Abstract

N-version programming is a fault tolerance technique that depends on a generic decision algorithm to determine a consensus result from the results delivered by two or more member versions of the software. In N-version programming, N teams of developers work independently on N unique but equivalent implementations of the same program. The major objectives of the NVP process are to maximize the independence of version development and to employ design diversity in order to minimize the probability that two or more member versions will produce similar erroneous results that coincide in time for a decision (consensus) action. But this fault-tolerance technique has been criticized for its statistical assumptions and high cost. A solution is proposed in which there are N versions of the software out of which t versions implement only a subset of the entire functionality which is highly critical while (N – t) versions implement the entire functionality. One of the biggest hurdles in using N version programming for fault tolerance is its high implementation cost. This proposed technique minimizes the cost of implementation while improving the efficiency and reliability of the system.
References


Index Terms

Computer Science
Programming Languages

Keywords

N version programming, fault tolerance, reliability, real time systems