
Ben Bergen
**Task-Graph and Functional Programming Models: The
New Paradigm**

P O Box 1663
MS B287
Los Alamos
NM 87545
`bergen@lanl.gov`

The Message Passing Interface (MPI) is an example of a distributed-memory communication model that has served us well through the CISC processor era. However, because of MPI's low-level interface, which requires the user to manage raw memory buffers, and its bulk-synchronous communication model, MPI will have great difficulty in scaling to exascale systems and beyond. Additionally, the MPI model cannot be easily extended to include the fault tolerance and resilience features that will be required to run at scale on modern computing architectures. A new approach is needed.

A trend that is gaining momentum in the computer and computational science communities is the use of data-driven models that employ task-graph runtimes to map data and tasks onto a distributed system. This approach offers a higher level of abstraction that frees the user from explicitly knowing the layout and location of the data regions accessed by a task. Task and data dependencies can be represented as a directed acyclic graph (DAG), allowing heuristics and auto-tuning strategies to be applied at runtime to optimize the execution of the DAG. This approach shows great promise. However, it will require a new understanding of how our existing numerical methods can be mapped into this model. At the same time, there is an opportunity for us to influence the features and requirements for implementations of task-graph models.

Although there are several efforts underway to develop useful task-graph implementations, the Legion project at Stanford University is a clear front-runner. This presentation is part one of a two-part series designed to introduce some of the Legion concepts to the audience. Here we discuss high-level task-graph concepts and provide an example of how a geometric multigrid algorithm might be implemented using a task-graph runtime.