Preface
DFM 2014

The Fourth International workshop on “Data-Flow Models (DFM) for extreme scale computing” was held in Edmonton, Canada, on the 24th of August 2014 in conjunction with PACT 2014. Its purpose was to bring together researchers interested in novel computational models based on Data-Flow/Data-Driven principles of execution.

In the past five decades, sequential computing has continued dominating the computer architecture landscape because designers were successful at building faster and faster computers by solely relying on improvements on fabrication technologies and architectural/organization optimizations. It is only its inability to tolerate long memory latencies that has slowed down the performance gains (this phenomenon is known as the Memory Wall). While various mechanisms have been implemented to overcome this wall (e.g., efficient hardware prefetch support), they only led to yet another “wall” which weakens the efficiency of modern chips: the Power Wall. This is why power considerations and heat dissipation issues have forced manufacturers to switch to multiple cores per chip and thus move into the concurrency era. New concurrent models/paradigms are needed in order to fully utilize the potential of multi-core chips. The switch to multi-core systems has significantly raised potentially concurrency, at the cost of introducing the major new difficulty of efficiently exploiting this available parallelism.

The Data-flow model is a formal model that can handle concurrency and tolerate memory and synchronization latencies. Data-Flow inspired systems can also be simpler and more power efficient than conventional systems. Recent work on Data-flow inspired systems has shown that the Data-Flow principles can be used to develop data-driven systems that can perform as well as, and in some cases outperform, systems based on conventional techniques, but running on commercial multi-core systems. Thus, it is time to revisit the research on Data-driven computation and bring it to the multi-core and extreme scale computing.

Professor José Nelson Amaral from the University of Alberta, Canada, delivered the Keynote speech “Relating the Partitioned-Global-Address-Space (PGAS) Programming Model with the Data-Flow Model” in which he reminded the audience of some historical points about the data-flow model and showed the evolution of the Partitioned-Global-Address-Space (PGAS) programming model, where the programmer typically specifies the distribution of data to address partitions, the tasks to be executed, the mapping of these tasks to the partitions, the coherence operation for shared data, and the synchronization amongst the tasks.

The first paper of Session 1 proposes a new framework for multi-core architectures called SPARTA (Stream-based Processor and Run-Time Architecture), and its program execution model PXM. The next paper presents a perspective on Swift as a dataflow language, applied as a top-level language useful for constructing large and complex applications for certain classes of parallel systems in which low-level parallelism is handled by pre-optimized libraries that match structured data to traditional architectures. In the third paper, the authors propose an alternative virtual memory system based on codelets and the 'fresh breeze' memory model, based on trees of fixed chunks of memory, each having a global handle. Finally, in the last paper of the session, Hierarchically Tiled Array are shown to be a good fit for data-flow computing in general and codelets in particular in the context of exascale computing.

Paper 1 of Session 2 proposes a configurable architecture for Exascale computing based on the dataflow model. Then, an extension to the dataflow Codelet model is shown that will support different application
demands and different types of hardware (e.g., accelerators). The StreamIt and ΣC Languages, two statically-scheduled stream languages, are compared and contrasted in the next paper and a method and tool to transform codes from StreamIt to SigmaC is described. Concrete proposals are presented to improve the performance and flexibilities of the languages.

In the last session (Session 3), the first paper compares the performance and hardware cost of dynamically scheduled static dataflow execution and statically scheduled static dataflow execution by implementing two models in FPGA design and simulating them with an RTL simulator for FPGA-based design. The second paper introduces a new intermediate representation for macro dataflow, DFGR. Essentially, it is a structured way of representing dataflow dependencies, using collections of dependencies relating primitive computation (steps) and communication (items). The last paper is a case study of the implementation of the fast multipole (FMM) method with various parallel execution models. After presenting a high-level view of the FMM algorithm and possible implementations, it discusses different ways to partition the input set.

This workshop and these proceedings are meant to appeal to researchers and practitioners interested in the design of new models of execution. We feel that data-flow principles are to be vital in the design not only of multi-core systems, but also in the much heralded Exascale systems. Therefore, we expect that this will be only the beginning of an exciting thread of research and we look forward to much successful work along these lines.

We thank the PACT 2014 organizing committee and the general Chair Jose Nelson Amaral in particular for their generous support of the workshop and thus allowing us to run a successful meeting.

We also thank all contributing authors, as well as the attendees, who have made this workshop quite an enjoyable event. Finally, we wish to thank Professor Amaral again for his inspiring keynote.