TACC Technical Report IMP-08

Resilience in the Integrative Model

Victor Eijkhout*

February 22, 2016

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

Abstract

As supercomputers grow to the exascale, resilience – the ability of a computation to withstand hardware failure – is becoming a serious issue. We outline a solution based on redundant computation.

1 Sketch

Hardware can fail in multiple ways, from messages arriving out of order, via memory corruption through cosmic rays, to processors dying. Here we consider the simple, and most disastrous, case of a processor completely dropping out of the computation. We assume that the failure develops in the middle of a computation, rather than during communication. This assumption is warranted if the network fabric or the communication library has some guarantee that a message is delivered in total or not at all.

We reason through our solution in the following steps.

1. Our solution is to duplicate all computation. This carries an obvious factor two overhead cost, which we will assume is justifiable.
2. Firstly we note that redundant computation is easily modeled in Integrative Model for Parallelism (IMP) through the use of non-disjoint distributions.
3. Redundant computation also brings with it a new complication: every message in the system is also duplicated. This means that a receiving process now has two possibilities to fulfill its outstanding receive request. IMP can accomodate this without further modifications by using tagged sends; see [2].
4. If every task is redundantly executed twice, every message is duplicated, but there will be two receivers.

   Uh Oh. How do you pair them up? Wildcard receives are easy. Wildcard sends not.
5. The flip side of the tagged send notion is that a process can now have an unmatched outstanding send, since the receive operation for the data it is sending may have been satisfied by the duplicate message. This is certainly the case if a processor has dropped out.

2 Failure detection

The concept of *containment domain* was proposed in [1]. This requires several user actions, including writing a test function to decide whether the code in the containment domain has failed. In [3] we showed how the IMP model has the possibility of detecting failed tasks automatically.

References

[1] Jinsuk Chung, Ikhwan Lee, Michael Sullivan, Jee Ho Ryoo, Dong Wan Kim, Doe Hyun Yoon, Larry Kaplan, and Mattan Erez. Containment domains: A scalable, efficient, and flexible resilience scheme for exascale systems. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, pages 58:1–58:11, Los Alamitos, CA, USA, 2012. IEEE Computer Society Press.

[2] Victor Eijkhout. Associative messsaging in the integrative model (under construction). Technical Report IMP-07, Integrative Programming Lab, Texas Advanced Computing Center, The University of Texas at Austin, 2014.

[3] Victor Eijkhout. Task execution in the integrative model. Technical Report IMP-04, Integrative Programming Lab, Texas Advanced Computing Center, The University of Texas at Austin, 2014.