# Implementing The APR Technique on Loosely coupled Multiprocessor system

Su-Nyung Huh

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

Feb.15 2000

**Keywords:** Dependable Software, Fault Tolerant Software, Replication,APR, FTAG.

There are many application domains where computer systems perform life-critical tasks. For such tasks, where failures can lead to catastrophes, the dependability of the computer systems responsible for these tasks is of up-most importance. Such systems include patient monitoring systems, nuclear power plant control systems, flight control systems, and traffic control systems. For some other applications that are being increasingly dependent on computer systems, such as telephone system, banking systems, stock exchange systems, etc., failures of computer systems can lead to great financial losses.

To justify our dependability on such systems, it is necessary for these systems to correctly perform their operations and to deliver the correct results even when the system experiences failures during its operation. In other words, these systems need to provide for fault-tolerance.

There are two approaches for enhancing the dependability of computer systems. One of these approaches is the fault avoidance approach. The goal of this approach is to prevent faults from occurring or getting introduced in the system by using for example highly reliable components. In this approach, no redundancy is introduced in the system. The other approach is the fault-tolerance approach. The goal of this approach is to provide correct services despite the presence of faults in the system. Though several techniques have been proposed for fault-avoidance, it is impossible to prevent faults from occurring or being introduced in the system. Thus, fault tolerance techniques are required for the development and implementation of highly dependable computer systems.

The dependability of computer systems received much attention in the literature. Much of the research work in the literature focused on providing and implementing dependability at the hardware level. With the increasing complexity of computer systems, the design and implementation of dependability at the software and middleware level become of primary importance in today and future computer systems.

The implementation of dependability at the software level is a relatively new research field. Its goal is to implement fault tolerant and dependable systems using software components. Several techniques have been proposed for implementing fault tolerant systems at the software level. Replication techniques proved to be one of the most powerful techniques for implementing such systems. Such techniques are widely used in implementing todays fault tolerant systems.

The active parallel replication technique (APR), has been proposed as a replication and replica management technique for implementing fault tolerant software. The APR technique is a highly efficient and flexible technique. It improves the efficiency of replication by reducing the cost induced by the redundancy introduced in the system. The APR technique makes an efficient use of parallel computing in order to reduce the overall computation time.

The APR technique, as proposed in the literature, lacks an in-depth study and investigation on the behavior and fault coverage of the technique in case of failures. Moreover, only a general discussion and architecture for the implementation of the technique on a loosely coupled multi processor system has been proposed. The modeling and detailed design of such an implementation has not been proposed in the literature.

In this research work, we studied the APR technique and identified a number of problems that can arise in different failure scenarios. We proposed a refinement of the APR technique to effectively address these problems. In particular we designed an Adaptive Computation Management Scheme for earlier failure detection and recovery. We validated the proposed scheme using Gantt Charts. We also developed a number of execution examples and showed that the approach proposed in this paper is effective.

Moreover, a detailed design for the implementation of the APR technique has been developed. We proposed a highly modular architecture and structure for the APR runtime system. We are using the functional programming language Objective Caml and the Ensemble Group Communication layer for the implementation of the APR technique.