

Appears in Proceedings of the Second IEEE International Conference on Autonomic Computi

to remedy that, but depends on a system, such as AutoPod, to provide the decoupling infrastructure that's needed.

To support the AutoPod abstraction design of secure and isolated namespaces on commodity operating systems, we employ a virtualization architecture that operates between applications and the operating system, without requiring any changes to applications or the operating system kernel. This virtualization layer is used to translate between the AutoPod namespaces and the underlying host operating system namespace. It protects the host operating system from dangerous privileged operations that might be performed by processes within the AutoPod, as well as protecting those processes from processes outside of the AutoPod on the host.

The key pod virtualization mechanisms used are a system call interposition mechanism and the `chroot` utility with file system stacking to provide each pod with its own file system namespace that can be separate from the regular host file system. Pod virtualization support for migration is based on Zap [2]. Pod virtualization is very light weight, imposing on average less than 10% performance hit on microbenchmarks, while imposing less than 2% overhead for real application scenarios such as outlined by TPC-W.

To support migration across different kernels, AutoPod use a checkpoint-restart mechanism that employs an intermediate format to represent the state that needs to be saved on checkpoint. On checkpoint, the intermediate format representation is saved and digitally signed to enable the restart process to verify the integrity of the image. Although the internal state that the kernel maintains on behalf of processes can be different across different kernels, the high-level properties of the process are much less likely to change. We capture the state of a process in terms of higher-level semantic information specified in the intermediate format rather than kernel specific data in native format to keep the format portable across different kernels. AutoPod is able to checkpoint and restart real applications much faster than restart could take on its own. While starting a full desktop environment from scratch can take nearly 20 seconds, checkpointing and restarting take only 851 ms and 942 ms respectively.

3 Autonomic System Status Service

Many operating system vendors provide their users with the ability to automatically check for system updates and to download and install them when they become available. Example of these include Microsoft's Windows Update service, as well as Debian based distribution's security repositories. User's are guaranteed that the updates one gets through these services are genuine because they are verified through cryptographic signed hashes that verify the contents as coming from the vendors. The problem with

these updates is that some of them require machine reboots; In the case of Debian GNU/Linux this is limited to kernel upgrades. We provide a simple service that monitors these security repositories. The autonomic service simply downloads all security updates, and by using the pod's checkpoint/restart mechanism enables the security updates that need reboots to take effect without disrupting running applications and causing them to lose state.

Commodity systems also provide information about the current state of the system. Subsystems, such as a hard disk's Self-Monitoring Analysis Reporting Technology (SMART), let an autonomic service monitor the system's hardware state. Similarly, the kernel on the machine monitors the state of the system, and if irregular conditions occur, such as DMA timeout or needing to reset the IDE bus, will log this occurrence. Our autonomic service simply monitors the kernel logs to discover these irregular conditions. When the hardware monitoring systems or the kernel logs provide information about possible pending system failures, the autonomic service simply checkpoint the pods running on the system. It then migrates the pod to a new system to be restarted on, ensuring that no state is lost.

4 Conclusions

The AutoPod system provides an operating system virtualization layer that decouples process execution from the underlying operating system, by running the process within a Pod. AutoPod can transparently migrate isolated applications across machines running different operating system kernel versions. This enables security patches to be applied to operating systems in a timely manner with minimal impact on the availability of application services. We have implemented AutoPod on Linux without requiring any applications changes.

References

- [1] D. E. Lowell, Y. Saito, and E. J. Samberg. Devirtualizable virtual machines enabling general, single-node, online maintenance. In *Eleventh International Conference on Architectural Support for Programming Languages and Operating Systems*, October 2004.
- [2] S. Osman, D. Subhraveti, G. Su, and J. Nieh. The Design and Implementation of Zap: A System for Migrating Computing Environments. In *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002)*, Boston, MA, Dec. 2002.
- [3] E. Rescorla. Security holes... Who cares? In *Proceedings of the 12th USENIX Security Conference*, Washington, D.C., Aug. 2003.
- [4] VMware, Inc. <http://www.vmware.com>.