Proving Correctness of a Controller Algorithm for the RAID Level 5 System

Mandana Vaziri* Nancy Lynch[a] Jeannette M. Wing

March 1998
CMU-CS-98-117

[a]MIT Lab. for Computer Science, 545 Technology Square, Cambridge, MA 02139

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This paper will appear in the *Proceedings of the International Symposium on Fault-Tolerant Computing*, 1998.

This research is sponsored in part by the Defense Advanced Research Projects Agency and the Wright Laboratory, Aeronautical Systems Center, Air Force Materiel Command, USAF, F33615-93-1-1330, and Rome Laboratory, Air Force Materiel Command, USAF, F306602-97-2-0031, and in part by the National Science Foundation under Grant No. CCR-9523972. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency Rome Laboratory or the U.S. Government.

Keywords: formal methods, RAID systems, formal specification, verification, correctness proofs, I/O automata

Abstract

Most RAID controllers implemented in industry are complicated and difficult to reason about. This complexity has led to software and hardware systems that are difficult to debug and hard to modify. To overcome this problem Courtright and Gibson have developed a rapid prototyping framework for RAID architectures which relies on a generic controller algorithm. The designer of a new architecture needs to specify parts of the generic controller algorithm and must justify the validity of the controller algorithm obtained. However the latter task may be difficult due to the concurrency of operations on the disks. This is the reason why it would be useful to provide designers with an automated verification tool tailored specifically for the RAID prototyping system. As a first step towards building such a tool, our approach consists of studying several controller algorithms manually, to determine the key properties that need to be verified.
This paper presents the modeling and verification of a controller algorithm for the RAID Level 5 System. We model the system using I/O automata, give an external requirements specification, and prove that the model implements its specification. We use a key invariant to find an error in a controller algorithm for the RAID Level 6 System.