Implementing Access to Data Distributed on Many Processors

NASA's Jet Propulsion Laboratory, Pasadena, California

A reference architecture is defined for an object-oriented implementation of domains, arrays, and distributions written in the programming language Chapel. This technology primarily addresses domains that contain arrays that have regular index sets with the low-level implementation details being beyond the scope of this discussion. The theoretical foundations are based upon the work "A Semantic Framework for Domains, Arrays, and Distributions in Chapel" by Hans Zima. What is defined is a complete set of object-oriented operators that allows one to perform data distributions for domain arrays involving regular arithmetic index sets. What is unique is that these operators allow for the arbitrary regions of the arrays to be fragmented and distributed across multiple processors with a single point of access giving the programmer the illusion that all the elements are collocated on a single processor. Today's massively parallel High Productivity Computing Systems (HPCS) are characterized by a modular structure, with a large number of processing and memory units connected by a high-speed network. Locality of access as well as load balancing are primary concerns in these systems that are typically used for high-performance scientific computation. Data distributions address these issues by providing a range of methods for spreading large data sets across the components of a system. Over the past two decades, many languages, systems, tools, and libraries have been developed for the support of distributions. Since the performance of data parallel applications is directly influenced by the distribution strategy, users often resort to low-level programming models that allow fine-tuning of the distribution aspects affecting performance, but, at the same time, are tedious and error-prone. This technology presents a reusable design of a data-distribution framework for data parallel high-performance applications. Distributions are a means to express locality in systems composed of large numbers of processor and memory components connected by a network. Since distributions have a great effect on the performance of applications, it is important that the distribution strategy is flexible, so its behavior can change depending on the needs of the application. At the same time, high productivity concerns require that the user be shielded from error-prone, tedious details such as communication and synchronization.

This program was written by Mark James of Caltech for **NASA's Jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*

This software is available for commercial licensing. Please contact Karina Edmonds of the California Institute of Technology at (626) 395-2322. Refer to NPO-42506.