

Efficient Execution of Computation Modules in a Model with Massive Data

Gary Kratkiewicz
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
kratkiewicz@bbn.com

Renu Kurien Bostwick
BBN Technologies
10 Moulton Street
Cambridge, MA 02138
renu@bbn.com

Geoffrey S. Knauth
BAE Systems
2361 J. Davis Hwy, Suite 706
Arlington, VA 22202
geoff@knauth.org

Extended Abstract

Models and simulations for analyzing difficult real-world problems often deal with massive amounts of data. The data problem is compounded when the analysis must be repeatedly run to perform what-if analyses. One such model is the Integrated Consumable Item Support (ICIS) Model developed for the U.S. Defense Logistics Agency (DLA). It models DLA's ability to satisfy future wartime requirements for parts, fuel, and food. ICIS uses a number of computation modules to project demands, model sourcing, and identify potential problem items for various commodities and military services. These modules are written in a variety of computer languages and must work together to generate an ICIS analysis. An analysis can include a million different types of items. ICIS utilizes 1000 input files in 200 different formats requiring many gigabytes of storage space, and generates a gigabyte of output data per analysis.

Drawing on the fact that similar analyses can contain much duplicated intermediate and output data, ICIS uses a novel data and computation management system called the Data Set Manager (DSM) to eliminate recomputation of identical intermediate results and duplication of data storage between analyses. This system controls the execution of modules, invoking only those modules that will generate intermediate results not previously computed. This system features data and module version traceability, storage of file and computation dependencies, a description-based file system, hashed identification of unique data files, an API interface for modules written in common modern programming languages, and a script-based means of wrapping legacy modules.

The version of the DSM used in ICIS was implemented using the Java and Jscheme languages. In certain common situations, the DSM reduces ICIS recomputation time from six hours to ten minutes and uses a fraction of the original storage space. This paper provides an overview of the ICIS model and describes the data and computation management system.

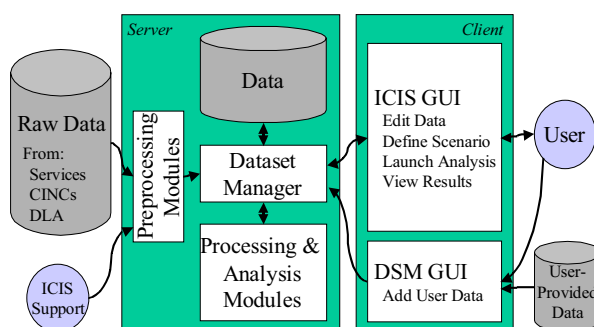


Figure 1. ICIS Architecture: Data storage and computations reside on a server. The user edits data, launches analyses, and views results from the user interface client code running in a Web browser. During an ICIS run the Data Set Manager (DSM) executes modules when necessary and reuses prior module results when possible to perform the analysis quickly and efficiently.

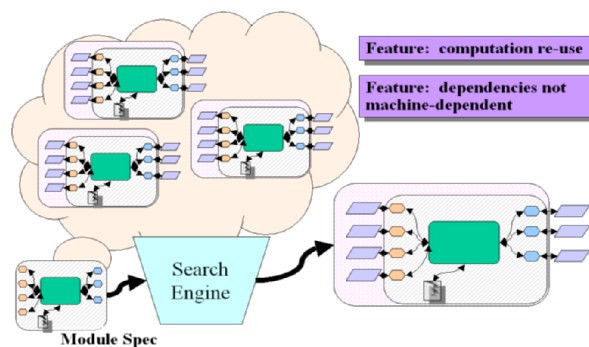


Figure 2. Computation Storage and Reuse: By storing information about what inputs went into which output calculations, the DSM allows for computation re-use. Computational trees can be built by running only those modules that need to be run and re-using results of modules that have been previously run.