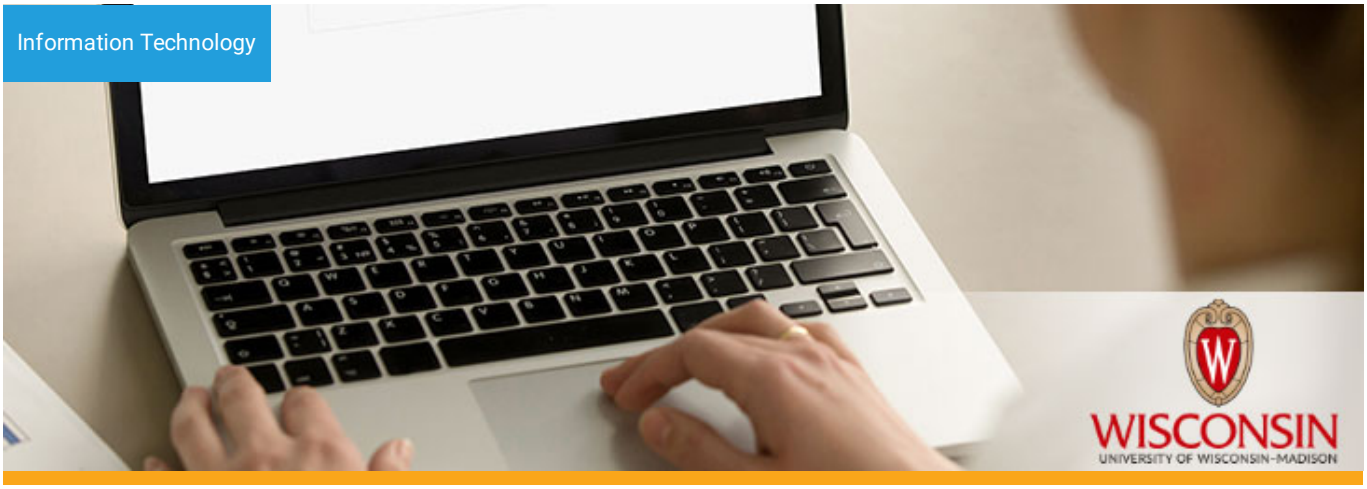
Information Technology



Dynamic Dependence-Based Parallel Execution of Software for Performance Optimization

View U.S. Patent No. 8,417,919 in PDF format.

WARF: P08192US02

Inventors: Gurindar Sohi, Matthew Allen

The Wisconsin Alumni Research Foundation (WARF) is seeking commercial partners interested in developing a mechanism that improves software performance by implementing parallelization while maintaining sequential program semantics.

Overview

Modern microprocessors integrate multiple cores onto a single chip. Improving software performance on these multicore processors requires complex parallel programming models. Parallelization significantly increases the cost and time required to implement robust software, and has been used traditionally in niche markets that can justify the extravagant cost. Due to the increasing prevalence of multicore processors, implementation of parallel execution in all types of software is needed to realize optimized performance.

One method of producing programs that run in parallel is for a programmer to divide the program into multiple threads designed to execute independently. Creating such a procedure is difficult, since access to shared data must be carefully synchronized to ensure only one thread at a time may access the data. Consequently, multi-thread programming is significantly more challenging and error prone than sequential programming. A method that retains the simplicity of sequential programming while improving software performance through parallelization is needed.

The Invention

UW-Madison researchers have developed a mechanism that maintains sequential program semantics while implementing a dynamic parallel execution of independent computations in a manner which reduces the likelihood of data races – instances in which multiple threads accessing shared data “race” one another to influence the output.

Using this model, programmers specify a serializer, a piece of code that when executed, maps a “function invocation” into a serialization set. The programmer may specify the serializer such that all function invocations operating on the same data are mapped to the same serialization set. Then, the execution of the function invocations are automatically parallelized by assigning them to a number of delegate threads. In this manner, function invocations in different serialization sets are assigned to different delegate threads. This ensures that operations on the same datum are executed in program order, avoiding data races and retaining the advantage of predictable, deterministic execution.

Applications

- Executing a program on a multi-processor computer with shared memory
- Exploiting fine-grain parallelization in common sequential programs in which different computational operations may write to the same data

Key Benefits

- Retains the simplicity and predictability of the sequential programming model

- Leverages multicore processors using parallelization while avoiding data races
- Permits pre-emptive detection of errors in the parallelization process

Stage of Development

The method of parallelized instrumentation has been evaluated by the researchers on real machines, including a multicore-based Sun Fire T2000 and a SMP Sun Fire V880. The fast communication afforded by multicore processors has been shown to yield significant speedup of parallelized instrumentation.

Additional Information

For More Information About the Inventors

- Gurindar Sohi

Related Technologies

- WARF reference number P07057US describes an automated way to parallelize the execution of a sequential computer program for multiple processors.

Publications

- Allen M.D., Sridharan S. and Sohi G.S. 2009. Serialization Sets: A Dynamic Dependence-Based Parallel Execution Model. ACM SIGPLAN Not. 44, 85-96.

Tech Fields

- Information Technology : Computing methods, software & machine learning

For current licensing status, please contact Emily Bauer at emily@warf.org or 608-960-9842