Next Generation Asynchronous Adaptive Specialization for Data-Parallel Functional Array Processing in SAC

Accelerating the Availability of Specialized High Performance Code

Clemens Grelck and Heinrich Wiesinger

University of Amsterdam
Informatics Institute
Science Park 904
1098XH Amsterdam, Netherlands
c.grelck@uva.nl h.m.wiesinger@student.uva.nl

Abstract

Data-parallel processing of multi-dimensional functional/immutable arrays is characterized by a fundamental trade-off between software engineering principles on the one hand and runtime performance concerns on the other hand. Whereas the former demand code to be written in a generic style abstracting from structural properties of arrays as much as possible, the latter require an optimizing compiler to have as much information on the very same structural properties available at compile time. Asynchronous adaptive specialization of generic code to specific data to be processed at application runtime has proven to be an effective way to reconcile these contrarian demands.

In this paper we revisit asynchronous adaptive specialization in the context of the functional data-parallel array language SaC. We provide a comprehensive analysis of its strengths and weaknesses and propose improvements for its design and implementation. These improvements are primarily concerned with making specializations available to running applications as quickly as possible. We propose four complementary measures to this effect. Bulk adaptive specialization speculatively waits for future specialization requests to materialize instead of addressing each request individually. Prioritized adaptive specialization aims at selecting the most profitable specializations first. Parallel adaptive specialization reserves multiple cores for specialization and, thus, computes multiple specializations simultaneously. Last but not least, persistent adaptive specialization preserves specializations across independent program runs and even across unrelated applications.