A programming model, system software and supporting architecture that are parallel must be developed in order to achieve parallel computing.

A programming model, system software and supporting architecture that are parallel must be developed in order to achieve parallel computing. Programming models will need to become more human-centric and researchers will need to use human subject experiments in order to determine the most efficient way to program manycore systems. Manycore systems, rather than today’s popular multicore systems, will be one of the first steps to achieving parallel computing. Parallel applications will be classified within the Thirteen Dwarfs, where if applications belong in a certain Dwarf, then the applications are similar in computation and data movement. Simple and small processors should be used, and autotuners should have a larger role to compilers in order to translate parallel programs.

Significant applications contain several dwarfs that each take up a part of the application’s computation – the performance of a large application will depend on each dwarf’s performance as well as how the dwarfs are composed together. Simple and small processors should be used and we want to minimize energy and peak power, while maximizing performance. The switch to parallel programming will require performance counters, which help computer architects evaluate their work, to become more important and developed than they are today. Parallel applications’ performance, however, will largely depend on the quality of the generated code – while this is traditionally done by the compiler, the difficulty in adding new optimizations to compilers will mean that autotuners should have a larger or complementary role to compilers. Embedded computers, which are vulnerable to viruses and other such attacks due to their connections to networks, will be rescued by virtual machines (allow an operating system with running applications to be treated as software). In order to determine the success level of parallel computation, we must take into account programmer productivity and final implementation efficiency.