Serious research in parallelism in software, hardware and programming models is needed to solve the most important computational problems of today and future.

This paper discusses possible solutions and proposes research areas in software and hardware for future parallel computing. They propose 13 “dwarfs” - abstract application kernels that are basis for the most important applications in the future - to guide innovation and evaluation of new prototypes and serve as a new type of benchmarks. As the limit of maximizing hardware performance by increasing clock speed has been reached, authors propose “manycore” homogenous or heterogenous system of 1000s of simple processors with emphasis on power efficiency. Furthermore due to the “memory wall” innovation is needed to increase memory bandwidth and reduce latency with hybrid interconnect design that can adapt.

Synchronization constructs such as transactional memory, full-empty bits or message passing as well as performance and energy counters should be build-in. In order to bridge the gap between applications and hardware a high priority is research in programming models with emphasis on human errors and programmer productivity. These models must support rich set of data sizes and types and different types of parallelism. Also new autotuners for parallel code instead of traditional compilers are needed.

This paper discusses possible solutions and proposes research areas in software and hardware for future parallel computing. Authors propose 13 “dwarfs” - general patterns of computation and communication specified in a high level of abstraction that are basis for the most important applications in the future. Their purpose is to guide innovation and evaluation of new prototypes and serve as a new type of benchmarks. They are: 1. dense linear algebra, 2. sparse linear algebra, 3. spectral methods, 4. N-body methods, 5. structured grids, 6. unstructured grids, 7. MapReduce (Monte Carlo), 8. combinational Logic, 9. graph traversal, 10. dynamic programming, 11. back-track and branch+bound, 12. graphical models, 13. finite state machine. They cover applications in embedded computing, general purpose computing, machine learning, graphics and games, databases.

The limit of maximizing hardware performance by increasing clock speed has been reached. Authors propose “manycore” die on which there are 1000s of small, relatively simple processors, floating point units, vector and Single Instruction Multiple Data processing elements. The question is still open if it is better to have homogeneous or heterogeneous manycores. The main concern in designing future processors will be power efficiency.

Since the cost of hardware is shifting from processing to memory new innovations are needed to increase memory bandwidth and reduce latency. Even more importantly, for half of the dwarfs the major obstacle to good performance is memory wall (it can take up to 200 cycles to access DRAM). Authors advocate hybrid interconnect design that uses circuit switches to adapt. Synchronization should be addressed with hardware, for example, using transactional memory, full-empty bits or message passing. They also advocate built-in hardware performance and energy counters.

In order to bridge the gap between applications and hardware a high priority is research in programming models with emphasis on human errors and programmer productivity. These models must support rich set of data sizes and types and different types of parallelism. Also new autotuners for parallel code instead of traditional compilers are needed. Additionally virtual machines will be used instead of grand Operating Systems.