Practicality and Feasibility of Improving Linux Container Utilization with Task Rebalancing Strategy

Laboratory for Software Design & Analysis
established in 2005

Pongsakorn U-chupala, Yasuhiro Watashiba,
Kohei Ichikawa, Susumu Date

Introduction

Linux container provides high-performance lightweight computing resource allocation and isolation [1]. Unlike virtualization, Linux container does not require hypervisor layer and guest os. Instead, each container runs natively on the host allowing it to achieve near-native performance. This technology has been adopted by leading high-performance computing company such as Google for many years [2]. By adopting the technology in the computer cluster, task spin-up time is significantly decreased. Since task spin-up time is a major cost of task migration, with reduced task spin-up time, task migration is cheaper resulting in task rebalancing strategy becomes a viable technique to improve cluster utilization.

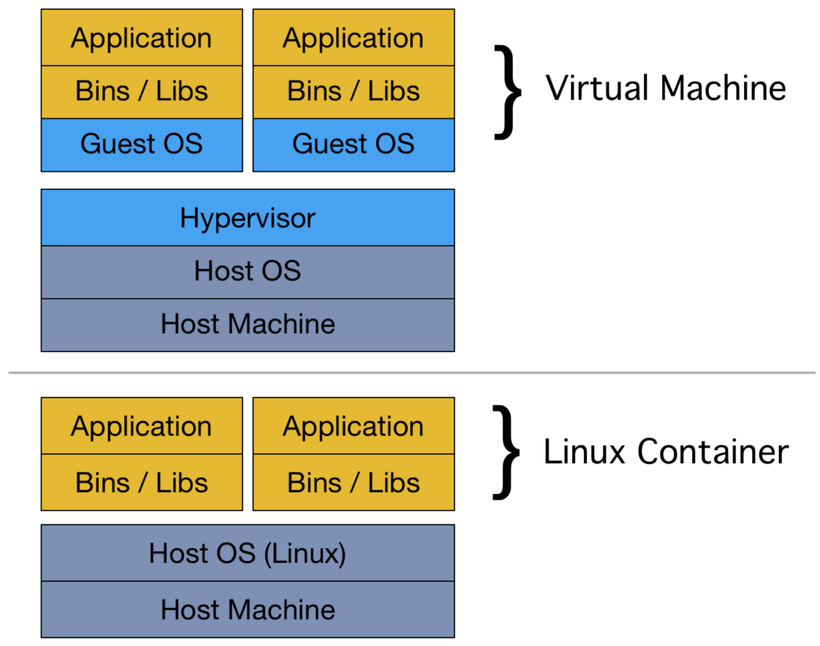

Figure 1. Comparison between virtual machine and Linux container

Task Rebalancing Strategy

Tasks in any cluster system could be categorized roughly into 2 types, short-lived batch task and long-lived service task. Although the majority of tasks in a cluster are batch task, they consume a small amount of resources. On the contrary, service tasks, although small in number, take up most of the resources of a cluster [3]. Therefore, optimizing resource allocation of service task can have a significant impact on the overall utilization of a cluster.

In traditional scheduling, a scheduler tries to find the most optimal placement for new task; allocated host machine then executes scheduled task until the end of task's life cycle. This strategy works fine for batch task. However, for service task, task placement calculated at the beginning of execution may not stay optimal during the course of task's lifetime, as the other tasks are moving in and out of the cluster. Task rebalancing improve the scheduling process by allowing scheduled task to be migrated to another host machine. Task placement is constantly reevaluated to reflect the current status of the cluster.

Task rebalancing is achieved by swapping only tasks on different host machines with comparable resources allocation and significant resource usage difference. This method is utilized to preserve amount of allocated resources and available resources on each host machine in order to minimize interference to the concurrently running scheduler. With task rebalancing, hosts utilization is load-balanced, resulting in higher possibility of overcommitting success thus increasing optimal overcommit factor and overall cluster utilization.

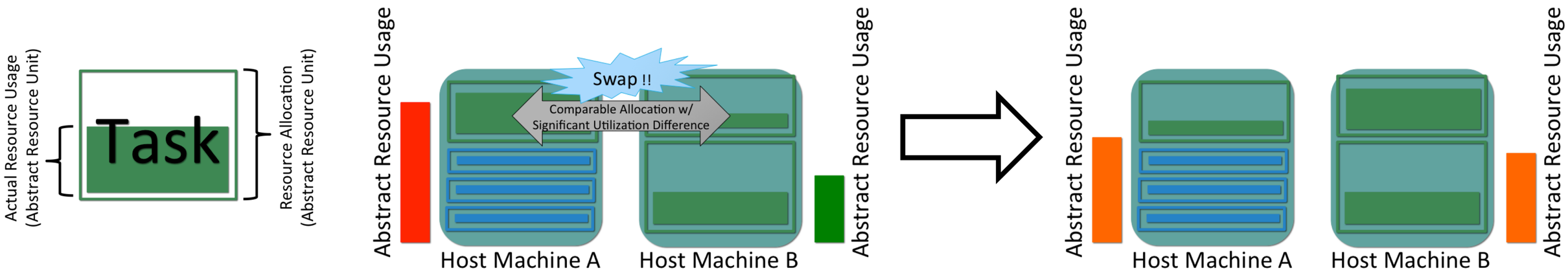

Figure 2. Example of task rebalancing in action

Evaluation & Preliminary Results

Although theoretically possible, feasibility and practicality of using task rebalancing strategy remain unproven. We implemented a simulation to assert the feasibility and practicality of using task rebalancing strategy with container cluster. The simulation is driven by real container cluster trace data provided by Google [4]. We are also looking for others trace data provider. While higher resolution simulations are still being performed, preliminary simulation results suggest an increase in service (long-running) task completion rate.

[1] S. Soltesz, S. Soltesz, H. Pötzl, H. Pötzl, M. E. Fiuczynski, M. E. Fiuczynski, A. Bavier, A. Bavier, L. Peterson, and L. Peterson, "Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors," SIGOPS Oper. Syst. Rev., vol. 41, pp. 275–287, 2007.

[2] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. a. Kozuch, "Heterogeneity and dynamicity of clouds at scale," Proc. Third ACM Symp. Cloud Comput. - SoCC '12, pp. 1–13, 2012.

[3] M. Schwarzkopf and A. Konwinski, "Omega: flexible, scalable schedulers for large compute clusters," EuroSys '13 Proc. 8th ACM Eur. Conf. Comput. Syst., pp. 351–364, 2013.

[4] J.Wilkes and C. Reiss. Details of the ClusterData-2011-1 trace, 2011. https://code.google.com/p/ googleclusterdata/wiki/ClusterData2011_1.