An MPI implementation and NSPorts implementation have similar mathematics to represent computation conducted in this instance. The MPI calls and NSPDO invocations both serve as transition functions. In the case of temporary objects, both mechanisms incur a cost of instansiating a set of listening "objects". NSPDO has one advantage for long term use. Namely, the listening objects can be left active. As such, the libraries can be invoked by many processes (including processes that had nothing to do with the starting of the listening objects). In the case of residual listening NSPDO, the cost of use includes discovery of the NSPDO and the proxy call. In case of MPI and non-residual NSPDO, the cost includes discovery of the proxy, proxy call, and instansiation of the objects.