# LaRCS: A Language for Describing Parallel Computations for the Purpose of Mapping *

Virginia M. Lo†, Sanjay Rajopadhye‡, Moataz A. Mohamed§,
Samik Gupta, Bill Nitzberg, Jan Arne Telle, Xiaoxiong Zhong
Dept. of Computer and Information Science
University of Oregon
Eugene, Oregon 97403-1202
email: lo@cs.uoregon.edu

**Abstract**

LaRCS is a graph description language which enables the programmer of parallel algorithms to specify information about the static and temporal communication behavior of parallel algorithms. The information contained in a LaRCS program is used for the mapping problem: to assign tasks in the parallel computation to processors, and to route inter-task messages along the links of the interconnection network. Many practical algorithms exhibit regular communication patterns and LaRCS provides a mechanism for describing this regularity in a compact, parameterized manner. Static communication topology is expressed in LaRCS code through node labels and simple communication functions on the nodes. Temporal communication behavior is represented by a notation we have developed called *phase expressions*.

This paper introduces the LaRCS language and compares LaRCS with several existing graph description languages. We introduce a new graph theoretic model of parallel computation based on LaRCS called the **T**emporal **C**ommunication **G**raph; and we show how information provided by LaRCS plays an important role in mapping.

*This research was sponsored by Oregon Advanced Computing Institute (OACIS), a consortium of academic, industrial, and government agencies in the state of Oregon.

†Partially supported by NSF grant CCR-8808532

‡Partially supported by NSF grant MIP-8802454

§Moataz Mohamed is currently a member of Computer Systems Lab, Stanford University

**Keywords**: configuration language, graph description language, regular parallel computation, mapping, parallel programming environments.

# Contents

# 1 Introduction

The *mapping problem* in message-passing parallel processors involves the assignment of tasks in a parallel computation to processors and the routing of inter-task messages along the links of the interconnection network. Most commercial parallel processing systems today rely on manual task assignment by the programmer and message routing that does not utilize information about the communication patterns of the computation. The goal of our research is *automatic* and *guided* mapping of parallel computations to parallel architectures in order to achieve portability and maximal performance from parallel software.

We have developed a unified set of software tools for automatic mapping of parallel computations to parallel architectures. Our system is called OREGAMI[1]. Two key features of our system are (a) the ability to take advantage of the regularity present in both the computation structure and the interconnection network and (b) the desire to balance the user's knowledge and intuition with the computational power of efficient combinatorial algorithms.

In this paper, we focus on one component of the OREGAMI system called **LaRCS** (**La**nguage for **R**egular **C**ommunication **S**tructures). LaRCS is a description language which enables the programmer of parallel algorithms to specify information about the static communication topology and temporal communication behavior of the computation to be mapped. The design of LaRCS was driven by the observation that many parallel computations exhibit regularity in both static communication topology and temporal communication patterns. LaRCS can be used to capture this regularity for use by OREGAMI's mapping algorithms.

Fig. 1 illustrates the relationship of LaRCS to the components of the OREGAMI system, MAPPER and METRICS. MAPPER, our mapping software, is a library of algorithms which map the computation to the architecture. The specific algorithm that is invoked depends on the information provided by the LaRCS description. METRICS is an interactive graphics tool which displays the mapping along with a range of performance metrics reflecting load balancing, communication contention, and communication overhead.

LaRCS is designed for use in conjunction with parallel programming languages in

[1]For University of **OREG**on's contribution to the elegant symmetric structures (contractions) produced by ori**GAMI** paper folding.