

発表概要

物理メモリ・オーバーコミット環境下における Docker コンテナのクラウドサーバ向けメモリ割当制御

緒方 一則^{1,a)} 中澤 里奈¹ 小野寺 民也¹

2016年10月26日発表

クラウド・データセンター事業者では、システムの容量や性能の向上だけでなく、運用効率を高めることも重要な課題である。運用効率を高めるための手法として、リソースのオーバーコミットがある。クラウドサーバでは、多くのユーザは割り当てられたリソースの一部しか実際には使用しないので、物理的に確保できる以上のリソースをユーザに割り当てることでリソースを有効利用する手法である。我々は、運用効率をさらに高めるために、リソース、特にメモリのオーバーコミット率を、プロセスの一部をスワップアウトする程度にまで高める研究をしており、多数の Web アプリケーション・サーバが動作しているシステムでの有効性を示してきた。Linux コンテナの一般的な実装では、スワップの積極度を決めるカーネル・パラメータ `swappiness` をコンテナごとに設定できることを利用し、コンテナの優先度に応じて `swappiness` を制御することで、システム全体のスループットやレスポンスタイムを維持する。本発表では、Web アプリケーション・サーバとバックエンドサービスのコンテナが共存する環境における、コンテナごとの `swappiness` の制御手法を示す。バックエンドサービスの例として SparkBench を用いた。

Memory Allocation for Docker Containers on a Cloud Server when Memory is Physically Overcommitted

KAZUNORI OGATA^{1,a)} RINA NAKAZAWA¹ TAMIYA ONODERA¹

Presented: October 26, 2016

Datacenter efficiency is an important factor for cloud service providers, as well as its capacity and performance. One of the techniques to improve efficiency is to overcommit resources, where the amount of resources allocated to users is larger than that are physically available on the servers. We study techniques to improve datacenter efficiency by overcommitting memory as high degree as a part of running processes is swapped out, while the degradation of system's overall throughput and response time remains within tolerable amount. We have reported such technology is effective when multiple Web application servers are running in a machine. Linux has a kernel parameter called "swappiness" to give a hint to the kernel how eager it swaps out memory, and we utilized a common implantation of Linux containers where we can set different swappiness value for each container. In this presentation, we will explain how to manage per-container swappiness when containers of Web application servers and those of backend services are running in a single machine. We used SparkBench as an example of the background services running in the containers.

¹ 日本 IBM 株式会社東京基礎研究所
IBM Research – Tokyo, IBM Japan Ltd., Chuo, Tokyo 103-8510, Japan

a) ogatak@jp.ibm.com