

FAUmachine: Virtual Systems and Networks

Dr.-Ing. Volkmar Sieh

Institut für Informatik 3: Rechnerarchitektur
Friedrich-Alexander-Universität Erlangen-Nürnberg

October 2006



Overview

- 1 FAUmachine
- 2 Hardware Configuration
- 3 User Interaction
- 4 Fault Injection
- 5 VHDL Scripts
- 6 Simplified Scripts
- 7 Summary and Outlook

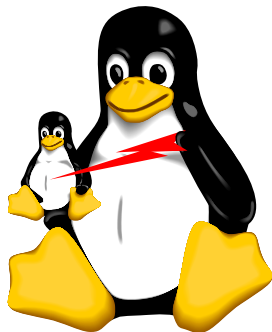


Outline

- 1 FAUmachine
- 2 Hardware Configuration
- 3 User Interaction
- 4 Fault Injection
- 5 VHDL Scripts
- 6 Simplified Scripts
- 7 Summary and Outlook



FAUmachine



- virt. PCs, serial terminals, modems, hubs, ...
- different simulation methods (mod. kernels, JIT, QEMU, ...)
- able to run variety of OSs (Linux, Windows, DOS, *BSD, ...)
- open source

more info at:

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



FAUmachine

The screenshot displays a virtual machine environment with several operating systems running simultaneously:

- FreeDOS-beta8**: A DOS-based system with a file explorer and system information.
- Debian-3.0-0**: A Linux system with a desktop environment and a login window for 'demo'.
- Win-XP-professional-german**: A Windows XP system with a desktop environment.
- FAUmachine**: A terminal window showing system boot logs and a process list.

The terminal window in the foreground shows the following output:

```

FAUmachine
File Machine inject Misc Send keycodes
On Off Reset Exit
Power: Disk: Network:
nothing grabbed

debian: # cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
gnupts:x:4:4:GNUts Bug-Reporting System (admin):/usr/lib/gnupts:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
sshd:x:100:10000:/:/var/run/ssh:/bin/false
demo:x:1000:1000:demo User, .../home/demo:/bin/bash

debian: # ps auxw
PID TTY STAT TIME COMMAND
 1 ? S 0:00 init [2]
 2 ? S 0:00 [kthreadd]
 3 ? S 0:00 [kupdate]
 4 ? S 0:00 [kswapd]
 5 ? S 0:00 [kworkerd]
146 ? S 0:00 /sbin/syslogd
149 ? S 0:00 /sbin/klogd
157 ? S 0:00 /usr/sbin/inetd
164 ? S 0:00 /usr/sbin/sshd
167 ? S 0:00 /usr/sbin/strd
170 ? S 0:00 /usr/sbin/cron
173 tty1 S 0:05 -bash
174 tty2 S 0:00 /sbin/getty 38400 tty2
175 tty3 S 0:00 /sbin/getty 38400 tty3
176 tty4 S 0:00 /sbin/getty 38400 tty4
177 tty5 S 0:00 /sbin/getty 38400 tty5
178 tty6 S 0:00 /sbin/getty 38400 tty6
222 tty1 R 0:02 ps auxw
debian: #

```



Performance

All instructions are compiled and executed natively.

Performance:

CPU/Memory/ROM: 5 times slower

Disk: 3 times slower (sometimes faster than host)

Network: 2 times slower (sometimes faster than host)

In general: 1/4 of performance of host system.



Outline

- 1 FAUmachine
- 2 Hardware Configuration**
- 3 User Interaction
- 4 Fault Injection
- 5 VHDL Scripts
- 6 Simplified Scripts
- 7 Summary and Outlook



FAUmachine System Configuration

PCs with Standard Hardware:

- CPU (x86/x86_64 with APIC)
- North Bridge
 - Memory Controller
 - PCI-Config-Space Controller
 - Host-Bus to PCI-Bus Bridge
- South Bridge
 - PIC, PIT, DMA
 - IDE, USB, SMBus Controller
 - Power Management
 - PCI-Bus to ISA-Bus/X-Bus Bridge
- BIOS, IOAPIC
- Keyboard Controller, Floppy Controller, Serial/Parallel Devices
- PCI-Bus, ISA-Bus
- ...



FAUmachine System Configuration

PCs can be configured:

- CPU (x86 / x86_64)
- Memory Module(s) (4 MByte - 2.5 GByte)
- Floppy Drive(s)
- IDE Controller(s)
- IDE Hard Drive(s) (20 MByte - ?)
- IDE CD/DVD Drive(s)/Burner(s)
- USB Controller(s)
- PCMCIA Controller(s)
- Video Adapter (Cirrus Logic, UMVGA)
- Network Adapter(s) (EEPROM100, NE2000)
- Sound Card(s) (SoundBlaster)
- ...



FAUmachine Network Configuration

Network can be configured:

- Each Virtual Machine: 0 - 6 Network Adapters
- Network Router
 - with built-in DHCP-Server
- Connections between Virtual Machines / Routers
- Connections from Virtual Machines / Routers to Real Network
 - via Slirp
 - via tun/tap Device



FAUmachine Configuration

Additionally:

- Serial Devices
 - Modems
 - Telephon Switches
 - Serial Terminals
- Parallel Devices
 - Printers
- ...



FAUmachine System Configuration

Configuration done by

- GUI
- VHDL



Outline

- 1 FAUmachine
- 2 Hardware Configuration
- 3 User Interaction**
- 4 Fault Injection
- 5 VHDL Scripts
- 6 Simplified Scripts
- 7 Summary and Outlook



Observability

Automatically observable:

- Contents of Monitor (ASCII-Characters, Pixel-Patterns)
- Contents of Serial Terminal (ASCII-Characters, Pixel-Patterns)
- LEDs (To do...)
- ...

Every pattern visible to real user observable by virtual user.



Controllability

Automatically controllable:

- Keyboard
- Mouse
- Reset- and Power-Button
- CD/DVD-Drive (insert/remove CD/DVD)
- ...

All actions controllable by real user usable by virtual user.



User Interaction

- Several Users Possible
- "Online" / "Live" User via GUI
- Definition of User via VHDL Script
 - Extensions for User Observations
 - Extensions for User Actions



Outline

- 1 FAUmachine
- 2 Hardware Configuration
- 3 User Interaction
- 4 Fault Injection**
- 5 VHDL Scripts
- 6 Simplified Scripts
- 7 Summary and Outlook



Fault Injection

Fault Injection:

- Bit Faults in Memory Cells (Flip und *Permanent*)
- Bit Faults in CPU Registers (Flip und *Permanent*)
- Disk Faults (Entire Disk, Single Defect Blocks)
- Network Package Loss (0%-100%)
- ...



Fault Injection

- "Online" / "Live" Fault Injection via GUI
- Definition of Fault Injection via VHDL Script
 - Type of Fault
 - Location of Fault
 - Time of Fault
 - Duration of Fault



Outline

- 1 FAUmachine
- 2 Hardware Configuration
- 3 User Interaction
- 4 Fault Injection
- 5 VHDL Scripts**
- 6 Simplified Scripts
- 7 Summary and Outlook



VHDL Scripts

VHDL has all features necessary to

- Define Hardware Setups
- Control in Parallel (using Signals)
- Observe in Parallel (using Signals)
- Influence (Fault Injection) in Parallel (using Signals)



VHDL Scripts

VHDL:

- Hardware Setup
- Software Setup
 - Defining a User Executing the Installation Process
- User Observations
- User Actions
- Fault Injection



VHDL Scripts

Example:

- Switch on power of virtual machine.
- If "Login:" is visible on screen, type "root".
- Wait until ... appears.
- Activate Fault ... of component ... after ... seconds for ... milliseconds.
- If ... is not visible within ... seconds, terminate simulation with message ...
- ...



Only the User Matters

Motto:
Any failure not seen by any user doesn't matter...!



VHDL-Skripte

Problems:

- VHDL Doesn't Define Fault Injection Signals
- VHDL Doesn't Define Pattern Matcher Signals

Solutions:

- "shortcut"s
 - Cables Directly Connected to Internal Signals of Components
 - Signals used for Observation
 - Signals used for Control



VHDL Scripts, "shortcut"s

Example: Fault Injection

```
signal active: boolean;  
active'shortcut_out("/pc1/eth0%recv_loss");  
  
wait for 5 min;  
active <= 90; -- now 90% package loss  
wait for 60 sec;  
active <= 0; -- no package loss any more
```



VHDL Scripts, "shortcut"s

Example Pattern Matcher:

```
signal match: boolean;  
match'shortcut_in("/monitor%match/0");  
  
match'shortcut_in_activate("Login:");  
wait on match until match for 60 sec;  
  
if not match then ... endif;  
  
match'shortcut_in_deactivate;
```



VHDL Example

TPCC with

- 2 Oracle Database Servers
- 4 Application Servers
- 12 Terminals for Application Users
- 4 Terminals for System Administrators



Reproducible Results

VHDL script contains *all* informations necessary to repeat experiments:

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

Other persons can reproduce experiments and results!



Outline

- 1 FAUmachine
- 2 Hardware Configuration
- 3 User Interaction
- 4 Fault Injection
- 5 VHDL Scripts
- 6 Simplified Scripts**
- 7 Summary and Outlook



Simplified Scripts

VHDL has many features

- => programming VHDL is time consuming
- => simplified macro language for simple applications



„faum-gen-vhdl” Example

Starting UltimateBootCD tests (e.g. "memtest86"):

wait_asc	Main Menu	1min
type	"\10"	
wait_asc	more mainboard	1min
type	"\13"	
wait_asc	Test #4	5min



Outline

- 1 FAUmachine
- 2 Hardware Configuration
- 3 User Interaction
- 4 Fault Injection
- 5 VHDL Scripts
- 6 Simplified Scripts
- 7 Summary and Outlook**



Summary

- FAUmachine provides virtual PCs and other components
- components can be observed
- components can be controlled
- high sophisticated fault injection facilities
- observations, actions, and fault injections can be programmed using VHDL and simplified scripts



Outlook

- support even more OS
- more PC components (e.g. PCIe components)
- more external components (e.g. laser printer)
- more faults
- improved pattern matcher



Thank You!

more info:

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

