TACC Technical Report IMP-07

Associative messsaging in the Integrative Model

Victor Eijkhout*

February 22, 2016

* eijkhout@tacc.utexas.edu, Texas Advanced Computing Center, The University of Texas at Austin

Abstract

IMP messages carry a higher semantic content than in other parallel programming systems. Rather than being arbitrary data, they are identical data on the sender and receiver; the transfer is induced by the fact that two distributions assign different ownership to the data on sender and receiver. This note makes that notion precise, and it discusses practical ramifications.

The following IMP reports are available or under construction:

IMP-00 The IMP Elevator Pitch
IMP-01 IMP Distribution Theory
IMP-02 The deep theory of the Integrative Model
IMP-03 The type system of the Integrative Model
IMP-04 Task execution in the Integrative Model
IMP-05 Processors in the Integrative Model
IMP-06 Definition of a 'communication avoiding' compiler in the Integrative Model
IMP-07 Associative messsaging in the Integrative Model (under construction)
IMP-08 Resilience in the Integrative Model (under construction)
IMP-09 Tree codes in the Integrative Model
IMP-10 Thoughts on models for parallelism
IMP-11 A gentle introduction to the Integrative Model for Parallelism
IMP-12 K-means clustering in the Integrative Model
IMP-13 Sparse Operations in the Integrative Model for Parallelism
IMP-14 1.5D All-pairs Methods in the Integrative Model for Parallelism (under construction)
IMP-15 Collectives in the Integrative Model for Parallelism
IMP-16 Processor-local code generation (under construction)
IMP-17 The CG method in the Integrative Model for Parallelism (under construction)
IMP-18 A tutorial introduction to IMP software (under construction)
IMP-19 Report on NSF EAGER 1451204.
IMP-20 A mathematical formalization of data parallel operations
IMP-21 Adaptive mesh refinement (under construction)
IMP-22 Implementing LULESH in IMP (under construction)

1 Discussion

In systems such MPI, a message is an amount of data moved from sender to receiver. In the Integrative Model for Parallelism (IMP), a message from processor q to p stems from two distributions α, β of the same object, such that $\alpha(q) \cap \beta(p) \neq \emptyset$. In other words, a message describes an index set $\alpha(q) \cap \beta(p)$, present on q, and required on p.

If the data with the α distribution was produced by kernel k, and the β data is the input for a kernel k', then the message is uniquely identified by the task identifier $\langle k, q \rangle$. Now we can assign the tasks $\langle k, q \rangle$ and $\langle k', p \rangle$ to arbitrary MPI processes, and instead of them waiting for a message from an identified MPI process, they wait for a message from any processor, with the correct tag attached.

This strategy has at least two interesting implications:

1. We can migrate tasks without any change to communication related data structures.
   - Trivially, a sending process still knows where to send its messages; but
   - a receiving process need not know that the sender has moved, since it receives by tag, not by sender.
2. If we add the receiver to the tag, this mechanism may make it easy to oversubscribe a processor with MPI tasks. In effect, the processor will have a task manager that listens for arbitrary messages, and activates the task for which a message has come in.
3. There is even a possibility of introducing resilience this way: if we redundantly duplicate tasks, receiving tasks can have multiple ways of satisfying an outstanding receive request. The request will be fulfilled based on its tag, not on who actually sends it.