Is There a Role for High-Performance Computing in GIS?

Marc P. Armstrong

The conventional wisdom held by many GIS users and researchers is that high–performance computing (HPC) environments are costly and difficult to use. However, improvements in hardware and software have changed the complexion of HPC during the past several years. Researchers and analysts who routinely use GIS and spatial analysis software can exploit these changes, and thus improve their ability to address computationally complex urban and regional problems. In this paper, I provide an overview of some trends in high-performance computing that have the potential to significantly impact GIS during the next few years. To realize this potential, GIS vendors and the GIS research community will have to adapt to computational challenges.

High–performance computers (HPC) are normally applied to problems that require large amounts (hours, and sometimes, days or weeks) of computer time on commodity-class (e.g., Pentium- or SPARC-based) workstations. In fact, a widely-adopted approach to solving a computationally burdensome problem is to "run it overnight." Consequently, many users are eager to upgrade their computer systems because of the perceived inadequacy of existing systems when performing selected tasks. In some instances, especially when complex models are part of a GIS-based analysis, users may send problems through an HPC system for quick turn–around. Note, however, that HPC is a fuzzy, mutable concept; technology is continuously improving and the size of the problems we wish to solve is also constantly increasing.

During the past 15 years, high-performance computers, such as those sold by Cray, have excelled at solving repetitive arithmetic calculations, especially when the calculations were performed inside of loops, by "pipelining" the execution of instructions (Patterson and Hennessy 1994). Since most high-level operations, such as adding or multiplying two numbers together, actually require the completion of several lower-level instructions before a result is produced (Karplus 1989, pp. 7-8), a pipeline sequences these low-level instructions in a loop so that several of them overlap and can be executed *simultaneously*. Because of this simultaneity, pipelining can yield an improvement in overall system performance, and is especially valuable for numerically intensive computations. In addition to pipelining, manufacturers of high-performance computers have focused on components that work at very high speeds. Together, pipelining and increased clock speeds led to the successful construction of ever-higher-performance supercomputers through the 1980s. And these same basic principles are also now widely used in the superscalar architectures used by the current generation of high-performance workstations (see, for example, Ryan 1994). In fact, the newest generation of processors typically contains several different pipelines that are optimized to perform tasks such as integer and floating point arithmetic.

Pipelines, however, cannot be extended to great lengths because of data dependencies and because they cannot handle the conditional branches that occur in many application programs (Patterson and Hennessy 1994). These branches normally contain different code blocks that are conditionally executed based on the result of a logical operation (e.g., if . . . then constructs). As a consequence, the instructions that need to be loaded into the pipeline are unknown before the logical expression is evaluated, and this restricts pipeline performance. Clock speeds also cannot be increased indefinitely to improve performance because some component-fabrication barriers are proving to be difficult and expensive to overcome. Consequently, computer manufacturers who formerly relied on pipelining and increased clock speeds to maintain dominance in high-performance computing markets have turned to architectures that are more explicitly parallel. For exam-

Marc P. Armstrong is an associate professor at The University of Iowa where he holds appointments in the Department of Geography and in the graduate program in Applied Mathematical and Computational Sciences. A primary focus of his research is on the use of parallel processing to improve the performance of analysis methods used in spatial decision-support systems. Armstrong is a co-leader of NCGIA research initiative 17 (Collaborative Spatial Decision-Making).

ple, new scalable systems, such as the IBM SP-2, are built from networked processors that are functionally identical to those used in RS/6000 workstations.

Parallel HPC architectures considerably extend the notion of processing simultaneity observed in pipelined systems. In addition, recent improvements in hardware and software have led to easy-to-use parallel processing environments. And with the promulgation of standard versions of high-level languages such as C*, Fortran 90 and High-Performance Fortran (Loveman 1993), application codes can now be ported across a variety of parallel HPC platforms. In fact, Fox *et al.* (1994: 29) assert that High-Performance Fortran is likely to become a *de facto* standard during the next several years.

What does the Future Hold?

The business of making high-performance computer systems is a risky one with a history of innovative ideas that have led down dead ends of commercial viability. Despite this uncertainty, broad trends can be discerned; and these trends point to the use of at least limited parallelism in GIS applications. These trends are evident not only in GIS, but in business computing as well. Consider, for example, a recent issue of *Information Week*, a weekly magazine with a target audience of business and technology managers. In the September 26, 1994 issue (p. 27), an advertisement touts the performance of one *parallel* database management software product over two others. The important point here is that there are several major database software vendors with existing parallel-processing strategies. Additionally, on page 56 of the same issue, a news article describes the changing corporate scientific computing strategy of Atlantic Richfield Co. (an oil company that uses geological analyses to support oil exploration) as it has moved from the use of "traditional" IBM 3090 and Cray Y-MP computers to a parallel, 16 processor Cray CS6400. Though these examples are admittedly anecdotal, it is now easy to see that parallelism has entered into the business computing world. Such stories are reported with increasing regularity and, consequently, parallelism is no longer considered to be esoteric in many sectors of the business data-processing community. As parallelism gains widespread acceptance in commercial applications, the GIS community must be prepared to respond.

Patterson and Hennessy (1994) define a continuum of advances in computer architectures that is bounded on one end by evolutionary developments, such as pipelining and caches, that require little change to algorithms when they are implemented. On the other end of the continuum, they define revolutionary developments, such as parallel processing, that may require algorithms to be rewritten to achieve optimal performance. Because of differences in both the relative capacity of individual processors in parallel computing systems as well as the types of interconnections among them, at present, application programmers must concern themselves with these characteristics if they are to use these systems efficiently.

Despite these challenges, parallel architectures can be applied readily to the types of problems that are normally faced by researchers using GIS and urban and regional models. During the past few years I have been involved in several projects that have evaluated the use of parallel computers in geographically-based applications ranging from locational modeling (Armstrong and Densham 1992) and spatial statistics (Armstrong, Pavlik and Marciano 1994a; 1994b; Armstrong and Marciano 1995) to spatial interpolation problems (Armstrong and Marciano 1993; 1994). Other researchers have also reported on the performance of GIS operations when parallel architectures are applied to vector-based polygon overlay (Franklin 1989; Hopkins and Healey 1990; Wang 1993), terrain triangulation (Puppo *et al.* 1994), intervisibility analysis (Mills, Fox and Heimbach 1992; DeFloriani, Montani and Scopigno 1994), neighborhood (focal) operators (Li 1994), cartographic name placement (Mower 1993) and line simplification (Li 1993a). In addition, graphical applications often require substantial amounts of rendering time, especially those that require the use of realistic (e.g., lighting models and hidden surface removal) three-dimensional graphics. Consequently, researchers have begun to explore the use of parallelism in polygon rendering and shading in general purpose settings (Ellsworth 1994; Whitman 1994) as well as in GIS-based terrain representation (Ding and Densham 1994).

The general consensus that has emerged from this growing body of work is that many geographical problems can be decomposed and recast so that they can be efficiently executed in parallel HPC environments. Indeed, many of the geographical problems reported in the literature are almost "embarrassingly parallel" (Fox, Williams and Messina 1994) and straightforward translations of sequential codes to their parallel counterparts can be performed. Consider an example problem from the work reported above, that of calculating interpolated values for a regular grid from an irregular distribution of control points (Armstrong and Marciano 1994). In this example, the value of each cell in the interpolated matrix can be calculated independently of those for other cells, and consequently, all of the available parallel processing elements can be used very efficiently. Many other types of geographical problems exhibit similar characteristics that make them well-suited to parallel environments.

In addition to the use of dedicated, and often expensive, parallel computer systems, other significant trends indicate that it is now possible to implement parallel processing environments on a "shoestring budget" by using collections of networked workstations. Several strategies have been suggested in the literature to accomplish this goal, but two conceptually similar approaches, Linda (Carriero and Gelernter 1990) and PVM (Beguelin *et al.* 1991; 1993), are in broad use. They work by establishing a virtual computer that is built from processing nodes (workstations) connected by a conventional network. Though very little work on the application of this distributed, message-passing model of computing has been done in GIS (see Li 1993b), Rokos and Armstrong (1996) describe a Linda implementation of an algorithm that calculates a measure of spatial autocorrelation for polygon data. The results indicate that this particular geographical problem, which is decomposable into coarse-grained parallel processes, can be adapted to a distributed parallel environment. If this finding is more generally applicable, this means that distributed parallelism is well within the economic and technical grasp of many GIS researchers who have access to networked workstations. However, additional work needs to be performed to determine the general effectiveness of this distributed approach to GIS-based computing.

One additional trend in high-performance computing is also linked to the availability of networks. A fundamental flaw in the historical development of computers has been the implicit assumption that all types of computing can best be accomplished using a single architecture. However, complex problems often contain a mix of computation types that each can best be executed using a different architecture. The goal of heterogeneous computing is to divide a problem into a set of sub-problems that are not only executed using different processors, but different architectures as well (Freund and Siegel 1993). A heterogeneous collection of computers must be tied together by a high-speed network to form a virtual, or meta-computer (Smarr and Catlett 1992). It is possible that many GIS-based analyses can be computed efficiently using a heterogeneous approach to computation. Densham and Armstrong (1994), for example, discuss the application of heterogeneous processing to location-allocation analysis. Because many geographical problems can be easily converted into a parallel form, a first step in heterogeneous processing has already been demonstrated. The remaining difficult task of assignment of sub-problems to particular architectures will require a deeper theory of geographical computing as well as considerable research to uncover empirical regularities in the computational characteristics of GIS-related problems.

If workstations are networked to implement distributed, heterogeneous high-performance computing environments, then they also can be used to meet the computational demands of analyses that are required when GIS software is used by groups. Because many public policy issues are resolved by committees and task forces, new types of "groupware" or collaborative spatial decision-making environments must be developed. A hallmark of group decision-making is the generation and evaluation of multiple scenarios. When several individuals simultaneously begin to generate different scenarios, this may induce a substantial computational burden on the system. The magnitude of this burden may be especially critical because most meetings have a finite (and often brief) duration (e.g., one hour). Consequently, required analyses must be computed quickly to permit discussion about relevant alternatives during the course of a meeting. Distributed parallelism can be applied to GIS-based analyses to improve performance with the important secondary effect of providing time to discuss the merits and limitations of a broader range of alternative approaches to problem-solving.

Conclusions

GIS has been transformed during the past decade by the widespread availability of low-cost computing resources. The current generation of workstations is now able to manage, process and display the large volumes of data that are routinely encountered in GIS applications. Nevertheless, there remain significant computational impediments to the timely analysis of large, detailed and complex GIS problems. These impediments can be overcome through the use of HPC. Recent software developments now make it possible to create low-cost parallel-programming environments that are built from collections of networked workstations. These linked computers are harnessed by software that searches for idle machines, and then assigns a portion of a computational load to them. In this way, the collections of workstations that are becoming commonplace in many offices can be used to solve computationally complex GIS problems. Before such activities become routine, however, researchers must investigate ways to decompose geographical problems for parallel computers, and GIS software vendors must follow the lead of database management software vendors and begin the development of commercial GIS software that is designed to exploit the opportunities presented by parallel-computing environments.

Acknowledgments

This research represents part of Research Initiative 17, "Collaborative Spatial Decision-Making", of the National Center for Geographic Information and Analysis, supported by a grant from the National Science Foundation (SBR 88-10917). Support by NSF is gratefully acknowledged.

References

Armstrong, M.P. and P.J. Densham. 1992. "Domain Decomposition for Parallel Processing of Spatial Problems." *Computers, Environment and Urban Systems*, Vol. 16 (6): 497–513.

Armstrong, M.P. and R. Marciano. 1993. "Parallel Spatial Interpolation." In *Proceedings of the Eleventh International Symposium on Computer-Assisted Cartography* (Auto-Carto 11), Bethesda, MD: American Congress on Surveying and Mapping, pp. 414–423.

Armstrong, M.P. and R. Marciano. 1994. "Inverse Distance Weighted Spatial Interpolation Using a Parallel Supercomputer." *Photogrammetric Engineering and Remote Sensing*, Vol. 60 (9): 1097–1103.

Armstrong, M.P. and R. Marciano. 1995. "Massively Parallel Processing of Spatial Statistics." *International Journal of Geographical Information Systems*, Vol. 9 (2): 169–189.

Armstrong, M.P., C.E. Pavlik and R. Marciano. 1994a. "Parallel Processing of Spatial Statistics." *Computers & Geosciences*, Vol. 20 (2): 91–104.

Armstrong, M.P., C.E. Pavlik and R. Marciano. 1994b. "Experiments in the Measurement of Spatial Association Using a Parallel Supercomputer." *Geographical Systems*, Vol. 1 (4): 267–288.

Beguelin, A., J. Dongarra, A. Geist, B. Manchek and V. Sunderam. 1991. *A User's Guide to PVM: Parallel Virtual Machine*. Technical Report ORNL/TM-11826. Oak Ridge, TN: Oak Ridge National Laboratory.

Beguelin, A., J. Dongarra, A. Geist and V. Sunderam. 1993. "Visualization and Debugging in a Heterogeneous Environment." *IEEE Computer*, Vol. 26 (6): 88–95.

Carriero, N. and D. Gelernter. 1990. *How to Write Parallel Programs: A First Course*, Cambridge, MA: The MIT Press.

DeFloriani, L., C. Montani and R. Scopigno. 1994. "Parallelizing Visibility Computations on Triangulated Terrains." *International Journal of Geographical Information Systems*, Vol. 8 (6): 515–531.

Densham, P.J. and M.P. Armstrong. 1994. "A Heterogeneous Processing Approach to Spatial Decision Support Systems." In *Advances in GIS Research: Proceedings of the Sixth International Symposium on Spatial Data Handling*, Vol. 1: 29–45, T.C. Waugh and R.G. Healey, (eds.), London, UK: Taylor and Francis Publishers.

Ding, Y and P. J. Densham. 1994. "A Loosely Synchronous, Parallel Algorithm for Hill Shading Digital Elevation Models." *Cartography and Geographic Information Systems*, Vol. 21 (1): 5–14.

Ellsworth, D.A. 1994. "A New Algorithm for Interactive Graphics on Multicomputers." *IEEE Computer Graphics and Applications*, Vol. 14 (4): 33–40.

Franklin, R. 1989. "Uniform Grids: A Technique for Intersection Detection on Serial and Parallel Machines." In *Proceedings of Auto-Carto 9*, Bethesda, MD: American Congress on Surveying and Mapping, pp. 100–109.

Fox, G.C., R.D. Williams and P.C. Messina. 1994. *Parallel Computing Works!* San Francisco, CA: Morgan Kaufman.

Freund, R.F. and H.J. Siegel. 1993. "Heterogeneous Processing." *IEEE Computer*, Vol. 26 (6): 13–17.

Hopkins, S., and R. G. Healey. 1990. "A Parallel Implementation of Franklin's Uniform Grid Technique for Line Intersection Detection on a Large Transputer Array." In *Proceedings of the Fourth International Symposium on Spatial Data Handling*. IGU Commission on GIS: Columbia, SC, pp. 95–104.

Karplus, W.J. 1989. "Vector Processors and Multiprocessors." In *Parallel Processing for Supercomputers and Artificial Intelligence*, K. Hwang and D. DeGroot, (eds.), pp. 1–27. New York, NY: McGraw Hill.

Li, B. 1993a. "Suitability of Topological Data Structures for Data Parallel Operations in Computer Cartography." In *Proceedings of Auto-Carto 11*, Bethesda, MD: American Congress on Surveying and Mapping, pp. 434–443.

Li, B. 1993b. "Developing Network-Oriented GIS Software for Parallel Computing." In *Proceedings of GIS/LIS '93*, Bethesda, MD: American Congress on Surveying and Mapping, pp. 403–413.

Li, B. 1994. "When to Use Parallel Processing in GIS?" In *Proceedings of GIS/LIS '94*, Bethesda, MD: American Congress on Surveying and Mapping, pp. 514–523.

Loveman, D.B. 1993. "High Performance Fortran." *IEEE Parallel and Distributed Technology*, Vol. 1 (1): 25–42.

Mills, K., G. Fox and R. Heimbach. 1992. "Implementing an Intervisibility Analysis Model on a Parallel Computing System." *Computers & Geosciences*, Vol. 18 (8): 1047–1054.

Mower, J.E. 1993. "Automated Feature and Name Placement on Parallel Computers." *Cartography and Geographic Information Systems*, Vol. 20 (2): 69–82.

Patterson, D.A. and J. L. Hennessy. 1994. *Computer Organization and Design: The Hardware / Software Interface*. San Francisco, CA: Morgan Kaufman.

Puppo, E., L. Davis, D. DeMenthon and Y. A. Teng. 1994. "Parallel Terrain Triangulation." *International Journal of Geographical Information Systems*, Vol. 8 (2): 105–128.

Rokos, D. and M.P. Armstrong. 1996. "Using Linda to Compute Spatial Autocorrelation in Parallel." Forthcoming in *Computers & Geosciences*.

Ryan, B. 1994. "Alpha Rides High." *Byte*, Vol. 19 (10): 197–198.

Smarr, L. and Catlett, C.E., 1992. "Metacomputing." *Communications of the Association for Computing Machinery*, Vol. 35 (6): 45–52.

Wang, F. 1993. "A Parallel Intersection Algorithm for Vector Polygon Overlay." *IEEE Computer Graphics and Applications*, Vol. 13 (2): 74–81.

Whitman, S. 1994. " Dynamic Load Balancing for Parallel Polygon Rendering." *IEEE Computer Graphics and Applications*, Vol. 14 (4): 41–48.