

# The Potemkin Virtual Honeyfarm

Jay Chen, Justin Ma, John McCullough, David Moore, Erik Vandekieft, Michael Vrible, Stefan Savage, Alex C. Snoeren, Geoffrey M. Voelker <http://www.ccied.org/>

## Achieving Scalability, Fidelity, and Containment

Networks of honeypot systems, or honeyfarms, are an excellent resource for studying network worms and other malware. However, honeyfarms exhibit a tension between

**Scalability:** How large a network can the honeyfarm serve?

**Fidelity:** How accurately does the honeyfarm capture the behavior of real systems?

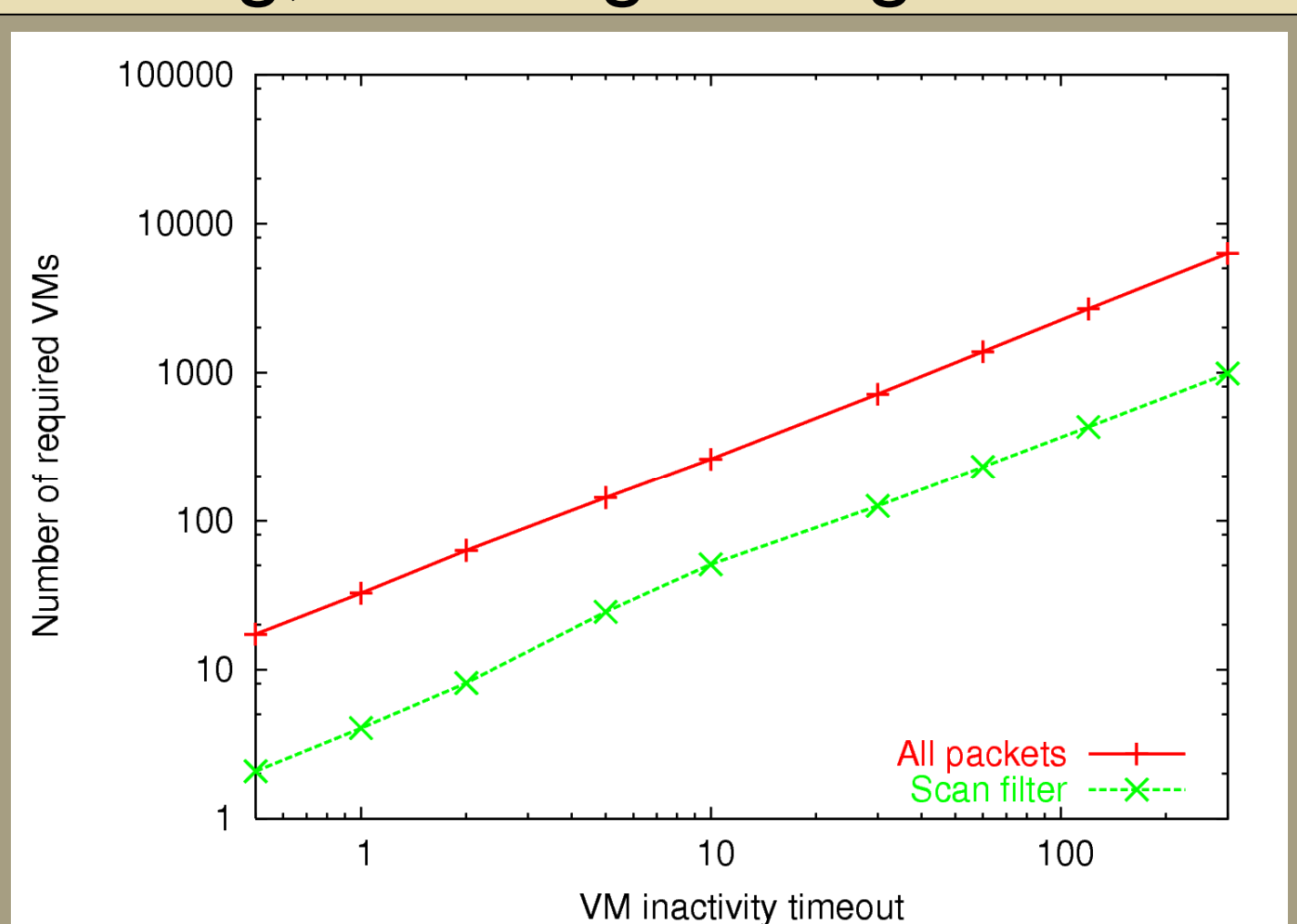
Potemkin aims to achieve both high scalability (monitor millions of addresses) and high fidelity (complete hosts) through the use of network filtering and hardware virtualization with the Xen virtual machine monitor.

### Potemkin Operation

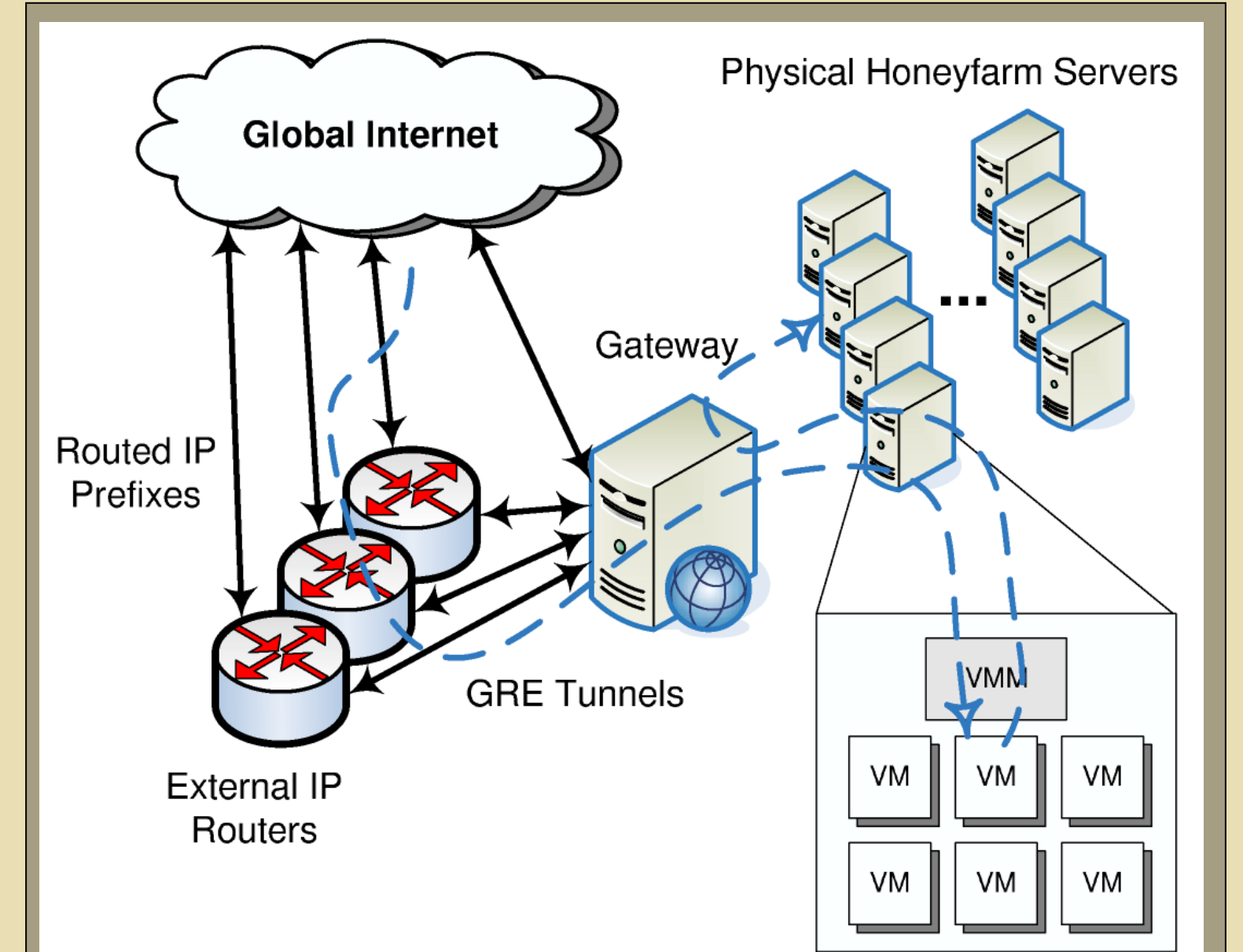
A network gateway forwards received traffic from the Internet to a dynamically-chosen honeyfarm server. Using *flash cloning*, the server instantiates a honeypot within a virtual machine, cloning it from a running reference honeypot. To achieve **containment**, outgoing connection attempts from the honeyfarm may be redirected by the gateway to another machine within the honeyfarm.

If no infection occurs, the VM is quickly recycled to reclaim resources. In the event of an infection, controlled propagation within the honeyfarm allows for immediate study, or infected VMs may be frozen for later analysis.

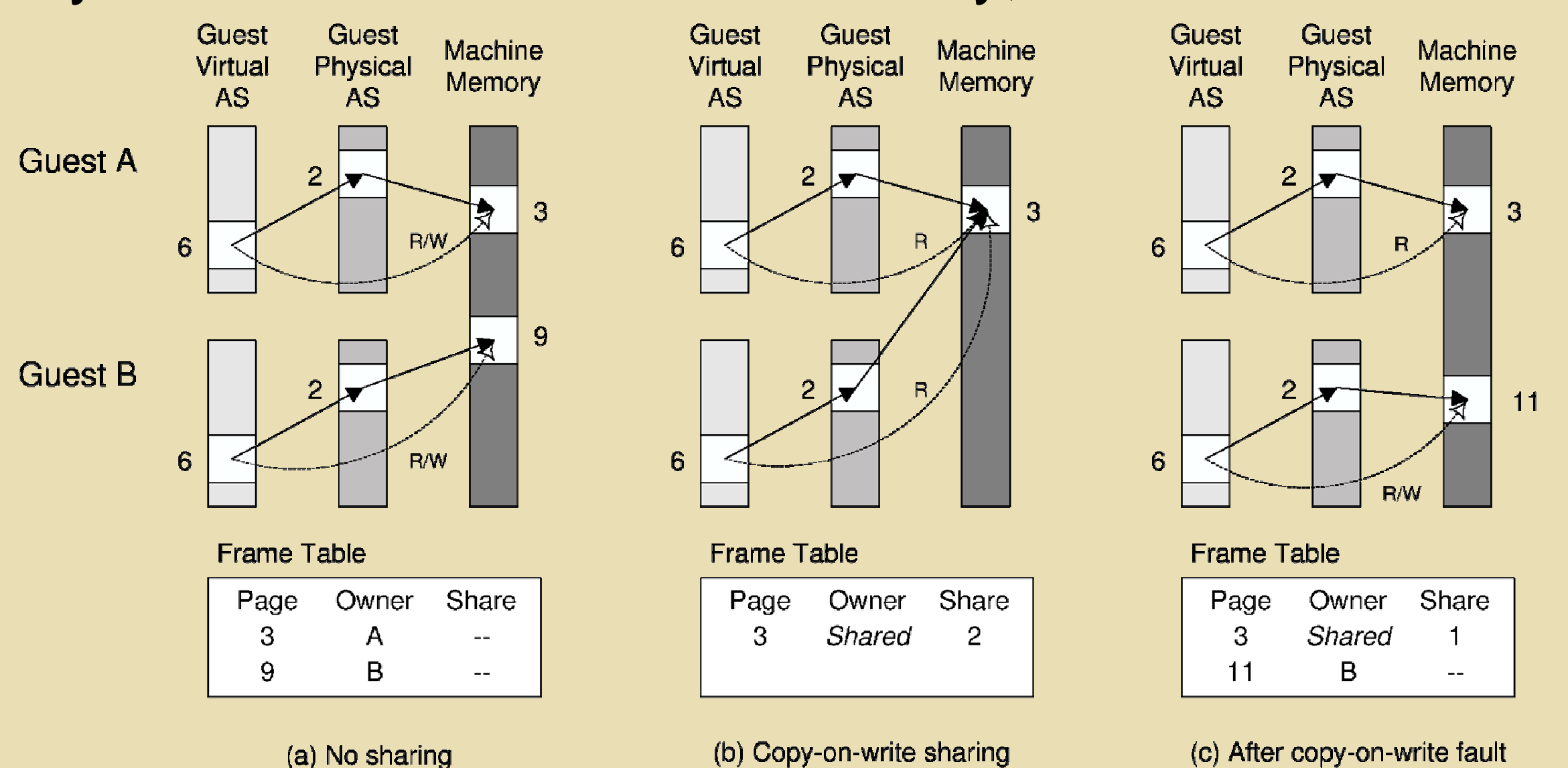
Using *delta virtualization*, or support for copy-on-write memory pages, cloned VMs may be created quickly and with maximal resource sharing, allowing for higher scalability.



**VMs needed for a /16 network. The network filtering is able to significantly increase system scalability.**



The Potemkin architecture using a gateway to direct traffic to virtual honeypots.



### Potemkin Implementation

We extended Xen to support flash cloning and delta virtualization. In Potemkin, virtual address spaces of cloned VMs reference a base VM image and use copy-on-write to exploit memory sharing among cloned VMs. The figure above illustrates the relationship of cloned VM address spaces in native Xen as well as Xen enhanced with copy-on-write in Potemkin.

### Potemkin Performance

Such techniques enable Potemkin to clone VMs very quickly and scale to support many active VMs. In our current prototype, Potemkin clones a new VM in 500 ms and each new clone only requires 2-10 MB of memory per VM, enabling 100s of active VMs. As a result, a Potemkin honeyfarm can use very high-fidelity honeypots to handle the traffic workload of networks containing IP address spaces of millions of hosts.

For more details: "Scalability, Fidelity, and Containment in the Potemkin Virtual Honeyfarm" by Michael Vrible, Justin Ma, Jay Chen, David Moore, Erik Vandekieft, Alex C. Snoeren, Geoffrey M. Voelker, and Stefan Savage. Presented at 20th ACM Symposium on Operating System Principles (SOSP), Brighton, UK, October 2005.