Towards agile and elastic bare-metal clouds

Yushi Omote†, Takahiro Shinagawa‡, Kazuhiko Kato†
†University of Tsukuba, ‡The University of Tokyo

Background & Problem

Bare-metal clouds emerging.
IaaS offering unshared physical machines.

\[\begin{align*}
\text{Agility} & \quad \text{Elasticity} \\
\text{No virtualization overhead.} & \quad \text{Suitable for HPC and Databases.}
\end{align*}\]

Users

No live migration, checkpointing. No elastic scale out (Slow OS deployment).

Current state-of-the-art

**Approach 1** Re-designing OS
Disturbs user’s self-customization of OSs.

**Approach 2** Enhancing firmware
Needs tremendous hardware extension.

**Approach 3** Reducing hypervisor overhead
Irreducible overhead (e.g. paging).

**Approach 4** P2V & V2P
No continuous virtualization overhead 😊
Causes downtime for P/V switching 😕

Goal

Agility & elasticity without OS modification, continuous overhead and downtime.

Approach

Temporarily-Virtualizable Hypervisor (TVH)

\[\begin{align*}
\text{OS} & \quad \text{I/Os} \\
\text{TVH} & \quad \text{Normally-off virtualization for bare-metal performance.} \\
\text{HW} & \quad \text{Temporarily virtualizes for agility & elasticity.} \\
\end{align*}\]

For quick OS deployment...

Network-boots and deploys OS.

For Live migration/checkpointing...

Captures HW/OS states and restores it.

Evaluation Results

**OS Deployment Test under DB workload**
- Guest OS is booted in 48secs.
- 32GB OS image is deployed in 14mins.
- Bare-metal performance after deployment.

**P/V Switching Test under DB workload**
- No downtime for switching.

Future work

Capturing/restoring HW state.
Challenge is untraceable states but they can be trivial. (e.g. error counters)

Authors’ e-mail: (Omote) omote@oss.cs.tsukuba.ac.jp, (Shinagawa) shina@ecc.u-tokyo.ac.jp, (Kato) kato@cs.tsukuba.ac.jp
Lab. Address: Laboratory of Advanced Research B Room 1128, University of Tsukuba, 1-1-1, Tenno-dai, Tsukuba, Ibaraki, 305-8573, Japan