

# Distributed Configuration Management for Reconfigurable Cluster Computing

Aju M. Jacob, Ian A. Troxel, and Alan D. George

High-performance Computing and Simulation (HCS) Research Laboratory  
Department of Electrical and Computer Engineering, University of Florida  
Gainesville, Florida 32611-6200

## 1. Introduction

Based on commodity parts and emerging technologies, reconfigurable systems for high-performance computing have the potential to achieve performance levels far exceeding those of conventional systems while maintaining advantages of cost, interoperability, etc. associated with industry standards and off-the-shelf components. In these systems, speedup and offload of various computation and communication tasks will be provided through dynamic hardware reconfiguration, with FPGA and CPU devices working in tandem to solve key tasks in a collaborative fashion. One of the limiting factors in building and exploiting such systems is the lack of a simple and effective management framework for applications, system services, programming models, and middleware within a broad range of platforms and tools. This paper focuses on concepts and options for Configuration Management (CM) within a new framework under development at Florida called CARMA (Comprehensive Approach to Reconfigurable Management Architecture) [1].

## 2. Distributed CM Design and Experimental Setup

Upon receiving a configuration request, the local CM on a given node checks its local configuration cache. If a cache miss, the local CM communicates with one or more remote CMs to obtain the configuration over the data network. CMs also manage FPGA resource access, configuration relocation, transformation, defragmentation, and caching. The CM in CARMA uses a Board Interface Module (BIM) to provide abstracted user access to RC boards and hardware independence to higher layers. To achieve a fault-tolerant CM service that scales, the CARMA CM is fully distributed. Four distributed CM schemes are proposed:

*Master-Worker (MW)*, *Client Server (CS)*, *Client Broker (CB)* and *Peer-to-Peer (PP)*; MW and CS are featured here.

Experiments have been conducted on a reconfigurable cluster of dual-Xeon servers with Tarari CPX2100 HPC boards in 64/66 PCI slots. Servers are connected by a 5.3 Gb/s data network (SCI) in a 2D torus topology and by a 1.0 Gb/s control network (Gigabit Ethernet). The focus is to determine overhead imposed on the system by a distributed CM, study tradeoffs, and provide a quantitative comparison between the MW and CS schemes. The performance metric used to compare these schemes is completion latency, the time interval starting when a configuration request is received at a local CM and ending when the configuration is loaded on the FPGA and thus fully prepared for execution.

## 3. Results and Conclusions

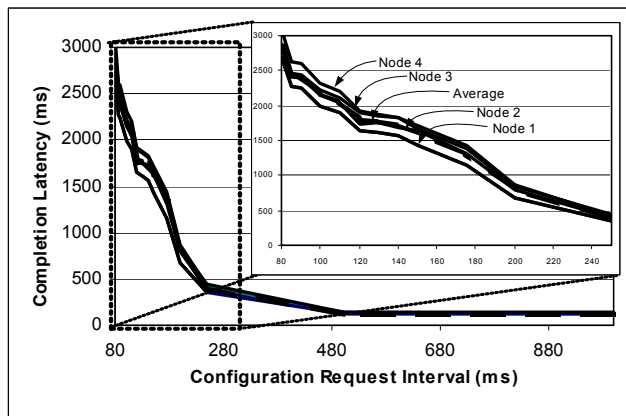
This study presents performance results for the MW and CS schemes in a head-to-head comparison on four nodes and shows CS outperforms MW in the region of interest (Figure 1). The distributed CM is found to impose minimal overhead on the system (i.e. less than 1% of completion latency in the worst case). A significant variability is observed between completion latencies of client nodes in the CS scheme due to client proximity to the server on the data network, a condition not experienced by MW nodes.

## 4. Acknowledgments

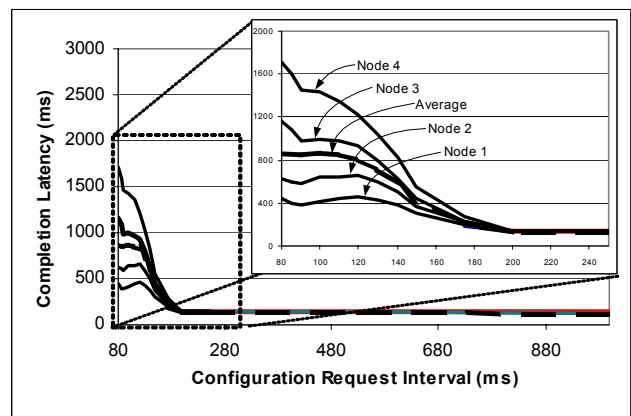
This work was sponsored in part by the U.S. Dept. of Defense, and supported by equipment and tools from Alpha Data, Celoxica, Cisco Systems, Dolphin, Intel, and Tarari.

## 5. References

- [1] Reconfigurable Computing Group Web Site at U. Florida [online], <http://www.hcs.ufl.edu/prj/rcgroup/teamHome.php>.



a) Master-Worker



b) Client-Server

Figure 1. Average Completion Latency vs. Configuration Request Interval