The Case for Distributed Execution Replay using a Virtual Machine

Nicolas Loriant    Jean-Marc Menaud

OBASCO research group
EMN/INRIA - LINA, Nantes, France

Emerging Technologies for Next-generation GRID
Motivations

GRID Computing
- Heterogenous ressource
- geographically scattered

Developping Distributed Grid applications
- Complex interactions between multiple tasks
- Transient errors: race conditions, locks

Execution Replay
- Monitor the execution
- Reproduce it help find the source of an error
Problems with execution replay

The probe effect

Monitoring requires to insert additional computation, the program behavior is modified

- bugs that would not have appear
- bugs that no longer appear

The irreproducibility effect

Some behavior are sometimes hard to reproduce
Proposal

**Virtual Machines** Xen, VMWare, QEMU, UMLinux...
- easily simulate thousands machines

**The probe effect**
The behavior of a virtual machine may be modified to compensate the probe effect

**The irreproducibility effect**
Everything is under control inside a virtual machine
A possible implementation

**Virtualization or Emulation**
- Virtualization performances, limited to same host/guest hardware
- Emulation: fine control of the host execution

**Using QEMU**
- Emulates a wide range of architectures
- Still great performances thanks to dynamic code generation
Monitoring

Target code

Translated/host code

inc %ebp
movl %eax, (cBut)
addl $4, cBut
inc $0x60, %eax
movl %eax, (cBut)
addl $4, cBut
movl (%ebp), (cBut)
addl $4, cBut
mov %eax, (%ebp)
movl label, (cBut)
addl $4, cBut
movl here, (cBut)
addl $4, (cBut)
movl next, cc
jmp executeNext

QEMU dynamic translator

label:
  inc %ebp
  inb $0x60, %eax
  movl %eax, (%ebp)
  jmp next

here:
  breakpoint @next
  > break @label
  > mode reverse
  > run

Monitoring
Reverse Execution

Nicolas Loriant

The Case for Distributed Execution Replay using a Virtual Machine
Replay Execution

Nicolas Loriant

The Case for Distributed Execution Replay using a Virtual Machine
Related Work

**Revirt**
- UMLinux (Virtualization)
- Checkpointing only

**Java Reverse Execution (J COOK)**
- Kaffe virtual machine
- both reverse and replay
Conclusion & Future Work

Summary
- using Virtual Machine for reverse and replay execution of distributed systems
- an outline of a possible implementation

Future Work
- How to manipulate a time traveling debugger?
- How to manipulate the replayed execution?
  - changing the ordering of network messages
  - changing the scheduling of tasks