Profile Guided Kernel Optimization for Individual Container Execution on Bare-Metal Container

Kuniyasu Suzaki, Hidetaka Koie, Ryousei Takano
National Institute of Advanced Industrial Science and Technology
{ k.suzaki | koie-hidetaka | takano-ryousei }@aist.go.jp

1. INTRODUCTION

Container technologies have become popular on super computers as well as data centers. Some tools are proposed (e.g., Singularity, Shifter, CharlieCloud) and deployed on real super computers (e.g., Shifter on "Piz Daint"). They use a container image to package an application, making it easy to customize the computing environment. Unfortunately, container technologies share the kernel for all processes of the containers. Users are not allowed to change the kernel and cannot benefit from kernel optimization, especially Profile Guided Kernel Optimization (PGKO), which optimizes a kernel for an application.

To solve this problem, we have developed Bare-Metal Container (BMC) [2] which is an OS provisioning system (e.g., Ironic of OpenStack and Kadeploy3) and runs a container image on a remote machine with a suitable Linux kernel. Furthermore, BMC allows changes to the kernel and the remote machine, thus facilitating performance comparisons. We utilized this feature and developed a mechanism to apply PGKO to BMC automatically. We measured the effects of PGKO on big data workloads (Apache and Redis) on Broadwell Xeon and Ivy Bridge i7 systems and found better performance using PGKO.

2. PGKO: PROFILE GUIDED KERNEL OPTIMIZATION

Profile Guided Optimization (PGO) is a popular optimization technology that reduces branch mispredictions, cache miss hits, and code size. PGO is supported on current compilers (e.g., Visual Studio, GCC, LLVM/Clang) and used on real applications (e.g., Google Chrome, Firefox). The PGO compiler first creates a binary to take a profile. When the binary runs, it creates a profile that is used by the PGO compiler to create an optimized binary.

The PGO technology is not limited to a user space application. It can also be used to optimize the Linux kernel [3] for the specific behavior of an application in a process called Profile guided kernel optimization (PGKO).

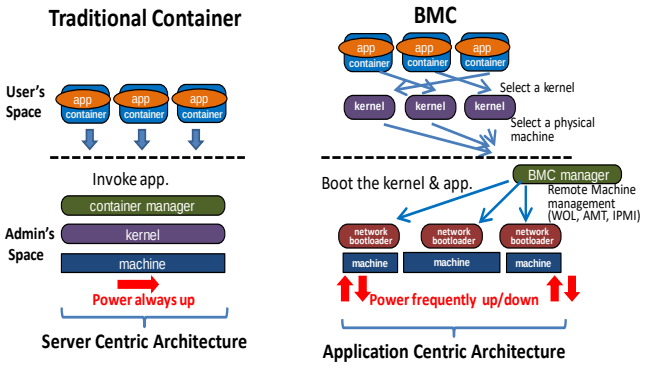

Fig.1. Invocation on tradtion container and BMC.

3. BARE-METAL CONTAINER

Bare-Metal Container (BMC) [2] is a technology to run a container image on a remote machine with a suitable Linux kernel. It allows an application to completely occupy the hardware resources and achieving full performance. BMC allows changes to the kernel and the machine easily. Therefore, the user can compare the effect of the kernel optimization on the machine. Figure 1 compares the invocation between traditional container and BMC.

BMC utilizes remote machine management technologies (IMPI, Intel AMT, and WakeupOnLAN) to power on and off a remote machine. BMC also utilizes network bootloader "iPXE" to control Linux booting. iPXE has an original scripting language and allows users to select a Linux kernel and initrd from HTTP/HTTPS servers. The container image is deployed on RamFS and used as the root file system.

BMC requires booting a Linux kernel for each trial, which is additional overhead. However, the performance increase from the optimized kernel used with HPC applications compensates for this overhead. The details of this is reported in [2].

4. BMC CUSTOMIZATION FOR PGKO

In order to use an optimized Linux kernel, PGKO requires two boot cycles. BMC allows the creation of a shell script to control these boot processes. During the first boot, a profile is taken with a test run of the application. This profile, which is stored at Linux DebugFS under /sys/kernel/debug/gcov/, is copied by the BMC. After that, the BMC re-compiles the kernel with the profile to create the optimized kernel. The second boot on BMC uses the optimized kernel with the application execution. These steps to optimize the execution environment are all automatically handled by the shell script.

5. EXPERIMENTS

PGKO was applied on an HTTP server benchmark "Apache Bench" and a database benchmark "Redis Bench". Profiles for Apache Bench and Redis Bench were taken for a benchmark size of 1,000,000 and 50,000,000 respectively. BMC measured the performance on a sever with Intel Broadwell Xeon E5-2630v4 2.20GHz and 64GB memory and a note PC with Intel Ivy Bridge i7-3520M 2.90GHz and 16GB memory. The results are shown in Figure 2 where Y-axis is the elapsed time and X-axis is the benchmark size. Five trials for each benchmark size were measured. The figures compared elapsed times on the normal kernel and optimized kernel. The optimized kernel was compiled on a separate machine and this overhead is omitted in order to assess the improvement only due to kernel replacement.

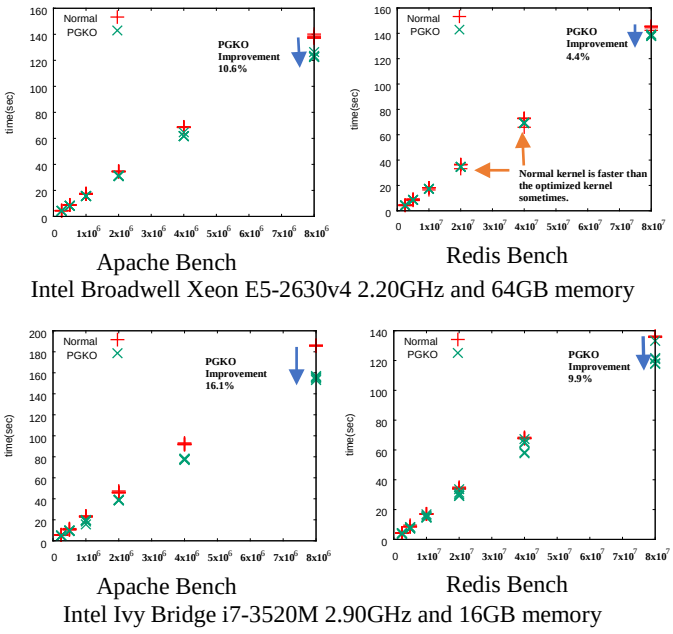

Fig 2. Performance comparison of Apache Bench and Redis Bench on the normal kernel and PGKO kernel.

The optimized kernel for Apache Bench improved the performance about 10.6% and 16.1% on the Xeon and i7, respectively. The time consumed by the kernel was measured by "strace -c" command. The result shows improved CPU time is 4.8% and 12.3% on Xeon and i7, respectively. The other improvements assumed to be waiting time of I/O.

The results on Redis were different from Apache. The results from the i7 showed clear improvement (9.9%), but the results from the Xeon showed uncertain improvement. The optimized kernel improved performance in general, but sometimes the native kernel showed better performance than the optimized kernel (e.g., at benchmark size $2x10^7$ and $4x10^7$).

6. DISCUSSION

The overhead of compiling an optimized kernel requires some time making it unsuitable for the single execution of an application. However, PGKO is useful when the target application is used repeatedly. Future work will investigate this further. We also discovered that the Intel Broadwell architecture improves branch prediction and may invalidate the software optimization. This may explain the discrepancy in the results when using Redis.

7. CONCLUSIONS

Bare-Metal Container (BMC) is a technology to run a container image on a remote machine with a suitable Linux kernel. This paper proposed a method to apply Profile Guided Kernel Optimization to BMC and found increased performance of the application. The source code of BMC is available as open source [1].

ACKNOWLEDGEMENT

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

REFERENCES

[1] BMC: https://github.com/baremetalcontainer/bmc

[2] K.Suzaki, H.Koie, and R.Takano, Bare-Metal Container: Direct Execution of a Container Image on a Remote Machine with an Optimized Kernel, High Performance Computing and Communications (HPCC) 2016.

[3] P. Yuan, Y. Guo, and X. Chen, Experiences in profile-guided operating system kernel optimization, Asia-Pacific Workshop on System (APSys), 2014.