ビッグメモリワークロードを考慮した GPU による仮想マシン移送高速化手法

直 井 由 樹 † 山 田 浩 史†

1. Backgroud

Live Virtual Machine (VM) migration is a promising approach for managing virtualized resources in cloud platforms such as IaaS environments. Live migration enables us to move a running VM between different physical machines without long downtime and loss of any running state such as network connections. VM replacement policies assuming the use of live migration such as load balancing and power saving have been studied in research communities.

However, live VM migration under modern workloads is a significant time- and resource-consuming task, resulting in ineffective data center operations even if the sophisticated policies output a better VM placement. Since emerging big memory applications such as in-memory databases and graph analysis requires a huge amount of memory, the memory size of the VMs becomes bigger and thus live migration takes longer to transfer the memory pages of such VMs.

2. Approach

To mitigate time- and resource-consumption of live migration, this paper presents an approach to accelerate the live migration execution with modern co-processors. Our approach exploits general purposed GPUs (GPGPUs) and borrows CUDA libraries sophisticated in the high performance community. In the approach, the hypervisor compresses/decompresses the VM's memory with the GPU on the source and destination respectively, and thus decreases an amount of transferred memory. Reducing network loads contributes to shorten migration time, leading to lower memory dirty rate in each iteration. The reduction also lowers CPU consumption related to TCP transfers, which is a major CPU usage of live migration. To the best of our knowledge, this is the first work to leverage GPU acceleration at the hypervisor.

To effectively utilize GPUs, we take asynchronism of between CPU and GPU execution. We change the architecture of the pre-copy live migration, which is a widely accepted live VM migration method, into a *pipelined architecture*. The execution of live VM migration on the source can be divided into three phases; 1). copy phase where pages of the migrated VM are copied into address spaces of the hypervisor, 2). serialize phase where the copied pages are formatted, and 3). send phase where the data is transferred. The regular pre-copy migration iterates the three phases sequentially, while our design pipelines each phase. Specifically, after the copy phase, we start another copy phase and the serialize phase. Our serialize phase performs compression of pages on a GPU, waits for compressed data, and then serializes it. Finally, the send phase transfers the serialized data to the destination. Our migration execution on the destination is also pipelined like the source one; 1). receive phase, 2). deserialize phase, and 3). restore phase. When receiving the transferred data, we decompress the data on a GPU and deserializes the uncompressed data. And then, the pages are mapped to restart the migrated VM on the source.

3. Current Status

A prototype has been implemented into Xen 4.2.1, uses a delta compression technique to compress/decompress VM memory like this work[2], and makes use of Gdev[1] to access a GPU. We conducted a preliminary experiment using NVIDIA Quadro 6000. In the experiment, we migrate a VM whose memory size is 16 GB. The result shows that migration time of the prototype is 3 to 4x faster than one of the Xen default migration and reduce CPU usage up to 60%.

† 東京農工大学
Tokyo University of Agriculture and Technology

参 考 文 献

1) S. Kato, M. McThrow, C. Maltzahn, and S. Brandt. Gdev: First-Class GPU Resource Management in the Operating System. In *Proc. of the USENIX ATC '12*, pages 401–412.
2) P. Sväard, B. Hudzia, J. Tordsson, and E. Elmroth. Evaluation of Delta Compression Techniques for Efficient Live Migration of Large Virtual Machines. In *Proc. of the VEE '11*, pages 111–120.