Summary
iWatcher: Efficient Architectural Support for Software Debugging

Venkatesan Packirisamy
April 11, 2005

1 To be completed before class

What are the problems solved by this paper? (50 words)
Software development costs are ever increasing. There has been a lot of tools to assist in debugging. Most of them incur a huge overhead. Also most of them are code based - the program is instrumented at several points to monitor the execution. The error could occur anywhere between the two points. This is not accurate. So there is a need for an efficient and accurate mechanism.

What are the approaches attempted by this paper? (50 words)
The paper proposes a hardware based monitoring mechanism. Here code segments are attached to specific memory locations and are executed when the location is accessed. Also the code segments are executed in parallel using Thread level speculation. Because of the TLS and the hardware based monitoring, iWatcher is very efficient. Using the mechanism iWatcher is able to track single locations or a range of locations.

What are the main conclusions of this paper? (50 words)
Since iWatcher is a memory based monitoring mechanism it can detect bugs more accurately. It is shown that it takes only 5-80% execution overhead to detect most of the bugs. Even if we monitor 20% of the loads, the overhead is going to be only 66 – 174% overhead. The iWatcher was able to detect most of the common bugs like uninitialized reads and data races.

2 To be completed after class

Did this paper address an important issue? Explain. (100 words)

Are the proposed approaches valid? Describe its strength and weakness. (100 words)

Do the results support the conclusions? Explain. (100 words)

Describe the potential future works? (100 words)