Fault Tolerant Parallel Software for MPPs by Converting Parallel Logic Programming Languages

Eiji Sugino
School of Information Science,
Japan Advanced Institute of Science and Technology

January 16, 1997

Abstract

Parallel computers are recently becoming available to many applications day by day. Since parallel computers are aimed to provide high performance, there is not enough investigation about fault tolerance in this area. Parallel application software need high performance and then long time to be solved essentially, nevertheless they are now running on a tight-rope.
The other hand, fault tolerant computers are constructed with special facilities, demand high prices, and put you an obligation to write dedicated fault tolerant program in return for efforts. It requires heavy burden to write such program, and parallelism make matters worse.
This thesis proposes a method converting programs to construct a fault-tolerant parallel software (FTPS) for MPPs without any dedicated facilities. Since to write FTPS by hand requires heavy burden for programmers, The author provides a mechanism which automatically converts original parallel programs into FTPS. This FTPS runs on MPPs based on the primary site approach. Any number of processors constructing sites is allowed to users and the user's programs can run on each site in parallel.
The target is currently a parallel logic programming language, but the method will be applicable to other parallel programming languages.
In this thesis, the author analyzes execution overhead of FTPS from experimentations using nCUBE2 since the overhead is crucial for applicability of the method. When nondeterministic portions in a program is limited, the overhead can be kept in low level. Even if a program has high nondeterminisity, the overhead can be decreased using high parallelism in it.
The author also shows that the MTTF of the system grows at a logarithmic rate by the number of sites. It means that the method improves the system availability by longer MTTF compared with the restriction of performance for fault tolerancy.

Key Words: **fault-tolerant software, parallel logic programming languages, primary site approach**