HPC, or high-performance computing, refers to the use of supercomputers or clusters of computers to solve difficult computational problems that typically arise through scientific inquiry. Computer users turn to HPC when a problem is too large to solve on a conventional laptop or workstation (because it requires too much memory or disk space) or runs too slowly (because the algorithm is complex, the dataset is large, or data access is slow). In some cases, the term HPC may also be applied to computing systems used by businesses for data warehousing and analysis.

Parallelism

HPC systems often derive their computational power from exploiting parallelism, meaning the ability to work on many computational tasks at the same time. An application or algorithm that does not exploit parallelism is usually called sequential or serial, because it has to execute tasks individually in a sequence or series.

Modern laptops and desktops have some limited support for parallelism because of the popularity of multi-core processors. These processors can run a handful of concurrent tasks, which improves the performance of multi-tasking, e.g. running your web browser, word processor, email client, and other desktop applications at the same time.

However, HPC systems typically offer parallelism at a much larger scale, with hundreds, thousands, or (soon) even millions of tasks running concurrently. Parallelism at this scale poses many challenges. Some algorithms have difficult-to-remove bottlenecks, or serialization points, where a single task must complete before any other tasks can begin. In other cases, tasks may need to share the same data or modify data in a carefully coordinated order. Writing parallel software can be challenging, and many existing software packages do not already support parallelism. Sometimes "parallelizing" an existing software package or algorithm requires a substantial investment in research and development.

In other cases, computational problems parallelize easily, because the nature of the problem is such that it can be broken into many small task that are independent of each other. A problem of this type is sometimes called "embarrassingly parallel," because it is so straightforward or simple to implement that it does not require substantial research.

Some Reasons to Use HPC at CCV

- You have a program that can be recompiled or reconfigured to use optimized numerical libraries that are available on CCV systems but not on your own system.
- You want to run a software package that is impractical to install or support on your own system.
- You have an "embarrassingly parallel" problem: for instance, you have a single application that needs to be rerun many times with different parameters.
- You have a serial application that you would like to run faster, and you are prepared to invest in rewriting or redesigning it to expose more parallelism; you may also need access to a parallel platform to test and debug the program during development.
- You have an application that has already been designed with parallelism and you would like to put it into production on CCV's Oscar cluster.
- You have an application that requires a large allocation of memory and it can fit on CCV's large-memory nodes, which have 64GB or 128GB of system memory.
- You have an I/O-intensive application that will benefit from CCV's high-performance filesystems, which can access data at much higher rates than conventional laptops and workstations.

Author: Mark Howison // Last Updated: 23 Sept 2010