Summary


Kiran Yellajyosula

April 13, 2005

1 To be completed before class

What are the problems solved by this paper? (50 words)

Debugging an application through software checks or dynamic execution monitoring slows down the execution by 10-100 times. Inserting dynamic checks to monitor code can also lead to unnecessary checks and due to aliasing variables some bugs may go undetected.

What are the approaches attempted by this paper? (50 words)

The authors propose a novel architectural scheme to support dynamic execution monitoring with low overhead. Code is instrumented through an interface to monitor a set of memory addresses and raise exceptions whenever an access to the memory location is made. The authors propose to use TLS to improve the performance.

What are the main conclusions of this paper? (50 words)

iWatcher is a novel technique which detects bugs in the software with a performance impact of about 4-80%. Employing TLS with iWatcher improves the performance by around 30%. iWatcher provides an interface which can be employed across various architectures.

2 To be completed after class

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

Debugging a program is conventionally done by use of assertions by the user or by using the compiler to instrument statically at a lot of locations or dynamically by the use of a debugger or by dynamic checkers or hardware assisted watchpoints as in Intel. All these methods reduce the speed of execution of a program by 10-100 times. The authors propose to use hardware support to monitor dynamic execution of a program, which reduces the overhead to around 80%. The hardware is triggered on or off through an interface which can function in various modes. The user can implement his/her own functions to correct or process the error.

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

Strength:

- The number of instructions being instrumented reduces.
- Aliasing problems can be easily handled as the location is used to trigger the exceptions.
- It can be used to correct the errors.
- Multiple functions can be linked to a particular address.
- Use of TLS reduces the overhead.
- The errors detected are actual errors occurring in the software.
• The technique is flexible and can be used to invoke a debugger when an error occurs.

Weakness:

• The technique uses extra hardware which increases complexity of chip construction. All the associated functions and the addresses are to be stored on the processor and cache lines are to be extended to support iWatch flags. The RWT and VWT also need to be constructed on the processor.

• TLS requires a lot more hardware and the hardware cost for performance while debugging a program is much high.

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

The authors use some benchmark codes with various types of errors added into the code to test the iWatcher based monitoring and iWatch detected all the errors. They also compared the performance with Valgrind and observed significant improvement in the overhead. The authors simulated the execution using TLS and observed further improvement in performance when the size or the number of monitoring functions increases.

Describe the potential future works? (100 words)

iWatcher based models may be extensible to multiprocessor or multithreaded programs, where interprocessor communications, deadlocks and barriers play a significant role. The iWatch interface functions may be inserted by the compiler or debugger directly by detecting an error and using rollback mode to edit the executable.

The authors proposed to test the approach for other server based applications and compare the performance with other dynamic checkers. They also proposed to combine iWatcher with invariant-inference tools like DIDUCE.