

発表概要

メモリオーバーコミット環境下における Docker コンテナの性能変動の低減

中澤 里奈^{1,a)} 緒方 一則¹ 小野寺 民也¹

2015年8月5日発表

軽量の仮想化技術である Linux コンテナが注目を集めている。クラウドプロバイダは物理マシン上に可能な限り多くのコンテナをつくり、リソースを効率良く使用したいと考えるため、大規模なマルチテナントと高速なデプロイを実現するコンテナ技術に期待している。一方で、オーバーコミットによってパフォーマンスが受ける影響や、その影響をどのように緩和させることができるかが十分に理解されていないため、プロバイダは、コンテナのパフォーマンス劣化を恐れてリソースのオーバーコミットを避け、保守的なリソース割当てを行う傾向にある。そこで、我々はメモリオーバーコミット環境下におけるコンテナの性能のばらつきに対して定量的な評価実験を行った。Linux コンテナの一実装である Docker コンテナに Web アプリケーションをデプロイし、それを複数たちあげ、そのすべてあるいは一部に対し 2 種類のレベルで負荷をかけた。実験の結果、メモリオーバーコミット環境下ではメモリアクセスの割合にかかわらずすべてのコンテナがスワップアウトしてしまっているため、負荷の低いコンテナ同様、負荷の高いコンテナのパフォーマンスのばらつきが著しいことが分かった。また、負荷の高いコンテナに対し cgroup のパラメータに適切な値を与えることは、スループットとレイテンシのばらつきを減らすのに非常に有効であることを確認した。

Taming Performance Variance of Docker Containers Under Memory Overcommitment

RINA NAKAZAWA^{1,a)} KAZUNORI OGATA¹ TAMIYA ONODERA¹

Presented: August 5, 2015

Linux containers are getting increasingly popular, due to its light-weight virtualization. Cloud providers embrace the technology as an enabler of massive multi-tenancy. While providers are eager to deploy as many containers in a physical machine as possible, they avoid overcommitting resources, fearing that it degrades the performance of all or some of the containers. In doing so, cloud providers tend to make conservative allocations of resources because it is not necessarily well understood how much the overcommitment makes impact on the performance and how the impact can be mitigated. This paper presents our quantitative study of performance variance of under memory overcommitted situation. Concretely, we deployed many instances of a Web application in Docker containers, and drove all or some of them at two different load levels. We observed significant performance variance under memory overcommitment for highly-loaded containers, as well as lightly-loaded ones, because all containers were swapped out, regardless of the memory access rate. We also found that setting an appropriate value of a cgroup parameter for the highly-loaded container was surprisingly effective in reducing the variance of the container both in throughput and latency.

¹ 日本アイ・ビー・エム株式会社東京基礎研究所
IBM Research - Tokyo

^{a)} rina@jp.ibm.com