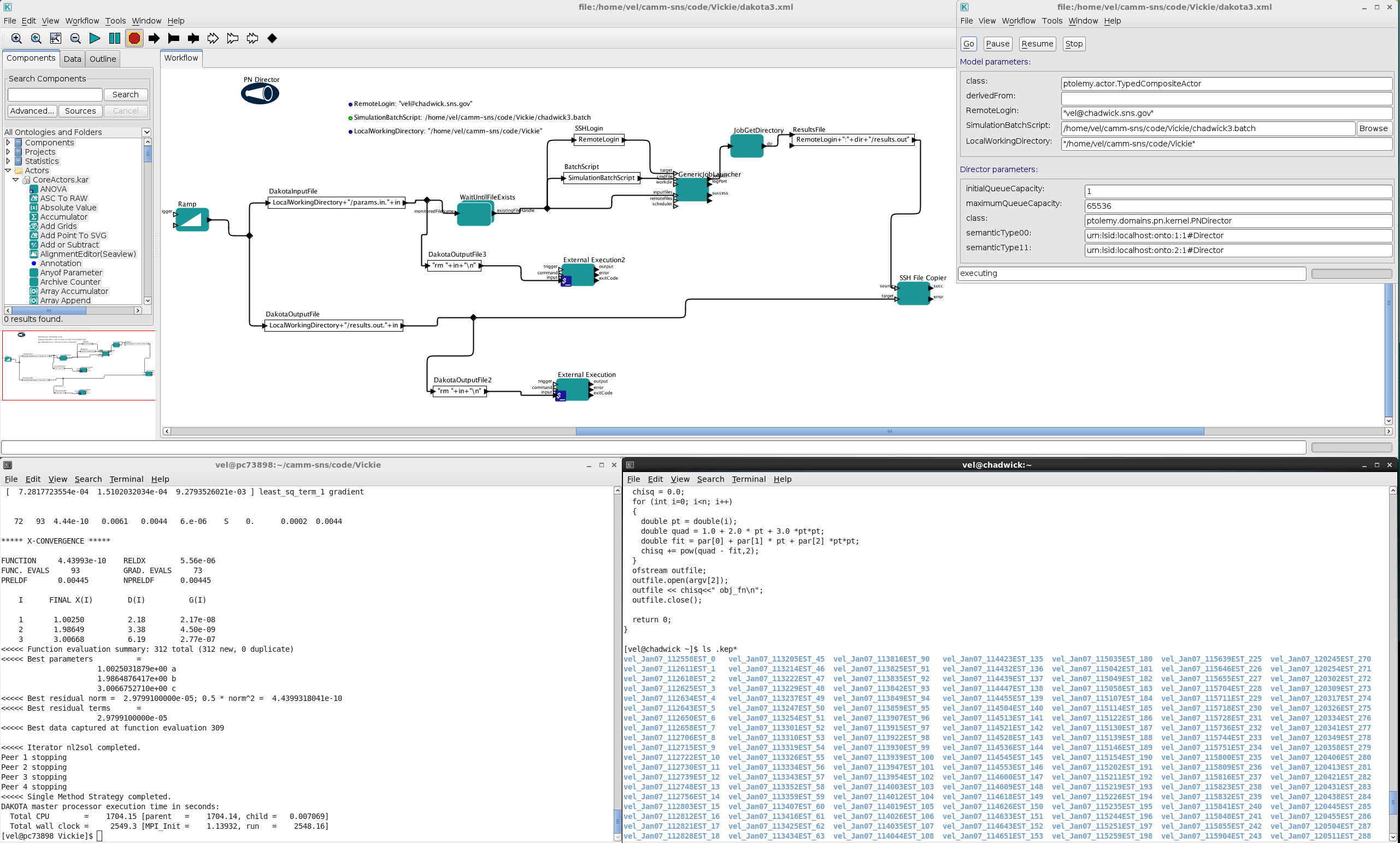
For optimizing the fit of the model to the experimental data, CAMM will use the Dakota toolkit [1] that has many options for optimization and sensitivity analysis. This software was written to be a flexible, extensible interface between simulation codes and analysis methods from least squares to genetic 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 optimization [Figure 1] that has been successfully run using this test system is a simple three parameter quadratic fit to artificial 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 certificate ready for testing when the Globus software is installed on Chadwick. Dakota writes the parameter files for concurrent evaluations and then waits until a file containing the fitness results returns. The optimal values are found using a least squares algorithm. Kepler’s simulation workflow checks for when a parameter files exists and creates a unique working directory on Chadwick for each evaluation. The simulation is run as a batch job on Chadwick and then the fitness file is copied from the working directory on Chadwick back to the workstation. Dakota uses these results to decide which parameters to test next. When the maximum number of iterations or the convergence tolerance is reached, Dakota stops the optimization.



*Figure 1: Top window shows the Kepler workflow running on the workstations with its GUI in the upper right. Lower left window shows Dakota at the end of the optimization on the workstation. Lower right window shows the function being optimized on Chadwick and all the working directories created for the simulation batch jobs.*

A second Kepler workflow has been written to read Dakota’s restart file and check 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, each 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 ActiveMQ 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, the Oak Ridge National Laboratory Cray XK7 high performance computer that is first on the Top 500 Supercomputer list.

[1] B. M. Adams, W. J. Bohnhoff, K. R. Dalbey, J. P. Eddy, M. S. Eldred, P. D. Hough, S. Lefantzi, L. P. Swiler, and D. M. Vigil, 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