed on Chadwick. The optimal values were found in 52 simulation calculations where most of the calculations were done to find numerical derivatives. Dakota writes the parameter files in parallel and then waits until a file containing the results for that function evaluation number returns. Kepler starts a simulation workflow when a parameter files exits. Multiple simulation codes are run simultaneously on Chadwick.

A second Kepler workflow has been written to read Dakota’s restart file and determine if a subset of the parameters has been previously evaluated. In that case, Kepler actors can retrieve simulation files from the working directory on the remote cluster and not have to redo the entire simulation workflow. For example, if the instrument resolution function changes but the simulation parameters do not, previous simulation data can be reused.

Future plans are to optimize the force constants for a NAMD simulation using Dakota and a Kepler workflow on our test system. With NAMD, the simulation will run in parallel using the cores of Chadwick. Then we will test using a grid certificate for communication between the machines and apply for a community grid certificate for SNS users. We also plan to use messaging for communication between the Kepler workflow and Dakota optimization instead of checking for file existence. When the test system is working well, we will move the simulations to Titan.

[1] Adams, B. M., Bohnhoff, W. J., Dalbey, K. R., Eddy, J. P., Eldred, M. S.,

Hough, P. D., Lefantzi, S., Swiler, L. P., and Vigil, D. M., 2011. DAKOTA: A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. Version 5.2 User’s Manual, Sandia Technical Report SAND2010-2183, updated Nov. 2011.

[2] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. Scientific workflow management and the kepler system. Concurrency and Computation: Practice & Experience, pp. 1039–1065, 2006