

Dependency analysis

Michael Sills-Lavoie
Department of Computer and Software Engineering

June 29th, 2010
École Polytechnique de Montreal



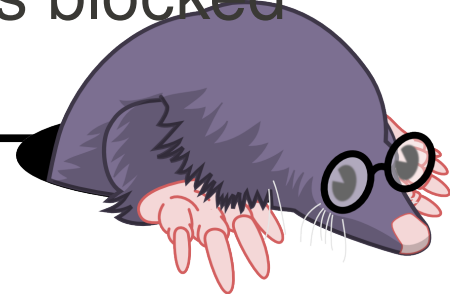
Content

- **What is dependency analysis**
- **Previous work**
- **Future work**



What is dependency analysis

- In compiler theory, dependence analysis produces execution-order constraints between statements/instructions. (Wikipedia)
- System wide, dependency analysis could be defined as a way to determine how a process reached a specific state in time by analyzing:
 - where the execution time was spent (user space, system call, IRQ, etc.)
 - where, how long and by who the program was blocked



Previous work

- Pierre-Marc Fournier :
 - Creation of a dependency analyzer plug-in for Ittv that :
 - analyze the trace events with state machines
 - create a text summary of where the process spent his time
 - Working, Blocked, Interrupted
 - create a text report about blocking causalities



Previous work

- Time spent summary

```
Process 438 [/usr/bin/xinit]
Total (1.309108247) <0>
  Blocked (1.294384393) <1>
    Syscall 23 [sys_select+0x0/0x16c] (1.294379908) <8>
    Syscall 57 [stub_fork+0x0/0x11] (0.000002859) <2>
    Syscall 59 [stub_execve+0x0/0xc0] (0.000001626) <4>
  Interrupted (0.009927037) <5>
  Scheduled out (0.009908719) <7>
  Waiting for schedule after blocking (0.000018318) <6>
  Working (0.004796817) <3>
```

- Blocking causality report

```
Process 438 [/usr/bin/xinit]
[...]
Blocked in RUNNING, SYSCALL 59 [stub_execve+0x0/0xc0], (times: [...], dur: 0.000002)
Woken up in context of 3 [migration/0] in WIB state UNKNOWN
Blocked in RUNNING, SYSCALL 23 [sys_select+0x0/0x16c], (times: [...], dur: 1.294232)
  Blocked in RUNNING, SYSCALL 2 [sys_open+0x0/0x17], (times: [...], dur: 0.013940)
    Woken up by a SoftIRQ: SoftIRQ 8 [rcu_process_callbacks+0x0/0x47]
  Blocked in RUNNING, SYSCALL 35 [sys_nanosleep+0x0/0x62], (times: [...], dur: 0.200006)
    Woken up by an IRQ: IRQ 239 []
  Blocked in RUNNING, SYSCALL 23 [sys_select+0x0/0x16c], (times: [...], dur: 0.203262)
    Woken up by a SoftIRQ: SoftIRQ 1 [run_timer_softirq+0x0/0x21a]
  Blocked in RUNNING, SYSCALL 35 [sys_nanosleep+0x0/0x62], (times: [...], dur: 0.010002)
    Woken up by an IRQ: IRQ 239 []
  Blocked in RUNNING, SYSCALL 35 [sys_nanosleep+0x0/0x62], (times: [...], dur: 0.010001)
    Woken up by an IRQ: IRQ 239 []
  Blocked in RUNNING, SYSCALL 35 [sys_nanosleep+0x0/0x62], (times: [...], dur: 0.010002)
    Woken up by an IRQ: IRQ 239 []
  Blocked in RUNNING, SYSCALL 35 [sys_nanosleep+0x0/0x62], (times: [...], dur: 0.010005)
    Woken up by an IRQ: IRQ 239 []
  Blocked in RUNNING, SYSCALL 35 [sys_nanosleep+0x0/0x62], (times: [...], dur: 0.010002)
    [...] 37 other nanosleeps() lasting 0.01 to 0.03 seconds [...]
Woken up in context of 440 [/usr/bin/Xorg] in WIB state WORKING
[...]
```



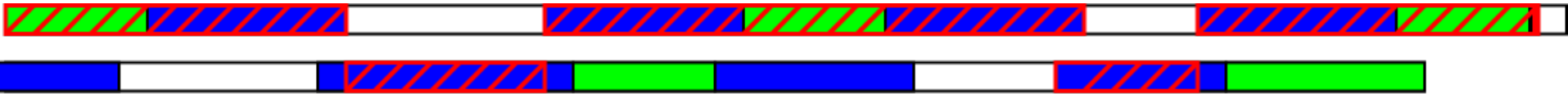
Previous work

- Problems
 - It's complicated and not user friendly
 - Causality reports are long and take time to understand
 - There isn't any user interface
 - It can't analyze dependencies between a host and a virtual machine
 - It can't analyze dependencies between hosts on a network



Future work

- Reimplementation of the dependency analyzer for TMF
 - Make use of the state system and control flow view
- Creation of a user interface able to show clearly and rapidly the dependency for a specific process



- Add the possibility to track dependencies between a host and a virtual machine and between distributed hosts



Questions ?



References

- [1] P.-M. Fournier, “Analyse automatisée des causes de blocage de processus à partir d'une trace d'exécution,” Master's thesis, Ecole Polytechnique de Montreal, March 2009.

