พร็อกซี่เซิร์ฟเวอร์ที่มีคุณสมบัติในการกระจายสมดุลภาระและความคงอยู่ได้ : กรณีศึกษาเครือข่ายคอมพิวเตอร์มหาวิทยาลัยสงขลานครินทร์

Load balancing and high availability proxy servers: case study the PSU campus network

วิบูลย์ วราสิทธิชัย*

บทคัดย่อ

ระบบบริการเว็บแคชเดิมมหาวิทยาลัยสงขลานครินทร์ใช้พร็อกซี่เซิร์ฟเวอร์จำนวนหนึ่งในการให้บริการ โดยใช้เทคนิคการกระจายการร้องขอบริการด้วยการวนรอบชื่อโดเมน (Round Robin DNS) ซึ่งเป็นวิธีการที่ใช้งานได้ดี เข้าใจหลักการได้ง่าย เนื่องจากไม่มีประสิทธิภาพมากนัก จึงทดแทนระบบดังกล่าวด้วยระบบคลัสเตอร์ที่มีเซิร์ฟเวอร์หลายตัวให้บริการเสมือนเป็นเซิร์ฟเวอร์เพียงเครื่องเดียวด้วยเทคโนโลยี Linux Virtual Server (LVS) บทความวิชาการฉบับนี้ อธิบายการทำงานของระบบพร็อกซี่เซิร์ฟเวอร์ที่มีคุณสมบัติของการกระจายสมดุลภาระและความคงอยู่ได้ด้วยการเชื่อมต่อแบบ IPVS/DR ซึ่งใช้ในกรณีที่เซิร์ฟเวอร์ทุกเครื่องวางอยู่บนเครือข่ายเดียวกัน

คำสำคัญ

Linux Virtual Server (LVS), IPVS/DR,Load Balancing, High Availability, Squid, Web cache

---

* **นักวิชาการคอมพิวเตอร์ 6** ศูนย์คอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

1. บทนำ

ในปัจจุบันเครือข่ายคอมพิวเตอร์ของมหาวิทยาลัยสงขลานครินทร์มีการเติบโตขึ้นเป็นอย่างมาก เนื่องจากมีการเพิ่มขึ้นของคณะ หน่วยงาน และจำนวนนักศึกษา สำหรับการใช้งานเวิลด์ไวด์เว็บซึ่งเป็นบริการหนึ่งที่มีผู้ใช้งานจำนวนมาก การใช้งานดังกล่าวจะถูกส่งผ่านไปยังพร็อกซี่เซิร์ฟเวอร์เพื่อทำหน้าที่ในการร้องขอข้อมูลแล้วส่งกลับมายังผู้ใช้งานพร็อกซี่เซิร์ฟเวอร์เพียงเครื่องเดียวไม่สามารถให้บริการกับผู้ใช้จำนวนมากได้ การเพิ่มจำนวนเซิร์ฟเวอร์และใช้เทคนิค Round Robin DNS เป็นวิธีการเพิ่มประสิทธิในการให้บริการ

วิธีการวนรอบชื่อโดเมนที่ให้บริการนี้ ยังมีข้อเสียในหลาย ๆ จุด เช่น การกระจายการร้องขอไปยังเซิร์ฟเวอร์ยังคงอยู่ ถึงแม้ว่าเซิร์ฟเวอร์นั้น ๆไม่สามารถทำงานในขณะนั้นได้ทำให้เกิดความผิดพลาดขึ้น และเซิร์ฟเวอร์แต่ละเครื่องอาจมีภาระไม่เท่ากัน

เพื่อให้สามารถใช้งานทรัพยากรได้อย่างคุ้มค่า จึงมีแนวคิดที่นำเอาวิธีการกระจายสมดุลภาระไอพี (IPVS Load Balancing) เข้ามาใช้งานแทนระบบการกระจายภาระแบบเดิม โดยวิธีการนี้จะมองเซิร์ฟเวอร์ที่ให้บริการทั้งหมดเป็นกลุ่มเดียวกันเรียกว่าคลัสเตอร์ (Cluster) โดยจะมีเซิร์ฟเวอร์หนึ่งเครื่องทำหน้าที่รับการร้องขอบริการ(Director) ที่เข้ามา จากนั้นจะทำการกระจายสมดุลภาระไปยังเซิร์ฟเวอร์ที่ให้บริการ

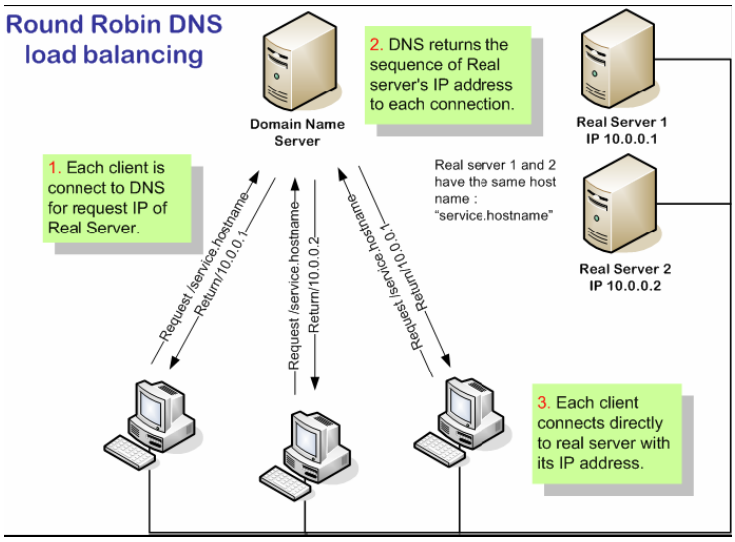
เพื่อให้มั่นใจว่าระบบสามารถให้บริการได้จึงเพิ่ม Director ขึ้นอีกหนึ่งเครื่องเพื่อทำหน้าที่สำรองในกรณีที่เซิร์ฟเวอร์ตัวหลักไม่สามารถทำงานได้ โดยการควบคุมจากโปรแกรมชื่อ heartbeat และเรียกคุณสมบัติของวิธีการนี้ว่า "ความคงอยู่ได้" (High Availability)

พร็อกซี่เซิร์ฟเวอร์ที่มีคุณสมบัติในการกระจายสมดุลภาระและความคงอยู่ได้ที่จะนำเสนอนี้ใช้ระบบ LVS ที่มีรูปแบบการเชื่อมต่อแบบ IPVS/DR

2. เทคโนโลยีที่เกี่ยวข้องกับการกระจายสมดุลภาระและความคงอยู่ได้

2.1 Round Robin DNS

Round Robin DNS [2] เป็นหนึ่งในหลายเทคนิคที่นิยมใช้กันอย่างกว้างขวางในการกระจายสมดุลภาระ [3]



รูปที่ 1 แสดงการทำงานของ Round Robin DNS

ใช้หลักการที่ว่า ชื่อโดเมนของเซิร์ฟเวอร์หนึ่งชื่อถูกกำหนดให้เป็นหมายเลขไอพีของเซิร์ฟเวอร์ทุกเครื่องในคลัสเตอร์ โดเมนเนมเซิร์ฟเวอร์จะทำการวนรอบส่งหมายเลขไอพีที่ต่างกันในแต่ละครั้งที่มีการร้องขอชื่อโดเมนเดียวกัน (ดูรูปที่ 1) วิธีการนี้ทำให้เกิดความยืดหยุ่นสูงในการเพิ่มจำนวนเซิร์ฟเวอร์ และผู้ใช้ไม่รู้ว่ากำลังติดต่อกับเครื่องใด ข้อด้อยของวิธีการนี้คือ DNS ไม่รู้ถึงความพร้อมในการให้บริการของเซิร์ฟเวอร์ในคลัสเตอร์ ไคลเอนต์จึงอาจได้รับหมายเลขไอพีของเครื่องดังกล่าว ปัญหานี้แก้ได้โดยการเลือกใช้ค่า Time To Live (TTL) ให้น้อย เพื่อให้ไคลเอนต์เก็บค่าแคชของหมายเลขไอพีนั้นไว้สั้น ๆ เช่น TTL เป็น 60 วินาที เป็นต้น

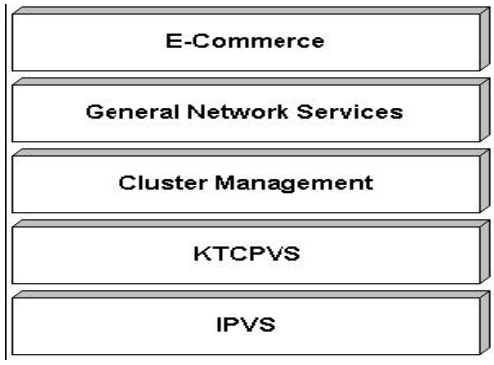
นอกจากนี้ยังเป็นการกระจายภาระที่ไม่สมดุล เนื่องจากเซิร์ฟเวอร์เครื่องหนึ่งอาจจะยังคงให้บริการอย่างหนักจากไคลเอนต์บางเครื่องที่ร้องขอในปริมาณมากอยู่แล้วต้องรับการเชื่อมต่อใหม่ที่เข้ามา ในขณะที่เซิร์ฟเวอร์อีกเครื่องอาจจะทำงานน้อยอยู่แต่ไม่อยู่ในจังหวะที่โดเมนเนมเซิร์ฟเวอร์ส่งหมายเลขไอพีไปยังไคลเอนต์ที่ร้องขอเข้ามาใหม่ เป็นต้น

2.2 The Linux Virtual Server Project

The Linux Virtual Server Project [4][5] พัฒนาโดย Wensong Zhang เป็นโครงงานที่ทำการศึกษาการสร้างระบบบริการเสมือน (Virtual Server System) ให้สามารถทำงานทางด้านการกระจายภาระและมีความคงทนสูงสำ หรับใช้ในเทคโนโลยีคลัสเตอริ่ง (Clustering Technology) เพื่อให้เป็นเครื่องบริการที่ใช้ ระบบปฏิบัติการลินุกซ์ ให้สามารถทำการบริการได้อย่างมีเสถียรภาพ มีความยืดหยุ่นสูง (Scalability Reliability and Serviceability)

2.2.1 LVS Framework

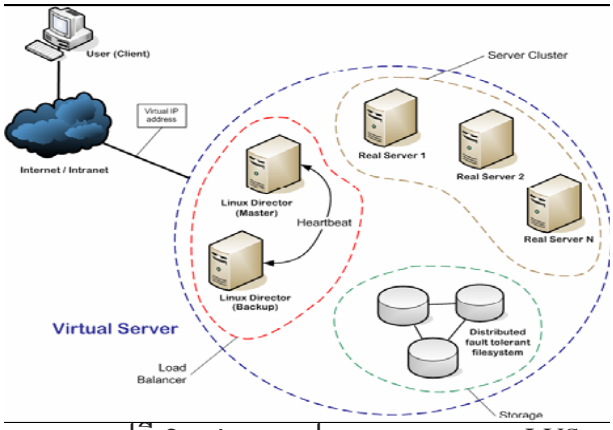
โครงการ LVS นี้พัฒนาซอฟต์แวร์เพื่อกระจายสมดุลภาระระดับไอพี (IP Load Balancing Software, IPVS) การกระจายภาระตามระดับของแอพลิเคชัน (Application-Level Load Balancing Software, TCPVS) และการจัดการคลัสเตอร์ ซึ่งเป็นส่วนประกอบในเฟรมเวิร์ก (Framework) ของระบบ LVS ดังรูปที่ 2



รูปที่ 2 แสดงส่วนประกอบใน LVS Framework

2.2.2 LVS Components

ระบบ LVS จะมีองค์ประกอบดังรูปที่ 3 ดังนี้



รูปที่ 3 แสดงองคประกอบของระบบ LVS

ส่วนแรกคือโหลดบาลานเซอร์ (Load Balancer) เรียกเซิร์ฟเวอร์ที่ใช้ในการกระจายภาระว่า "ไดเรกเตอร์ เซิร์ฟเวอร์" (Director Server) หรือลินุกซ์ไดเรกเตอร์ (Linux Director) ส่วนที่สองคือเซิร์ฟเวอร์คลัสเตอร์ (Server Cluster) เรียกเซิร์ฟเวอร์ที่ให้บริการว่า "เรียลเซิร์ฟเวอร์" (Real Server) และส่วนที่สามคือส่วนของระบบที่ทำหน้าที่ในการเก็บข้อมูล (Storage)

ดังนั้นเมื่อมองระบบคลัสเตอร์นี้ทั้งระบบ จะเสมือนว่าไคลเอนต์ได้ทำการติดต่อกับเซิร์ฟเวอร์เพียงเครื่องเดียวจึงเรียกระบบนี้ว่า "ระบบบริการเสมือน" (Virtual Server) และเมื่อระบบนี้ใช้ระบบปฏิบัติการลินุกซ์ในการทำงานจึงเรียกได้อีกชื่อว่า "ระบบบริการเสมือนโดยใช้ระบบปฏิบัติการลินุกซ์" (Linux Virtual Server, LVS)

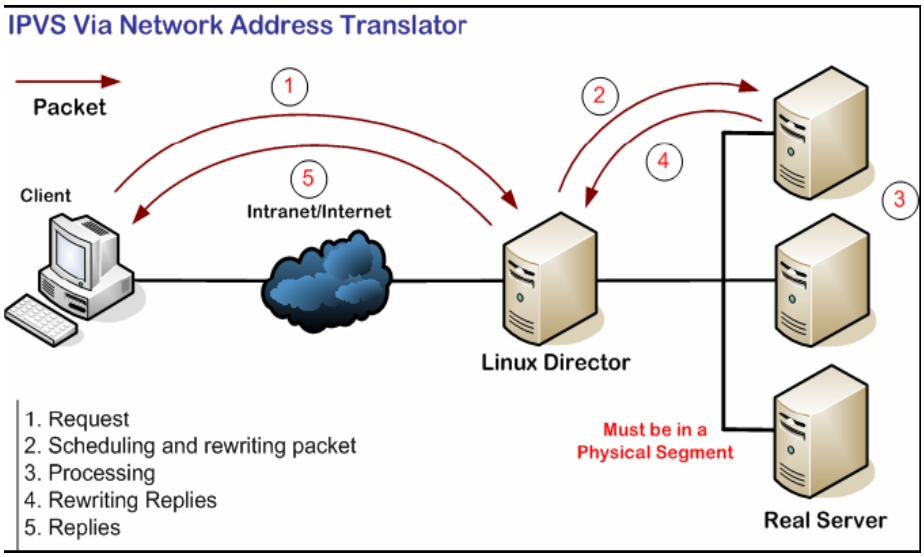
2.2.3 IPVS

ระบบ LVS แบบ IPVS นี้กระจายสมดุลภาระระดับไอพีของบริการในเครือข่ายโดยทำงานเป็นเลเยอร์ 4 สวิตชิ่ง (Layer 4 Switching) ที่รองรับโปรโตคอลทีซีพี (TCP) และยูดีพี (UDP) ซึ่งทำให้สามารถกระจายภาระไปยังเรียล เซิร์ฟเวอร์ที่ใช้ระบบปฏิบัติการที่แตกต่างกันได้ถ้าระบบปฏิบัติการนั้นรองรับโปรโตคอลทีซีพีและยูดีพี

IPVS จะมีรูปแบบการเชื่อมต่ออยู่สามแบบคือ IPVS via Network Address Translation (IPVS/NAT), IPVS via Direct Routing (IPVS/DR) และ IPVS via IP Tunneling (IPVS/TUN) แต่ละแบบมีหลักการทำงาน ดังนี้

IPVS/NAT

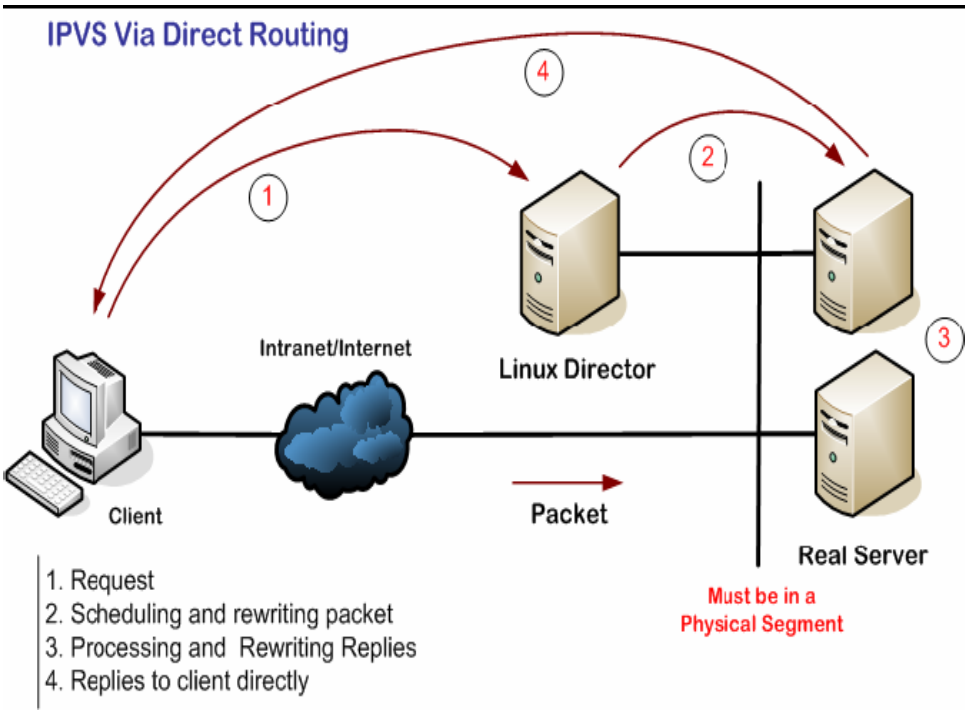
ลินุกซ์ไดเรกเตอร์และเรียลเซิร์ฟเวอร์ต่าง ๆ ที่อยู่ภายใน ระบบคลัสเตอร์จะทำการเชื่อมต่อกันโดยสวิตช์ (Switch) หรือฮับ (Hub) ในรูปที่ 4 ลินุกซ์ไดเรกเตอร์จะทำการเลือก เซิร์ฟเวอร์ภายในคลัสเตอร์ที่จะทำการประมวลผล จากนั้นหมายเลขไอพีและหมายเลขพอร์ตของแพ็กเก็ตที่เข้ามาจะถูกเขียนขึ้นใหม่ให้เป็นหมายเลขไอพีและหมายเลขพอร์ตของเซิร์ฟเวอร์ที่ได้เลือกไว้และส่งออกไป เมื่อลินุกซ์ไดเรกเตอร์ได้รับแพ็กเก็ตตอบกลับจากเรียลเซิร์ฟเวอร์ ระบบจะเขียนหมายเลขไอพีและหมายเลขพอร์ตให้เป็นหมายเลขของไอพีเสมือนเพื่อส่งกลับไปยังไคลเอนต์



รูปที่ 4 แสดงการเชื่อมต่อและลำดับการทำงานของการส่งต่อแพ็กเก็ตแบบ IPVS/NAT

IPVS/DR

วิธีการนี้เมื่อลินุกซ์ไดเรกเตอร์ได้รับสัญญาณร้องขอบริการแล้ว จะทำการเปลี่ยนฮาร์ดแวร์แอดเดรส (MAC Address) ในเดตาเฟรม (Data Frame) ให้เป็นของเซิร์ฟเวอร์ภายในคลัสเตอร์ที่ได้เลือกไว้ จากนั้นจึงส่งแพ็กเก็ตงกล่าวไปตามเครือข่าย เมื่อเซิร์ฟเวอร์ที่ถูกเลือกได้รับแพ็กเก็ตก็จะพบว่ามีหมายเลขไอพีปลายทางอยู่ที่ลูปแบ็คอินเตอร์เฟสของตัวมันเอง หลังจากที่ทำงานกับสัญญาณร้องขอที่ได้แล้วจึงส่งผลลัพธ์กลับคืนไปยังไคลเอนต์โดยตรง ดูรูปที่ 5 ประกอบ

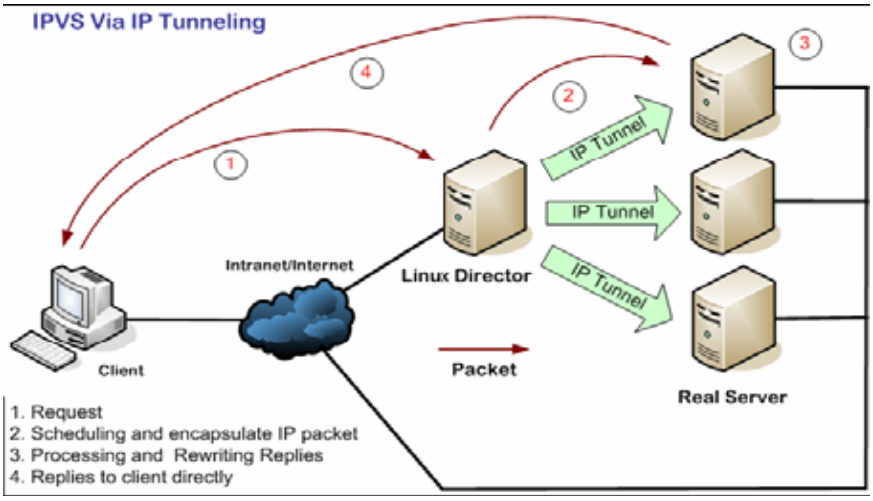


รูปที่ 5 แสดงการเชื่อมต่อและลำดับการทำงานของการส่งต่อแพ็กเก็ตแบบ IPVS/DR

IPVS/TUN

วิธีการนี้จะเป็นเทคนิคที่ทำการห่อหุ้มไอพีเดตาแกรม (IP Datagram Encapsulate) ภายในไอพีเดตาแกรมอีกทีหนึ่ง

ลินุกซ์ไดเรกเตอร์ได้รับการร้องขอจะทำการห่อหุ้มไอพีดาต้าแกรมของแพ็กเก็ตที่ได้มาไว้ภายในไอพีเดตาแกรมใหม่แล้วทำการส่งแพ็กเก็ตนั้นออกไปยังเซิร์ฟเวอร์ที่ถูกเลือก เมื่อเรียลเซิร์ฟเวอร์ได้รับแพ็กเก็ตแล้วจึงทำการคลายการห่อหุ้ม (Decapsulate) จึงพบว่าหมายเลขไอพีปลายทางตรงกับหมายเลขไอพีเสมือนของอินเตอร์เฟสแบบ Tunnel ที่เซิร์ฟเวอร์นั้นได้ติดตั้งไว้ เมื่อทำงานเสร็จเซิร์ฟเวอร์ส่งผลลัพธ์กลับไปยังไคลเอนต์โดยตรง ดังแสดงในรูปที่ 6



รูปที่ 6 แสดงการเชื่อมต่อและลำดับการทำงานของการส่งต่อแพ็กเก็ตแบบ IPVS/TUN

2.2.4 Scheduling Algorithm

ลินุกซ์ไดเรกเตอร์กระจายสมดุลภาระไปยังเซิร์ฟเวอร์ที่อยู่ภายในคลัสเตอร์โดยการส่งข้อมูลที่เป็นไปตามโปรโตคอลทีซีพีและยูดีพี ซึ่งกระบวนการเหล่านี้ได้ถูกนิยาม ไว้ในเคอร์เนลของลินุกซ์ โดยเรียกว่ารายการกฎเกณฑ์ (Scheduling Algorithm) มีอยู่หลายกฎเกณฑ์ เช่น

1. Round-Robin (rr)
2. Least - Connection (lc)
3. Weighted Least - Connection (wlc)
4. Destination Hashing (dh)
5. Source Hashing (sh) และอื่น ๆ

2.3. ldirectord

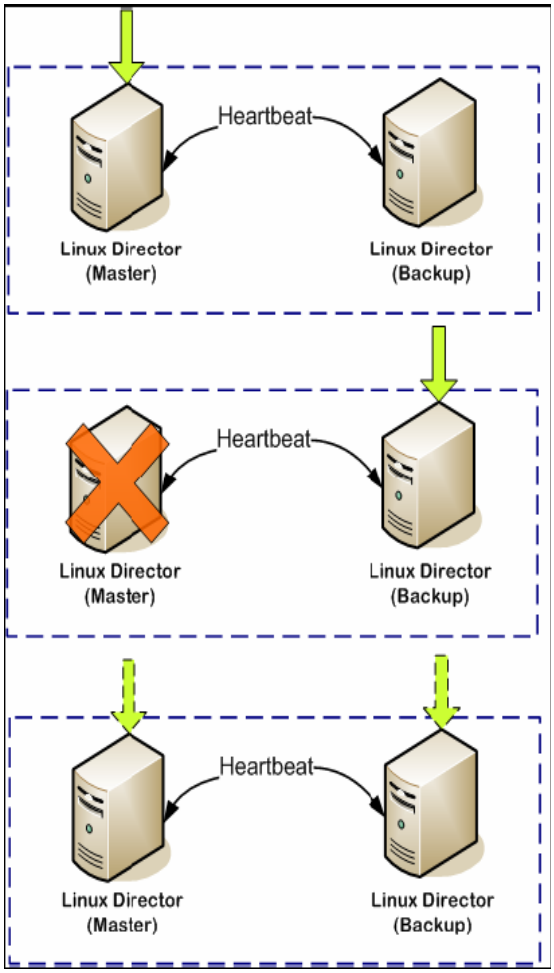
ลินุกซ์ไดเรกเตอร์เดมอน (Linux Director Daemon:ldirectord) เป็นโปรแกรมทำหน้าที่มอนิเตอร์และจัดการกับเรียลเซิร์ฟเวอร์ที่อยู่ในคลัสเตอร์ของ LVS เดมอนนี้ทำงานอยู่ในลินุกซ์ไดเรกเตอร์ มันจะมอนิเตอร์และตัดสินใจว่าเรียลเซิร์ฟเวอร์เครื่องใดยังคงอยู่ก็นำมาใส่ไว้ในรายการหากเครื่องใดไม่ทำงาน ก็จะถูกนำออกจากรายการ

ldirectord ใช้มอนิเตอร์บริการ (services) ดังนี้ ftp,smtp, http, pop, pops, nntp, imap, imaps, ldap,https, dns, mysql, pgsql, sip รวมทั้งโปรโตคอลทีซีพีและยูดีพีอื่น ๆ ด้วย โดยการตรวจสอบจะไม่ใช้วิธีการระบุชื่อบริการโดยตรง เช่น ตรวจสอบทีซีพีพอร์ต 8080 บนเรียลเซิร์ฟเวอร์ว่ายอมรับการติดต่อ (Connect) หรือไม่ เป็นต้น

2.4. Heartbeat

วิธีการแก้ปัญหาลินุกซ์ไดเรกเตอร์หยุดทำงานซึ่งจะทำให้บริการหยุดไปด้วย จะใช้วิธีการติดตั้งลินุกซ์ไดเรกเตอร์อีกเครื่องหนึ่งเป็นเครื่องสำรอง (Backup) และทำงานร่วมกันแบบ Active/Standby โดยวิธีการนี้ใช้โปรแกรมมอนิเตอร์ชื่อ heartbeat [6] ที่อยู่บนเครื่องทั้งสองทำหน้าที่ตรวจสอบการทำงานซึ่งกันละกันในช่วงเวลาหนึ่ง ๆ

เมื่อใดก็ตามที่ลินุกซ์ไดเรกเตอร์ที่ทำหน้าที่เป็นเครื่องสำรองไม่ได้รับสัญญาณตอบรับจากลินุกซ์ไดเรกเตอร์หลัก (Master) ในช่วงเวลาที่กำหนดไว้ ลินุกซ์ไดเรกเตอร์สำรองจะดึงหมายเลขไอพีบริการที่เครื่องหลักถืออยู่มาเป็นของเครื่องสำรองแทน (Take over Resource) และจะทำหน้าที่แทนเครื่องหลักต่อไป เมื่อเครื่องหลักกลับทำงานได้เป็นปกติ สามารถกำหนดการทำงานได้เป็นสองวิธีคือ เครื่องหลักที่กลับมาทำงานได้นั้นกลายเป็นเครื่องสำรอง หรือ เครื่องหลักกลับมาให้บริการเช่นเดิม

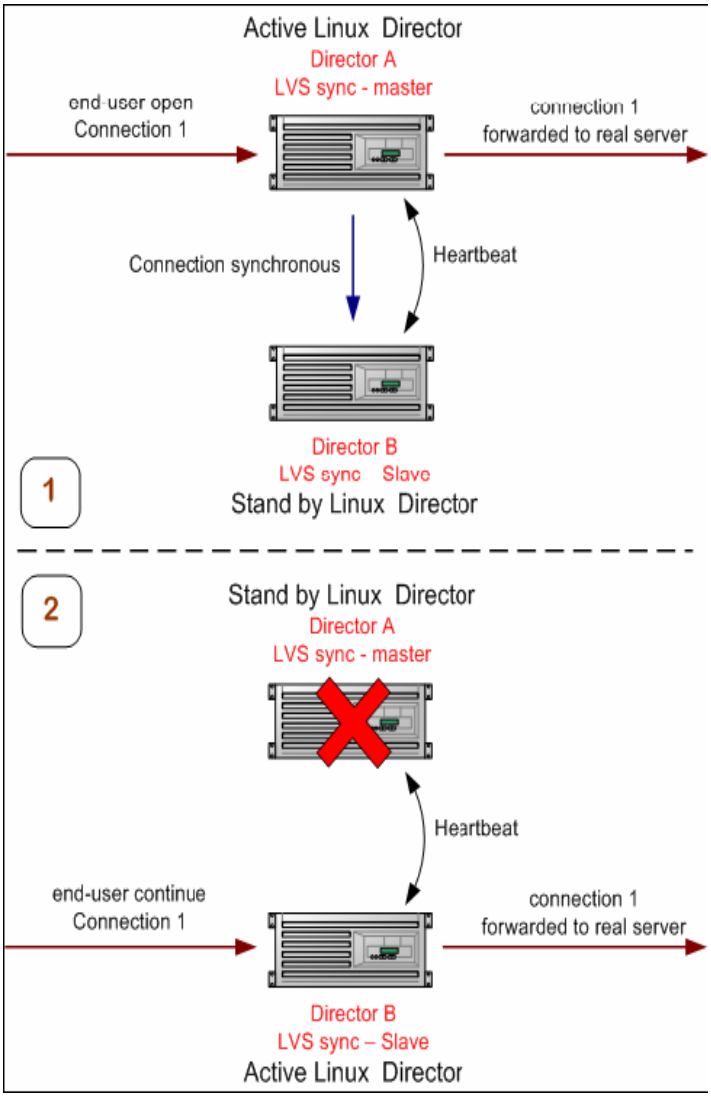


รูปที่ 7 แสดงการทำงานแบบ Active/Standby

2.5. Connection Synchronization

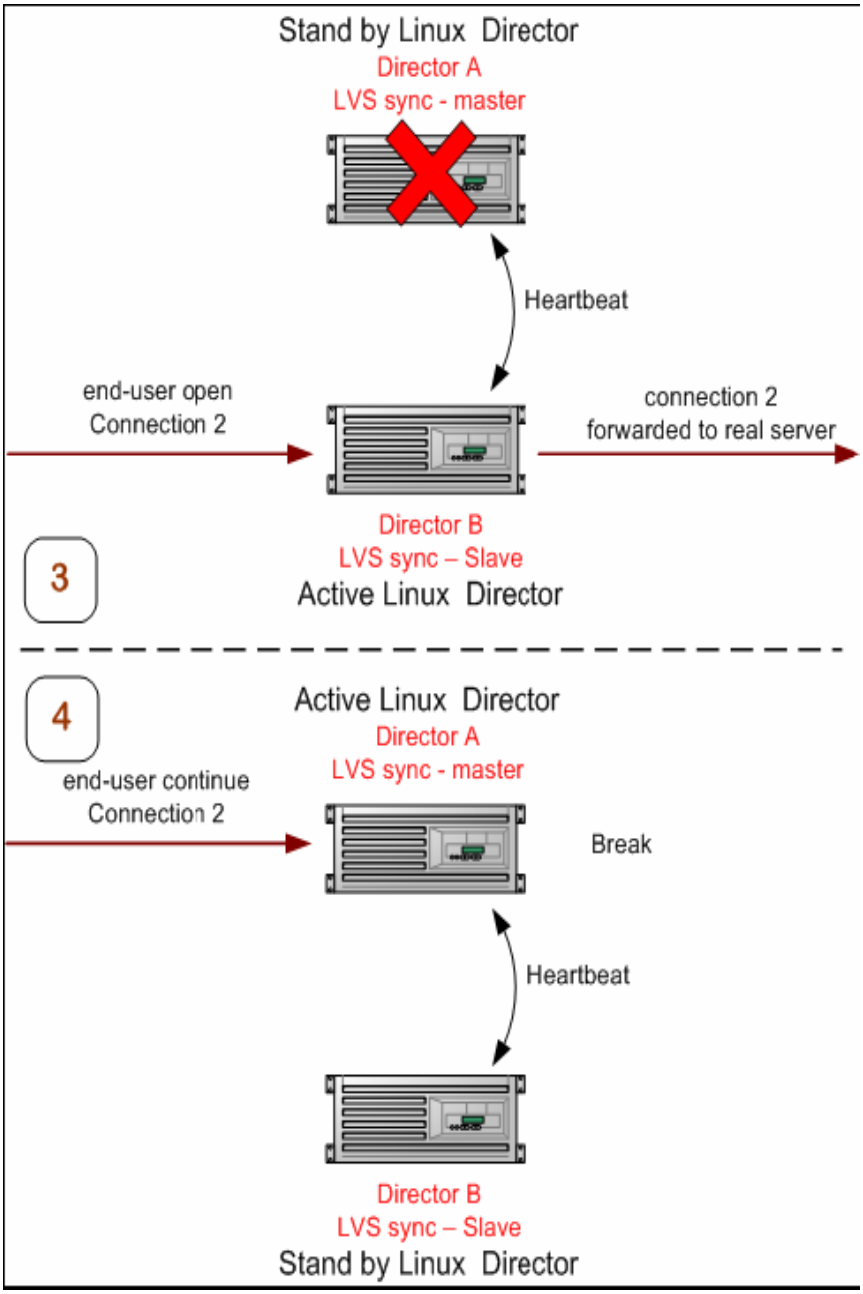
การเข้าจังหวะการเชื่อมต่อ จะเป็นการเก็บข้อมูลการสถาปนาการเชื่อมโยงแล้ว (Established Connection) ไว้ที่ลินุกซ์ไดเรกเตอร์ตัวหลักและตัวสำรอง เมื่อใดก็ตามที่ตัวสำรองให้บริการแทนตัวหลัก มันจะยังคงมีข้อมูลการเชื่อมต่อ ณ ปัจจุบัน และทำงานส่งแพ็กเก็ตนั้นต่อไปได้

ผู้ใช้เปิดการเชื่อมต่อที่ 1 กับเครื่องหลักและถูกเข้าจังหวะกับเครื่องสำรอง เมื่อเครื่องหลักหยุดทำงาน การเชื่อมต่อของผู้ใช้ดำเนินต่อไปได้ ดูรูปที่ 8



รูปที่ 8 แสดงวิธีการทำงานของการเข้าจังหวะการเชื่อมต่อ

ผู้ใช้เปิดการเชื่อมต่อที่ 2 กับเครื่องสำรองในขณะเครื่องหลักหยุดทำงาน เมื่อเครื่องหลักกลับทำงานได้ การเชื่อมต่อของผู้ใช้จะไม่สามารถดำเนินต่อไป เพราะไม่ได้เข้าจังหวะการเชื่อมต่อ ดูรูปที่ 9

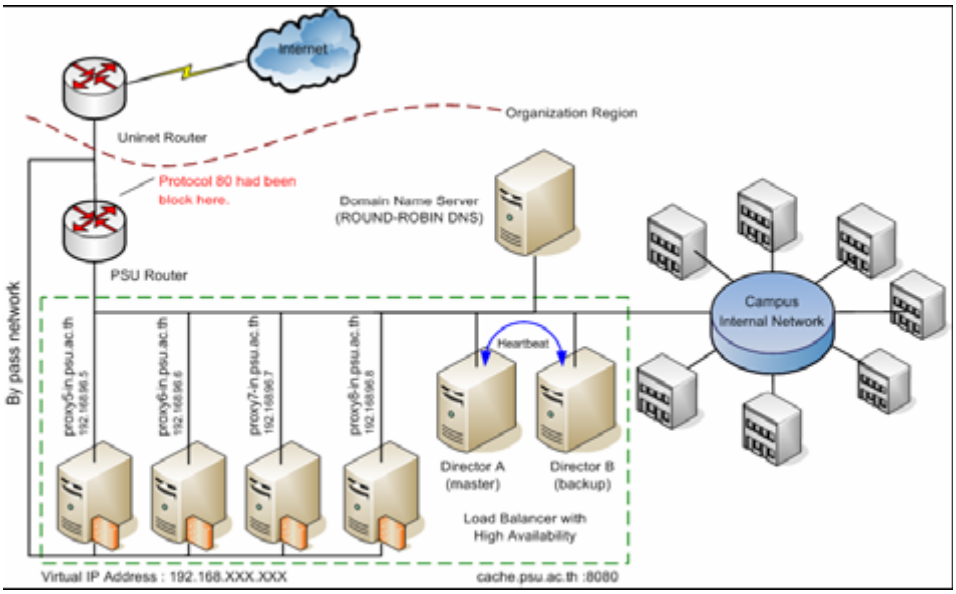


รูปที่ 9 แสดงวิธีการทำงานของการไม่เข้าจังหวะการเชื่อมต่อ

3. การติดตั้ง

3.1 การใช้งานที่มหาวิทยาลัยสงขลานครินทร์

เริ่มศึกษาระบบ LVS ตั้งแต่ประมาณเดือนมีนาคมพ.ศ. 2549 และเริ่มทดลองใช้งานจริงเมื่อ 30 พฤษภาคม พ.ศ. 2550 โดยติดตั้งลินุกซ์ไดเรกเตอร์จำนวน 2 เครื่องเพื่อทำงานในลักษณะ Active/Standby ร่วมกับพร็อกซี่เซิร์ฟเวอร์จำนวน 4 เครื่องที่ให้บริการเว็บแคชอยู่ก่อนแล้ว ดูรูปที่ 10



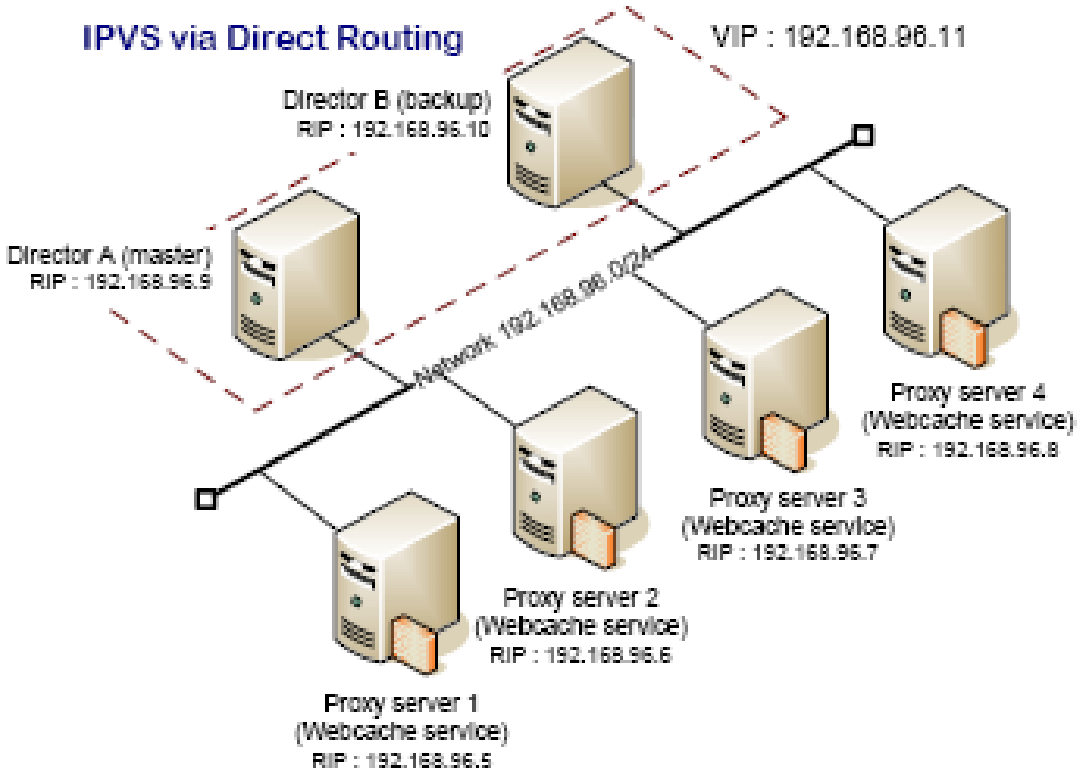
รูปที่ 10 แสดงระบบเครือข่ายที่พร็อกซี่เซิร์ฟเวอร์ตั้งอยู่

ข้อมูลคอมพิวเตอร์ที่ใช้ในการทดลองประกอบด้วย เครื่องคอมพิวเตอร์สำหรับลินุกซ์ไดเรกเตอร์ จำนวน 2 เครื่อง คือ Dual-Core Intel Xeon 5110 1.60 GHz Processor, 4 MB L2 cache, 1066 MHz FSB,RAM 1 GB, HD 72 GB, 10,000 rpm with 10/100/1000 2 ports

เครื่องคอมพิวเตอร์สำหรับพร็อกซี่เซิร์ฟเวอร์จำนวน 4 เครื่อง คือ Pentium IV 3.06 GHz processor, 512MB L2 cache, 533 MHz FSB, RAM 2 GB, HD 80 GB ATA 7,200 rpm with 10/100/1000 2 ports

3.2 ข้อกำหนดในการติดตั้ง

ชื่อโดเมนของการใช้บริการเว็บแคช คือ cache.psu.ac.th มีหมายเลขไอพี 192.168.96.11 สำหรับเป็นหมายเลขไอพีเสมือนของระบบนี้ ลินุกซ์ไดเรกเตอร์เครื่องหลักชื่อ lbproxy1 มีหมายเลขไอพี 192.168.96.9 ลินุกซ์ไดเรกเตอร์เครื่องสำรองชื่อ lbproxy2 มีหมายเลขไอพี 192.168.96.10 พร็อกซี่เซิร์ฟเวอร์ในคลัสเตอร์จำนวน 4 เครื่อง มีหมายเลขไอพีคือ 192.168.96.5 - 192.168.96.8 ตามลำดับ หมายเลขพอร์ตคือ 8080 ใช้รูปแบบการเชื่อมต่อ แบบ IPVS via Direct Routing ดูรูปที่ 11



รูปที่ 11 แสดงพร็อกซี่เซิร์ฟเวอร์ที่มีคุณสมบัติในการกระจายสมดุลภาระและความคงอยู่ได้โดยใช้เทคนิค IPVS/DR

ระบบปฏิบัติการที่ใช้บนลินุกซ์ไดเรกเตอร์คือ Fedora Core 6 เคอร์เนลเวอร์ชั่น 2.6.20-1.2948.fc6 และระบบปฏิบัติการที่ใช้บนพร็อกซี่เซิร์ฟเวอร์คือ Fedora Core 5 เคอร์เนลเวอร์ชั่น 2.6.15-1.2054_FC5smp

ซอฟต์แวร์ที่ติดตั้งในลินุกซ์ไดเรกเตอร์ คือ ipvsadm-1.24-8.1.i386.rpm ,ldirectord-2.0.8-1.fc6.i386.rpm และ heartbeat-2.0.8-1.fc6.i386.rpm ซอฟต์แวรที่ติดตั้งเพื่อใช้ทำพร็อกซี่เซิร์ฟเวอร์คือ squid

2.6.STABLE4

3.3 ขั้นตอนการติดตั้ง

ที่มหาวิทยาลัยสงขลานครินทร์จะมี Linux YUM Repository Server เก็บ Linux Fedora Core Distribution ช่วยให้ผู้ดูแลระบบสามารถติดตั้งและปรับปรุงซอฟต์แวร์ได้สะดวก และเนื่องจากลินุกซ์ที่มีเคอร์เนลตั้งแต่เวอร์ชั่น 2.6 ขึ้นไปสนับสนุน LVS อยู่แล้วจึงติดตั้งด้วยคำสั่ง yum ได้เลย

ในขั้นตอนการติดตั้ง หากต้องการติดตั้งให้มีคุณสมบัติการกระจายสมดุลภาระเพียงอย่างเดียว (ใช้ลินุกซ์ไดเรกเตอร์ 1 เครื่อง) ให้ติดตั้งซอฟต์แวร์ ipvsadm และ ldirectord หากต้องการติดตั้งให้มีคุณสมบัติทั้งการกระจายสมดุลภาระและความคงอยู่ได้ (ใช้ลินุกซ์ไดเรกเตอร์ 2 เครื่อง) ให้ติดตั้งซอฟต์แวร์ hearbeat เท่านั้นก็พอ ซึ่งโปรแกรมที่ติดตั้งซอฟต์แวร์จะทำการติดตั้ง ipvsadm และ ldirectord ให้โดยอัตโนมัติ อย่างหลังเป็นวิธีการที่เลือกใช้ในการทดลองนี้

นอกจากการติดตั้งซอฟต์แวร์แล้ว ต้องเพิ่มลูปแบ็คอินเตอร์เฟส lo:0 ที่มีหมายเลขไอพีเสมือน 192.168.96.11 ให้กับพร็อกซี่เซิร์ฟเวอร์ทุกเครื่องด้วยเพื่อให้รับแพ็กเก็ตที่ส่งมาจากลินุกซ์ไดเรกเตอร์ได้

ลินุกซ์ไดเรกเตอร์เครื่องหลักจะได้รับการสร้างอินเตอร์เฟสสำหรับหมายเลขไอพีเสมือนโดยอัตโนมัติโดยโปรแกรม heartbeat เมื่อใช้อินเตอร์เฟส eth0 สำหรับหมายเลขไอพีประจำเครื่อง ก็จะได้อินเตอร์เฟส eth0:0สำหรับหมายเลขไอพีเสมือน 192.168.96.11 ส่วนเครื่องสำรองจะไม่มีอินเทอร์เฟสดังกล่าวในขณะที่เครื่องหลักยังทำงานได้

เมื่อพร็อกซี่เซิร์ฟเวอร์ทุกเครื่องมีหมายเลขไอพีเสมือนเดียวกันกับลินุกซ์ไดเรกเตอร์จะทำให้มีปัญหาในเรื่องการตอบสนองต่อเออาร์พีเดทาแกรมร้องขอ (ARP Request) พร้อมกัน

เมื่อมีเครื่องที่ต้องการติดต่อกับหมายเลขไอพี 192.168.96.11 ร้องขอเข้ามา พร็อกซี่เซิร์ฟเวอร์ที่ได้เพิ่มลูปแบ็คอินเทอร์เฟสจะตอบฮาร์ดแวร์แอดเดรสกลับไปด้วย ทำให้ในตารางเออาร์พี (ARP Table) มีฮาร์ดแวร์แอดเดรสมากกว่า 1 แอดเดรสที่มีหมายเลขไอพีเสมือนเดียวกัน แสดงดังตัวอย่างผลลัพธ์ที่ได้จากคำสั่ง tcpdump

```
[root@lbproxy2 ~]# tcpdump -i eth0 not host 192.168.2.41 | grep arp | grep 192.1
68.96.11
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
17:28:03.398511 arp who-has 192.168.96.11 tell 192.168.96.4
17:28:03.398511 arp reply 192.168.96.11 is-at 00:1a:4b:de:9f:b0
17:28:03.398511 arp reply 192.168.96.11 is-at 00:13:21:1b:70:67
```

เนื่องจากเคอเนลเวอร์ชั่น 2.4 และ 2.6 ขึ้นไปไม่ได้จัดการปัญหาดังกล่าวให้ ดังนั้นจึงใช้เทคนิคซ่อนลูปแบ็คอินเทอร์เฟส [7] ของพร็อกซี่เซิร์ฟเวอร์ทุกเครื่องไม่ให้ตอบสนองเออาร์พีเดทาแกรมร้องขอ ด้วยการปรับแต่งระบบโดยเพิ่มบรรทัดคำสั่งในแฟ้ม /etc/sysctl.conf ดังนี้

net.ipv4.conf.all.arp_ignore = 1

net.ipv4.conf.all.arp_announce = 2

วิธีการกระจายสมดุลภาระที่เลือกใช้คือแบบ Least-Connection (lc) เพื่อกระจายภาระงานให้พร็อกซี่เซิร์ฟเวอร์ทุกตัวเท่า ๆ กัน เนื่องจากฮาร์ดแวร์ที่ใช้มีคุณลักษณะเดียวกัน ใช้วิธีการตรวจสอบทีซีพีพอร์ต 8080 โดยทำการสถาปนาการเชื่อมโยง หากทำสำเร็จก็ถือว่าเซิร์ฟเวอร์นั้นยังให้บริการได้อยู่

```
[root@lbproxy1 ~]# ipvsadm -L -n
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port           Forward Weight ActiveConn InActConn
TCP  192.168.96.11:8080 lc
  -> 192.168.96.8:8080            Route   1      13382      4
  -> 192.168.96.7:8080            Route   1      13381      0
  -> 192.168.96.5:8080            Route   1      13380      5
  -> 192.168.96.6:8080            Route   1      13381      5
```

ในขั้นตอนการติดตั้งคุณสมบัติความคงอยู่ได้ของระบบนี้จะใช้วิธีการให้โปรแกรม heartbeat ที่ทำงานบนลินุกซ์ไดเรกเตอร์ทั้งสองเครื่องมอนิเตอร์ซึ่งกันและกันโดยการส่งแพ็กเก็ตชนิดยูนิคาสต์ยูดีพีพอร์ต 694

เพื่อให้มีการเข้าจังหวะการเชื่อมต่อระหว่างลินุกซ์ไดเรกเตอร์ทั้งสองเครื่อง ได้ติดตั้งคำสั่งที่เครื่องหลักดังนี้

ipvsadm --start-daemon=master--mcast-interface=eth0

และติดตั้งคำสั่งที่เครื่องสำรองดังนี้

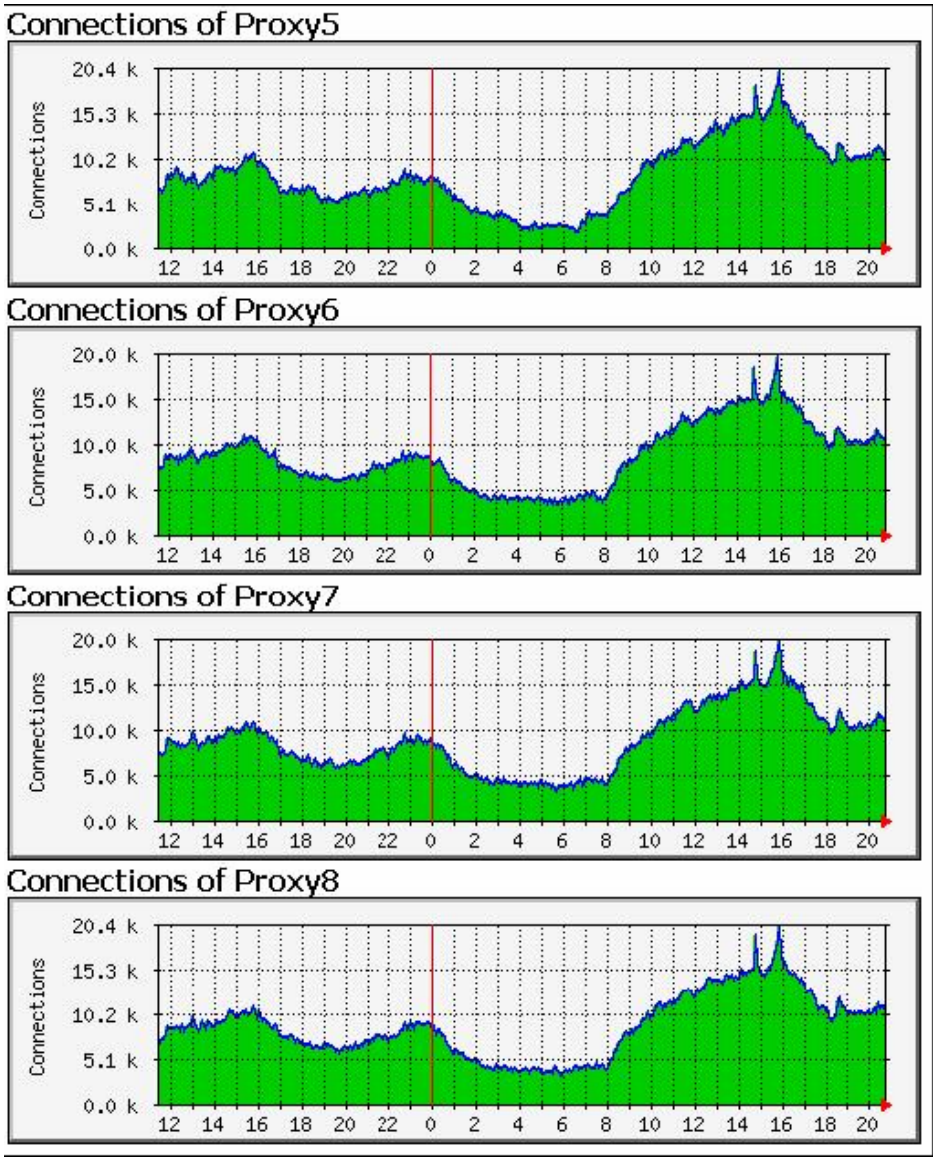
ipvsadm --start-daemon=backup--mcast-interface=eth0

และหลังจากที่มีการติดตั้งคำสั่ง พบว่ามีการส่งแพ็กเก็ตชนิดมัลติคาสต์ ดังตัวอย่างผลลัพธ์ที่ได้จากคำสั่ง tcpdump

```
[root@lbproxy2 ~]# tcpdump -nn -i eth0 not port 22
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
12:55:18.573130 IP 192.168.96.10.32771 > 192.168.96.9.694: UDP, length 207
12:55:18.573140 IP 192.168.96.10.32771 > 192.168.96.9.694: UDP, length 207
12:55:18.816717 IP 192.168.96.9.32771 > 224.0.0.81.8848: UDP, length 1420
12:55:18.816727 IP 192.168.96.9.32771 > 224.0.0.81.8848: UDP, length 1420
12:55:18.816730 IP 192.168.96.9.32771 > 224.0.0.81.8848: UDP, length 844
12:55:19.316745 IP 192.168.96.9.32768 > 192.168.96.10.694: UDP, length 207
```

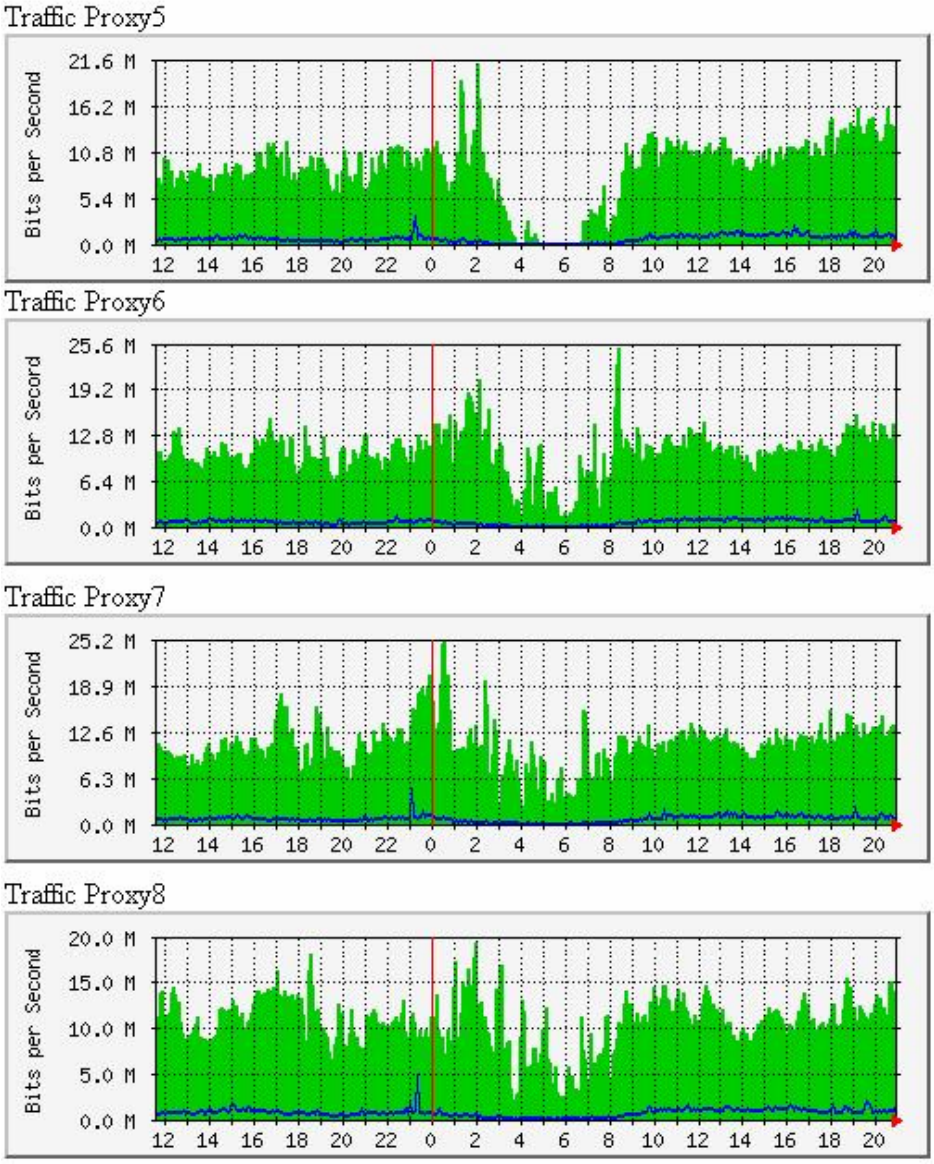
4. ผลที่ได้จากการทดลอง

พบว่าหลังจากนำระบบนี้มาใช้ พร็อกซี่เซิร์ฟเวอร์แต่ละตัวมีจำนวนการเชื่อมโยง (Connections) ใกล้เคียงกัน โดยเฉลี่ยอยู่ที่ประมาณ 10,000 รายการ/เครื่อง ดังรูปที่ 12



รูปที่ 12 แสดงจำนวนรายการการเชื่อมโยงของพร็อกซี่เซิร์ฟเวอร์ทั้ง 4 เครื่อง

และเมื่อดูจากกราฟ Traffic จะเห็นว่าปริมาณดาวน์โหลดของพร็อกซี่เซิร์ฟเวอร์ทั้ง 4 เครื่อง มีปริมาณใกล้เคียงกัน โดยเฉลี่ยอยู่ที่ประมาณ 12 เมกะบิตต่อวินาที/เครื่อง ดูรูปที่ 13 ซึ่งไม่ได้ลดลงจากก่อนหน้าการทดลองใช้งานระบบ LVS



5. บทสรุปและข้อเสนอแนะ

การจัดทำพร็อกซี่เซิร์ฟเวอร์ที่มีคุณสมบัติในการกระจายสมดุลภาระและความคงอยู่ได้ สามารถลดปัญหาการกระจายสมดุลภาระให้กับพร็อกซี่เซิร์ฟเวอร์ทั้ง 4 เครื่องที่ไม่เท่ากันซึ่งเป็นสาเหตุหนึ่งที่ทำให้เกิดความล่าช้าในการแสดงข้อมูลเว็บเพจในเครื่องที่มีภาระงานมากเกินไปบนสมมติฐานว่าช่องทางสื่อสารต่างประเทศมีขนาดแบนด์วิดธ์เพียงพอนอกจากนี้ยังช่วยเพิ่มเสถียรภาพของระบบในกรณีที่เกิดปัญหาขึ้นกับพร็อกซี่เซิร์ฟเวอร์เครื่องใดเครื่องหนึ่ง ระบบ LVS นี้จะถอดเครื่องดังกล่าวออกจากรายการโดยอัตโนมัติทำให้เกิดการใช้งานอย่างต่อเนื่อง และมีความยืดหยุ่นสำหรับผู้ดูแลระบบสามารถเพิ่ม/ลด

เซิร์ฟเวอร์ในระบบได้อย่างง่ายดายโดยไม่ต้องแก้ไขชื่อโดเมนตามวิธีการ Round Robin DNS แบบเดิม ยิ่งไปกว่านั้นการใช้ลินุกซ์ไดเรกเตอร์ถึง 2 เครื่องช่วยให้ความน่าจะเป็นที่จะมีโอกาสเกิด Single Point of Failure ของระบบน้อยลง

ในมุมมองของผู้เขียน พบว่าการสร้างระบบนี้ขึ้นมาได้และจะสามารถดูแลได้ตลอดไปนั้น ผู้ดูแลระบบจำเป็นต้องมีความรู้เรื่องสถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพีเป็นอย่างดี ต้องมีความรู้เรื่องระบบปฏิบัติการยูนิกซ์ในวิธีการใช้ Command Line และ Shell Script รวมทั้งความรู้ในเรื่อง Linux Network Administration เป็นอย่างดี เพื่อที่จะสามารถแก้ไขปัญหาได้อย่างรวดเร็ว และปรับแต่งระบบให้ดียิ่งขึ้นต่อไป

6. เอกสารอ้างอิง

[1] สุรศักดิ์ สงวนพงษ์. 2543. **สถาปัตยกรรมและโปรโตคอลทีซีพี/ไอพี.** กรุงเทพฯ : ซีเอ็ดยูเคชั่น.

[2] Round robin DNS - Wikipedia, the free encyclopedia.Available: http://en.wikipedia.org/wiki/Round_robin_DNS

[3] Making applications scalable with Load Balancing Available: http://1wt.eu/articles/2006_lb/

[4] The Linux Virtual Server Project - Linux Server Cluster for Load Balancing.Available: http://www.linuxvirtualserver.org

[5] LVS-HOWTO homepage Available: http://www.austintek.com/LVS/LVSHOWTO/HOWTO/

[6] The High Availability Linux Project Available: http://www.linux-ha.org/

[7] Using the "hidden" device feature in Linux 2.2 and 2.4 Available : http://www.ssi.bg/~ja/hidden.txt

****