pMapper: An Automatic Parallelization Toolbox for Array-based Algorithms
Technology #12483

Applications
Applications for this toolbox are found in scientific computing.

Problem Addressed
The pMapper Automatic Parallelization Toolbox maps algorithms onto parallel processors to improve algorithm efficiency. By automatically determining how to parallelize complex algorithms, this toolbox dramatically increases the speed and performance of computationally intensive software. This toolbox enables efficient parallelization and will become an essential tool for software developers as the computing world moves towards larger datasets and multiple processors.

Technology
pMapper finds efficient maps - in other words, it automatically assigns blocks of data to processing elements. The toolbox requires an underlying global array library or library that implements operations on parallel arrays. pMapper uses a two-phase mapping architecture: first, it collects data about the performance of the algorithm. This timing data is then used to generate maps for how to assign arrays to different processors. pMapper uses ‘lazy evaluation’, meaning that it delays execution until necessary. This approach enables pMapper to have the greatest possible amount of information about the program. pMapper also uses global optimization to ensure that the entire algorithm is running as efficiently as possible, rather than any individual component. pMapper has been tested on many benchmark algorithms and achieved speedup factors between 2.6 and 17, dramatically increasing algorithm performance. By improving efficiency without requiring detailed changes to the code, this toolbox will both improve performance and save time for software engineers.

Advantages

- Uses dynamic code analysis during runtime, enabling the toolbox to use information from the runtime environment to optimize efficiency
- Performs global optimization over all operations, further improving algorithm performance
- Improves algorithm performance without requiring significant changes to the code, saving time for programmers

Categories For This Invention:
Lincoln Laboratory
Software (Copyright)
Other (Software)
Intellectual Property:
Copyright Software

Inventors:
Nadya Bliss
Sanjeev Mohindra

Publications:
Addressing the Multicore Trend with Automatic Parallelization
Lincoln Laboratory Journal
2007

External Links:
Lincoln Laboratory
http://www.ll.mit.edu/

Image Gallery: