Building a Hierarchical Defense:

An Immune System Paradigm for the Design of Fault-Tolerant Systems

Algirdas Avižienis and Rimas Avižienis

A. Avižienis and Associates, Inc.
Santa Monica, CA 90403, USA

IFIP WG 10.4
5 January 2002
St. John, US VI

The Challenge #1

In the first 50 years of our computer age,
hardware has not been adequately exploited
(not even close to its full potential)
to assure system dependability or survivability.

Key weaknesses of contemporary COTS systems:

1. Hardware defenses are commingled with software at various levels of design hierarchy
2. Unprotected “hard core” elements exist in both hardware and software
3. There is no built-in support for tolerating design faults: software “bugs” and hardware “errata”

CAUTION: They will get worse in emerging technologies!

Current Fault Tolerance Solutions

1. Clustering of complete systems, using COTS software (Microsoft Cluster Service, Extreme Linux, etc.)

Problems: long recovery time, cost of complete replication

2. Hot standby duplexing of critical subsystems (CPU, power supply, etc.), plus ECC, RAID, N+1 sparing, etc., for others

Problems: There remain unprotected “hard core” elements

An example: Ziatech high availability architecture

3. Intelligent Platform Management (IPM) provides hardware and firmware for monitoring, plus software for inventory and logging

Problems: hard to make the IPM elements fault-tolerant (the IPM Interface Spec. has 395 pages)

An example: Pentium II Xeon Server Platform IPM

Another Deficiency:
No Support for Multichannel Computing

1. The abandonment of Pentium and P6 processor “FRC” (master/checker) mode has left no support for comparison of two channels in hardware.

2. There is no hardware support for majority voting of three channels in COTS processors.

3. The proliferation of hardware design faults (“errata”) will necessitate tolerance of design faults by the use of N-version design diversity in hardware and software, at least in life-critical applications.

An example: Consider the Pentium III:

a) The first specification update in March 1999 listed 44 errata; of which 36 remained unfixed by May 2001.
b) 35 new errata were announced between March 1999 and May 2001; 22 of them were not fixed by May 2001.

The Direction #1 for a Long-term Solution:
A generic, hierarchical fault-tolerant hardware infrastructure.

that:

1. Can communicate to the “client” system’s software,
   but is not dependent on its support
2. Is compatible with the clients’ diverse hardware and transparent
   to the clients’ software
3. Is not susceptible to software attacks and to the clients’ design faults
4. Supports the client system’s multichannel (TMR, etc.) computing,
   including diverse hardware and software channels to provide design
   fault tolerance for the client system

Challenge # 2

Explain the concept of the fault tolerance infrastructure
by means of an analogy that is easily understood, and challenges
both fault tolerance experts and non-experts,
especially systems designers and their customers.

Direction (of Solution) # 2

Develop and employ the immune system analogy
to explain the concept of the fault tolerance infrastructure.

The Immune System Paradigm (ISP):

Why and How ?

Why: It provides a convenient analogy to explain a set of design principles for fault-tolerant systems that are based on the separations of hardware and software mechanisms for fault tolerance

How: (1) Identify the key properties of the human immune system

(2) Set up the analogies that relate the immune system to fault tolerance

(3) List the attributes that an implementation of fault tolerance must have in order to justify the immune system analogy

The Human Immune System

1. Detects and reacts to threats continuously and autonomously, independent of cognition.
2. Is distributed throughout the body, serving all organs.
3. Has own communication links-network of lymphatic vessels
4. Its cells, organs and vessels are self-defended, redundant, and in several cases, diverse

The Analogies Are:

1. The body is analogous to hardware
2. Cognitive processes are analogous to software
3. The immune system is analogous to a fault tolerance implementation, called the “f.t. infrastructure” (FTI)

To Justify the Immune System Analogy, the FTI must:

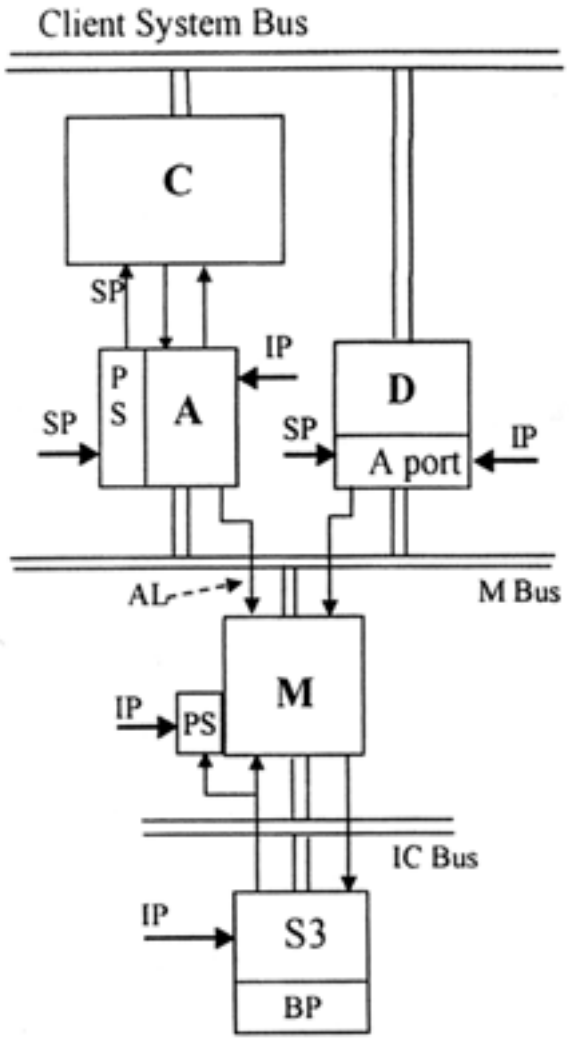
1. Consist of hardware elements only

2. Be compatible with diverse COTS hardware and be transparent to their software

3. Support multichannel computing, including diverse hardware and software channels

4. Be fully fault-tolerant itself

The Fault Tolerance Infrastructure (FTI)



SP: System Power
IP: Infrastructure Power
BP: Backup Power
PS: Power Switch
C: Computing Node
A: Adapter Node
D: Decision Node
M: Monitor Node
S3: Startup, Shutdown, Survival Node
AL: A-Line

Note: Redundant Nodes are not shown