

UC Berkeley

UC Berkeley Previously Published Works

Title

Foreword

Permalink

<https://escholarship.org/uc/item/9g47786h>

ISBN

9781509052141

Authors

Morris, K
Yelick, KA
Zheng, Y
[et al.](#)

Publication Date

2017-01-30

DOI

10.1109/PAW.2016.004

Peer reviewed

Foreword

In an effort to enable extreme-scale computing, system architectures and technologies are undergoing dramatic transformations, involving complex memory hierarchies, hardware heterogeneity, many- and multi-core architectures, and different levels of parallelization. In order to effectively exploit these transformations when producing scalable applications, scientific researchers must increasingly have cross-cutting technical expertise in hardware, software, and algorithm development. Given this landscape, programming models that provide scientific researchers with a more effective approach for developing their parallel applications are of the utmost importance.

The partitioned global address space (PGAS) parallel programming model effectively combines the productive data reference semantics of shared memory systems with a strong locality model in which programmers can reason about the mapping of data to distributed memories. This combination of features simplifies programming complexity while enhancing performance by supporting shared namespaces in a way that exposes data locality in support of scalability. Many PGAS models rely on the single-program, multiple-data (SPMD) programming model commonly used for distributed memory programming. Others provide more of a global view of execution in which language concepts are used to map data and control structures to the target nodes.

There are many programming languages within the PGAS family, including Fortran 2008, Unified Parallel C (UPC), X10, and Chapel. Such languages provide users with rich syntax for expressing locality-aware parallel computations. By embedding PGAS concepts into a language's type system and execution semantics, compilers can be leveraged to help with error checking and optimization. PGAS models also take the form of meta-languages and libraries, such as Unified Parallel C++ (UPC++), Coarray C++, OpenSHMEM, MPI-3, and Global Arrays. These have the benefit of being integrated with existing languages, simplifying the learning curve for existing programmers.

PGAS Applications Workshop (PAW) includes work from case studies of PGAS programming models in the context of real-world applications as a means of better understanding practical applications of PGAS technologies. The work characterizes the scalability and performance, expressiveness and programmability, as well as any downsides or areas for improvement in existing PGAS models.

Karla Morris - Sandia National Laboratories
Katherine A. Yelick - Lawrence Berkeley National Laboratory
Yili Zheng - Google
Salvatore Filippone - Cranfield University
Bradford L. Chamberlain - Cray Inc.
Bill Long - Cray Inc.