

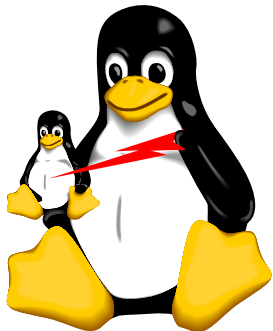
FAUmachine

Volkmar Sieh, Mario Dal Cin

Department of Computer Science 3
Computer Architecture
Friedrich-Alexander-Universität Erlangen-Nürnberg

June 2007





- virt. PCs, serial terminals, modems, hubs, ...
- able to run variety of OSs (Linux, Windows, DOS, *BSD, ...)
- sophisticated fault injection capabilities
- open source

more info at:

- <http://www.faumachine.org/>
- <mailto:info@faumachine.org>



Fault Injection

Fault Types:

- Memory / CPU Register
 - transient bit-flips
 - *permanent* stuck-at faults
 - *permanent* coupling faults
- Disk / CDROM
 - transient/permanent block faults
 - transient/permanent whole disk faults
- Network
 - transient send/receive faults
 - intermittent send/receive faults
 - permanent send/receive faults
- ...

Many other faults can be implemented very easily.



Latest Experiment: Testing Linux Software-Raid

- Hardware
 - standard PC with 3 IDE disks:
 - disk 1: OS
 - disk 2, 3: Raid containing Linux kernel sources (mirroring)
- Software
 - Debian-4.0r0 Linux standard installation (RAID: multiple devices; "md")
- Workload
 - compiling Linux kernel sources
- Faults
 - permanent whole disk faults at random time
 - permanent disk block faults at random time
 - one fault per experiment



Latest Experiment: Testing Linux Software-Raid

- setup script (=description) contains about 1000 lines of code
 - automatically reproducible!
- setup time: about one day
- time for one experiment: about one hour
- 450 experiments executed

Result:

- no Raid failures for 445 runs (injected faults are tolerated)
 - 5 Raid failures detected (whole disk faults)
- => 1.11 % failure probability on hardware malfunctioning



VHDL scripts contain *all* informations necessary to repeat experiments:

- Hardware Setup
- Software Setup (Installation Guide)
- User Interaction (Which Action, When, ...)
- Fault Injection (Which Fault, When, Where, ...)

But: multiple FAUmachine simulators on multiple hosts?

Task A: make net of FAUmachine simulator *deterministic*

=> Other people can reproduce experiments and results!



Task B: use VHDL-modeled components for FAUmachine's virtual machine

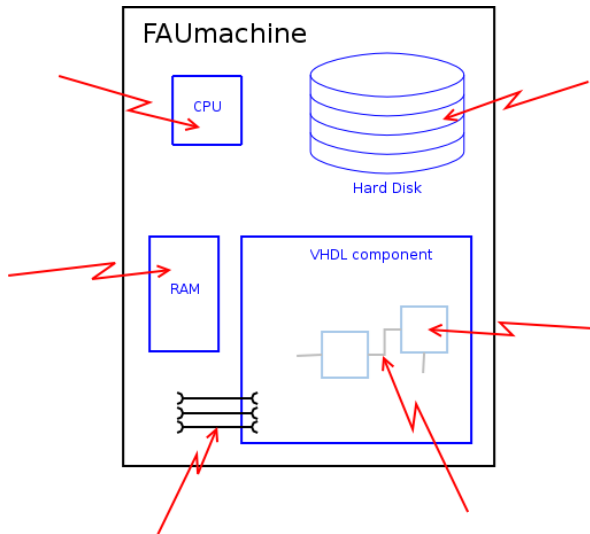
- virtual machine: high performance simulation
- VHDL component: detailed simulation w/wo fault injection
- run original Windows/Linux/... driver code
- run original/synthesizable hardware VHDL code
- run estimated fault load
- automated testing

without the need of any hardware prototype

- rapid prototyping
- sophisticated method to estimate fault tolerance attributes



Current Work / Hardware/Software-Co-Design



More Info:

- <http://www.faumachine.org/>
- <mailto:info@faumachine.org>

