A Middleware Framework for Adaptive Migration of Virtualized Applications and Datasets

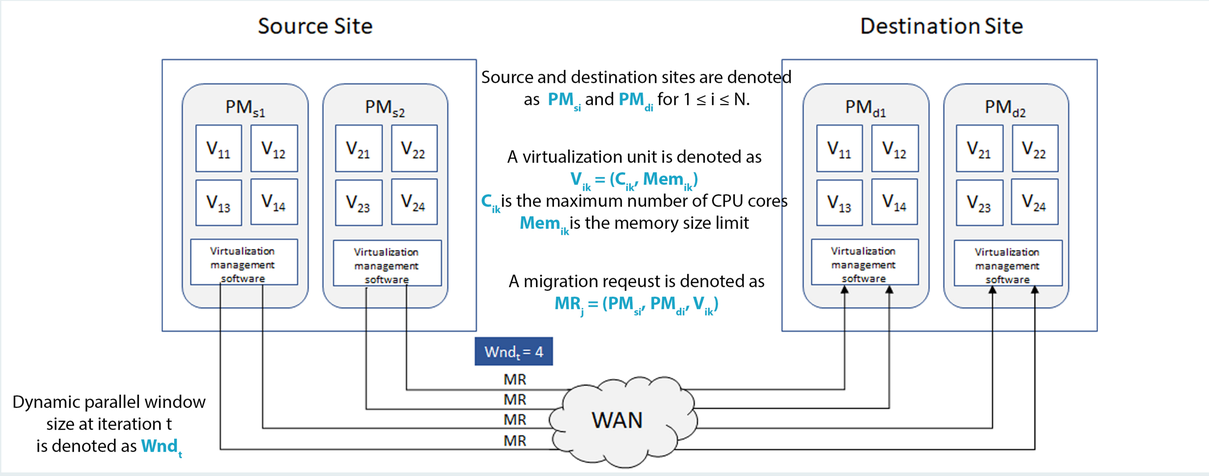
Wongsatorn Thongthaworn, Prapaporn Rattanatamrong

Department of Computer Science, Faculty of Science and Technology, Thammasat University

Motivation

- Application migration in Wide Area Network (WAN) is needed in many scenarios: **disaster recovery** and **service hand-off in edge cloud.**
- We focus on **parallel multi-container migration in WAN environments** by utilizing multiple TCP connections over a single direct path (a.k.a parallel TCP).
- We propose **a middleware framework** for adaptively migrating virtualized applications across dynamically changing networks that is generalized and not restricted to any particular virtualization technology implementation.

Problem Statement



To achieve the best performance combination between total migration time (TMT) and average individual migration time (TMT)

1. What is the **appropriate sizes of the parallel window (Wnd_t)**?
2. For each parallel migration, how to **schedule migration requests** so that traffic loads are balanced among hosts?

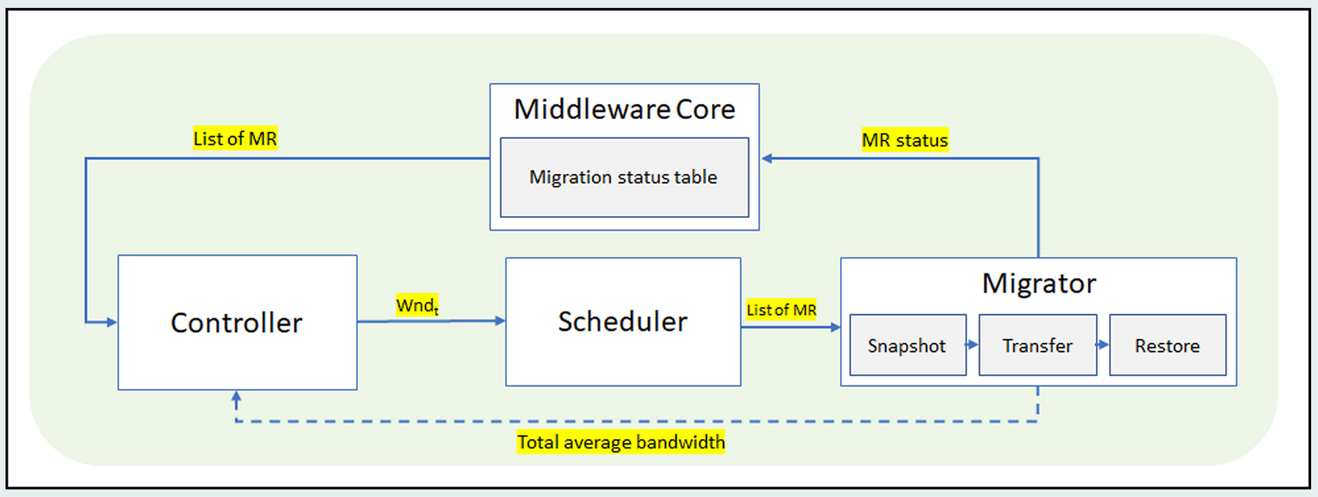
Middleware Design

1. The middleware core

The middleware core contacts another middleware core at the migration destination to request for a list of mapped hosts in the destination site. Then, it creates migration requests for all virtualization units and updates the status of each migration request in a migration status table.

2. The feedback-based controller

A closed-loop feedback control to find an appropriate number of parallel migration requests at a different point in time that can efficiently utilize current network



3. The load balancing scheduler

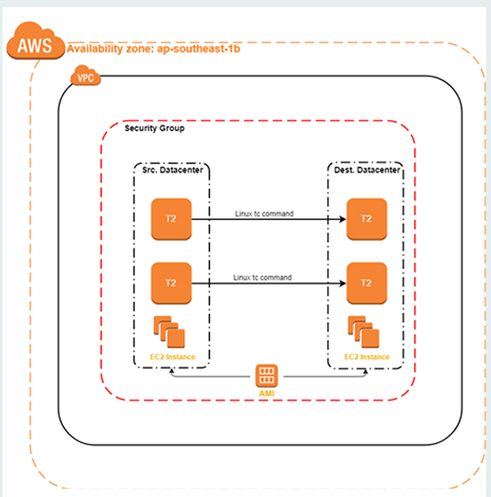
The scheduler is designed to choose a number of requests indicated by the parallel window size for each time window in such a way that balances out migration workload across hosts.

4. The migrator

The migrator executes a number of migration request chosen by the scheduler in each time window. The execution of a migration request starts with assistance of the virtualization management software in the source host to snapshot, transfer and restore virtualzation unit to specified in the migration request.

Performance Evaluation

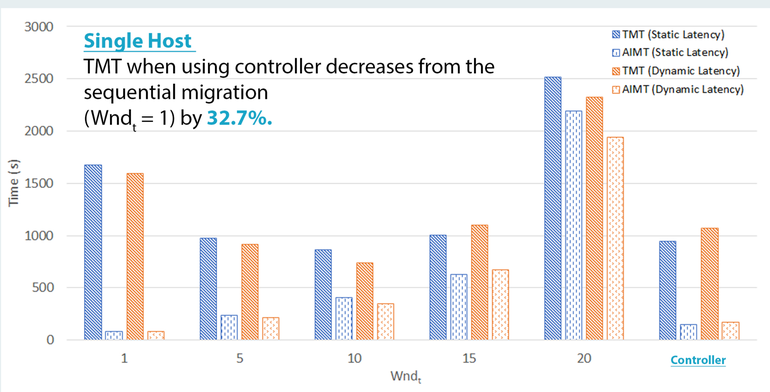
Our middleware prototype is implemented by shell scripts and LXC commands.

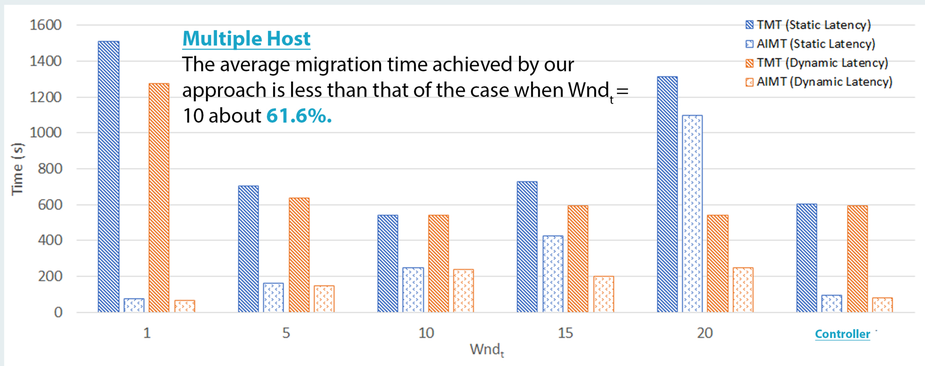


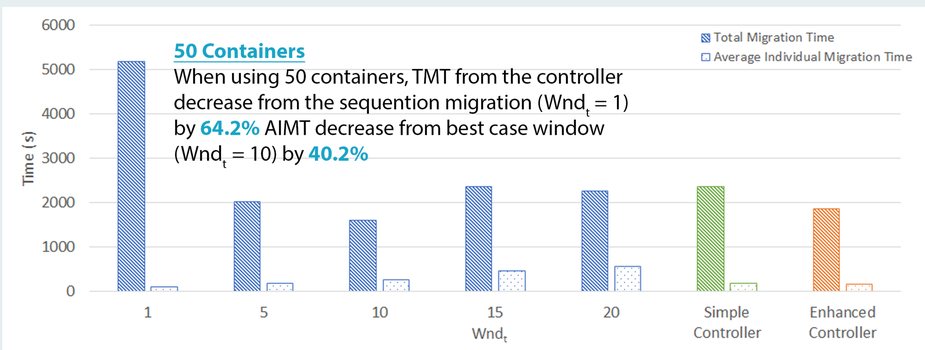
mirgate 20 containers
(each container size is 600MB)

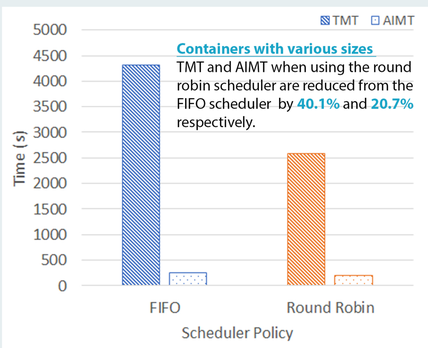
Experimental Setup

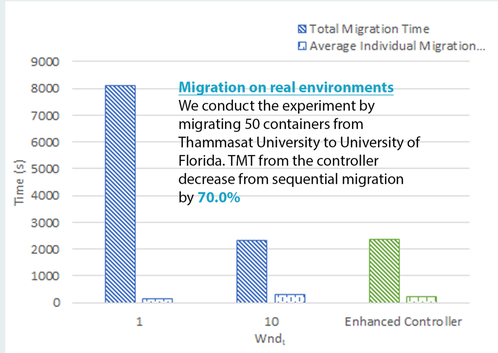
The setup includes two clusters. Each site install a middleware node and two LXC container hosts. We have two network settings: static and dynamic networks. In both network settings, there is no packet loss.











Conclusion & Future Work

- Our approach also improves the combination of total migration time and average individual migration time in comparison to using the best case fixed parallel window for both static and dynamic networks.
- We are improving the controller to react more promptly or proactively to changing network behavior. This will increase the chance to avoid network congestion.
- Scheduling parallel migration on heterogeneous hardware or virtualization units require further investigation in the future.

Refences

[1] T. S. Kang, M. Tsugawa, A. Matsunaga, A. B. J. Fortes and T. Hirofuchi, "Reducing the Migration Times of Multiple VMs on WANs," in IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum, 2013.
[2] T. S. Kang, M. Tsugawa, A. Matsunaga, T. Hirofuchi and A. B. J. Fortes, "Design and Implementation of Middleware for Cloud Disaster Recovery via Virtual Machine Migration Management," in IEEE/ACM 7th International Conference on Utility and Cloud Computing, 2014.
[3] A. Machen, S. Wang, K. K. Leung, B. J. Ko and T. Salonidis, "Live Service Migration in Mobile Edge Clouds," in IEEE Wireless Communications, vol. 25, no. 1, pp. 140-147, February 2018.

Pacific Rim Applications and Grid Middleware Assembly (PRAGMA 37), San Diego, CA, USA, September 11-14 2019

Acknowledgement

Funded through a grant from the Thailand Research Fund (TRF) and Thai Network Information Center Foundation (THNIC Foundation) #TRG5580020.