

氏 名 高橋 公俊

学位(専攻分野) 博士(情報学)

学位記番号 総研大甲第 2114 号

学位授与の日付 2019 年 9 月 27 日

学位授与の要件 複合科学研究科 情報学専攻
学位規則第6条第1項該当

学位論文題目 A Study on Portable Load Balancer for Container Clusters

論文審査委員 主 査 教授 合田 憲人
准教授 竹房 あつ子
准教授 鯉淵 道紘
准教授 栗本 崇
教授 横山 重俊
群馬大学総合情報メディアセンター

(様式3)

博士論文の要旨

氏 名 高橋 公俊

論文題目 A Study on Portable Load Balancer for Container Clusters

Today, most of the people in the world can not spend a day without using smartphones or PCs. They use these devices to access services provided by web applications on the Internet. These services include e-mail, social media, search engines, shopping site, etc., everything provided through the Internet. As these services become an indispensable part of the daily lives, portability of the application becomes very important.

For example, those who provide these services need to be able to recover from a disaster by migrating their web applications to different locations. They also need to be able to expand their businesses to different countries, once the web service is successful in one country. It is also preferable for them to be able to migrate their services without a hassle at their convenience in order to avoid lock-ins.

For the portability of web applications, providing a web application consisting of a cluster of Linux containers is a promising candidate, since Linux containers can run on any Linux system regardless of the infrastructures. A container orchestrator (also called container cluster management system) is a tool to simplify the management of a cluster of containers that are launched on multiple servers. And it is expected to provide a uniform platform for container clusters by functioning as a middleware, which will improve the portability of web applications. However, none of the existing container orchestrators meets the expectation, because none of them has a standard way to set up the route for ingress traffic from the Internet automatically. Users need to set up a route for ingress traffic manually every time they start a new web application, depending on the type of the infrastructure. The lack of this standardized automation is one of the most critical problems that prevent container orchestrators from serving as a common middleware that facilitates the portability of web applications. Without solving this problem, the migration of a web application will never be easy, and will always require manual adjustment to the infrastructures.

In this dissertation, the author addresses this problem by proposing an architecture using a portable software load balancer that can run on any infrastructure. The author provides a cluster of software load balancers in containers that can be launched as a part of web applications for Kubernetes. The architecture is also capable of setting up the route for the ingress traffic automatically by using standard protocols. For this, Equal Cost Multi Path (ECMP) routes are populated through Border Gateway Protocol (BGP) in order to provide redundancy and scalability at the same time. By

using the proposed architecture, web application clusters no longer depend on the load balancers provided by infrastructures. And hence, container orchestrators become being able to better serve as a common middleware.

The author has implemented a containerized software load balancer using Linux kernel's IPVS, and carried out experiments with the following criteria: 1) verify if the proposed load balancer works correctly both in the cloud and the on-premise data center. 2) verify if the proposed load balancer has a sufficient performance level for 1 Gbps and 10 Gbps networks. 3) verify if the proposed redundancy architecture using ECMP with BGP properly functions.

From the results of the experiment, it has been shown that the proposed load balancers can run in an on-premise data center, Google Cloud Platform (GCP), and Amazon Web Service (AWS). Therefore, the proposed load balancers can be said to be portable.

In the case of 1 Gbps network environment, the throughput of the IPVS in a container with Layer 3 Direct Server Return(L3DSR) setting has been about 1.5 times higher than that of existing iptables DNAT rules, which is prepared by Kubernetes's daemons as an internal load balancer. And it has been shown that the proposed load balancer has more than enough throughput to fill up 1 Gbps bandwidth. In the case of 10 Gbps network environment, while a single IPVS load balancer in the container can provide only 1/4 of required throughput, ECMP setups using more than four of them can deal with 10 Gbps equivalent of the traffic. Therefore, the proposed load balancer has been proven to be portable with sufficient performance in both 1 Gbps and 10 Gbps network environments.

The author has also verified that ECMP routes are properly created on the upstream router, upon launch of new load balancer containers. The update of the ECMP routing table was correct and quick enough, i.e., within 10 seconds, throughout 20 hours experiment. The maximum performance level of a cluster of load balancers has scaled linearly up to four times as the number of the load balancer containers has been increased up to four. The maximum aggregated throughput obtained through the experiment is 780k [req/sec], which is limited by the CPU performance of the benchmark client and can be improved using better hardware in the future experiment. Therefore, the author has proved that proposed load balancer has the capability of the automatic setup of ingress traffic in a redundant and scalable manner.

Sooner or later, the day when the network in a data center becomes all 100 Gbps will come. Therefore, it is essential to improve the performance of the portable load balancers in future work. The author has started to implement a novel software load balancer using eXpress Data Path (XDP) technology. The preliminary result, where the maximum throughput is about 390K [req/sec] with single-core packet processing, indicates that this technology is very promising. The author estimates that about five of the software load balancers using this technology with 16 core packet processing can

provide enough throughput in 100 Gbps environments in the future.

The proposed load balancer has been verified to be portable while providing sufficient throughput in 10 Gbps environment. And the proposed redundancy architecture using ECMP with BGP has also been verified to function properly. As a consequence, the proposed architecture with this load balancer will help improve the portability of web applications.

The outcome of this study will benefit users who want to improve the portability of web applications and deploy them anywhere they want. Moreover, the result of this study will potentially benefit users who want to use a group of different cloud providers and on-premise data centers across the globe seamlessly. In other words, users will become being able to deploy a complex web service on aggregated computing resources on the earth, as if they were starting a single process on a single computer.

博士論文審査結果

Name in Full
氏名 高橋 公俊

論文題目 A Study on Portable Load Balancer for Container Clusters

本学位論文は、「A Study on Portable Load Balancer for Container Clusters」と題し、全7章から構成されている。

第1章「Introduction」では、本研究の背景と目的を述べている。ウェブサービスを実装するために、計算機仮想化技術の一つであるコンテナを複数束ねたコンテナクラスタシステム上でアプリケーションプログラムを実行する方式が普及している。出願者は、BCP対策等の目的でコンテナクラスタシステムを異なる計算基盤（クラウドやオンプレミスシステム）間で容易に移動可能とすることを目的としたポータビリティの重要性に着目するとともに、既存技術ではポータビリティの向上にロードバランサに起因する問題があることを指摘し、本学位論文の目的が、この問題点を解決するためのロードバランサアーキテクチャの提案であることを述べている。

第2章「Background」では、本学位論文が対象とするコンテナクラスタシステム、およびそれに関連するネットワーク、ロードバランサ技術について解説している。

第3章「Architecture and Implementation」では、既存技術におけるコンテナクラスタシステムのポータビリティの問題点を解説するとともに、問題点を解決するためのロードバランサのアーキテクチャ、ならびにその実装手法を提案している。既存技術では、コンテナクラスタシステムを異なる計算基盤に移動する度に、異動先の計算基盤の仕様に合わせて、管理者がロードバランサに関する通信の設定を実施する必要があり、ポータビリティを低下させる要因となっていた。これに対して提案手法では、ロードバランサをコンテナクラスタシステム内に実装し、アプリケーションと同時に異なる計算基盤へ移動するとともに、通信設定を自動化している。また、ロードバランサの冗長化やスケーラビリティの確保を図ることにより、アプリケーションの実行性能の低下を抑えている。

第4章「Performance Evaluation」ならびに第5章「Performance in faster network」では、提案手法の性能評価結果を示し、その有効性を述べている。性能評価では、提案手法を用いて実装されたコンテナクラスタシステム上のアプリケーションが、オンプレミスシステムならびに二つのパブリッククラウド上でそれぞれ同等の性能で実行可能であることを示している。また、ロードバランサの冗長化により高負荷時でも高い性能を維持できること、ならびに急激な負荷の上昇に対しても対応可能でスケーラビリティを確保できることを示している。最後に、これらの実験結果より、提案手法が、既存技術に比べて同等の性能を保ちつつ、コンテナクラスタシステムのポータビリティの向上に成功していると結論づけている。

第6章「Related Work」では、本研究の関連研究について解説し、第7章「Conclusions」では、本研究を総括し、得られた成果および今後の課題について述べている。

以上を要するに本学位論文は、コンテナクラスタシステムのポータビリティを向上させることにより、計算基盤間でのウェブサービスの移動を容易にする手法を提案するとともに、提案手法の実装ならびに性能評価により、その有効性を示したものである。これらの成果は、様々な IT サービスの BCP 対策やベンダロックインという課題の解決につながるものである。また、本学位論文の成果は、学術雑誌論文 1 件、フルペーパー査読付国際会議 1 件他として発表され、学術的な貢献も認められる。以上の理由により、審査委員会は、本学位論文が学位の授与に値すると判断した。