William Josephson

11 March, 2004

In *A Case for NOW*, Anderson, Culler, and Patterson describe the basic architecture and motivation for Networks of Workstations. The author's underlying observation is that COTS hardware is not only cheaper than the special purpose hardware found in supercomputers and mainframes, but it also enjoys a better price-performance ratio. This advantage, together with commonly recognized technology trends such as Moore's law for microprocessors, slower growth in the size of DRAM and disk, the growing separation between processor speed and I/O throughput, then recent improvements in networking technology and the advantage of mass production put workstations in a unique position to challenge high-performance computers in a variety of tasks. For instance, fast networks make the aggregate DRAM available in a NOW attractive as a disk cache since the latency to access remote RAM is competitive with accessing local disk, largely because disk *capacity* and not performance is improving. Despite these technological reasons in favor of NOW, the authors claim that the challenge is to make NOW a win for *all* users: their architecture does not call for dedicated clusters as are now common, but rather a form of cycle-stealing on idle workstations throughout an administrative domain. This means that implementors must provide not only low-overhead, low-latency networking primitives, but also a method for sharing idle resources in the network in a manner that doesn't penalize normal workstation users while still providing good performance for NOW users at all times. The authors propose to do this using a protected virtual OS on commodity hardware and operating systems using software-fault isolation. They then limit the impact of swapping out a workstation user's context by controlling the selection of ''victim'' hosts and the frequency with which a host is chosen. Despite the lack of details about GLUnix, the software layer built on commodity operating systems, or about xFS, the distributed filesystem for NOW, the paper does present a motivation for NOW that is rooted in technological trends that have subsequently been borne out by the market place. Perhaps workstations, now displaced by PCs, are so cheap that even the cycle-stealing features of GLUnix layer are not worth the effort: dedicate PCs are easier to manage and more readily accepted by users.

Wide-Area Computing by Grimshaw *et al.* describes one approach to extending NOW-like ideas to the wide-area. Legion, the system they describe is essentially a set of CORBA-like objects for supporting grid-computing, although they don't use the term. Although they couch the discussion in terms of a wide-area distributed operating system, what they describe appears to be more a system for providing a consistent and secure interface to a limited number of kinds of distributed resources (there is insufficient technical detail to say much about their operating system analogy). Consistent, secure access to remote computing resources is an attractive application, but the authors do not discuss in detail what kinds of problems are best suited to their system. Some problems are embarrassingly parallelizeable and are almost entirely compute-bound, operating on small data sets (*e.g.* SETI at Home). The number of problems that fit in this category seems small, however, and wide-area distribution seems less useful when large amounts of data must be transported. Thus, although the authors do not say so explicitly, it would seem that Legion-like grid-computing systems are best suited to those scenarios where it makes sense to move computations closer to the data rather than vice-versa.