การประเมินสมรรถนะและเปรียบเทียบประสิทธิภาพอินเกรสคอนโทรลเลอร์บนระบบคลัสเตอร์คูเบอร์เนทิส

PERFORMANCE EVALUATION AND COMPARISON OF INGRESS CONTROLLERS ON KUBERNETES CLUSTER

อาธิป พวงลำใย[1] และ ชัยพร เขมะภาตะพันธ์[2]

[1] นักศึกษาปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์, arthip.p@gmail.com

[2] อาจารย์ที่ปรึกษา สาขาวิชาวิศวกรรมคอมพิวเตอร์และโทรคมนาคม วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์, chaiyaporn@dpu.ac.th

---

บทคัดย่อ

ปัจจุบันมีระบบโหลดบาลานซ์บนสภาพแวดล้อมการทำงานของคอนเทนเนอร์คลัสเตอร์ด้วยคูเบอร์เนทิสหลายรูปแบบ แต่ละแบบมีการทำงานและความสามารถที่แตกต่างกันไป ดังนั้นบทความนี้ได้นำเสนอการติดตั้งและทดสอบ เลเยอร์-7 โหลดบาลานซ์ คอนโทรลเลอร์ ที่ทำหน้าที่เพื่อกำหนดเส้นทางในการเข้าใช้บริการของระบบที่ให้บริการอยู่บนการประมวลผลแบบคอนเทนเนอร์คลัสเตอร์คูเบอร์เนทิส จำนวน 4 รูปแบบ คือ 1) Nginx Ingress controller 2) Traefik Ingress controller 3) Voyager Ingress controller และ 4) GCE L7 load balancer controller (GLBC) เพื่อประเมินสมรรถนะและเปรียบเทียบประสิทธิภาพ ทางด้านความสามารถในการให้บริการและความเร็วในการตอบสนอง

ผลการทดสอบพบว่า ในด้านความสามารถในการให้บริการ GCE L7 load balancer controller (GLBC) สามารถให้บริการได้มากกว่า 5,000 requests/second บนระบบที่ใช้ทำการทดสอบ โดยมี Traefik Ingress controller, Voyager Ingress controller และ Nginx Ingress Controller เป็นลำดับถัดมา ที่สามารถให้บริการได้ เฉลี่ย 2,600 requests/second, 2,560 requests/second และ 2,200 requests/second ตามลำดับ ในด้านความเร็วในการตอบสนอง GCE L7 load balancer controller (GLBC) ใช้เวลาในการตอบสนองน้อยที่สุด ตามด้วย Traefik Ingress controller, Voyager Ingress controller และ Nginx Ingress controller ตามลำดับ และความเร็วในการตอบสนองมีแนวโน้มสูงขึ้นเมื่อจำนวนครั้งในการร้องขอข้อมูลต่อหน่วยเวลาเพิ่มมากขึ้น อย่างไรก็ตามถ้าพิจารณาถึงความสะดวกต่อการใช้งานในกรณีที่เลือกใช้ Public Cloud ของ Google Cloud Platform แล้ว GCE L7 load balancer controller (GLBC) จะเป็นตัวเลือกที่สามารถเลือกใช้งานได้สะดวกที่สุด เนื่องจากมีการติดตั้งไว้บน Google Kubernetes Engines อยู่แล้ว โดยที่ไม่ต้องติดตั้งระบบ Ingress Controller เพิ่มเติม

คำสำคัญ: การประมวลผลแบบกลุ่มเมฆ, คอนเทนเนอร์, คูเบอร์เนทิส, โหลดบาลานซ์, อินเกรสคอนโทรลเลอร์

ABSTRACT

Nowadays, there are many methods for load balancing on container clustering using Kubernetes which have a different operation and feature. Thus, this article presents an installation and test of layer-7 load balance controller which functions as an HTTP/HTTPS load balancer for accessing service on Kubernetes containers cluster. There are 4 types as follows: 1) Nginx Ingress controller 2) Traefik Ingress controller 3) Voyager Ingress controller 4) GCE L7 load balancer controller (GLBC). The study will evaluate and compare the performance and efficiency based on traffic throughput and responses time.

The results from this study revealed that the most efficient method in case of throughput is GCE L7 load balancer controller (GLBC) with accounting rate for more than 5,000 requests/second. Traefik Ingress controller, Voyager Ingress controller and Nginx Ingress controller can be accountable for 2,600, 2,560, 2,200 requests/second, respectively. Regarding the response time, it was found that GCE L7 load balancer controller (GLBC) takes the least time to response, followed by Traefik Ingress controller, Voyager Ingress controller, and Nginx Ingress controller, respectively. Additionally, the response time has a high tendency depending on the requests per time unit. However, GCE L7 load balancer controller (GLBC) is the best option for public cloud (Google cloud platform) in terms of its convenience since it was already installed on Google Kubernetes Engines without additional Ingress Controller installation.

Keywords: Cloud Computing, Containers, Kubernetes, Load Balance, Ingress controller

1. บทนำ

เทคโนโลยีสารสนเทศ (Information Technology) เป็นเครื่องมือสำคัญประการหนึ่งในการดำเนินธุรกิจของหน่วยงาน บริษัท หรือผู้ให้บริการ การประยุกต์ใช้เทคโนโลยีสารสนเทศได้อย่างเหมาะสม จะส่งผลให้การดำเนินการมีประสิทธิภาพเพิ่มมากขึ้น สามารถลดระยะเวลาในการทำงานได้อย่างมีนัยยะสำคัญ ลดการใช้ทรัพยากร ลดการใช้พลังงานไฟฟ้า แต่จะสามารถเพิ่มความง่าย ความสะดวกในการใช้งาน และเพิ่มผลผลิตได้มากยิ่งขึ้น

ผู้ให้บริการในปัจจุบันจึงได้ปรับเปลี่ยนรูปแบบในการให้บริการโดยประยุกต์ใช้เทคโนโลยีสารสนเทศ เพื่อเป็นการให้บริการผ่านอินเตอร์เน็ต (Online Services) ในรูปแบบเว็ปแอพพลิเคชั่น โมบายแอพพลิเคชั่น หรือการให้บริการในรูปแบบออนไลน์อื่น ๆ มากยิ่งขึ้น เพื่อให้เหมาะสมกับโลกยุคเทคโนโลยีในปัจจุบัน โดยนิยมที่จะแบ่งบริการออกเป็นส่วนย่อยหลาย ๆ ส่วน (Microservices) และนำมาให้บริการต่อผู้ใช้บริการตามแต่ละลักษณะหน้าที่ที่กำหนดไว้

เมื่อมีการแบ่งบริการออกเป็นส่วนย่อยหลาย ๆ ส่วนแล้ว นั้น การทำงานของระบบเบื้องหลัง จึงต้องมีการปรับปรุง โดยการประยุกต์ใช้ระบบประมวลผลแบบกลุ่มเมฆ (Cloud Computing) ร่วมกับระบบการประมวลผลแบบคอนเทนเนอร์ (Containers Technology) เพื่อความสะดวกในการดูแลและจัดการระบบการให้บริการ สามารถจัดการระบบให้มีความเหมาะสมกับภาระงานเพื่อประหยัดการใช้พลังงานในศูนย์ข้อมูลและสิ่งที่สำคัญที่สุดคือเพื่อให้บริการแก่ผู้เข้าใช้บริการด้วยการเข้าถึงที่รวดเร็วและสามารถเข้าถึงบริการได้ตลอดเวลา

ระบบบริหารจัดการการประมวลผลแบบคอนเทนเนอร์ที่เป็นที่นิยมใช้ในปัจจุบัน คือ คูเบอร์เนทิส (Kubernetes) ซึ่งถูกพัฒนาโดย Google Inc. (Kubernetes, 2018) โดยใช้ประสบการณ์ในการดูแลศูนย์ข้อมูลที่ได้ดำเนินการมาอย่างยาวนานเป็นแนวคิดและบทเรียนในการพัฒนาเพื่อให้ระบบมีประสิทธิภาพมากที่สุด การทำงานของคูเบอร์เนทิสนั้น จะทำงานในรูปแบบที่นำเครื่องคอมพิวเตอร์แม่ข่าย หรือเครื่องคอมพิวเตอร์เสมือนมาทำงานร่วมกันในลักษณะคลัสเตอร์ ที่ใช้คอมพิวเตอร์หลาย ๆ เครื่องช่วยกันประมวลผล เพื่อลดระยะเวลาในการทำงานและเพิ่มความมั่นคงต่อระบบการให้บริการในกรณีที่มีคอมพิวเตอร์เครื่องใดเครื่องหนึ่งขัดข้อง

การเข้าใช้งานระบบที่ให้บริการอยู่บนคลัสเตอร์คูเบอร์เนทิสที่ทำงานอยู่บนระบบประมวลผลแบบกลุ่มเมฆสามารถที่จะเข้าถึงได้หลายรูปแบบ เช่น ExternalIP, NodePort หรือ Load Balancer ซึ่งแต่ละวิธีก็จะมีข้อดีข้อเสียที่แตกต่างกันไป แต่วิธีที่เป็นที่นิยมใช้งานเมื่อให้บริการบนอินเตอร์เน็ตนั้น คือการเข้าใช้งานผ่านอินเกรสคอนโทรลเลอร์ (Ingress) ซึ่งเป็นโหลดบาลานซ์ คอนโทรลเลอร์ ที่ทำงานในระดับแอปพลิเคชันเลเยอร์ (เลเยอร์-7)

งานวิจัยฉบับนี้ จึงนำเสนอการติดตั้ง ทดลองและทดสอบ เลเยอร์-7 โหลดบาลานซ์ คอนโทรลเลอร์ ที่ทำหน้าที่เพื่อกำหนดเส้นทางในการเข้าใช้บริการของระบบที่ให้บริการอยู่บนการประมวลผลแบบคอนเทนเนอร์คลัสเตอร์คูเบอร์เนทิส จำนวน 4 รูปแบบ คือ

1) Nginx Ingress controller รุ่นที่พัฒนาโดยชุมชน Kubernetes (Kubernetes, 2018)

2) Træfik Ingress controller (Traefik, 2018)

3) Voyager Ingress controller ที่ทำงานบนพื้นฐานของ HAProxy (AppsCode, 2018)

4) GCE L7 load balancer controller (GLBC) ซึ่งเป็นระบบโหลดบาลานซ์ที่เป็นค่าปริยายบน Google Cloud Platform (Kubernetes, 2018)

โดยจะทำการทดสอบบนระบบ Public Cloud (Google Cloud Platform) เพื่อประเมินสมรรถนะและเปรียบเทียบประสิทธิภาพ ทางด้านความสามารถในการให้บริการ ความเร็วในการตอบสนองและนำข้อมูลที่ได้จากการทดลองมาพิจารณา วิเคราะห์และสรุปผล เพื่อนำผลการทดลองไปใช้ประกอบการตัดสินใจในการเลือกใช้รูปแบบที่เหมาะสมกับระบบที่จะให้บริการ

2. วัตถุประสงค์การวิจัย

เพื่อทดสอบและปรียบเทียบประสิทธิภาพของ เลเยอร์-7 โหลดบาลานซ์ คอนโทรลเลอร์ หรืออินเกรสคอนโทรลเลอร์ (Ingress controller) บนระบบคลัสเตอร์คูเบอร์เนทิส

3. การดำเนินการวิจัย

3.1 สถาปัตยกรรม

ในการทดสอบนี้จะดำเนินการสร้างระบบประมวลผลคอนเทนเนอร์คลัสเตอร์คูเบอร์เนทิส บน Public Cloud ของ Google Cloud Platform (Google Kubernetes Engine) ใช้โซน asia-southeast1-a (Singapore) โดยมี Worker Node จำนวน 3 เครื่อง ที่มีทรัพยากรที่เท่า ๆ กันทุกเครื่อง โดยมีรายละเอียด แสดงดังตารางที่ 1

ตารางที่ 1 การกำหนดทรัพยากรเครื่อง Worker Node

Components	Server
	Worker Node
Processor	2 vCPUs
Processor Platform	Intel Broadwell
Memory	2.5 GB
Disk	100 GB, Standard persistent disk
OS	Ubuntu Linux

3.2 ขั้นตอนการทดสอบ

1) สร้างระบบประมวลผลคอนเทนเนอร์คลัสเตอร์คูเบอร์เนทิส บน Google Kubernetes Engine โดยกำหนดทรัพยากรของเครื่อง Worker Node ตามรายละเอียดในหัวข้อที่ 3.1 สถาปัตยกรรมและตารางที่ 1

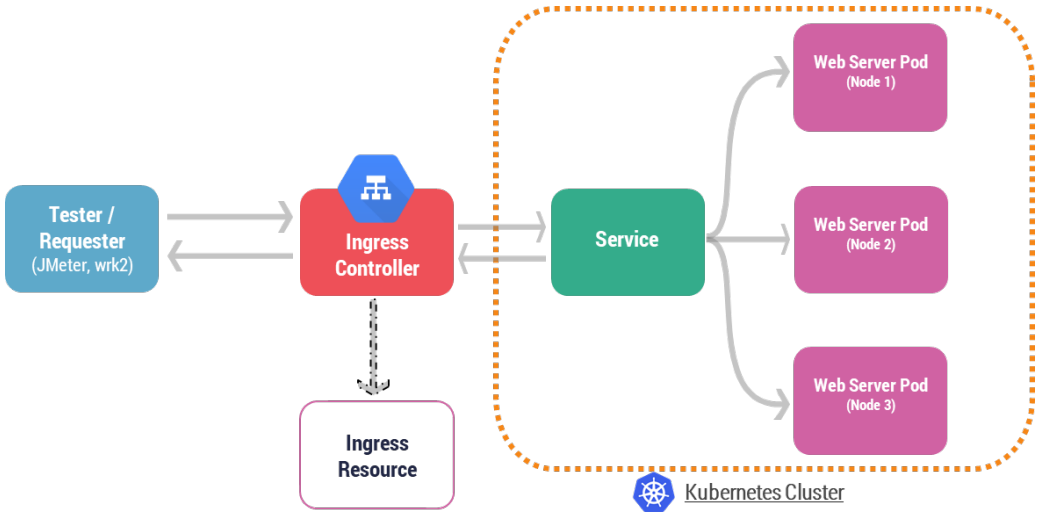
2) เมื่อทำการสร้างระบบประมวลผลคอนเทนเนอร์คลัสเตอร์คูเบอร์เนทิสพร้อมใช้งานแล้ว ขั้นตอนต่อไปก็เป็นการสร้าง pod เพื่อใช้ทำงานเป็น Web Server เพื่อตอบกลับ request ด้วยชื่อของเครื่องที่ให้บริการ (Hostname) หรือชื่อของ pod จำนวนเท่ากับจำนวน Worker Node ในการทดลองนี้ คือ 3 pods

3) สร้าง Service ในระบบคูเบอร์เนทิส เพื่อเป็นส่วนติดต่อระหว่าง External network กับ Cluster network

4) ติดตั้ง Ingress controller ที่จะทำการทดสอบบนคลัสเตอร์คูเบอร์เนทิส เพื่อทำหน้าที่กำหนดเส้นทางในการเข้าใช้งาน Service บนคลัสเตอร์

5) ดำเนินการทดสอบประสิทธิภาพของ Ingress controller และบันทึกผลการทดสอบ

โดยมีภาพแสดงระบบที่ใช้ในการทดสอบ แสดงในรูปที่ 1



รูปที่ 1 แสดงระบบที่ใช้ในการทดสอบ Ingress controller

3.3 การทดสอบประสิทธิภาพและจำนวนชุดข้อมูล

การทดสอบจะดำเนินการทดสอบเพื่อพิจารณาประสิทธิภาพของอินเกรสในหัวข้อ ดังต่อไปนี้

- ความสามารถในการให้บริการ (Throughput)

คือความสามารถในการให้บริการของระบบต่อหน่วยเวลา ในการวัดประสิทธิภาพของ Web Service จะนิยมวัดเป็น จำนวนที่สามารถให้บริการได้ต่อหน่วยเวลาเป็นวินาที (requests per second) ซึ่งการทดสอบในงานวิจัยนี้ จะวัดความสามารถในการให้บริการ (Throughput) ในหน่วย requests per second โดยทำการทดสอบการเข้าใช้บริการตั้งแต่ 100 – 5,000 requests per second ด้วยโปรแกรม wrk2 (Gil Tene, 2018) และบันทึกผลที่ระบบสามารถให้บริการได้จริง

โดยกำหนดค่าพารามิเตอร์ที่ใช้ในการทดสอบ ของโปรแกรม wrk2 ดังนี้

- t : จำนวน CPU Thread ที่ใช้ในการทดสอบ = 2 threads
- c : จำนวน Connection = 10 connections
- d : ระยะเวลาที่ใช้ในการทดสอบ = 30 วินาที (s)
- R : จำนวน Requests ต่อ วินาที ที่ทดสอบ= 100, 200, 500, 1000, 1500, 2000, 2500, 3000, 4000 และ 5000 requests/second

- ความเร็วในการตอบสนอง (Response time)

คือเวลาที่ใช้ในการติดต่อระหว่างเครื่องผู้ใช้งาน (client) กับระบบที่ให้บริการ (server) จนถึงเวลาที่ได้รับข้อมูลตอบกลับ

Response time = 2 * Transit Time + Processing time (at server) **(1)**

โดยที่ **Transit Time** คือ เวลาที่ใช้ในการส่งข้อมูลถึงจุดหมายผ่านทางเครือข่าย

โดยทั่วไป Response time จะมีหน่วยเป็น มิลลิวินาที (ms) ในการทดสอบนี้จะทำการทดสอบการเข้าใช้บริการระบบ โดยใช้โปรแกรม Apache JMeter (Apache Software Foundation, 2018) ทดสอบและบันทึกผลเวลาที่ระบบใช้ในการตอบสนองของอินเกรสคอนโทรเลอร์ 4 รูปแบบ ในกรณีต่าง ๆ ดังนี้

- จำนวน 100 requests ในเวลา 1 วินาที
- จำนวน 200 requests ในเวลา 1 วินาที
- จำนวน 500 requests ในเวลา 1 วินาที
- จำนวน 1,000 requests ในเวลา 1 วินาที
- จำนวน 1,500 requests ในเวลา 1 วินาที

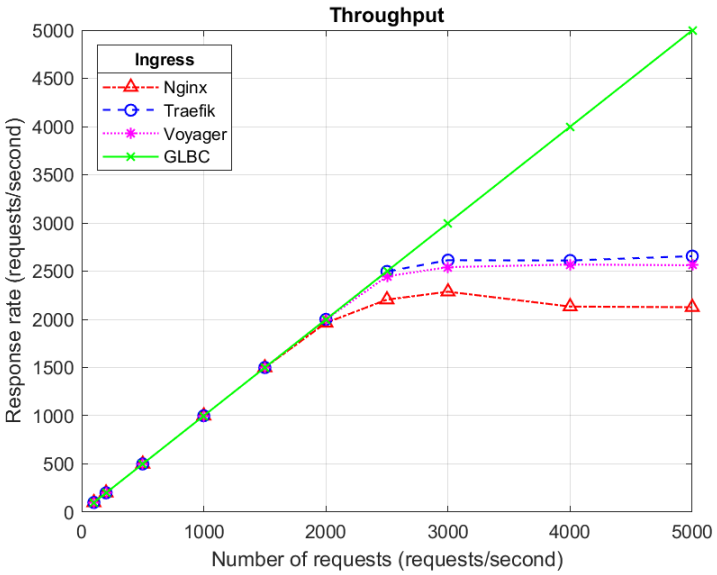
จำนวณกรณีที่ทำการทดสอบ เท่ากับ 4 x 5 = 20 กรณี

4. ผลการวิจัย

จากการทดสอบสมรรถนะและประสิทธิภาพของอินเกรสคอนโทรลเลอร์ จำนวน 4 รูปแบบ ที่นำมาทดสอบบนระบบคลัสเตอร์คูเบอร์เนทิส ที่มี Worker Node จำนวน 3 Node ที่ทำงานบน Google Cloud Platform มีผลการทดสอบในแต่ละหัวข้อ ดังนี้

4.1 ความสามารถในการให้บริการ (Throughput)

ในรูปที่ 2 แสดงผลการเปรียบเทียบความสามารถในการให้บริการของระบบ หรือ Throughput ผลการทดสอบพบว่า Nginx Ingress controller สามารถให้บริการได้น้อยที่สุด ที่ค่าเฉลี่ยประมาณ 2,200 requests/second ลำดับที่มีประสิทธิภาพดีขึ้นมา คือ Voyager Ingress controller ที่สามารถให้บริการได้ที่ค่าเฉลี่ย 2,560 requests/second และ Traefik Ingress controller ที่สามารถให้บริการได้ที่ค่าเฉลี่ย 2,600 requests/second ตามลำดับ โดยที่ GCE L7 load balancer controller (GLBC) สามารถให้บริการได้มากกว่า 5,000 requests/second ซึ่งสูงกว่าจำนวนที่สูงที่สุดที่ใช้ในการทดสอบ

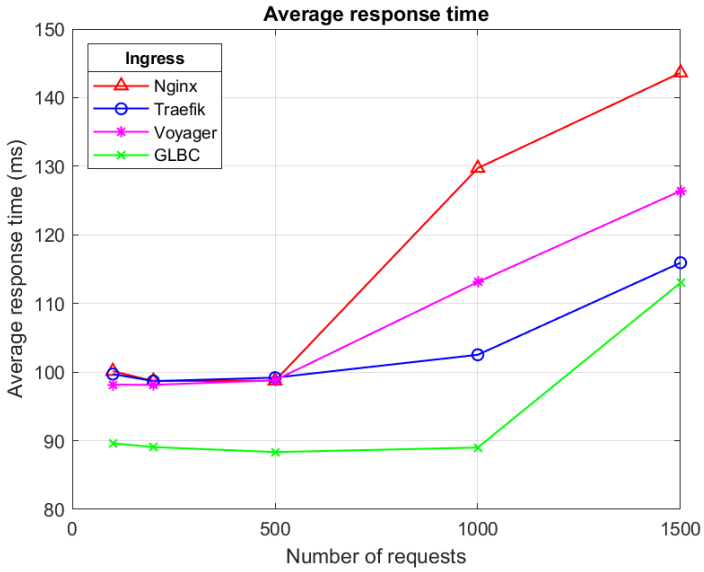


รูปที่ 2 แสดงผลการทดสอบความสามารถในการให้บริการ (Throughput)

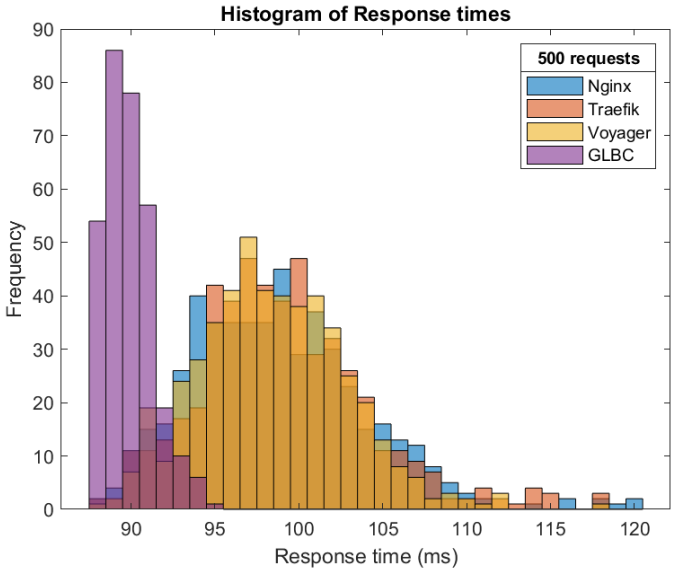
4.2 ความเร็วในการตอบสนอง (Response time)

ในรูปที่ 3 แสดงค่าของความเร็วในการตอบสนองเฉลี่ย หรือ Average response time ที่ทำการทดสอบเป็นจำนวน 100, 200, 500, 1000 และ 1500 ครั้ง ผลการทดสอบพบว่า GCE L7 load balancer controller (GLBC) มีค่าความเร็วในการตอบสนองเฉลี่ยน้อยที่สุด ซึ่งค่าน้อยหมายความว่ามีประสิทธิภาพที่ดี ลำดับที่มีประสิทธิภาพที่ดีถัดมา คือ Traefik Ingress controller และ Voyager Ingress controller ตามลำดับ และ Nginx Ingress controller มีค่าความเร็วในการตอบสนองเฉลี่ยสูงที่สุด

แนวโน้มจากการทดสอบพบว่าเมื่อจำนวนครั้งในการร้องขอข้อมูลเพิ่มมากขึ้นจะทำให้ความเร็วในการตอบสนองเฉลี่ยสูงขึ้นในทุก ๆ อินเกรสคอนโทรลเลอร์



รูปที่ 3 แสดงผลการทดสอบความเร็วในการตอบสนองเฉลี่ย (Average response time)



รูปที่ 4 แสดงการแจกแจงความถี่ของผลการทดสอบความเร็วในการตอบสนอง ที่ 500 requests/second

จากรูปที่ 4 ที่แสดงการแจกแจงความถี่ ของผลการทดสอบความเร็วในการตอบสนองที่จำนวน 500 requests/second ในทุก ๆ อินเกรสคอนโทรลเลอร์ จากรูปที่ 4 จะพบว่า GCE L7 load balancer controller (GLBC) มีจำนวนความถี่ที่ 70 – 90 ครั้ง อยู่ที่ค่าระหว่าง 89 – 90 มิลลิวินาที ซึ่งน้อยกว่าของอินเกรสคอนโทรลเลอร์อื่น ๆ โดยที่ Traefik Ingress controller, Voyager Ingress controller และ Nginx Ingress controller มีจำนวนความถี่ ประมาณ 40 ครั้ง อยู่ที่ค่าประมาณ 96 - 102 มิลลิวินาที จึงแสดงว่ามีค่าความเร็วในการตอบสนองที่ช้ากว้า GCE L7 load balancer controller (GLBC) (ค่า Response time น้อย แสดงถึงการมีประสิทธิภาพที่ดี)

5. บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการทดสอบ

งานวิจัยนี้ทำการทดสอบประสิทธิภาพของอินเกรสคอนโทรลเลอร์บนระบบคอนเทนเนอร์คลัสเตอร์คูเบอร์เนทิส ผลการทดลองพบว่า

ในด้านความสามารถในการให้บริการ (Throughput) พบว่า GCE L7 load balancer controller สามารถให้บริการได้มากกว่า 5,000 requests/second ซึ่งสูงกว่าจำนวนที่สูงที่สุดที่ใช้ในการทดสอบในงานวิจัยนี้ (ที่กำหนดไว้ที่ 5,000 requests/second) รองลงมาคือ Traefik Ingress controller ที่สามารถให้บริการได้ เฉลี่ย 2,600 requests/second, Voyager Ingress controller ที่สามารถให้บริการได้ เฉลี่ย 2,560 requests/second และ Nginx Ingress controller ที่สามารถให้บริการได้ เฉลี่ย 2,200 requests/second ตามลำดับ

ในด้านความเร็วในการตอบสนอง (Response time) พบว่า GCE L7 load balancer controller สามารถตอบสนองได้เร็วที่สุด ในทุก ๆ การทดสอบ (ทุก ๆ จำนวน requests ต่อ วินาที) อินเกรสคอนโทรลเลอร์ที่ตอบสนองได้เร็วเป็นลำดับถัดมา คือ Traefik Ingress controller, Voyager Ingress controller และ Nginx Ingress controller ตามลำดับและความเร็วในการตอบสนองมีแนวโน้มสูงขึ้นเมื่อจำนวนครั้งในการร้องขอข้อมูลต่อหน่วยเวลาเพิ่มมากขึ้นในทุก ๆ อินเกรสคอนโทรลเลอร์

5.2 ข้อเสนอแนะ

1) ในการนำอินเกรสคอนโทรลเลอร์ไปใช้งานควรคำนึงถึงคุณสมบัติอื่น ๆ ที่อำนวยความสะดวกในการใช้งานที่นอกเหนือจากขอบเขตของการทดสอบในงานวิจัยนี้ (Throughput และ Response time) เช่น การรองรับ Web socket, การรองรับการเข้ารหัส TLS, การทำ SSL Termination, การเลือกหรือปรับแต่ง Load balancing algorithm ขั้นสูง, ระบบ Logging และมีระบบจัดเก็บและแสดงผลข้อมูลสถิติ (Dashboard) เพื่อใช้ประกอบในการตัดสินใจเลือกใช้อินเกรสคอนโทรลเลอร์

2) การทดสอบนี้ดำเนินการบน Google Cloud Platform ซึ่งอาจจะมีผลการทดสอบที่คลาดเคลื่อนเมื่อนำไปทดสอบหรือใช้งานบนระบบ Public Cloud อื่น ๆ เนื่องจากประสิทธิภาพของระบบ Cloud load balancer และประสิทธิภาพของระบบเครือข่ายบนระบบ Public Cloud นั้น ๆ แนวทางการดำเนินงานวิจัยต่อไปจะดำเนินการทดลองบน Public Cloud ระบบอื่นๆ เช่น Amazon web services, Microsoft Azure และ IBM Cloud เป็นต้น

เอกสารอ้างอิง

Apache Software Foundation. (2018). Apache *JMeter™*. Retrieved June 15, 2018, from https://jmeter.apache.org/

AppsCode. (2018). Voyager *Secure HAProxy Ingress Controller for Kubernetes.* Retrieved June 15, 2018, from https://appscode.com/products/voyager/

Gil Tene. (2018). A *constant throughput, correct latency recording variant of wrk.* Retrieved June 15, 2018, from https://github.com/giltene/wrk2/

Kubernetes. (2018). Concepts - *Kubernetes.* Retrieved June 15, 2018, from https://kubernetes.io/docs/concepts/

Kubernetes. (2018). Ingress *controller for Google Cloud.* Retrieved June 15, 2018, from https://github.com/kubernetes/ingress-gce/

Kubernetes. (2018). NGINX *Ingress Controller*. Retrieved June 15, 2018, from https://kubernetes.github.io/ingress-nginx/

Traefik. (2018). Kubernetes *Ingress Controller*. Retrieved June 15, 2018, from https://docs.traefik.io