Brief Announcement: Open Cilk

Tao B. Schardl  
MIT CSAIL  
Cambridge, MA, USA  
neboat@mit.edu

I-Ting Angelina Lee  
Washington University in St. Louis  
St. Louis, MO, USA  
angelee@wustl.edu

Charles E. Leiserson  
MIT CSAIL  
Cambridge, MA, USA  
cel@mit.edu

ABSTRACT
Open Cilk is a new open-source platform to support Cilk multithreaded programming, especially for researchers and teachers. Open Cilk aims to provide a full-featured implementation of Cilk that is easy to modify and extend. Based on the award-winning Tapir/LLVM compiler, Open Cilk will provide a streamlined run-time system and feature comprehensive static instrumentation for dynamic-analysis tools. As a community-infrastructure project, Open Cilk encourages contributions from researchers in the areas of languages, compilers, runtime systems, tools, libraries, and benchmarks.

CCS CONCEPTS
• Theory of computation → Parallel algorithms; • Computing methodologies → Parallel programming languages; • Software and its engineering → Parallel programming languages; Compilers; Runtime environments; Software testing and debugging;

KEYWORDS
Cilk; multicore programming; parallel algorithms; parallel languages; productivity tools

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM Reference Format:

1. Introduction
For years Cilk [6, 11, 19, 33, 42] has offered software developers a high-performance multithreaded programming environment while providing researchers with a platform for investigating a wide range of topics in multithreaded computing. When Cilk research began at MIT in 1994, it was conceived [5, 6, 10, 11, 26, 41] as a parallel-programming extension to the C programming language [28]. In 2006, at the advent of the multicore revolution, MIT spun off Cilk into the start-up Cilk Arts, Inc., which produced the open-source Cilk++ platform [33] as an extension to C++ [51]. Intel Corporation acquired Cilk Arts in 2009, added vectorization directives, and rechristened Cilk++ as Cilk Plus [19, 42], making it available commercially in the ICC compiler and in open-source implementations for the GCC [2, 13] and LLVM [21] compilers.

But Intel was unable to maintain a strong internal Cilk-development group. By 2013, attrition among Intel’s Cilk team caused Intel to cease active development of the Cilk Plus product and focus solely on maintenance. In August 2017, Intel announced [23] that it would no longer even maintain Cilk Plus. Since Intel also maintained the open-source GCC implementation, GCC announced it was dropping its support for Cilk as well [14].

Cilk Hub and Open Cilk
In January 2018, a group of us decided to address the flagging support for Cilk by creating a community-driven organization called Cilk Hub (http://cilkhub.org) to maintain and develop the Cilk technology. Cilk Hub provides a technically superior Open Cilk system to serve the needs of software developers, multicore and parallel-computing researchers, and teachers of parallel computing. Open Cilk features a new compiler based on Tapir/LLVM [45], comprehensive static instrumentation, and a suite of productivity tools. Cilk Hub provides external documentation of the Open Cilk system, offers workshops and tutorials, and coordinates annual meetings. Cilk Hub has formed an Advisory Board of leading researchers in multicore computing to set directions and priorities.

Cilk Hub will be active in developing Open Cilk as an open-source community project. Project enhancements include a lean-and-mean runtime system, new linguistics, new productivity tools, and libraries and benchmarks. Cilk Hub will manage integration of community-proposed enhancements to Open Cilk to ensure that it remains an efficient and malleable platform for multicore-programming research. Cilk Hub will produce a cloud VM that will allow researchers, teachers, and programmers to use Open Cilk technology — including the compiler, runtime system, libraries, and tools — without elaborate installation.

2. The Open Cilk system
Open Cilk is a full-featured, open-source implementation of Cilk designed to run on Linux and other Unix-like systems. Open Cilk provides a platform onto which researchers and teachers currently using Cilk Plus can migrate their ongoing research and educational materials with minimal effort. Open Cilk also serves as a vehicle for future research on Cilk programming and parallel-language technology.

Language
Open Cilk is fully compatible with Cilk Plus, minus its vector notation. Open Cilk also enhances Cilk Plus by, for example, providing support for spawning statement blocks, instead of just functions. Going forward, we plan to improve the Cilk language based on input...
from the community. As we develop Cilk, we will strive to maintain its simplicity and ensure that all new language features work seamlessly with existing features. Some of the language features to be considered include simple linguistics for reducer hyperobjects [12] and a more efficient implementation of reducer hyperobjects in the Open Cilk runtime system [31], as well as linguistic and runtime support for on-the-fly pipeline parallelism [32], speculative parallelism [8], async-finish parallelism as in X10 [7], amorphous parallelism [39], and vectorization and data parallelism [17].

**Compiler**

Open Cilk’s compiler is based on the Tapir/LLVM compiler [45] and its Cilk/Clang front end. Tapir/LLVM embeds fork-join parallelism into LLVM’s intermediate representation (IR). The resulting simple extension to LLVM’s IR, called Tapir, allows compiler optimizations for serial code to operate on parallel code with minimal changes. Integrating Tapir into LLVM required only adding or modifying about 6000 source lines of code, compared to LLVM’s 4-million-line codebase. These few changes enable the Open Cilk compiler to produce more efficient parallel executables than other compilers for parallel languages.

**Runtime system**

Although the Open Cilk system currently uses the Cilk Plus runtime system, we plan to develop a new open-source work-stealing runtime system for Open Cilk to suit the needs of researchers studying parallel runtime systems. The Open Cilk runtime system will implement the core runtime functionality of the existing Cilk Plus runtime system, but feature a smaller, simpler codebase and extensive documentation. We plan to enhance the runtime system with new capabilities. Among the considerations are support for long-latency operations such as file I/O [37] and interoperability with the system environment, including debuggers [1] and Pthreads [18].

**Comprehensive static instrumentation**

To support productivity tools for parallelism, the Open Cilk compiler provides comprehensive instrumentation based on the CSI framework [44]. CSI provides a simple application program interface (API) consisting of hooks (functions) that are automatically called at salient points throughout the program-under-test, such as function calls, basic blocks, and memory operations. A tool writer can implement a “CSI-tool” as a library, without modifying the compiler, by simply defining the semantics of the various API hooks. Undefined hooks are elided, minimizing the overhead of the CSI-tool. Whereas the original CSI framework operated on serial programs, the Open Cilk compiler extends CSI to instrument Tapir, giving tool writers additional hooks that correspond to `cilk_spawn`, `cilk_sync`, and other important points in parallel programs.

**Productivity tools**

The Open Cilk system includes two open-source CSI-tools: the Cilkscan nondeterminacy detector, and the Cilkscale scalability analyzer. Cilkscan implements the core functionality of Intel’s closed-source Cilkscan tool [20], and Cilkscale implements the core functionality of Intel’s closed-source Cilkview tool [16]. We plan to enhance Open Cilk’s suite of productivity tools with performance and functionality enhancements [38, 40, 54, 56] to existing tools, as well as with implementations of additional productivity tools, such as the Cilkprof scalability profiler [43] and a tool to support record and replay [30, 55].

**Libraries and benchmarks**

Open Cilk will incorporate an expansive collection of libraries and benchmark programs. We plan to build this collection by drawing from the extensive body of community-developed parallel libraries and benchmarks [3, 4, 9, 11, 15, 22, 24, 25, 27, 29, 34–36, 46–50, 52, 53, 57].

3. The architecture of Open Cilk

Open Cilk is architected to provide a robust platform for professional application development and for next-generation multicore-computing research and education. The design and implementation of Open Cilk exhibits five central software qualities: compatibility, open source, componentization, integration, and reliability.

**Compatibility**

Open Cilk provides support for modern C++ features and existing Cilk Plus applications and libraries. To ensure that Cilk can parallelize large applications, Open Cilk supports crucial features, such as using C++ exceptions in Cilk programs. Open Cilk provides support for existing Cilk Plus libraries, including reducer hyperobjects [12] and deterministic parallel random-number generators [35].

**Open source**

The entire Open Cilk implementation is open source. Open Cilk thus allows researchers to develop enhancements to all parts of the system: compiler, runtime system, libraries, and productivity tools. Cilk Hub manages the process of integrating community-proposed innovations into Open Cilk so that applications researchers and professional developers can exploit the enhancements.

**Componentization**

The Open Cilk system is divided into distinct software components with well-defined interfaces. This design allows researchers to experiment with alternative implementations of different individual components.

**Integration**

The components of Open Cilk are integrated to ensure that they all work together correctly. Cilk Hub aims for consistency and compatibility among all components with each system release. Cilk Hub will produce a cloud VM so that researchers, teachers, and programmers can use Open Cilk technology in the cloud without elaborate installation, confident that all components will interoperate.

**Reliability**

To ensure that Open Cilk releases are stable, fast, and free of serious bugs, Open Cilk includes an extensive suite of unit tests, regression tests, and benchmarks, all of which it will continue to develop. We also plan to develop a cloud-based testing infrastructure for continuous testing and integration of changes to Open Cilk.

4. Conclusion

Cilk Hub welcomes contributions to Open Cilk. To stay abreast of further developments, please visit http://cilkhub.org.