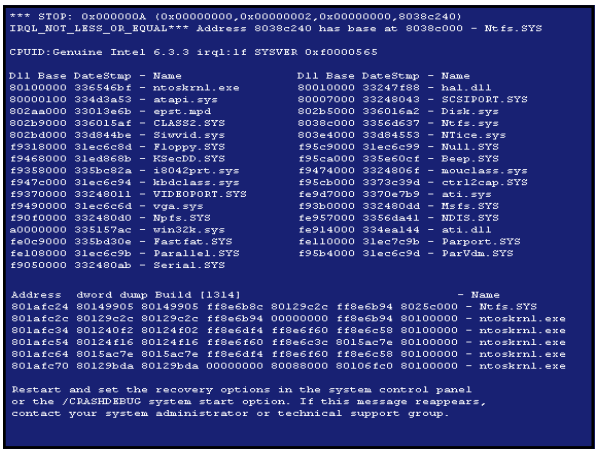
# Initial Evaluation of a User-Level Device Driver Framework


Stefan Götz
Karlsruhe University
Germany
*sgoetz@ira.uka.de*

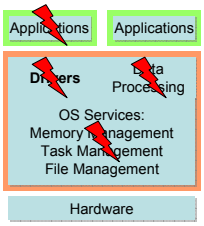
Kevin Elphinstone
National ICT Australia
University of New South Wales
*kevine@cse.unsw.edu.au*

```
*** STOP: 0x000000A (0x00000000,0x00000002,0x00000000,8038c240)
IRQL_NOT_LESS_OR_EQUAL*** Address 8038c240 has base at 8038c000 - Ntfs.SYS

CPUID:Genuine Intel 6.3.3 irql:1f SYSVER 0xf0000565

Dll Base DateStmp - Name                 Dll Base DateStmp - Name
80100000 336546bf - ntoskrnl.exe         80010000 33247f88 - hal.dll
80000100 334d3a53 - atapi.sys            80007000 33248043 - SCSIPORT.SYS
802aa000 33013e6b - epst.mpd             802b5000 336016a2 - Disk.sys
802b9000 336015af - CLASS2.SYS           8038c000 3356d637 - Ntfs.sys
802bd000 33d844be - Siwvid.sys           803e4000 33d84553 - NTice.sys
f9318000 31ec6c8d - Floppy.SYS           f95c9000 31ec6c99 - Null.SYS
f9468000 31ed868b - KSecDD.SYS           f95ca000 335e60cf - Beep.SYS
f9358000 335bc82a - i8042prt.sys         f9474000 3324806f - mouclass.sys
f947c000 31ec6c94 - kbdclass.sys         f95cb000 3373c39d - ctrl2cap.SYS
f9370000 33248011 - VIDEOPORT.SYS        fe9d7000 3370e7b9 - ati.sys
f9490000 31ec6c6d - vga.sys              f93b0000 332480dd - Msfs.SYS
f90f0000 332480d0 - Npfs.SYS             fe957000 3356da41 - NDIS.SYS
a0000000 335157ac - win32k.sys           fe914000 334ea144 - ati.dll
fe0c9000 335bd30e - Fastfat.SYS          fe110000 31ec7c9b - Parport.SYS
fe108000 31ec6c9b - Parallel.SYS         f95b4000 31ec6c9d - ParVdm.SYS
f9050000 332480ab - Serial.SYS

Address  dword dump Build [1314]                                  - Name
801afc24 80149905 80149905 ff8e6b8c 80129c2c ff8e6b94 8025c000 - Ntfs.SYS
801afc2c 80129c2c 80129c2c ff8e6b94 00000000 ff8e6b94 80100000 - ntoskrnl.exe
801afc34 801240f2 80124f02 ff8e6df4 ff8e6f60 ff8e6c58 80100000 - ntoskrnl.exe
801afc54 80124f1f 80124f16 ff8e6f60 ff8e6c3c 8015ac7e 80100000 - ntoskrnl.exe
801afc64 8015ac7e 8015ac7e ff8e6df4 ff8e6f60 ff8e6c58 80100000 - ntoskrnl.exe
801afc70 80129bda 80129bda 00000000 80088000 80106fc0 80100000 - ntoskrnl.exe

Restart and set the recovery options in the system control panel
or the /CRASHDEBUG system start option. If this message reappears,
contact your system administrator or technical support group.
```

## Why do systems crash?

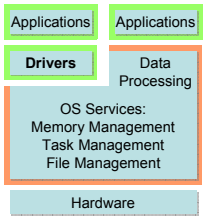
- Privileged device drivers in monolithic systems
  - No protection
    - → system crashes
- [Chou, Engler et al., 01]
  - Many flaws in driver code
  - Large driver code-base
- Run un-trusted drivers?






## Goals

- Improve OS reliability
- Achieve protection
  - Isolate drivers
  - De-privilege drivers
- Application properties
  - → user-level drivers
- Maintain performance





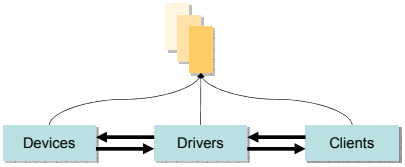

## Related Work

- Driver signing
- Isolated but privileged drivers (Nooks)
- "Safe" kernel extensions
- User-level drivers
  - Incomplete bottom-up analysis so far
  - Virtual machines
  - Potential for formal system verification



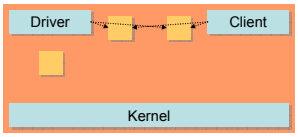
## Basic Driver Model

- Data transfer & buffer management
- Event notification



# Data Transfer

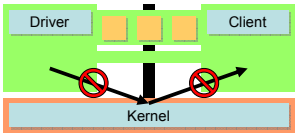
- Copying expensive → pass-by-reference
- Transfer via shared memory
- Amortization of setup costs






# Event Notification

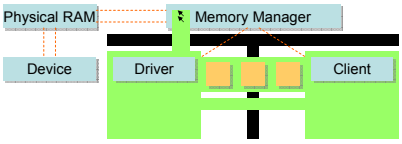
- Kernel interaction too expensive
- User-level messaging via shared memory
- Batching





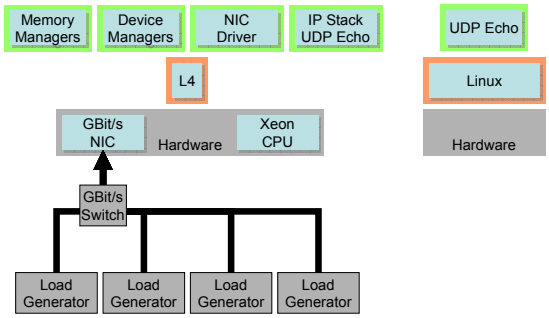

# Buffer Address Translation

- Physical buffer addresses for DMA
- Translation data unavailable to driver
- Secure state sharing
- Amortization of costs



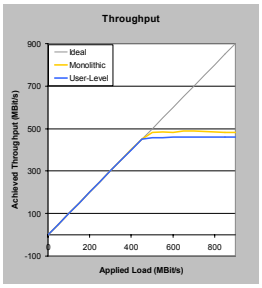


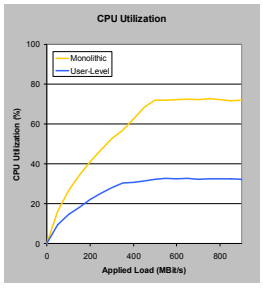

# Evaluation: Test Setup






# Evaluation: Benchmark Results




Similar throughput at lower CPU utilization than Linux



# Summary

- Poor OS reliability due to device drivers
- Isolated and de-privileged drivers
  - Fault isolation
  - Competitive performance
- Future work
  - Application-level benchmarks
  - Impact of DMA-safe hardware (IO-MMUs)
- *Punch line / Take-home message?*

# Thank You

- [Chou, Engler et al., 01]: Chou, A., Yang, J., Chelf, B., Hallem, S., Engler, D. *An empirical study of operating systems errors*. In Proceedings of the 18th Symposium on Operating Systems Principles, 2001



# Session Abstraction

- Interaction and authentication context
- Connects interacting components
- Chainable
- Associates with shared memory
- Relatively long-lived
- Allows for
  - batching
  - granularity vs. performance trade-off



# Buffer Memory Pinning

- Time-based pinning
  - Pin time guaranteed by memory manager
  - Part of translation cache entries
  - Correct estimates in drivers difficult
- State sharing
  - Pinning requests by drivers
  - Advisory bit in translation cache entries
  - Resource limits enforceable
  - Translation caches writable for drivers



# Interrupt Handling

- Interrupt delivery to user-level via kernel
- Synchronous IPC as light-weight abstraction
- Overhead from kernel interaction
  - Small compared to off-chip device handling
- Interrupt hold-off techniques
  - Batching in hardware
  - Latency vs. throughput & CPU utilization



# Execution Time Profile