[CP-02-003] High-Performance Computing and GIS

Abstract

High-performance computing (HPC) involves using multiple interconnected computers combined with parallel processing approaches to solve problems that are too large or complex for a single computer. Generally, HPC systems, such as supercomputers and clusters, use high-bandwidth, low-latency network interconnects to enable fast and efficient communication across processes running on the HPC system. In geographic information science (GIScience), growth in data size and analytical complexity drive more demand for computing power and the need for HPC. Researchers and practitioners use HPC to process more geospatial data, run finer-grained simulations, and explore problems at spatial and temporal scales that were previously impossible. The key challenge in using HPC is coordinating and synchronizing dozens, hundreds, or thousands of processors simultaneously.

Keywords: cluster

Author & citation

Shook, E. (2026). High-Performance Computing and GIS. The Geographic Information Science & Technology Body of Knowledge (Issue 1, 2026 Edition), John P. Wilson (ed). DOI: 10.22224/gistbok/2026.1.2

Explanation

1. Introduction
2. Architecture Foundations
3. Processor Architectures
4. Challenges and Opportunities

1. Introduction

High-performance computing (HPC) involves using multiple interconnected computers, combined with parallel processing, to solve problems that are too large or complex for a single computer. Achieving suitable performance with HPC systems demands technical skills and a deep understanding of the problems you are trying to solve, because these problems must be decomposed into multiple sub-problems to be solved simultaneously by dozens, hundreds, or thousands of processes, while maintaining synchronous communication across all processes. This approach is called parallel programming (see GIS & Parallel Programming, Shook 2019). A hallmark of geospatial problems that require HPC is that they involve highly synchronous processing, meaning processes must communicate frequently, which is common in complex analyses and models with significant spatial and/or temporal dependencies. This is distinct from High-Throughput Computing (HTC), which allows tasks to be mainly processed independently of one another. For an in-depth comparison between HPC and HTC, see High-Throughput Computing (Carbajales-Dale et al.,

2025). HPC is also distinct from Cloud Computing, which provides on-demand computing that may or may not be high-performance, with low-latency/high-bandwidth interconnects, large memory, or accelerators. A prime example of cloud computing is Amazon Web Services (AWS).

2. Architecture Foundations

HPC systems are not a single computer but rather a collection of computers, commonly referred to as a cluster, designed to work together to solve computational problems. Very large clusters that feature high-end storage and interconnects are referred to as a supercomputer. For a list of the largest supercomputers on the planet with millions of processing cores and requiring tens of kW of power to execute the world's most computationally demanding tasks, see the Top 500 List (https://top500.org/).

Clusters have two common types of nodes, another name for computers. A Head node is used to allow users to login, compile code, move data, and to tell the cluster to run code. Importantly, Head nodes are not used to run code. Instead, users submit a "job" using the Head node, which provides instructions on running your code. A Compute node is where your code actually runs. Large clusters may have one or two Head nodes, but hundreds or thousands of Compute nodes.

Job management systems are responsible for allowing multiple users to submit jobs that request compute resources and scheduling jobs across the available resources as efficiently as possible. Slurm is a common job management system used in many clusters and supercomputers. Common features of job management systems include a job submission, which allows users to request resources; a job queue, which allows jobs to wait until resources are available; a job scheduler, which prioritizes and sequences job execution and allocates them across resources; and a job monitoring system, to ensure jobs are functioning properly.

Interconnects connect nodes together using network cables. Infiniband and Ethernet are two common networking technologies used in HPC systems. Interconnects commonly feature low-latency, meaning that signals are sent between nodes quickly, and high-bandwidth, meaning that many signals can be sent between nodes. Specialized interconnects can be a major cost in clusters and supercomputers to improve overall performance and allow the cluster of computers to act as a single processing system. Interconnects not only allow nodes to communicate for coordination, but also allow for data to be transferred quickly and efficiently between nodes. HPC systems oftentimes use different network topologies to achieve improved performance, such as fat tree and torus.

3. Processor Architectures

Processing power is central to HPC systems. There are different types of processors used in HPC. Below, three types of processors are summarized. HPC systems can comprise different types of processors. If an HPC system supports a single type of computing unit, such as only CPUs, it has a homogeneous architectures. However, if HPC systems support a combination of different processing units, including CPUs, GPUs, and accelerators, they

have heterogeneous architectures. Programming for heterogeneous architectures is considered more difficult, but it can yield better performance.

Central Processing Units (CPUs) are the core processing units of a computer and handle general-purpose processing. While CPUs are found in laptop and desktop computers, CPUs in HPC systems usually contain more processing cores and more cache memory. A core is the smallest computing component in a CPU, for example. The number of cores generally determines how many computations can be performed simultaneously. It is common for Compute nodes in HPC systems to have 32, 64, or 128 cores.

Graphics Processing Units (GPUs) were initially used solely for displaying graphics on monitors. However, now they are designed as highly parallel architectures used to process massive data and run AI models. GPUs are massively-parallel architectures that work best with highly structured data such as rasters. Unlike CPUs that have 128 cores, high-end GPUs can have thousands of cores. For more information on programming GPUs, see GPU Programming for GIS Applications (Mower, 2018).

Accelerators are a broader class of specialized processing units designed to accelerate domain-specific components of a problem. A common characteristic of accelerators is they diverge from the seminal von Neumann architecture model which is used by CPUs. This traditional architecture model features control units and logic units that are separated from memory units. In contrast, accelerators integrate control, logic, and memory in unique ways to maximize performance for specific problems. However, a common tradeoff to these more exotic architecture models is they are more difficult to program (Shi, 2014).

4. Challenges and Opportunities

Three key challenges exist for the GIS&T community to use HPC. Firstly, is the modality of development and use of computational resources. Many GIScientists use graphical user interfaces (GUIs) to conduct geospatial analytics and modeling. For example, commercial GIS tools such as Esri ArcPro and open-source tools such as QGIS are GUIs. However, for many HPC resources, the primary mode of interaction is through command-line interfaces. Secondly, almost all clusters and supercomputers use Linux or Unix-based operating systems (OSs). Many GIScientists are familiar with Windows-based operating systems. Switching OSs can break existing code and may require learning new software and libraries. Thirdly, parallel programming is generally required to effectively use HPC and learning parallel programming can be considered a challenge (Shook et al., 2016). For example, high-resolution environmental data such as temperature and precipitation at the country or global scale can be terabytes in size, yet analyzing it is a typical GIS task. Conducting potentially multi-terabyte-scale analyses on a desktop computer can be impractical or impossible due to storage and memory limitations. To overcome these limitations, GIS professionals can use data parallelism and functional parallelism to decompose the problem into multiple sub-problems that can be solved simultaneously. For more information on parallel programming, see GIS & Parallel Programming (Shook, 2019).

Three key opportunities exist. Firstly, cyberinfrastructure (CI) can lower barriers to accessing HPC. In the United States, ACCESS is a National Science Foundation (NSF) supported cyberinfrastructure that provides researchers and students free access to multiple HPC systems across the US. Another example of CI in the US is the I-GUIDE project

(I-GUIDE Project, n.d.). Secondly, HPC removes computational barriers for GIS&T, enabling scientists and professionals to tackle larger, more complex geospatial problems worldwide (Armstrong, 2000). Thirdly, advances in software systems are lowering barriers. One example is Open OnDemand (https://www.openondemand.org/), which allows for web-based access to traditional GUI software running on HPC hardware. Open OnDemand and similar systems can run traditional GIS software such as QGIS, reducing a key barrier in needing to learn and use command-line interfaces.

References

Armstrong, M. P. (2000). Geography and Computational Science. Annals of the Association of American Geographers, 90(1), 146–156.

Carbajales-Dale, P., Cleveland, S., and Post, G. (2025). High Throughput Computing and GIS. The Geographic Information Science & Technology Body of Knowledge (2025 Edition), John P. Wilson (ed.). DOI: 10.22224/gistbok/2025.1.1

I-GUIDE. (n.d.). Institute for Geospatial Understanding through an Integrated Discovery Environment (I-GUIDE). Last accessed March 12, 2026.

Mower, J. (2018). GPU Programming for GIS Applications. The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2018 Edition), John P. Wilson (Ed.).

Shi, X., Lai, C., Huang, M., & You, H. (2014). Geocomputation over the Emerging Heterogeneous Computing Infrastructure. Transactions in GIS, 18, 3-24.

Shook, E. (2019). GIS and Parallel Programming. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2019 Edition), John P. Wilson (Ed.).

Shook, E., Hodgson, M. E., Wang, S., Behzad, B., Soltani, K., Hiscox, A. and Ajayakumar, J. (2016). Parallel cartographic modeling: a methodology for parallelizing spatial data processing. International Journal of Geographical Information Science 30 (12):2355–2376.