

# Through an Event Log, Darkly

Keith Douglas

Statistics Canada

[philosopher.animal@gmail.com](mailto:philosopher.animal@gmail.com)

# Outline

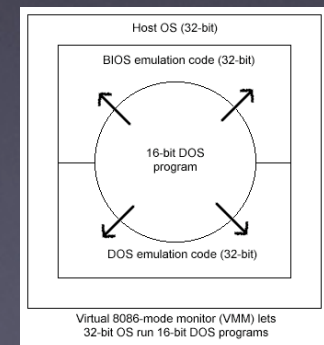
- Introduction
- Simulation of computers by computer
- Simulating a timer: some metaphysics
- Epistemological lessons
- Conclusion

# Introduction

- Start by looking briefly at the philosophy of simulation literature
- Well established in philosophy of mind for decades (Hofstadter and Dennett 1981)
- Enormous literature also in philosophy of science and technology generally (Lehtinen and Kuorikoski 2007)
- Gap: Where is the literature on simulation of computers by computers?

# Simulation of Computers by Computers?

- Not a new activity - often called *emulation*
- Two main sorts
  - Additional hardware (PC Transporter, Ile card)
  - Software (Bernie ][ The Rescue, OS/2's DOS Emulation)
- Special type of emulation in latter: *virtualization*



# Simulation of computers By Computers (cont'd)

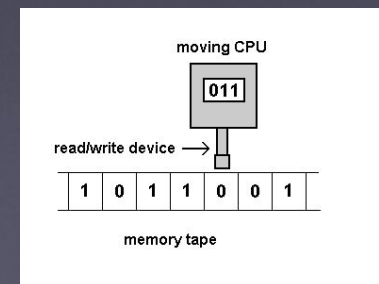
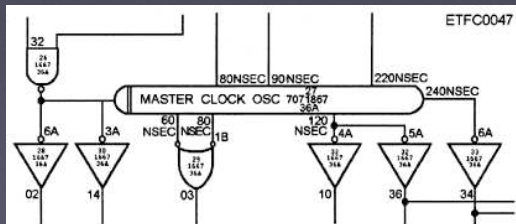
- “Better DOS than DOS” and v8086 mode
- Traditional uses of emulation:
  - Run legacy software in new system (OS/2's DOS emulation)
  - Run software from a competing platform (SoftWindows)
  - Have fun with old stuff (Bernie ][ The Rescue)

# Simulation of computers By Computers (cont'd)

- But: recent change in design in microprocessors allows something novel
- Virtualization of the entire CPU, not just a legacy part (no OS/2 in OS/2 this way)
- Now can run virtualize Windows in Windows
- Many reasons to do this (security, better use of resources, ease of maintenance)
- Big commercial example: VMWare

# Simulating a Timer

- AKA, Why Turing Equivalence is Not Enough
- Turing and “On Computable Numbers ...”
- The UTM is the “logician’s way” of understanding virtualization
- Abstracts from time variable



# Simulating a Timer (cont'd)

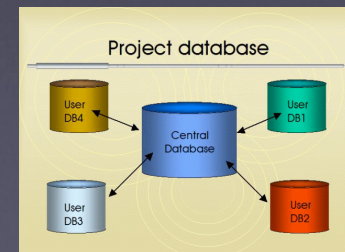
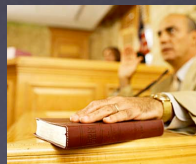
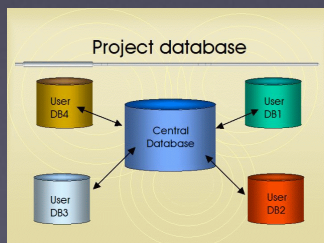
- Virtualized hardware is **USED** like a real machine
- Can install any software in it (e.g. BizTalk Server 2006)
- What's the problem? Simulating the timer.

# Simulating a Timer (cont'd)

- Oscillator
- Also emulated - relies on that of the host
- **CANNOT** be emulated perfectly
- Hence conflict between the world “outside” and in the VM or between VMs
- Problem then occurs with any sensitive timing

# Simulating a Timer

- Database mirroring in SQL Server 2005
  - A “complicated ping” needed for witness server
  - Role of principal, mirror, witness
  - So witness ideal for virtualization since tiny machine needed, right?

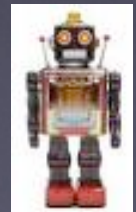


# Simulating a Timer

- To answer the question from the last slide: wrong!
- Suppose the timing is “out of wack”
- Servers lose track of each other
- In particular, witness loses the principal and initiates failover
- NB: **NOTHING** to do with resource limitations - we’re not simulating a 64 bit dual core machine on an XT

# Simulating a Timer

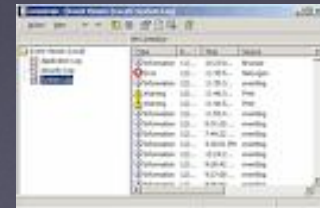
- Metaphysical lesson / area of investigation:  
The Robot Reply
- Not a appeal to mysterious “causal powers”  
but concrete
- But it gets worse: epistemology affected by  
this little piece of applied metaphysics





# Epistemological Lessons

- Is this a banal conclusion? VMWare seems to say so
- But how do we discover that we indeed have tripped a problem? VMWare can't exhaustively test all software under all conditions
- Black smoke? No, consult log files



# Epistemological Issues

- Banal point that is nevertheless vital:
- *Errors written to an error log have to be provoked by a condition that the program writing the log has been designed to detect and report*
- Hence we have a “black box”, or “we see through an event log, darkly”
- Is this a case of the Duhem-Quine problem?



# Epistemological Issues

- Actually: version of Collins' "experimenter's regress"
- Internal tools presuppose being used in a real environment
- ping (proper) generally right: software plays "tricks"
- But imperfect: so "latency" is sometimes reported in the thousands of milliseconds when it should be on the order of one ms

# Epistemological Issues

- But it gets worse
- Interacting with system can “clear up” the very problem being investigated (cf. Bunge 1967)
- Mechanism: VM idle so VM throttled; then application needs more time slices so timing worse; **then** host kicks in more slices because tasks running; admin sees problem, pokes around provoking more slices



# Epistemological Issues

- Consequence: by the time one calls server support or the network team, things are back to normal
- Thus two puzzles
- Solutions from the philosophy of science to the analogous cases don't quite work for second: no objective version of a subjectivist misinterpretation
- First has possible area for solution

# Epistemological Issues

- Not all epistemic resources “on the inside” (Franklin)
- Consilience (Wilson) / Convergence (Bunge)
- Must reproduce what we already know, at least in part
- Can use other parts of a system to check, including humans
- Our own app helps here too



# Epistemological Issues

- Solve also by measuring the same value in different ways (Orion)
- Raises issues of “the same value” (subnetting)
- Agreement in this case
- Popperianism: attempt to exonerate VMWare with over-all latency measurement
- Popperianism 2: Try the witness elsewhere



# Conclusions

- Sketched answer to “log regress” problem
- Left open interaction problem
- Metaphysical issue also open
- Aim was merely to sketch areas of interesting future research

Questions?