Summary Part 1

iWatcher: Efficient Architectural Support for Software Debugging

Mrinal Nath (ID: 3307043)
April 11, 2005

What are the problems solved by this paper?

Software debugging can be difficult due to several limitations of the debugging tools. For e.g. static tools are limited by aliasing and other compile time limitations, and dynamic checkers are computationally intensive and limited by imperfect variable disambiguation. Bugs that are not caught and end up in the final software can ultimately cause a lot of economic damage. This paper proposes iWatcher, an architectural support for automatic, flexible and low overhead dynamic execution monitoring to ease the task of software debugging and help catch bugs faster.

What are the approaches attempted by this paper?

iWatcher is a 'location controlled monitoring' tool implemented using a combination of hardware (WatchFlags, Range Watch Table, main check function register, ROB and LSQ flags, etc.) and software (on/off system calls, check table, etc.). When a monitored location is accessed, the associated monitoring function is automatically triggered and executed. The monitoring function can be supplied with almost the entire state of the machine through the parameters that are passed to it by the architecture. In case of a SMT or CMP, TLS can be used to reduce the overhead incurred by speculatively executing the main program in parallel with the monitoring function. TLS can also be used to support roll-back of the program to a checkpointed state.

What are the main conclusions of this paper?

iWatcher is able to detect all accesses to a watched memory locations while other tools have some limitations. The overhead associated with iWatcher is relatively low (up to 0%). Even when 20% of the dynamic loads in a program are monitored, the overhead is only up to 174%. Due to such low overhead, iWatcher can be used to detect various types of bugs during run-time and also as a basis for interactive debugging.