การจัดสรรทรัพยากรบนระบบประมวลผลแบบกลุ่มเมฆ โดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือน

บุญอนันต์ ปอศรี[1]* และ บวร ปภัสราทร[2]

บทคัดย่อ

การจัดสรรทรัพยากรเป็นปัจจัยสำคัญในความสำเร็จของระบบประมวลผลแบบกลุ่มเมฆ โดยปกติแล้วการจัดสรรทรัพยากรบนระบบประมวลผลแบบกลุ่มเมฆนั้นทำได้โดยการเพิ่มจำนวนเซิร์ฟเวอร์เสมือนแต่เนื่องจากวิธีนี้ระบบจำเป็นที่จะต้องใช้เวลาคัดลอกอิมเมจไฟล์ การเตรียมระบบรวมถึงการเริ่มต้นระบบใหม่ ซึ่งใช้เวลารวมกันประมาณ 2-5 นาที ในงานวิจัยฉบับนี้นำเสนอการจัดสรรทรัพยากรบนระบบประมวลผลแบบกลุ่มเมฆโดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือน วิธีนี้จะทำการเพิ่มจำนวนหน่วยประมวลผลและหน่วยความจำที่เซิร์ฟเวอร์ใช้ได้แทนการสร้างเซิร์ฟเวอร์เสมือนใหม่ จากการทดลองภายใต้สภาพแวดล้อมเดียวกัน พบว่า การจัดสรรทรัพยากรตามวิธีที่เสนอมีเวลาในการตอบสนองน้อยกว่าการสร้างเซิร์ฟเวอร์เสมือนขึ้นใหม่ ซึ่งมีเวลาในการตอบสนองที่มากกว่าข้อตกลงถึง 22.5 เปอร์เซ็นต์ ด้วยวิธีการจัดสรรทรัพยากรโดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือนนี้สามารถนำไปประยุกต์ใช้ในระบบประมวลผลแบบกลุ่มเมฆที่ต้องการให้เวลาในการตอบสนองการประมวลผลน้อยกว่าเวลาตามข้อตกลงในทุกสถานการณ์

คำสำคัญ: การจัดสรรทรัพยากร การประมวลผลแบบกลุ่มเมฆ

[1] นักศึกษา คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

[2] รองศาสตราจารย์ คณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

* ผู้นิพนธ์ประสานงาน โทรศัพท์ 08-0952-9089 อีเมล: 52440339@st.sit.kmutt.ac.th

รับเมื่อ 19 มีนาคม 2555 ตอบรับเมื่อ 17 พฤษภาคม 2555

Cloud Computing Resources Provisioning Using Virtual Server Size Expansion Method

Bunanun Posri[1]* and Borworn Papasratorn[2]

Abstract

Resource Provisioning is a key success factor of Cloud Computing. Generally new virtual server will be created and attached to application environment that requires more resources. Using this method, it takes time to move image of virtual server to physical server, prepare environment in physical server and start the virtual server. These processes normally take about 2-5 minutes. In this paper, we present Virtual Server Size Expansion Method to reduce resource provisioning time. The proposed method will increase number of processors and memory size of virtual server, instead of create new virtual server. In the same experimental environment, it is found that the response of the proposed method is lower than method with new virtual server that response time more than 22.5% of SLA. The proposed Virtual Server Side Expansion method can be applied in cloud computing that requires response time under limit given in service level agreement in all situations.

Keywords: Resource Provisioning, Cloud Computing

[1] Student, School of Information Technology, King Mongkut's University of Technology Thonburi.

[2] Associate Professor, School of Information Technology, King Mongkut's University of Technology Thonburi.

* Corresponding Author, Tel. 08-0952-9089, E-mail: 52440339@st.sit.kmutt.ac.th

Received 19 March 2012; Accepted 17 May 2012

1. บทนำ

ปัจจุบันมีการใช้การประมวลผลแบบกลุ่มเมฆในระบบสารสนเทศขององค์กรขนาดใหญ่ และมีการเปลี่ยนแปลงการใช้ซอฟต์แวร์ให้เป็นการบริการ ซึ่งต้องการการปรับเปลี่ยนทรัพยากรของระบบอย่างคล่องตัวและรวดเร็ว โดยที่ไม่ต้องเพิ่มเซิร์ฟเวอร์หรือสร้างศูนย์ข้อมูลแห่งใหม่ [1] โดยเกิดจากการใช้เทคโนโลยีจำลองเสมือนจริง (Virtualization) ที่ภายในประกอบด้วยเซิร์ฟเวอร์เสมือน (Virtual Machine) และเมื่อนำเซิร์ฟเวอร์เสมือนเหล่านี้เชื่อมต่อกันผ่านเครือข่าย ผู้ใช้สามารถปรับเพิ่มหรือลดทรัพยากรได้ตามความต้องการ ในระบบประมวลผลแบบกลุ่มเมฆเราสามารถแบ่งกลุ่มของเซิร์ฟเวอร์เสมือนออกเป็นสองส่วนส่วนแรกเป็นส่วนของหน่วยประมวลผลจะประกอบไปด้วยเซิร์ฟเวอร์เสมือนซึ่งทำหน้าที่เหมือนหรือต่างกัน เช่น เว็บเซิร์ฟเวอร์ (Web Server) ดาต้าเบสเซิร์ฟเวอร์ (Database Server) หรือไฟล์เซิร์ฟเวอร์ (File Server) เป็นต้น มาทำงานร่วมกัน รวมเรียกว่าแอพพลิเคชันเซิร์ฟเวอร์ (Application Server) เพื่อให้บริการแก่ผู้ใช้งาน โดยสามารถมีได้หนึ่งหรือหลายแอพพลิเคชันภายใต้ระบบประมวลผลแบบกลุ่มเมฆนั้น

ในส่วนที่สองเป็นระดับควบคุมการให้บริการ (Cloud Service Provider) ทำหน้าที่ดูแล ควบคุมให้การทำงานของแต่ละแอพพลิเคชันเซิร์ฟเวอร์เป็นไปตามข้อตกลงการให้บริการ (Service Level Agreement: SLA) ที่กำหนดไว้กับผู้ใช้งาน ในกรณีที่เกิดเหตุผิดปกติเช่น เวลาในการตอบสนองของระบบลดต่ำลง ส่วนควบคุมการให้บริการมีหน้าที่ในการเพิ่มทรัพยากรให้แก่แอพพลิเคชันเซิร์ฟเวอร์ [2] ในปัจจุบันการเพิ่มทรัพยากรให้กับแอพพลิเคชันเซิร์ฟเวอร์บนระบบประมวลผลแบบกลุ่มเมฆทำได้โดยสร้างเซิร์ฟเวอร์เสมือนขึ้นมาใหม่และนำไปใช้เพิ่มทรัพยากรระบบให้กับแอพพลิเคชันเซิร์ฟเวอร์ที่ต้องการ แต่ทั้งนี้การสร้างเซิร์ฟเวอร์เสมือนนั้นประกอบไปด้วยขั้นตอนดังต่อไปนี้

1. ขั้นตอนในการคัดลอกอิมเมจไฟล์ (Image File) จากเซิร์ฟเวอร์ควบคุมไปยังเซิร์ฟเวอร์จริง (Physical Server)

2. ขั้นตอนในการเริ่มต้นระบบปฏิบัติการของเซิร์ฟเวอร์เสมือน

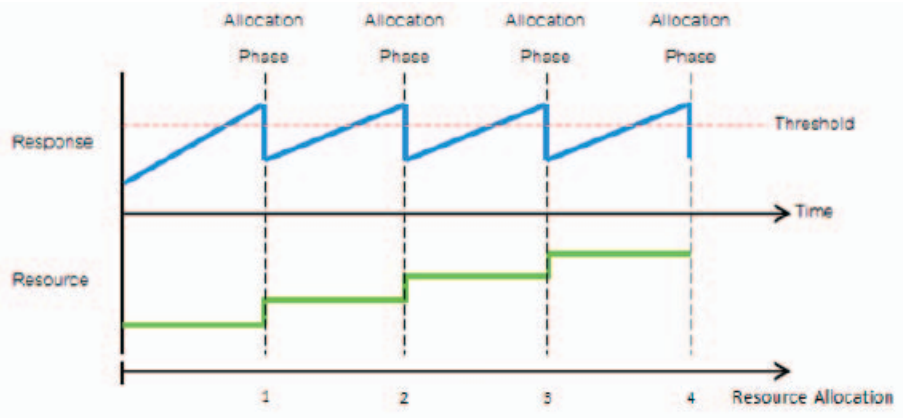
3. ขั้นตอนในการเริ่มต้นแอพพลิเคชันเซิร์ฟเวอร์

ทั้งสามขั้นตอนนี้จำเป็นจะต้องใช้เวลาในการเคลื่อนย้ายและประมวลผล โดยเฉลี่ยแล้วประมาณ 2-5 นาที [1] ขึ้นกับชนิดของระบบปฏิบัติการบนเซิร์ฟเวอร์เสมือน แอพพลิเคชันที่ติดตั้งบนเซิร์ฟเวอร์เสมือน รวมถึงจำนวนผู้ใช้งาน ณ ขณะนั้น ด้วยเหตุผลเหล่านี้ทำให้การเพิ่มทรัพยากรให้แก่แอพพลิเคชันเซิร์ฟเวอร์ที่ร้องขอนั้นเกิดความล่าช้าและไม่ทันต่อการเปลี่ยนแปลงของจำนวนผู้ใช้งาน เป็นเหตุทำให้เวลาในการตอบสนองการร้องขอการประมวลผลช้ากว่าที่ตกลงไว้กับผู้ใช้งาน ซึ่งการผิดข้อตกลงนี้อาจเป็นผลทำให้ผู้ให้บริการจำเป็นที่จะต้องจ่ายค่าปรับแก่ผู้ใช้งานอีกด้วย

งานวิจัยนี้นำเสนอการจัดสรรทรัพยากรบนระบบประมวลผลแบบกลุ่มเมฆ โดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือน (เพิ่มหน่วยประมวลผลและขยายขนาดหน่วยความจำ) แทนการเพิ่มจำนวนเซิร์ฟเวอร์เสมือน

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เป้าหมายของการจัดสรรทรัพยากรบนระบบประมวลผลแบบกลุ่มเมฆนั้นคือการจัดสรรทรัพยากรให้เพียงพอต่อความต้องการของผู้ใช้งาน ณ ขณะนั้น ได้มีการนำเสนอการจัดลำดับงานที่ร้องขอจากผู้ใช้ เพื่อต้องการประมวลผลบนแอพพลิเคชันเซิร์ฟเวอร์ ซึ่งเปรียบเทียบการจัดลำดับโดยใช้วิธีต่างๆ ดังนี้ วิธีมาก่อนได้ก่อน (FCFS) การจัดลำดับงานโดยเรียงลำดับตามความสำคัญของชนิดของงาน (Head-of-the-Line Priority) และการจัดลำดับของงานโดยคำนวณจากชนิดของงานที่มีอัตราส่วนระหว่างงานที่ประมวลผลเสร็จและใช้เวลาในการตอบสนองน้อยกว่าข้อตกลง กับจำนวนงานทั้งหมดที่ประมวลผลเสร็จ โดยงานที่มีอัตราส่วนน้อยกว่าจะได้ประมวลผลก่อน (Probability Dependent Priority: PDP) [2] ซึ่งการจัดสรรลำดับของงานแบบ PDP นั้นมีประสิทธิภาพสูงสุดในทั้งสามวิธีที่กล่าวมา แต่เมื่อผู้ใช้งานมีจำนวนมากขึ้น การจัดลำดับความสำคัญของงานอย่างเดียวจึงไม่เพียงพอ



รูปที่ 1 แสดงส่วนของการสังเกตการณ์และส่วนของการขยายขนาด

ประกอบกับความสามารถของระบบประมวลผลแบบกลุ่มเมฆที่สามารถขยายทรัพยากรของระบบได้อย่างยืดหยุ่น จึงได้มีการนำเสนอการจัดสรรทรัพยากรโดยพิจารณาจากจำนวนผู้ใช้งาน ณ ขณะนั้นเปรียบเทียบกับจำนวนผู้ใช้งานสูงสุดที่เซิร์ฟเวอร์เสมือนจะรองรับได้ ในกรณีที่จำนวนผู้ใช้งานที่เข้าใช้เซิร์ฟเวอร์เสมือนเครื่องใดเครื่องหนึ่งมากกว่าค่าที่ได้กำหนดไว้ ระบบจะทำการเพิ่มเซิร์ฟเวอร์เสมือนเพื่อกระจายผู้ใช้งานไปยังเซิร์ฟเวอร์เสมือนเครื่องต่างๆ และเป็นผลทำให้เวลาในการตอบสนองของแอพพลิเคชันเซิร์ฟเวอร์โดยรวมลดลง ในทางกลับกันเมื่อจำนวนผู้ใช้งานที่เข้าใช้งานเซิร์ฟเวอร์เสมือนเครื่องใดเครื่องหนึ่งลดต่ำลงกว่าค่าที่ได้กำหนดไว้ ระบบจะทำการปิดเซิร์ฟเวอร์เสมือนเครื่องนั้นเพื่อลดการใช้ทรัพยากรของระบบ [3] แต่ด้วยวิธีการนี้เมื่อมีผู้ใช้งานเพิ่มขึ้นอย่างรวดเร็วระบบจะไม่สามารถเพิ่มทรัพยากรให้เพียงพอต่อความต้องการของผู้ใช้งานได้ทันท่วงที นอกนี้มีการนำเสนอการกระจายผู้ใช้งานโดยวิธีการย้ายเซิร์ฟเวอร์เสมือนไปยังเครื่องเซิร์ฟเวอร์ที่มีปริมาณการใช้งานน้อยกว่า (Live Migration) เพื่อลดภาระของเครื่องเซิร์ฟเวอร์ที่มีเซิร์ฟเวอร์เสมือนรวมกันอยู่มากหรือทำงานหนัก โดยใช้วิธีการสร้างเซิร์ฟเวอร์เสมือนขึ้นมาใหม่และทำการคัดลอกข้อมูลในหน่วยความจำของเซิร์ฟเวอร์เสมือนเครื่องเก่าไปยังเซิร์ฟเวอร์เสมือนเครื่องใหม่ [7]

3. การจัดสรรทรัพยากรบนระบบประมวลผลแบบกลุ่มเมฆโดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือน

การจัดสรรทรัพยากรด้วยวิธีนี้ประกอบด้วยสองส่วน คือ ส่วนของการสังเกตการณ์ (Observation Phase) และส่วนของการขยายขนาด (Allocation Phase) ดังแสดงไว้ในรูปที่ 1

3.1 ส่วนของการสังเกตการณ์

ในส่วนของการสังเกตการณ์นี้เราจะทำที่ระดับควบคุม ซึ่งทำการตรวจสอบเวลาที่เซิร์ฟเวอร์เสมือนแต่ละเครื่องใช้ในการประมวลผลงานที่ผู้ใช้งานได้ทำการร้องขอไปยังเซิร์ฟเวอร์โดยแยกตามประเภทของงาน และนำเวลาที่เซิร์ฟเวอร์เหล่านั้นใช้ประมวลผลมาหาเวลาการตอบสนองเฉลี่ยดังแสดงในสมการที่ 1 [4]

$$T_{response} = N/R - T_p \qquad (1)$$

โดยที่

N : จำนวนผู้ใช้งานพร้อมกัน

R : จำนวนการร้องขอการประมวลที่เซิร์ฟเวอร์ได้รับต่อหนึ่งวินาที

T_p : ค่าเฉลี่ยของเวลาในการประมวลผล (วินาที)

$T_{response}$: ค่าเฉลี่ยของเวลาในการตอบสนอง (วินาที)

เวลาที่ได้นั้นจะถูกนำมาเปรียบเทียบกับเวลาที่

ได้ตกลงไว้กับผู้ใช้งาน ซึ่งถ้าเวลาในการตอบสนองนั้นมากเกินกว่าเวลาที่ได้ตกลงไว้ ระบบก็จะทำการขยายขนาดของเซิร์ฟเวอร์เสมือนซึ่งจะทำให้เวลาในการตอบสนองเฉลี่ยลดลง

3.2 ส่วนการขยายขนาด

เมื่อเวลาในการตอบสนองเฉลี่ยมากขึ้นเกินกว่าค่าที่กำหนดไว้ แสดงให้เห็นว่าทรัพยากรที่เซิร์ฟเวอร์เสมือนมีอยู่ไม่เพียงพอต่อการประมวลผลงานที่ผู้ใช้ร้องขอ ระบบในส่วนควบคุมจะทำการค้นหาเซิร์ฟเวอร์เสมือนที่มีทรัพยากรน้อยที่สุดและยังสามารถเพิ่มทรัพยากรได้ จากนั้นจะทำการสั่งให้เซิร์ฟเวอร์เสมือนนั้นเพิ่มหน่วยประมวลผลและขยายขนาดหน่วยความจำที่ใช้ได้ ผ่านทางเซิร์ฟเวอร์ที่ทำการเรียกใช้เซิร์ฟเวอร์เสมือนเครื่องนั้นอยู่ โดยกระบวนการในส่วนของการสังเกตการณ์และส่วนของการขยายขนาดนั้นใช้อัลกอริทึมดังรูปที่ 2

จากรูปที่ 2 เมื่อเวลาในการตอบสนองเฉลี่ยมากขึ้นเกินกว่าข้อตกลง ระบบจะทำการค้นหาเซิร์ฟเวอร์เสมือนที่ถือครองทรัพยากรน้อยที่สุด และส่งคำสั่งไปยังซอฟต์แวร์เซิร์ฟเวอร์เสมือน ทั้งนี้การที่จะสั่งให้เซิร์ฟเวอร์เสมือนสามารถใช้หน่วยประมวลผลและหน่วยความจำเพิ่มขึ้นได้ ซอฟต์แวร์เซิร์ฟเวอร์เสมือนจำเป็นที่จะต้องรองรับคุณสมบัติการทำ CPU Hot plug และ Memory Hot plug เช่น ซอฟต์แวร์ KVM เวอร์ชัน 63 ขึ้นไปหรือ Xen เวอร์ชัน 3.0 ขึ้นไป ซึ่งในการทดลองที่จะกล่าวถึงในถัดไปนั้นเลือกใช้ซอฟต์แวร์ KVM

4. การทดลอง

การทดลองในงานวิจัยนี้ทำขึ้นเพื่อทดสอบประสิทธิภาพของการจัดสรรทรัพยากรบนระบบประมวลผลแบบกลุ่มเมฆ ระหว่างการจัดสรรทรัพยากรโดยใช้วิธีเพิ่มจำนวนเซิร์ฟเวอร์เสมือน และการจัดสรรทรัพยากรโดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือน และวัดผลความเร็วในการประมวลผลของทั้งสองระบบ การทดลองนี้ใช้เครื่องเซิร์ฟเวอร์ทั้งหมดสามเครื่องดังตารางที่ 1

Input

N : Number of concurrent users
R : Number of request per second
$T_{response}$: Average response time
T_p : Average process time
Th_{upper} : Upper threshold of response time
Th_{lower} : Lower threshold of response time
VM_i : Virtual Machine instance i

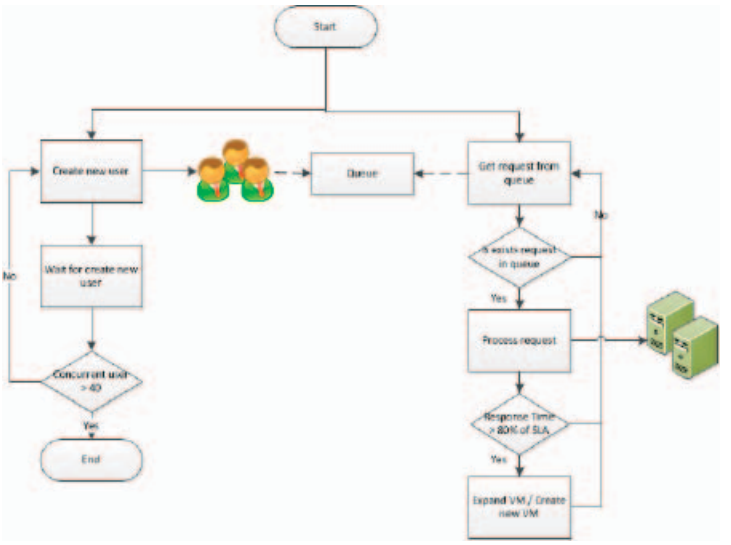
```
Start:
Calculate average response time:
T_response = N/R – T_p
if (T_response > Th_upper) then
        VM_i = get virtual machine that have lowest
               resource
        if (VM_i can expand resource) then
                Expand resource of VM_i
        else
                Create new Virtual Machine
                instance
else if (T_response < Th_lower) then
        VM_i = get virtual machine that have highest
               resource
        if (VM_i can decrease resource) then
                Decrease resource of VM_i
        else
                Shutdown VM_i
End :
```

รูปที่ 2 วิธีการจัดสรรทรัพยากรบนระบบประมวลผลแบบกลุ่มเมฆโดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือน

ตารางที่ 1 แสดงคุณสมบัติของเซิร์ฟเวอร์ที่ใช้ในการทดลอง

Server	CPU	Memory	Network	OS	Virtual Machine
Server 1	Intel Core2 Duo 2.6 GHz	4 GB	100 Mbps	Ubuntu Server 11.04 64bits	KVM
Server 2	Intel Core i5 2.4 GHz	8 GB	100 Mbps	Ubuntu Server 11.04 64bits	KVM
Server 3	Intel Core2 2.2 GHz	4 GB	100 Mbps	Win-dows 7	-

การทดลองนี้แบ่งเซิร์ฟเวอร์ออกเป็นสามเครื่องเซิร์ฟเวอร์เครื่องที่ 1 เป็นส่วนของระดับการจัดการใช้ซอฟต์แวร์



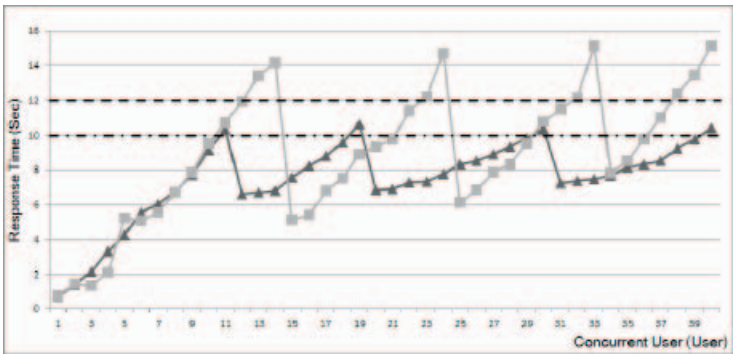
รูปที่ 3 แสดงขั้นตอนวิธีทำการทดลอง

Open Nebula เพื่อเป็นโปรแกรมในการจัดการบริหารระบบประมวลผลแบบกลุ่มเมฆ โดยทำหน้าที่ควบคุมการสร้าง ปิด เพิ่มขนาด และลดขนาดเซิร์ฟเวอร์เสมือน ภายในเซิร์ฟเวอร์เครื่องนี้ทำการสร้างอิมเมจไฟล์สำหรับติดตั้งบนเซิร์ฟเวอร์เสมือน โดยภายในอิมเมจไฟล์ติดตั้งระบบปฏิบัติการ Linux Ubuntu Server 11.04 64bits โดยขนาดของอิมเมจไฟล์มีขนาดประมาณ 500 MB ภายในเครื่องเดียวกันนี้ติดตั้งซอฟต์แวร์กระจายการใช้งาน (Load Balance) โดยใช้โปรแกรม Nginx เวอร์ชัน 1.0.14 เพื่อกระจายการร้องขอจากผู้ใช้งานไปยังเซิร์ฟเวอร์เสมือนทุกเครื่อง เซิร์ฟเวอร์เครื่องที่ 2 ทำหน้าที่ในการรองรับการทำงานของเซิร์ฟเวอร์เสมือน โดยติดต่อกับเครื่องควบคุมผ่าน Secure Shell (SSH) เซิร์ฟเวอร์เครื่องที่ 3 ทำหน้าที่ในการจำลองการร้องขอการประมวลผลไปยังแอพพลิเคชันเซิร์ฟเวอร์ โดยการทดลองมีขอบเขตดังนี้

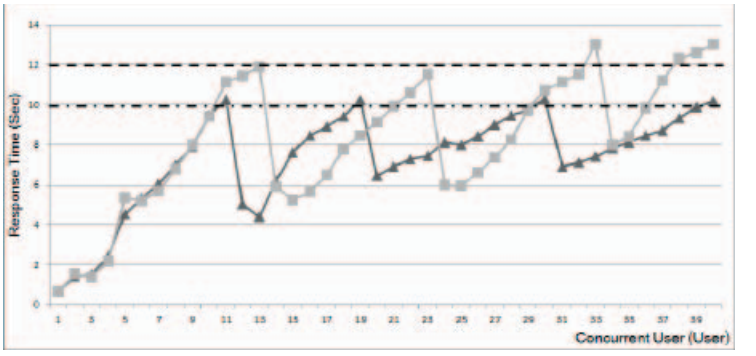
- งานที่ผู้ใช้ทำการร้องขอไปยังเซิร์ฟเวอร์มีเพียงประเภทเดียวซึ่งเป็นการประมวลอัลกอริทึม WheatStone ซึ่งเป็นอัลกอริทึมที่ใช้ในการวัดประสิทธิภาพการประมวลผลเลขจำนวนเต็มของคอมพิวเตอร์
- ข้อตกลงการให้บริการคือ 12 วินาที โดยระบบจะทำการเพิ่มทรัพยากรเมื่อเวลาในการตอบสนองเฉลี่ยมากกว่า 9.6 วินาที หรือ 80% ของข้อตกลงการให้บริการ
- เริ่มการทดลองจากผู้ใช้งาน 1 คนและหยุดทำการทดลองเมื่อผู้ใช้งานมีจำนวน 40 คน
- การขยายของเซิร์ฟเวอร์เสมือนจะทำการขยายขนาดขึ้น 25% จากขนาดเดิม

จากรูปที่ 3 ซึ่งแสดงวิธีการทดลองเมื่อเริ่มต้นการทดลองโดยแบ่งออกเป็นสองส่วนด้วยกันคือ ส่วนของการจำลองผู้ใช้งาน และส่วนของการขยายขนาดเซิร์ฟเวอร์เสมือน

เมื่อเริ่มต้นการทดลองในส่วนของการจำลองผู้ใช้งานจะทำการจำลองผู้ใช้งานขึ้น โดยที่ผู้ใช้งานแต่ละคนนั้นจะทำการร้องขอการประมวลผลไปแอพพลิเคชัน เซิร์ฟเวอร์เมื่อแอพพลิเคชันเซิร์ฟเวอร์ประมวลผลเรียบร้อยและทำการส่งผลลัพธ์กลับไปยังผู้ใช้งานแล้ว ผู้ใช้งานจะทำการร้องขอไปยังแอพพลิเคชันเซิร์ฟเวอร์ใหม่ และทำแบบเดิมจนกว่าจะสิ้นสุดการทดลองและในส่วนของเซิร์ฟเวอร์เมื่อเริ่มต้นการทดลอง Open Nebula จะทำการคัดลอกอิมเมจไฟล์ไปยังเครื่องเซิร์ฟเวอร์ที่ใช้สำหรับติดตั้งเซิร์ฟเวอร์เสมือน เมื่อเสร็จสิ้นกระบวนการคัดลอก Open Nebula จะสั่งให้เซิร์ฟเวอร์เสมือนเริ่มแสดงการเพิ่มทรัพยากรโดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือนแสดงการเพิ่มทรัพยากรโดยใช้วิธีเพิ่มจำนวนเซิร์ฟเวอร์เสมือนแสดงเวลาการตอบสนอง



รูปที่ 4 กราฟเปรียบเทียบเวลาการตอบสนองเฉลี่ย ระหว่างวิธีขยายขนาดและวิธีเพิ่มจำนวนเซิร์ฟเวอร์เสมือน (ผู้ใช้งานเพิ่ม 1 คนทุก 20 วินาที)



แสดงการเพิ่มทรัพยากรโดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือน
แสดงการเพิ่มทรัพยากรโดยใช้วิธีเพิ่มจำนวนเซิร์ฟเวอร์เสมือน
แสดงเวลาการตอบสนองเฉลี่ยที่มากที่สุดที่ยอมรับได้ตามที่ตกลงไว้กับผู้ใช้งาน (12 วินาที)
แสดงเวลาที่ระบบจะทำการขยายขนาดทรัพยากร หากเวลาในการตอบสนองเฉลี่ยมากกว่าที่กำหนด (9.6 วินาที)

รูปที่ 5 กราฟเปรียบเทียบเวลาในการตอบสนองเฉลี่ย ระหว่างวิธีขยายขนาด และวิธีเพิ่มจำนวนเซิร์ฟเวอร์เสมือน (ผู้ใช้งานเพิ่ม 1 คนทุก 40 วินาที)

เฉลี่ยที่มากที่สุดที่ยอมรับได้ตามที่ตกลงไว้กับผู้ใช้งาน (12 วินาที) แสดงเวลาที่ระบบจะทำการขยายขนาดทรัพยากร หากเวลาในการตอบสนองเฉลี่ยมากกว่าที่กำหนด (9.6 วินาที) ทำงาน ในการทดลองนี้ได้ทำการแก้ไขโปรแกรม Nginx ในส่วนของ Load Balance เพื่อเพิ่มอัลกอริทึมในการขยายขนาดเซิร์ฟเวอร์เสมือน

โดยอัลกอริทึมนี้จะหาเวลาการตอบสนองเฉลี่ยของแอพพลิเคชันเซิร์ฟเวอร์ และนำเวลาที่ได้มาเปรียบเทียบกับข้อตกลงการให้บริการ หากมีค่ามากกว่าจะทำการเพิ่มทรัพยากรให้กับแอพพลิเคชันเซิร์ฟเวอร์ ในการทดลองทำการแก้ไขโปรแกรม Nginx ให้สามารถติดต่อกับ OpenNebula ผ่านทาง Open Cloud Computing Interface (OCCI) เพื่อให้ Nginx สามารถสั่งการให้ OpenNebula ทำการเพิ่มทรัพยากรให้กับแอพพลิเคชันเซิร์ฟเวอร์

แบ่งการทดลองออกเป็นสองส่วนด้วยกันคือ เปรียบเทียบระหว่างการขยายขนาดเซิร์ฟเวอร์เสมือน

และการเพิ่มจำนวนเซิร์ฟเวอร์เสมือน โดยมีผู้ใช้งานเพิ่ม 1 คน ทุกๆ 20 วินาที ผลลัพธ์แสดงไว้ในรูปที่ 4 และเปรียบเทียบระหว่างการขยายขนาดเซิร์ฟเวอร์เสมือนและการเพิ่มจำนวนเซิร์ฟเวอร์เสมือน โดยมีผู้ใช้งานเพิ่ม 1 คน ทุกๆ 40 วินาที ผลลัพธ์แสดงไว้ในรูปที่ 5 จากรูปที่ 4 การเพิ่มทรัพยากรโดยใช้วิธีเพิ่มจำนวนเซิร์ฟเวอร์เสมือนนั้นมีเปอร์เซ็นต์ของเวลาการตอบสนองเฉลี่ยที่มากกว่าข้อตกลง (เส้นสัญลักษณ์สี่เหลี่ยม) สูงถึง 22.5% เปรียบเทียบกับกราฟสีน้ำเงินซึ่งเป็นการทดลองโดยการขยายขนาดทรัพยากร ซึ่งเวลาของการตอบสนองเฉลี่ยไม่เกินค่าที่ตกลงไว้ จากรูปที่ 5 การเพิ่มทรัพยากรโดยใช้วิธีเพิ่มจำนวนเซิร์ฟเวอร์เสมือนนั้นมีเปอร์เซ็นต์ของเวลาการตอบสนองเฉลี่ยที่มากกว่าข้อตกลง 7.5% เปรียบเทียบกับกราฟสีน้ำเงินซึ่งเป็นการทดลองโดยการขยายขนาดทรัพยากรซึ่งเวลาของการประมวลผลเฉลี่ยไม่เกินค่าที่ตกลงไว้

5. สรุป

ในงานวิจัยนี้ ได้ศึกษาเกี่ยวกับการเพิ่มประสิทธิภาพของการจัดสรรทรัพยากรบนระบบประมวลผลแบบกลุ่มเมฆโดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือน และได้ทำการทดลองเปรียบเทียบระหว่างการเพิ่มทรัพยากรโดยใช้วิธีขยายขนาดเซิร์ฟเวอร์เสมือนและวิธีเพิ่มจำนวนเซิร์ฟเวอร์เสมือน โดยผลลัพธ์ที่ได้แสดงให้เห็นว่าการจัดสรรทรัพยากรโดยใช้วิธีขยายขนาดนั้นจะมีประสิทธิภาพดีกว่าเมื่อจำนวนของผู้ใช้งานเพิ่มขึ้นอย่างรวดเร็ว แต่จะมีประสิทธิภาพใกล้เคียงกันเมื่อจำนวนผู้ใช้เพิ่มขึ้นอย่างช้าๆ ซึ่งสอดคล้องกับบทวิจัย “Above the Clouds: A Berkeley View of Cloud Computing” ที่การเริ่มต้นการทำงานของเซิร์ฟเวอร์เสมือนนั้นใช้เวลา 2-3 นาที

จากงานวิจัยนี้สามารถนำไปสู่การศึกษาเพิ่มเติมในเรื่องของการนำการจัดสรรทรัพยากรด้วยวิธีเพิ่มจำนวนเซิร์ฟเวอร์เสมือนและวิธีขยายขนาดเซิร์ฟเวอร์เสมือนมาใช้ร่วมกันโดยพิจารณาจากช่วงเวลาที่เหมาะสม และศึกษาเพิ่มเติมในเรื่องของวิธีลดค่าใช้จ่ายของผู้ใช้งานระบบประมวลผลแบบกลุ่มเมฆจากการใช้การจัดสรรทรัพยากรแบบต่างๆ และเพื่อเพิ่มความชัดเจนของผลการทดลอง ผู้ทดลองเสนอให้เพิ่มเครื่องคอมพิวเตอร์ที่ใช้ในการทดลองหรือทดลองกับระบบประมวลผลแบบกลุ่มเมฆที่เปิดให้บริการอยู่ในปัจจุบัน

เอกสารอ้างอิง

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, “Above the Clouds: A Berkeley View of Cloud Computing,” *Technical Report No.UCB/EECS-2009-28*, EECS Department, U.C. Berkeley, February 2009.

[2] Ye Hu, Johnny Wong, Gabriel Iszlai, and Marin Litoiu, “Resource Provisioning for Cloud Computing,” *CASCON ‘09 Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research*, November 2009, pp.101-111.

[3] Trieu C. Chieu, Ajay Mohindra, Alexei A. Karve, and Alla Segal, “Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment,” *IEEE International Conference on e-Business Engineering, 2009*, pp.281-286.

[4] Oracle, Average Response Time. http://docs.oracle.com/cd/E19575-01/821-0178/abfch/index. html.

[5] Kernel-based Virtual Machine (KVM), Change Log http://www.linux-kvm.org/page/ChangeLog.

[6] Wikipedia, Comparison of platform virtual machines, http://en.wikipedia.org/wiki/ Comparison_of_platform_virtual_machines.

[7] Ashima Agarwal and Shangruff Raina, “Live Migration of Virtual Machines in Cloud,” *International Journal of Scientific and Research, vol. 3, June 2012*.