Predictive Application Failures in Software Code Errors

"Lead Inventor: Angelos D. Keromytis, PhD.

Tech Ventures Reference: IR M04-065, IR M05-081, IR M05-082

Intelligent software application instrumentation for code failure:
Software applications are among those things we expect to be instrumented, i.e. metered and controlled. However, the state of the art remains rudimentary. To become more intelligent, software application instrumentation needs to do much more than report a failure or send out an alert. It must be able to learn about an application’s performance, break it down into segments, recover when failures happen, ‘fix itself’, and ‘know’ how to predict when something might go wrong.

Virtualized code segments mask and fix code errors while continuing software operation:
This technology includes software and processes for ‘instrumenting’ applications so that errors, unhandled by code logic, can be immediately isolated. The software then automatically adapts the applications execution, keeps its services available, and begins a process of ‘self-healing’. Once root cause for the error is identified, this technology automatically fixes the error and resumes normal application execution. This technology effectively ‘virtualizes’ code segments so that it can continue operation, and so it can emulate the recreation and errors. In this virtualized mode, it masks the effects of the error, maintains application function and availability, performs error recovery, and then returns to steady state. This is done using a recursive method for isolating an application fault, automatically emulating the fault’s conditions and sorting up the hierarchy of code segments in search of an error’s trigger. Once the error is found, it creates a ‘vaccine’ for the error and then ‘immunizes’ itself, automatically returning the application to normal operation and mapping out the error-causing condition. If the error shows indications of reappearing in the future, this code can implement preventative measures to avoid it. This technology also includes the analytical methods for determining which code segments need to be instrumented.

Applications:
• Application integrity in production
• Automated recovery due to code error
• Automated recovery due to security breach or attack

Advantages:
• High availability, fault tolerant
• Failure prediction
• Self-healing

Patent Status: US Pending (US 7,490,268) ~ see link below.
Inventors

Angelos D. Keromytis