USGS
science for a changing world

Beowulfs' Place in the Computational Infrastructure for Geographic Modeling

By David I. Donato, U.S. Geological Survey (USGS)

Choosing the best computational platform for a demanding application in geographic modeling means finding the best balance among costs, throughput, turnaround time, and security. For applications which require high security, low unit computational costs, and rapid turnaround, the best platform may be a *Beowulf — an interconnected cluster of ordinary serial computers*.

1. Platforms for scientific computation

Although the numerous configurations of hardware for high-performance computing have so far resisted any convincing taxonomy, the following six computational platforms are representative of the variety of options:

Parallel platforms:

- Computational Grids
- Virtual machines
- Beowulfs (and other clusters)
- Loosely coupled systems

Serial platforms:

- Serial supercomputers
- Workstations

The first four of these are parallel platforms and the last two are serial. Parallelization in both hardware and software has become increasingly important as growth in serial-processor throughput has begun to plateau, though the computational speedups achievable through parallelization are limited, as shown in **figure 1**.

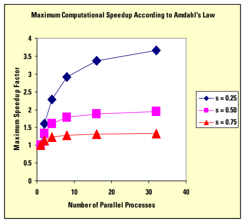

Figure 1. Amdahl's Law predicts that the maximum speedup achievable through parallelization is constrained by the inherently serial component of a computation:

$$S(p) = \frac{1}{s + \frac{1-s}{p}},$$

where $S(p)$ is the maximum speedup, s is the serial portion of a computation, and p is the number of parallel processes.

A Note on Computational Costs and Platform Choice

There is surprisingly little published information addressing the practical matter of choosing a computational platform to reduce computational costs. This may be due in part to the reluctance of computer professionals to relinquish control over these choices by involving managers; however, it may also be due to widespread management perceptions that computation and modeling are not among their respective organizations' core competencies and therefore require little management attention and no specific capital budgets.

2. How geographic modeling differs from general scientific computation

While general scientific computation tends to be processor bound and requires only modest inputs and produces only modest amounts of output, geographic modeling (like many other applications in geoinformatics) often takes in and puts out voluminous digital geographic data files, digital map files, and remote-sensing image files. Arguably, geographic modeling is *driven* by the availability of large geographic datasets and digital-image collections.

Data-driven computation (such as geographic modeling) benefits from placement on platforms such as clusters which provide fast and inexpensive data access. **Proximity of data to the processor**—a term coined here to mean structuring and positioning data so as to use the fastest kinds of machine memory—is essential to high-performance, data-driven computation. **Figure 2** quantifies proximity of data to processor for various major types of machine memory.

3. A major trend in scientific computation

Whether market mechanisms for capturing the value of currently idle computational capacity will develop is an open question, but this much is certain:

1. Scientific computation is a niche — not a mass — market. The bulk of the requirements of scientific computation will be met indirectly by improvements aimed at the business, personal, and entertainment markets.
2. For the remainder of this decade, gains in computational throughput will derive more from better-crafted software and parallelization of hardware and software than from improvements in serial processor hardware (though multicore, 64-bit processors with larger cache Random Access Memories (RAMs) will contribute significantly to increased throughput).

Comparisons of Features of Alternative Platforms

Platform Category	Description and Characteristics
Computational Grid	Provides computation on demand from a large pool of computational resources. Largely experimental; fluctuating standards; unproven economic viability; high network overhead; major security issues; and uncertain costs.
Virtual Machine	Uses spare cycles from machines on a local network. Low total cost but relatively slow turnaround; high network overhead; and security issues.
Beowulf (Cluster)	Uses idle resources in dedicated setting. Low unit cost, fast turnaround, and good security, but significant setup costs.
Loosely Coupled Systems	Works well when cooperation among systems is good. Acceptable performance when division of work keeps communications overhead to a minimum. Costs vary and there are security issues.
Serial Supercomputer	Provides expensive and rationed service. Available programming languages are limited and there are long waits in job queues.
Workstation	Requires relatively high initial capital outlay, but has relatively low administrative and unit costs. Good security and low network overhead, but comparatively long turnaround time for parallelizable computations.

4. Choosing a computational platform for an application

Choosing a computational platform for a particular application involves tradeoffs, notably between cost and performance. As with most real-world problems, there is usually no single, unequivocal best choice, but all choices can be ranked on various factors and a rational choice can be made based on application-specific requirements.

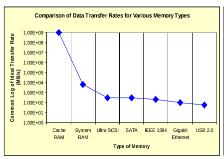

Figure 2. Data transfer rates are key determinants of computational turnaround time and throughput. Notice the difference of **five orders of magnitude** in transfer rates between cache RAM and system RAM and one order of magnitude between system RAM and high-performance magnetic disk drives. Shifting data access to the left in this Figure improves proximity of data to the processor.

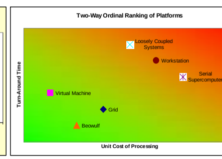

Figure 3. In this two-factor diagram, choices in the lower left (green) area are generally superior to choices in the upper right (red) because of shorter turnaround time and lower unit costs. There is no single, optimal choice for all applications. The ordinal ranks shown here were determined informally and are subject to debate.

5. Beowulfs in the infrastructure for scientific computation

A Beowulf — a cluster of new or surplus commodity computers — provides supercomputer throughput with low capital outlay and moderate setup costs for some classes of significantly parallel or parallelizable computations. The controlled environment of a Beowulf allows for security, short turnaround times, and predictability of throughput. By virtue of these features, Beowulfs have earned a place in the infrastructure for scientific computation in general and for geographic modeling in particular.

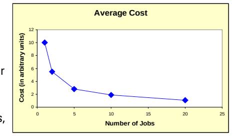

Figure 4. Hypothetical costs in this figure illustrate the principle that average cost quickly converges to marginal cost as available computational capacity is put into use. Apportioning fixed setup costs over as few as 10 or 20 jobs brings the per-job cost down to a competitive level.

Beowulfs are especially well suited to single-program/multiple-data (SPMD) parallel applications in geographic modeling because a cluster (in which each processor has its own dedicated memory resources) often provides greater proximity of data to the processor than the alternative parallel-computing platforms.

Windows®, Linux, and Beowulfs

The Windows ® and Linux operating systems in the 1990s jointly were enabling technologies for Beowulfs. Windows® created a pool of serviceable computers no longer suitable for desktop use because of a series of hardware-bound software upgrades. Linux provided an efficient operating environment suitable for low-cost application development and computation using former Windows ® desktop PCs.

6. Lessons from the USGS Eastern Geographic Science Center Beowulf

1. The economic value of a cluster can be multiplied by enabling remote access via a communications network, but the additional costs of security-related administration also multiplies cluster setup and administrative costs.
2. Operating-system upgrades should not be made more often than every six months nor less often than every 18 months.
3. Running a Beowulf requires someone willing and able to make hardware repairs on a monthly basis.
4. Operating a Beowulf is best undertaken by someone with a strong interest and a clear vision of the possible.

Empirical data from various sources support these assertions.

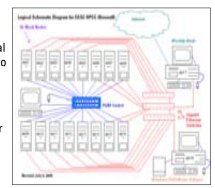

Figure 5. The USGS Eastern Geographic Science Center (EGSC) High-Performance Computing Cluster (HPCC) is in the "guarded Beowulf" configuration which balances access and security. An operational Web site provides a Web-based interface to interactive cluster applications.

The Future of Computing Technology

Computational Hardware

Problems of heat dissipation and current leakage have significantly reduced the pace of increase in processor speeds due to increases in clock rates. Evolutionary improvements in computational hardware will, however, continue through the foreseeable future. These advances will probably consist of the following:

- Increased use of multicore processor chips;
- Incremental improvements in internal processor optimizations;
- Larger on-chip and on-die cache RAMs;
- Use of 64-bit processors (providing faster integer arithmetic and larger address space);
- Faster front-side memory buses and faster Dynamic Random Access Memory (DRAM);
- Incremental increases in magnetic storage density and disk speed; and
- Improvements in data density and data-access rates for optical media.

SOA, RPC, and Microcharge Infrastructure

Despite extensive work on developing a commercial infrastructure for distributed software components*, there is still no effective infrastructure supporting the automatic microcharges (for a fraction of a cent or a few cents) needed in order to enable commercial software-level component services. Without such an infrastructure, commercial services from computational grids may prove economically infeasible, particularly if revenue derives only from unmetered, level-of-access charges. Service Oriented Architecture (SOA) and Remote Procedure Call (RPC) capabilities, which have potential applications in geographic modeling, are suitable for use with Beowulfs and loosely coupled systems; the suitability of SOA and RPC applications for computational grids, however, will remain unproven as long as grid standards remain in flux.

* Software-component technologies include the following: Common Object Request Broker Architecture (CORBA); Enterprise Java Beans (EJB); Distributed Component Object Model (DCOM); and Simple Object Access Protocol (SOAP).

Needed: A Simulator for Parallel Platforms and Applications

Design of parallel algorithms is challenging, to say the least. Two of the many reasons are the following:

- data structures, software components, and other logical design features often do not map to the physical features of hardware, and
- parallel-programming extensions to general-purpose, high-level, procedural programming languages impose alternative models of parallelism that significantly constrain parallel-software developers as they attempt to realize algorithms in working software.

A general-purpose parallel-software simulator — capable of conveniently modeling software performance for any of a variety of parallel-hardware configurations and algorithm elements — is needed for comparing alternative designs. If detailed, processor-level prediction is not required, a relatively simple discrete-time systems simulator (with queues, stores, servers, and transactions specialized for software design) would satisfy this need.

This poster was prepared during April and May, 2006, for *Geoinformatics 2006*, May 10-12, Reston, Virginia. It was revised during June, 2006, for clarity and readability.