Tell me, I will forget, Show me, I may remember, Involve me, I will understand.”
Confucius

Finding parallelism is more important than ever, but parallelizing a program still remains as an almost manual process. Thus, there is an urgent need for new tools that make parallel programming easier. Parallware Trainer is a new software tool for high-productivity HPC training using OpenMP and OpenACC.

TECHNICAL FEATURES

- Interactive, real-time editor GUI
- Assisted code parallelization using OpenMP and OpenACC
- Transparent, local/remote execution and benchmarking
- Support for the C/C++ and Fortran programming languages
- Detailed report of the parallelism discovered in C code
- Support for multiple compilers.

BENEFITS

- Faster, more effective learning
- Reduced learning curve
- All-in-one learning tool for parallel programming
- Immediate use of parallel programming
- Support for multicore processors and GPUs

EXPERIENTIAL LEARNING

Education and training is organized today mostly around courses, workshops and hackathons. As shown in the pyramid, in courses the participants passively listen to a lecture or presentation, and apply parallel programming concepts to simple example codes in hands-on sessions. Workshops increase learning retention through interactive training activities between the participants. And hackathons are events where people come together to solve problems, participating in groups of about 2-5 individuals that take out their laptops and dive into their own problems. Hackathons allow experiential learning as participants work with a coach to learn through immediate practice in the optimization and parallelization of their own code. However, hackathons are expensive small events that train only a few people throughout the year.

Parallware Trainer aims at bringing the benefits of workshops and hackathons to a broader STEM audience.
Berkeley algorithmic motifs\cite{1,2} describe the science of the applications. Motifs aim at helping in writing parallel programs so that it is as easy as writing codes for sequential computers. The bridge is used to illustrate that software is the problem in bridging the gap between user applications and the parallel hardware industry. The new tower represents how Parallware approaches this software challenge. It highlights that the features of the code implemented by the programmer directly impact on the productivity of the parallelization process.

Production-level Compilers (e.g., Intel ICC, GNU GCC, NVIDIA PGI) are not of practical use to understand the failure reasons behind the parallelization of Applications. Compilers typically report the true/anti/output dependences that prevent the parallel execution of a code, being difficult for programmers to know how to rewrite their codes to enable parallelism. In contrast, Parallware reports the parallel patterns found in the code, hiding the complexity of the dependences found by the compiler. The screenshot of Parallware Trainer\cite{4} shows an example code from the motif Sparse Linear Algebra. The ATMUX code computes the product of a transposed sparse matrix and a vector, which corresponds to a Parallel Sparse Reduction in terms of Parallware parallel patterns.

References:

Best practices on parallel programming typically make coding recommendations, such as, to use stride-1 memory accesses and to prefer structures-of-arrays instead of arrays-of-structures.