Vector Fabrics

Parallelize your C program for multicore x86

vfThreaded-x86

- *Utilize the performance of the latest multicore x86 processors*
- *Develop scalable programs for dual-core, quad-core, and manycore processors*
- *Parallelize your C program in hours instead of months*
- *Point and click user interface shows known-good parallelization strategies*
- *Draw rapid conclusions from an intuitive graphical user interface*
- *Predict the speed of your parallelized program before implementing*
- *Don't waste time on ineffective strategies*
- *Get step-by-step implementation instructions*
- *Avoid introducing subtle, hard-to-find bugs*
- *Parallelize programs having millions of lines of code*
- *Simple subscription reduces your up-front costs and risks*

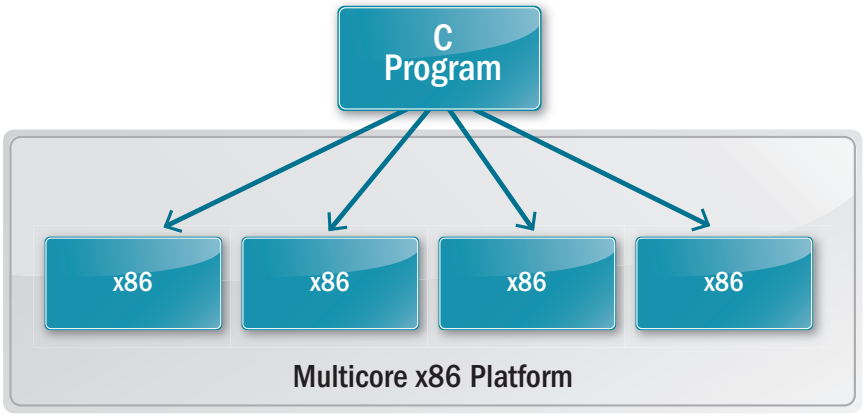
Even today's low-end x86 processors have multiple cores, but writing multi-threaded software that takes advantage of this is a labor-intensive and error-prone manual process. It can take months to complete, and your results might not even work correctly. There are always subtle code constructs that work in a sequential version but which create problems when parallelized.

Even if you manage to get it working, your implementation might not reach the required performance, or may even be slower than the original. All this gets even harder if you're working with code written by someone else, like open-source or legacy code.

vfThreaded-x86 changes all that. It is the only tool that gives you the precise information you need to make sure that your parallelized software will work correctly and have the highest performance on your multicore x86 machine.

vfThreaded-x86 saves you months of guesswork

- Find the best multi-threading strategy using precise information about your C program
- Identify and eliminate barriers to parallelization
- Understand the data communication requirements of different parallelization options
- See the specific impact of your target x86 processor on the performance and overhead of your strategy
- Ensure that you have exercised all critical behaviors in your program
- Quickly implement a correct-by-construction parallel version of your program



Let vfThreaded-x86 guide you

vfThreaded-x86 removes the guesswork and thoroughly but quickly analyzes your program behavior on your specific x86 target platform and turns that knowledge into specific information you can use to make decisions. You can get results in hours instead of months, even for programs you didn't write yourself. And you can have confidence that the parallel version will work correctly.

Don't waste time on dead-end strategies

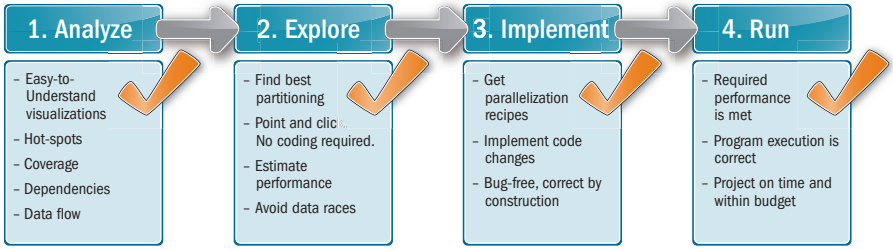
vfThreaded-x86 can find parallelization opportunities in your program that will preserve the program semantics and will not result in race conditions or other problems. You can then spend your time evaluating the different options and deciding which one you want to use.

Estimate your performance

Multi-threading can add overhead to your program. You may have to synchronize data, and that takes time. vfThreaded-x86 estimates the actual performance of your program for a specific parallelization strategy, taking the x86 architecture's traits into account.

Implement your strategy

vfThreaded-x86 will give you step-by-step instructions, which we call recipes, for implementing your parallelization strategy. After any step, you can always test what you've done to make sure it's all still working correctly before moving on to the next step. A vfStream thread communication and synchronization library is provided, keeping you focused on how your program works instead of on the intricacies of parallelization.



Partitioning can be accomplished in four simple steps. vfThreaded-x86 does the analysis based on your test cases; you can then explore the partitioning options until you achieve the performance you want. You then use the recipes to implement the partitioning and run the results.

The intuitive GUI shows you at a glance where the hotspots are as well as the dependencies that may make some partitioning choices expensive or impossible. You can also easily examine your test coverage and core utilization. Recipes provide detailed implementation instructions that help you avoid introducing bugs.

Know that you've done a complete job

vfThreaded-x86 learns how your program works by watching it as it executes test data that you provide. Subtle dependencies and interactions can be discovered only by making sure that test data is comprehensive. vfThreaded-x86 provides coverage reports so that you can be confident your parallel will program operate correctly under all conditions.

Take advantage of the cloud

Vector Fabrics' vfCloud implementation gives you abundant, secure computing power. You don't have to install any software, manage a compute farm, or licenses, and you don't have to manage revisions. It just works, in any environment.

System Requirements: vfEmbedded runs under Windows, Mac or Linux inside a web browser that supports Flash.

To see for yourself just how much time and work you can save, visit us at **www.vectorfabrics.com**.

Vector Fabrics
www.vectorfabrics.com
Paradijslaan 28, 5611 KN Eindhoven, The Netherlands
Tel: +31 40 8200960 Fax: +31 40 2366403