Towards agile and elastic bare-metal clouds

Yushi Omote
University of Tsukuba
omote@osss.cs.tsukuba.ac.jp

Takahiro Shinagawa
The University of Tokyo
shina@ecc.u-tokyo.ac.jp

Kazuhiko Kato
University of Tsukuba
kato@cs.tsukuba.ac.jp

Bare-metal clouds are an emerging form of Infrastructure-as-a-Service (IaaS) that offers not virtual machines (VMs) but physical machines (*bare-metal instances*) without using hypervisors. By eliminating the virtualization layer, customers of bare-metal clouds can obtain several benefits of directly running their operating systems (OSs) on unshared physical hardware [1]. For example, bare-metal instances are suitable for highly CPU-intensive and/or I/O-intensive workload such as HPC applications and database servers, where virtualization overhead significantly reduces performance and performance predictability [7]. They also have greater flexibility in hardware configurations such as using dedicated GPU accelerators and specific SSD products.

While abandoning hypervisors brings these benefits, it loses key useful features of virtualization: *agility* and *elasticity*. For example, bare-metal instances do not support *live migration* and *checkpointing* because there is no virtualization layer. In addition, the deployment of bare-metal instances takes tens of minutes to copy OS images and reboot [4] while starting-up VMs usually takes only within ten minutes. The start-up time of instances is an important factor to assist temporary use and responsive scale out of instances in proportion to requests.

Modifying OSs to provide agility and elasticity [2, 6] will lose another IaaS benefit, *OS transparency*; customers are forced to use special OS configurations and disturbed to use self-customized OSs. Enhancing firmwares [7] has difficulty in offering hypervisor-class functionalities like live migration without largely extending hardware. Reducing hypervisor overhead [3] leaves obstacles to bare-metal performance due to, e.g., nested page walk. On-demand virtualization [5] incurs downtime for virtual-physical transitions or largely loses OS transparency due to the changes of hardware interfaces.

Our goal is to provide agility and elasticity without losing OS transparency and continuous virtualization overhead. To this end, we design a *temporarily-virtualizable* hypervisor that normally turns off virtualization and temporarily turns on it on demand to provide the useful features. To seamlessly turn on/off the virtualization, the hypervisor exposes physical hardware interfaces to the guest OS. Instead of emulating devices, it monitors I/Os and carefully modifies the guest OS behavior in accordance with the specification of physical hardware.

To improve agility and elasticity, the hypervisor supports fast start-up of bare-metal instances that improves location transparency by redirecting disk I/Os of the guest OS to the storage server over network and performing background copy to deploy the entire OS image to the local disk, making the guest OS gradually shift from remote to local execution. The hypervisor also supports live migration and checkpointing by monitoring I/Os on the source machine to capture the hardware state and restoring the state on the destination. During normal execution (no start-up and migration), the hypervisor turns off the virtualization for bare-metal performance.

We confirmed our prototype hypervisor booted 32GB-image OSs in a few minutes, finished background copy within 14 minutes under database workloads and seamlessly turned off the virtualization, allowing bare-metal performance. It could seamlessly turn on the virtualization again and intercept I/Os and memory accesses. Our next step is to implement the mechanism to determine the physical hardware state for live migration and checkpointing. The challenge is that some states might be untraceable or unrestorable by I/O observation in the hypervisor. However, based on our current study, we expect they should be mostly uncritical (e.g. statistic registers of hardware errors). Therefore, we believe our approach will achieve agile and elastic bare-metal clouds that do not lose OS transparency and bare-metal performance.

References

[1] BUTLER, B. New bare metal cloud offerings emerging. Network World, Oct 2012.

[2] DAVID, C., ET AL. OS Streaming Deployment. In *Proc. of IPCCC'10* (2010).

[3] GORDON, A., ET AL. ELI: bare-metal performance for I/O virtualization. In *Proc. ASPLOS'12* (2012).

[4] KANG, M., ET AL. General Bare-Metal Provisioning Framework. The OpenStack Summit, Oct 2012.

[5] KOOBURAT, T., AND SWIFT, M. The best of both worlds with on-demand virtualization. In *Proc. of HotOS'11* (2011).

[6] KOZUCH, M. A., ET AL. Migration without virtualization. In *Proc. of HotOS'09* (2009).

[7] MOGUL, J. C., ET AL. The NIC is the hypervisor: bare-metal guests in IaaS clouds. In *Proc. of HotOS'13* (2013).