

Unmodified Device Driver Reuse and Improved System Dependability via Virtual Machines
Joshua LeVasseur, Volkmar Uhlig, Jan Stoess, Stefan Gotz (University of Karlsruhe)

One liner: This paper presents the notion of using virtual machines to provide device driver reuse (to extend driver-scarce OSes) and isolation (to improve fault tolerance and availability); the authors show that their implementation incurs relatively low overhead.

The strength of this paper is that it discusses many thorny issues related to porting device drivers to virtual machines and thus gives the reader a good deal of information should he want to do this himself. Of the few figures provided, the performance graphs are very clear and help to illustrate the authors' points in that area. Otherwise, the idea of putting device drivers in VMs is novel enough to warrant me putting it in the strengths section.

One of the biggest weaknesses of this paper is that the reader is drowned in a sea of language that is foreign to all but the L4/EROS/microkernel community. All the discussion of virtualization, paravirtualization, client and server VMs without any figures or diagrams illustrating their architecture is asking way too much of the reader. Another weakness is that the authors managed to implement essentially only three drivers in VMs: network, disk and PCI. Though CPU consumption for these devices generally does not affect throughput, I'd like to see what happens when between 50-100 device drivers coexist in their own VMs on a system. Something tells me that the CPU consumption will be harder to swallow. I'm not terribly concerned by the DD/OS memory consumption, but I am concerned about the overhead of the drastic techniques used to reduce it. Perhaps this will become more critical as memory becomes scarce in the system—a test that they authors did not run. Finally, I would have liked to see some example code from the modifications that the authors made to run their system with VMs. Although they say it did not take much time to write, the students doing the work were clearly familiar with the concept, so I'd like to see how much time it takes an undergrad to do the same work.

In general, this seems like a good idea, but I'm afraid that the authors did not make the case that their work is general, as it was applied only to a limited case of Linux on top of L4, and it's hard to say how well this would work for porting a Windows device driver (sans source code) to Linux on top of L4. Something tells me that it would not quite work, though the grandiose claims in some sections of this paper would seem to indicate otherwise.