Paper summary

Virtualization is helpful to partition the ample resources to provide high server utilization. Xen is a type 1 hypervisor that allows multiple commodities operating systems to share conventional hardware safely with little overhead and provide increased functionality. Xen supports unmodified application binaries and does time-sensitive tasks more efficiently. Xen provides an interface to the guest OS different from the underlying hardware, which may not support full virtualization (such as x86). It utilizes the concept of paravirtualization in which the guest OS is "Xen-aware." The guest OS is modified (runs in ring 1) to not invoke any privileged instruction and traps to Xen in ring 0. Just like system call, the guest OS makes an hypercall to XEN in a synchronous manner. Guest OS manages the page table in read-only mode. Xen exists in a 64MB section at the top of every address space to avoid TLB flushes during context switches. When guest OS needs to update, an hypercall to Xen is done to renew the page table in batched fashion. A shared asynchronous ring buffer is used to transfer data between Xen and guest OS. Hence Xen supports multiple operating systems to be virtualized by avoiding the drawbacks of full virtualization and achieves the goal of hosting up to 100 virtual machine instances simultaneously on a modern server.

Strengths

1. User-level application modification is not required.

2. Xen reserves the top 64MB of every address space to save TLB flushes on each context switch giving performance boast.

3. Xen provides a sandboxed environment for hostile workloads.

4. Detailed evaluation results are provided.

5. Effectively utilizes the hardware in data centers without sacrificing reliability and performance.

Weaknesses

1. A more thorough explanation of data transfer using the I/O ring could have been there.

2. Modifying the kernel for paravirtualization can add extra complexity. The OS community would require to maintain two versions of the same OS.

3. The tight coupling between the hypervisor and the guest OS can cause an issue for software updates.

4. Validation of privilege operation by Xen could be a significant overhead.

5. Guest OS runs on ring 1, application on ring 4, and hypervisor on ring 0. However, architectures with only two protection rings can be an issue.

6. Non-open-source operating systems can be an issue. (Later on, Xen 3.0.2 added Intel VT-x and AMD-V support to run the unmodified guest OS on HVM Compatible Processors).

7. Only XenoLinux is tested. Testing for different OS's is needed.

Comments for author

1. Can collaboration between guest os scheduler and hypervisor os scheduler help in scheduling more efficiently?

2. Hardware-assisted virtualization can solve a lot of issues in a fully virtualized environment. So is paravirtualization still relevant?