

The Eureka Programming Model for Speculative Task Parallelism (Artifact)

Shams Imam and Vivek Sarkar

Rice University, Houston, TX, USA
{shams,vsarkar}@rice.edu

Abstract

This artifact includes a Java-based library implementation of the Eureka programming model (EuPM) that simplifies the expression of speculative parallel tasks. Eureka-style computations are especially well-suited for parallel search and optimization applications. The artifact includes implementations of the eureka patterns that are supported by our **Eureka** API. These patterns include search, optimization, convergence, N-version programming, and soft real-time deadlines. These

different patterns of computations can also be safely combined or nested in the EuPM, along with regular task-parallel constructs, thereby enabling high degrees of composability and reusability. We also include source code of the different benchmarks presented in the paper. The interested reader can use the artifact to experiment with various eureka-style applications and custom **Eureka** variants in the EuPM.

1998 ACM Subject Classification D.1.3 [Programming Techniques]: Concurrent Programming – Parallel Programming

Keywords and phrases Async-Finish Model, Delimited Continuations, Eureka Model, Parallel Programming, Speculative Parallelism, Task Cancellation, Task Termination

Digital Object Identifier 10.4230/DARTS.1.1.6

Related Article Shams Imam and Vivek Sarkar, “The Eureka Programming Model for Speculative Task Parallelism”, in Proceedings of the 29th European Conference on Object-Oriented Programming (ECOOP 2015), LIPIcs, Vol. 37, pp. 421–444, 2015.

<http://dx.doi.org/10.4230/LIPIcs.ECOOP.2015.421>

Related Conference 29th European Conference on Object-Oriented Programming (ECOOP 2015), July 5–10, 2015, Prague, Czech Republic

1 Scope

The artifact includes the source code of our Java-based library implementation of the different **Eureka** variants and benchmarks. Our implementation is based on the cooperative runtime [1] of Habanero-Java library [2]. The artifact does not include the source code of the Kilim-based [4] cooperative runtime which supports cooperatively aborting tasks. The runtime is instead available as a binary dependency that is automatically downloaded by Maven during the build process.

The artifact is designed to support repeatability of all the experiments of the companion paper [3]. In addition, we encourage users to implement their eureka-style applications and other **Eureka** variants using our **Eureka** runtime. The EuPM has also been taught in the introductory parallel programming class for second-year undergraduate students at Rice University (COMP 322). Additional documentation and code examples on the EuPM, are available in the course lecture and lab materials at <http://wiki.rice.edu/confluence/display/PARPROG/COMP322>.

2 Content

The artifact is a Maven [5] project customized to include the Kilim bytecode weaving to support delimited continuations while building the project. The project includes:



© Shams Imam and Vivek Sarkar;
licensed under Creative Commons Attribution 3.0 Germany (CC BY 3.0 DE)

Dagstuhl Artifacts Series, Vol. 1, Issue 1, Artifact No. 6, pp. 06:1–06:2



DAGSTUHL ARTIFACTS SERIES
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

06:2 The Eureka Programming Model for Speculative Task Parallelism (Artifact)

- a README file with general instructions;
- a Maven `pom.xml` file;
- source code for implementations and unit tests of the Eureka variants;
- source code for benchmarks;
- source code for programs to generate scripts to run the benchmarks; and
- source code for programs to process the data produced by running the benchmarks.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). The artifact has also been made available in the Rice's server at <http://shams.web.rice.edu/papers/ecoop15-eureka-artifact.zip>.

4 Tested platforms

The artifact works on any platform running a Java 7 compliant JDK. The results of the paper have been verified on the IBM POWER7 and AMD Opteron 6276 processors.

5 License

EPL-1.0 (<http://www.eclipse.org/legal/epl-v10.html>)

6 MD5 sum of the artifact

534d17ab271f3a88f4fcc01f1543192

7 Size of the artifact

344 KB

Acknowledgements. We are grateful to the anonymous reviewers, Prasanth Chatarasi and Deepak Majeti for their comments and suggestions on improving the artifact.

References

- 1 Shams Imam and Vivek Sarkar. Cooperative Scheduling of Parallel Tasks with General Synchronization Patterns. In *ECOOP '14*, 2014.
- 2 Shams Imam and Vivek Sarkar. Habanero-Java Library: a Java 8 Framework for Multicore Programming. In *PPPJ '14*, 2014.
- 3 Shams Imam and Vivek Sarkar. The Eureka Programming Model for Speculative Task Parallelism. In *ECOOP '15*. LIPIcs, 2015.
- 4 Sriram Srinivasan and Alan Mycroft. Kilim: Isolation-Typed Actors for Java. In *ECOOP '08*, 2008.
- 5 The Apache Software Foundation. Welcome to Apache Maven. <http://maven.apache.org>.