Paper summary

ESX Server is a small type 1 hypervisor designed to divide hardware resources efficiently among virtual machines running unmodified commodity OS. ESX Server allocates more machine memory than required. When there is a memory shortage, the ESX server reclaims the memory by using various techniques like ballooning, idle memory tax, Content-based page sharing, and hot I/0 page remapping. The balloon driver is installed in the user space of the guest os, and hence no modification is required. It is a pseudo device driver which dynamically inflates or deflates the memory like a balloon, causing guest OS to call its native memory management algorithm and return memory from its free list. It is similar to the self-paging technique in the Nemesis system. Similarly, deflating the balloon frees up memory for the Guest OS. To eliminate repetitive copies of pages in various VM's, it incorporates a content-based page sharing that uses hashing to destroy the copies and put them into shared pages without the involvement of guest OS. ESX reclaims the inactive memory by charging the client with idle tax memory where idle pages are taxed more than active pages with a config parameter defaulting to 75%. ESX ensures a range of target memory allocation, ensuring that at least the "min size" is available even during overcommitment and the "max size" to cap the distribution. The admission control policy ensures enough memory is available for the guest OS to boot. The benchmarks demonstrate an outstanding result for adopting the ESX hypervisor.

Strengths

1. Unlike Disco and Xen, no modification is required in the guest OS.
2. ESX uses the hash table(O(1) time) to find duplicate pages.
3. No guest OS modification.
4. Detailed benchmarks.

Weaknesses

1. An untrusted malicious host OS can uninstall or disable the Ballon driver causing security issues.
2. The randomized page replacement policy needs to be replaced by a better page policy.

Comments for author

A variety of different tax rate and their performance impacts could have been explored. Also, I feel that the ballooning mechanism could have been given in more detail. Other than that, I really liked the idea of using a device driver instead of doing any modification in the guest OS.