RELIABILITY AND INTEGRITY OF DISTRIBUTED COMPUTING SYSTEMS

by B. Randell

Computing Laboratory, University of Newcastle upon Tyne,
Newcastle upon Tyne, UK

Abstract

A brief outline is provided of recent work in an on-going research project sponsored by the UK Science Research Council.

Outline

The specific problem that we are addressing is that of designing distributed computing systems which can cope effectively with situations in which it is beliedly detected that erroneous data has been allowed to spread from computer to computer in the system. It is now widely recognised that the complexity of large computing systems is a major cause of unreliability. We have therefore sought system structuring techniques which would allow fault tolerance measures to be incorporated in a system without undue increase in its complexity. One technique we have based much of our work on uses the idea of structuring a system into a hierarchy of interfaces (or levels of abstraction). This approach involves extending a given (possibly hardware) interface through the provision of programs (or object managers) which implement new abstract objects and provide operations to manipulate these objects.

Our early work addressed the provision of backward error recovery in multi-level systems, since this form of recovery has the attractive property of being able to provide automatic recovery even from unanticipated design faults. In a distributed system the abstract objects available on an interface may have concrete representations on a number of separate systems (or nodes). We have been investigating the structuring of a distributed system so that backward error recovery can be achieved for such objects. While a program can directly access the object managers which reside on the same (local) node, object managers on remote nodes have to be accessed indirectly by sending a message to a process executing on the required node. Our model of recoverability in multi-level systems provides us with a simple and coherent method for achieving recovery in such situations.

These ideas form the basis for the current project's first major implementation experiment. This experiment has involved providing the abstraction of a recoverable (and distributed) file system to a UNIX process, replacing the standard UNIX file system interface, in a configuration involving several PDP-11s connected using a Cambridge ring.

Multiple levels, each composed of one or more object managers, constitute a structuring of the actual system. They imply a corresponding structuring of the system's activity. Such a structuring is provided by the notion of an atomic activity, i.e. a set of operation invocations constituting some portion of a larger activity which, from the viewpoint of the rest of the activity, happens instantaneously. Atomic activities can be pre- and post-planned (and thus involve synchronisation actions and possible delays) or can arise dynamically. The experimental distributed file system provides backward error recovery based on the use of pre-planned atomic activities. The problem with dynamically arising atomic activities is finding them when they are needed. This is the problem that our basic work on the chase protocols addressed. This work has been extended to cover the decentralised provision of recovery control in a distributed system in which communicating processes establish and commit recovery points asynchronously with respect to the other processes, and never hold up their execution waiting for the validation of data from another process.

Another major area of work has concerned programming issues related to exception handling. A formal definition has been given of the correctness and robustness, with respect to their specifications, of programs incorporating exception handling. Verification conditions have been derived for an acceptance test to be "precise" in the sense that it evaluates to false when and only when an exception should occur, and heuristics given for the design and placement of such tests.

Further details of these, and a number of other project activities, are to be found in [1] and various published papers.

References