# A New, Architectural Paradigm for High-performance Computing

**Published** Mar 1, 2001

**David A. Bader**

## Abstract

At first thought, one may not realize the need for a special issue of Parallel and Distributed Computing Practices focused on Cluster Computing when our community has already invested years of research and spent a wealth of resources on traditional Big Iron supercomputers, for instance, the SGI/Cray Origin and IBM SP. Academia, industry, and national laboratories are still using, and for the foreseeable future, will continue to use, supercomputers to solve both grand-challenge scale and high-throughput applications. Then why the interest and new popularity of research into clusters for high-performance parallel and distributed applications? Clusters incorporate the tremendous successes that are attributable to our parallel and distributed processing community.

Over the next decade, clusters will span the entire range of high-performance computing platforms. Instead of being restricted solely to supercomputers for achieving high-performance computing, the spectrum of available platforms will spread out to Beowulf-class clusters, to superclusters, and on to supercomputers. Moving along this continuum represents increasing computational resources, which in turn allows the capability for solving larger classes of applications or instances of problems. Compute-bound applications on workstations often may take advantage of the extra horsepower provided by symmetric multiprocessors. Applications with even larger computational requirements and a high-degree of concurrency may use Beowulf clusters, that is, clusters that are home-built from mass-market, commercial off-the-shelf workstations and networks, for instance, PC's interconnected by Fast Ethernet. A supercluster may contain commodity SMP nodes interconnected with scalable, low-latency, high-bandwidth networks, and is integrated by a system vendor rather than the computational scientist. Applications with greater complexity, for example, grand challenges from the sciences, demand scalable systems with high-performance, interconnects, perhaps coupled with terascale I/O subsystems, and thus, are appropriate for these superclusters. Finally, supercomputers, the largest of the clusters, may provide the ultimate platform for the highest-fidelity models and simulations and data-intensive applications. At each step up this ladder of cluster architectures, the system cost increases approximately by an order of magnitude, but having the high-performance capability to solve the problem of interest may require an entry point at any of these levels, the Beowulf, or supercluster, or even supercomputer platform.

Cluster computing with Beowulf- or superclusters, thus, may be one or two orders of magnitude less expensive than a supercomputer, and provide a cost-effective solution and capability that was not available on a workstation. But this cost comparison measures just the basic system price. What about the applications? As recently as a few years ago, the lack of a canonical high-performance architectural paradigm meant that migrating to a new computer system, even from the same vendor, typically required redesigning applications to efficiently use the high-performance system. Thus, the application must be redesigned with parallel algorithms and libraries that are optimized for each high-performance system. This process of porting, redesigning, optimizing, debugging, and analyzing, the application may in fact be prohibitively expensive, and certainly frustrating when the process is completed just in time to greet the next generation system! That was then, this is now. With the successful standardization of system-level interfaces, many of these issues no longer hinder our expedited use of these new, high-performance clusters.

The message-passing interface (MPI) is a clear example of a standard that has contributed significantly to the success of cluster computing. Supercomputer vendors working together with the high-performance computing community embarked on a process to standardize the user-level interface and behavior for passing messages between concurrent tasks. Now, using ordinary sequential compilers and vendor-supplied or freely-available implementations of this message-passing interface, applications can be implemented efficiently on one supercomputer and ported with little or no effort to other parallel machines. The message-passing paradigm, thus, provides a reasonably good canonical high-performance model and methodology for implementing high-performance applications. An algorithm that exhibits efficient execution and scaling on one machine then is likely to perform well on another with little or no additional programming effort. Using this cluster computing paradigm, algorithms typically are optimized by minimizing the number of remote memory accesses (reducing the amount of messages passed), and carefully organizing the data placement to increase memory locality. Thus, high-performance algorithms and applications that have been optimized for supercomputers, using the same cluster paradigm, may be appropriate for Beowulf and superclusters. Cluster computing, realizing the importance of a standard high-performance programming environment that applies equally well to both ends of the architectural spectrum, highly leverages these and other successes from parallel and distributed processing research and development.

This is in an exciting, new era for high-performance computing where the canonical architectural paradigm stretches from home-built Beowulf clusters, to superclusters, and on to supercomputers. But cluster computing faces some challenges ahead of it. Continued research is required for designing 1) scalable, high-performance algorithms, applications, and methodologies, that can tolerate the communication overheads, and adapt to hybrid, SMP, and heterogeneous, nodes; 2) high-performance computer architectures, scalable interconnection networks, and I/O subsystems; and 3) scalable systems software, compilers, and monitoring tools. Cluster computing has come of age, and it is high time for the parallel and distributed processing community to lay the groundwork for our own legacy of success stories.

David A. Bader
University of New Mexico

Issue

Vol 2 No 2 (1999) (https://www.scpe.org/index.php/scpe/issue/view/35)

Section

Editorial

Current Issue

Vol 20 No 2 (2019) (/index.php/scpe/issue/current)

(https://www.scpe.org/index.php/scpe/gateway/plugin/WebFeedGatewayPlugin/atom)

(https://www.scpe.org/index.php/scpe/gateway/plugin/WebFeedGatewayPlugin/rss2)

(https://www.scpe.org/index.php/scpe/gateway/plugin/WebFeedGatewayPlugin/rss)

Information

For Authors (https://www.scpe.org/index.php/scpe/information/authors)

For Librarians (https://www.scpe.org/index.php/scpe/information/librarians)

ISSN

**1895-1767**

Social media

(https://www.facebook.com/groups/251002403356/)

**Published by:**


(https://www.uvt.ro/)