SOFTWARE METRICS FOR PARALLEL PROGRAMS

Imtiaz Ahmad and Mansur H. Samadzadeh

Computer Science Department, Oklahoma State University, Stillwater, OK 74078

1. Introduction

Software engineering has several fields one of which is the study of static measurements of programs as indicators of their maintainability, reliability, and clarity. Several metrics have been proposed and evaluated for qualitatively measuring the conceptual complexity of sequential programs. Corresponding metrics for parallel software are noticeably absent. This spawned the need to develop parallel software metrics. There are several ways to explore the feasibility and plausibility of parallel software metrics. One way is by proposing new metrics exclusively developed for parallel software. Another way is by applying the existing software metrics, originally proposed for sequential programs, and analyzing their applicability. We have adopted both ways in this study.

The test suite for this study is a compendium of parallel programs [2]. The compendium contains over 30 programs ranging in length from 500 to 14,000 lines, written by graduate students. All programs in the compendium were written in C on Intel's iPSC/1 or iPSC/2 parallel computers.

2. Software Metrics

In this study we considered five metrics: Size [3], Halstead Software Science [4], McCabe's cyclomatic complexity V(G) [5], Samadzadeh and Edwards' residual complexity [7], and the newly proposed communication complexity. What follows is a brief discussion of the newly proposed communication complexity metric. In an attempt to modify the cyclomatic complexity for a parallel algorithm, we can consider all message passing/receiving commands as virtual conditional statements. This assumption can be intuitively supported by the argument that for any message passing/receiving command the program control jumps to another location, thus increasing the difficulty of comprehending the program.

3. Experimental Methodology

The framework for experimentation defined by Basili et al. [1] has been used in the design of this study. A questionnaire-based subjective evaluation of the static code of parallel programs in the compendium was planned. The programs were also objectively evaluated using the metrics mentioned earlier. Correlation analysis was used to study possible relationships between the chosen metrics and the experts' perception of relative comprehensibility. The collection of data was partially made possible with the help of PC-METRIC [6], the rest of data collection was achieved by programs developed on iPSC/2. To get the experts' perception of the relative complexity of the applications, a questionnaire was devised. The questionnaire has 10 questions. The first four questions evaluate the expertise level of the person who is evaluating the programs in the compendium. Questions 5 through 10 deal with the experts' perception regarding each application. The final analysis is currently under way based on various statistical tests.

4. References