

# Parallel Computing: Background

Parallel computing is the Computer Science discipline that deals with the system architecture and software issues related to the concurrent execution of applications. It has been an area of active research interest and application for decades, mainly the focus of high performance computing, but is now emerging as the prevalent computing paradigm due to the semiconductor industry's shift to multi-core processors

## A Brief History of Parallel Computing

The interest in parallel computing dates back to the late 1950's, with advancements surfacing in the form of supercomputers throughout the 60's and 70's. These were **shared memory** multiprocessors, with multiple processors working side-by-side on shared data. In the mid 1980's, a new kind of parallel computing was launched when the Caltech Concurrent Computation project built a supercomputer for scientific applications from 64 Intel 8086/8087 processors. This system showed that extreme performance could be achieved with mass market, off the shelf microprocessors. These massively parallel processors (**MPPs**) came to dominate the top end of computing, with the ASCI Red supercomputer computer in 1997 breaking the barrier of one trillion floating point operations per second. Since then, MPPs have continued to grow in size and power.

Starting in the late 80's, **clusters** came to compete and eventually displace MPPs for many applications. A cluster is a type of parallel computer built from large numbers of off-the-shelf computers connected by an off-the-shelf network. Today, clusters are the workhorse of scientific computing and are the dominant architecture in the data centers that power the modern information age.

Today, parallel computing is becoming mainstream based on **multi-core processors**. Most desktop and laptop systems now ship with dual-core microprocessors, with quad-core processors readily available. Chip manufacturers have begun to increase overall processing performance by adding additional CPU cores. The reason is that increasing performance through parallel processing can be far more energy-efficient than increasing microprocessor clock frequencies. In a world which is increasingly mobile and energy conscious, this has become essential. Fortunately, the continued transistor scaling predicted by Moore's Law will allow for a transition from a few cores to many.

## Parallel Software

The software world has been very active part of the evolution of parallel computing. Parallel programs have been harder to write than sequential ones. A program that is divided into multiple concurrent tasks is more difficult to write, due to the necessary synchronization and communication that needs to take place between those tasks. Some standards have emerged. For MPPs and clusters, a number of application programming interfaces converged to a single standard called **MPI** by the mid 1990's. For shared memory multiprocessor computing, a similar process unfolded with convergence around two standards by the mid to late 1990s: **pthread**s and **OpenMP**. In addition to these, a multitude of competing parallel programming models and languages have emerged over the years. Some of these models and languages may provide a better solution to the parallel programming problem than the above "standards", all of which are modifications to conventional, non-parallel languages like C.

As multi-core processors bring parallel computing to mainstream customers, the key challenge in computing today is to transition the software industry to parallel programming. The long history of parallel software has not revealed any "silver bullets," and indicates that there will not likely be any single technology that will make parallel software ubiquitous. Doing so will require broad collaborations

across industry and academia to create families of technologies that work together to bring the power of parallel computing to future mainstream applications. The changes needed will affect the entire industry, from consumers to hardware manufacturers and from the entire software development infrastructure to application developers who rely upon it.

Future capabilities such as photorealistic graphics, computational perception, and machine learning really heavily on highly parallel algorithms. Enabling these capabilities will advance a new generation of experiences that expand the scope and efficiency of what users can accomplish in their digital lifestyles and work place. These experiences include more natural, immersive, and increasingly multi-sensory interactions that offer multi-dimensional richness and context awareness. The future for parallel computing is bright, but with new opportunities come new challenges.