



TECHNISCHE  
UNIVERSITÄT  
DRESDEN

Faculty of Computer Science Institute for System Architecture, Operating Systems Group

# OS Paper Reading Group

## Winter Term 2008/09

Dresden, 2008-10-15

- Website: [www.tudos.org](http://www.tudos.org) -> Teaching -> Reading Group
- Mailing List:  
<http://os.inf.tu-dresden.de/mailman/listinfo/paper-reading-group>
- Modus
  - 1 paper each week
    - Related to systems research
    - 10-15 min presentation
      - Paper content
      - Questions regarding the paper / the general topic
    - ***Discussion***
  - Choice of paper
    - *Staff members* propose 3 alternatives, voting on the mailing list.
    - *Students* pick a paper either from the list on the web site or any other source.
  - Pizza anyone?

- Send a paper summary to [doebel@tudos.org](mailto:doebel@tudos.org) until the day before the presentation (23:59:59)
  - Explain what you understood from the paper.
  - Ask questions about things you did not understand.
  - Mention things you like/dislike about the paper.
  - ...
- Present one paper yourself during the term.
  - English (though this is not a test of your language skills!)
  - Show that you understood the paper.
  - Prepare questions for discussion.
  - Extended knowledge (e.g., related work) can be a plus.



# Configuration Debugging as search – Finding the Needle in the haystack (A. Whitaker et al.)

presented by Bjoern Doebel

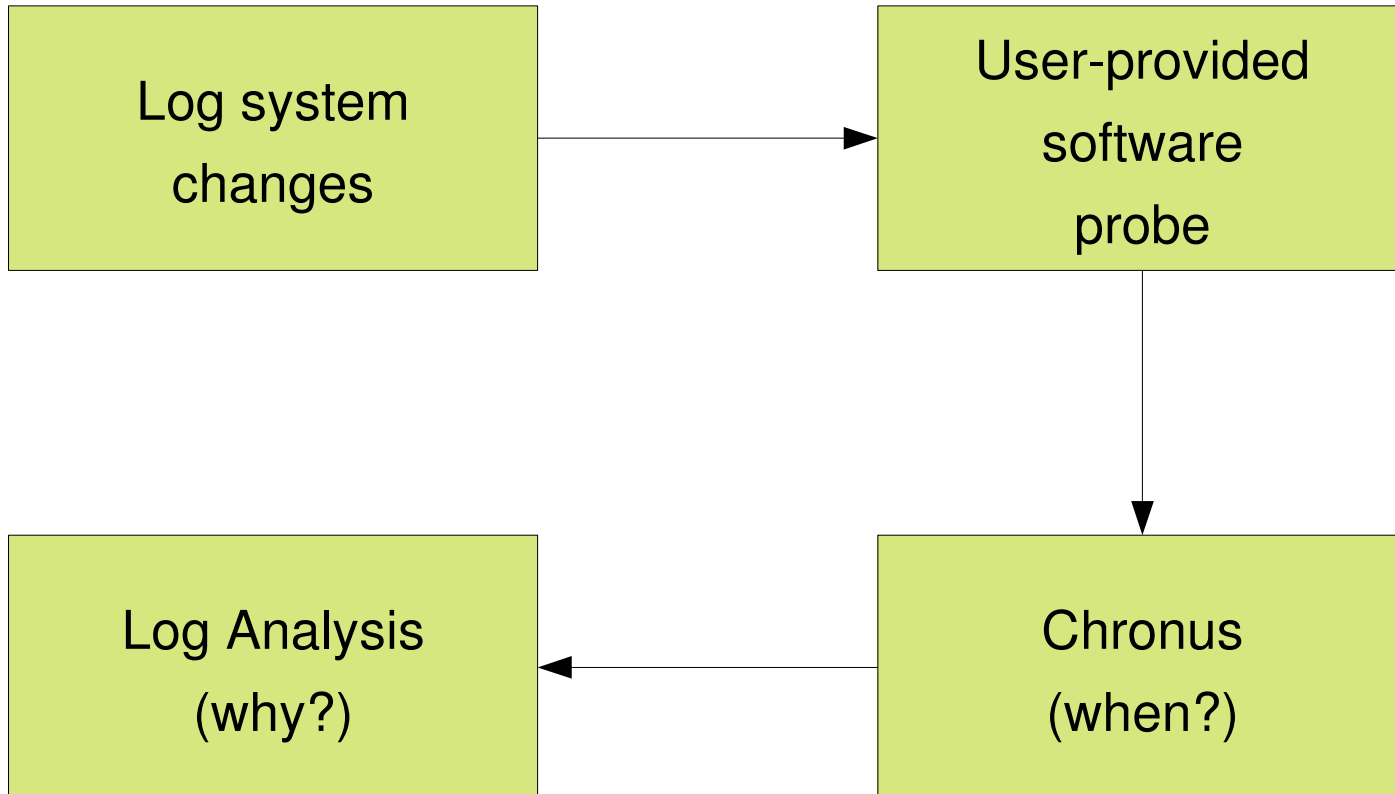
Dresden, 2008-10-15

- Systems often break after configuration changes
  - Software patch
  - Modification of security policy
  - Administrator actions
    - Gray, 1985: 42% of all failures are human-induced.
- Experts needed to find bugs.

Goal: Automate diagnosis



# Chronus workflow

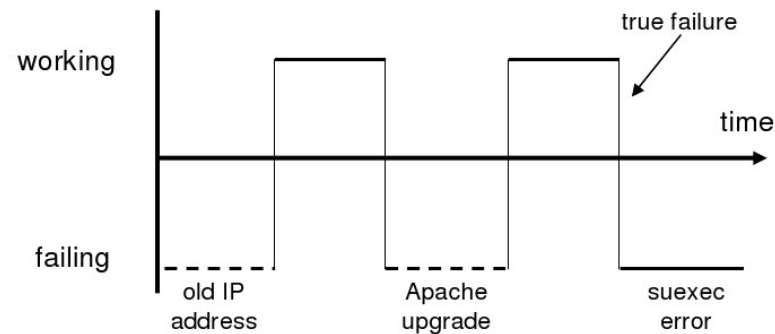


- Traditional checkpointing
  - Persistent + transient state
  - Slow at runtime
- Incremental logging
  - Faster at runtime
  - Slow at state reinstantiation
- Focus on persistent state (disk)
- Dedicated mechanism for tracking changes at block level -> **Time-Travel-Disk**

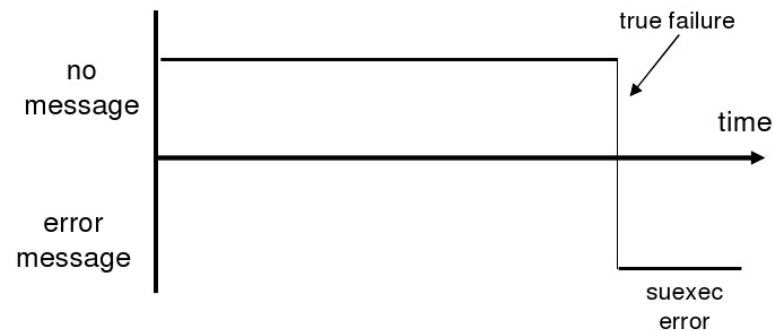
- User-defined interval
  - Start & end state need to differ wrt to bug
- Reinstantiation of system states using VMM
  - Cheaper than real HW
  - Lose insight into some error causes (drivers...)
  - Run user-defined probes
- Binary search through states in interval
  - Search in  $O(\log n)$
  - Problem: multiple changes in interval?
    - Simulated annealing (random restart, random intervals)



- User-defined probes
- External vs. internal probes
- Success-oriented vs. failure-oriented probes



**a) Success-directed probe timeline**



**b) Failure-directed probe timeline**



- Observation: TT-disk overhead negligible most of the time
- Logs increase heavily
  - Compression helps (at the cost of reinstatiation overhead)
  - Disable logging for certain operations
- Debug execution time  $\sim$  minutes

- TT-disk enable us to reinstantiate the system at every block-level modification.
  - Millions of possible states.
  - Not guaranteed that the system is even bootable in each of these states.
  - My intuition: could get less states and the same feature set using FS-level checkpointing.
    - e.g., only create a state after a file has been completely written
    - `git bisect` as a real world example
  - However: we wouldn't need TT-disks then...

- More complex systems -> Log size increase -> Search time increase
- Most of Chronus' test run time spent rebooting
- Systems get more dynamic
  - Kernel modules are old already
  - Kernel updates at runtime (Makris, EuroSys 2007)
  - SysFS tuning of kernel parameters
- *Hypothesis:* future configuration changes will not necessarily rely on block-level interaction
  - Always restart from an initial CoW virtual machine
  - Re-apply delta-checkpoints of dynamic configuration changes

Could this be faster than cold-rebooting the VM?  
Is this relevant?