Referee Report

Paper Number:
Reviewer's Name: Jesse Bannon
Name of Paper: MetaMorph: A Library Framework fo Interoperable Kernels on Multi- and Many-core Clusters
Author(s): Ahmed Helal, Paul Sathre, Wu-chun Feng

Section I. Overview

A. Reader Interest

1. Which category describes this manuscript?

(x) Practice / Application / Case Study / Experience Report
( ) Research / Technology
( ) Survey / Tutorial / How-To

2. How relevant is this manuscript to the readers of this periodical? Please explain your rating under IIIA.

( ) Very Relevant
(x) Relevant
( ) Interesting - but not very relevant
( ) Irrelevant

B. Content

1. Please explain how this manuscript advances this field of research and / or contributes something new to the literature. Please explain your answer under IIIA. Public Comments.

2. Is the manuscript technically sound? Please explain your answer under IIIA. Public Comments.

( ) Yes
(x) Appears to be - but didn't check completely
( ) Partially
( ) No

C. Presentation

1. Are the title, abstract, and keywords appropriate? Please explain your answer under IIIA. Public Comments.

(x) Yes
( ) No

2. Does the manuscript contain sufficient and appropriate references? Please explain your answer under IIIA.

(x) References are sufficient and appropriate
( ) Important references are missing; more references are needed
( ) Number of references are excessive

3. Does the introduction state the objectives of the manuscript in terms that encourage the reader to read on? Please explain your answer under IIIA. Public Comments.

( ) Yes
( ) Could be improved
(x) No

4. How would you rate the organization of the manuscript? Is it focused? Is the length appropriate for the topic? Please explain your answer under IIIA. Public Comments.

( ) Satisfactory
(x) Could be improved
( ) Poor

5. Please rate and comment on the readability of this manuscript. Please explain your answer under IIIA.

(x) Easy to read
( ) Readable - but requires some effort to understand
( ) Difficult to read and understand
( ) Unreadable

Section II. Summary and Recommendation

A. Evaluation

Please rate the manuscript. Please explain your answer under IIIA. Public Comments.

( ) Award Quality
( ) Excellent
( ) Good
( ) Fair
(x) Poor

B. Recommendation

Please make your recommendation. Please explain your answer under IIIA. Public Comments.

( ) Accept with no changes
( ) Author should prepare a minor revision
( ) Author should prepare a major revision for a second review
(x) Reject

Section III. Detailed Comments

A. Public Comments (these will be made available to the author)

Explanation for the Recommendation

MetaMorph aims to solve one of the most inherently difficult problems in HPC. However, there is not enough information to support its practical use, which is why this paper has been rejected. MetaMorph's main objective to abstract compute accelerators fails to address how it will continue to stay updated with future parallel architectures. Any developer interested in this API needs reassurance that MetaMorph has the ability to evolve with new hardware and backends.

Another issue with the presentation of MetaMorph is that it was too focused on fluid dynamic related problems, where parallelism can be described as a grid that must communicate with neighboring partitions. Describing a library framework should not be limited to a single subset of distributed problems. Graph and sparse linear algebra problems are examples of inherently different problems in terms of load balancing and communication, and should be mentioned somewhere in the text and how to address these types of problems when using MetaMorph.

Lastly, the performance results fail to characterize the affect MetaMorph's middleware has on runtime. The introduction states MetaMorph is designed to extract as much computational capability as possible from exascale computing systems. Comparing only the computational portion of MetaMorph to a serial implementation does not provide the reader a fair assessment of performance loss compared to a GPU or multithreaded implementation on the same hardware. There should be performance metrics that compare a MetaMorph's configuration (CUDA, OpenMP, OCL-AMD) with its respective counterpart designed specifically for that architecture.

Summary of the Paper and Assessment

MetaMorph is a middleware technology which aims to homogenize the use of accelerator technologies such as GPUs, Intel MICs, multiprocessors, and FPGAs within a cluster into a single framework. With rapidly evolving parallel architectures, developers can not afford to continually rewrite software to adapt. MetaMorph aims to abstract this concept while still being able to guarantee performance.

What the paper immediately fails to address is how MetaMorph will continue to adapt to new hardware and architectures. The problem being addressed relieves the developer of translating their codes to new architectures, but someone is still responsible for keeping MetaMorph up to date. Developers interested in MetaMorph need clearer reassurance that MetaMorph is capable of evolving to new technologies.

There are three layers within MetaMorph. From top to bottom includes the adaptivity, abstraction, and interoperability layer. These layers aim to modularize performance critical code into separate shared object libraries which can be translated to any hardware accelerator.

MetaMorph includes basic communication functions similar to MPI, and a family of functions related to 'grid' computation, where each process owns performs computation and must communicate with its neighbors. This type of problem is emphasized throughout the rest of the paper including their case study and performance results. This is a common problem structure in HPC, however, it is unfair to asses an API using a single problem type which includes specialty functions without considering other types of problems.

Lastly, the experimental results fail to demonstrate the effect MetaMorph has on performance. All results do not include inter-node data transmission time. This is, in many cases, one of the most expensive portions of a distributed algorithm. The impact middleware has on this crucial step can have serious implications to performance. Additionally, the only non-MetaMorph performance metric was ran serially on an Intel Xeon E5-2697. It is unfair to compare only that against MetaMorph using GPUs and OpenMP. Again, these metrics fail to compare the impact of MetaMorph's middleware when using the same hardware configurations.