**WorkShop : Load Balance Web Server (Server High Avaibility)**

**Workshop Scenario:**

**Load Balance Server (LBS)
 Internet Gateway + Workshop Desktop
lbs.testlab**

**eth0 : internet dhcp(nat)**

**eth1 : intranet 192.168.99.1**



**Internet**



**Web Server 2
webserver2.testlab**

**eth0 : intranet 192.168.99.20**

**Web Server 1
webserver1.testlab**

**eth0 : intranet 192.168.99.10**

**วิธีการ Import ไฟล์ ova เข้าโปรแกรม Oracle VM Virtual Box**

1. เปิดโปรแกรม Oracle VM VirtualBox ที่อยู่บน Desktop

2. ไปที่ File->Import Appliance เลือกไฟล์ที่อยู่ใน drive d:\workshop.ova

3. จะปรากฎหน้าแสดงรายการ VM จากนั้นกด Import

4. รอประมาณ 5-10 นาที ในการ Import

**Workshop Guideline:**

**Overall Guide:**

http://sysadmin.psu.ac.th/2014/05/21/workshop-load-balance-web-server-server-high-availability/

**Section 1: เรียนรู้เกี่ยวกับ Server High Availability**

**Goal:**

1. เข้าใจถึงคำว่า HA และ Uptime

2. เข้าใจถึงวิธีการทำ HA แบบต่าง ๆ

3. สามารถเข้าใจถึงวิธีการทำ Load Balance แบบต่าง ๆ

**Guide:**

http://sysadmin.psu.ac.th/2014/05/22/server-high-availability/

***“เคยได้ยินคำว่า Server เรามี Uptime ถึง 99% ไหม”***

 High Availability มีชื่อย่อในวงการสั้น ๆ ว่า HA องค์กรหรือหน่วยงานมากมายในปัจจุบันมีการวัดประสิทธิภาพการให้บริการด้วย HA โดยจะมีหน่วยการวัดที่เรียกว่า Uptime ซึ่งคิดจากเวลาเป็น % ของการให้บริการ โดยมีชื่อเรียกระดับการให้บริการตามตารางดังนี้

|  |  |  |  |
| --- | --- | --- | --- |
| **Availability %** | **Downtime per year** | **Downtime per month\*** | **Downtime per week** |
| 90% (“one nine”) | 36.5 days | 72 hours | 16.8 hours |
| 95% | 18.25 days | 36 hours | 8.4 hours |
| 97% | 10.96 days | 21.6 hours | 5.04 hours |
| 98% | 7.30 days | 14.4 hours | 3.36 hours |
| 99% (“two nines”) | 3.65 days | 7.20 hours | 1.68 hours |
| 99.5% | 1.83 days | 3.60 hours | 50.4 minutes |
| 99.8% | 17.52 hours | 86.23 minutes | 20.16 minutes |
| 99.9% (“three nines”) | 8.76 hours | 43.8 minutes | 10.1 minutes |
| 99.95% | 4.38 hours | 21.56 minutes | 5.04 minutes |
| 99.99% (“four nines”) | 52.56 minutes | 4.32 minutes | 1.01 minutes |
| 99.999% (“five nines”) | 5.26 minutes | 25.9 seconds | 6.05 seconds |
| 99.9999% (“six nines”) | 31.5 seconds | 2.59 seconds | 0.605 seconds |
| 99.99999% (“seven nines”) | 3.15 seconds | 0.259 seconds | 0.0605 seconds |

 ระดับที่หลายหน่วยงานหรือหลายอุปกรณ์พยายามจะโฆษณา จะอยู่ที่ระดับ Five nines หรือ Uptime 99.999% นั่นเอง ซึ่งจะเห็นได้ว่าจะมี Downtime แค่ 5.26 นาที ต่อปีเท่านั้น ซึ่งมองในด้านอุปกรณ์เครือข่าย จะมีปัจจัยน้อยที่จะทำให้เกิด Downtime แต่ในส่วนของ Server นั่นจะมีปัจจัยมากกว่าเพราะประกอบด้วยอุปกรณ์มากชิ้น แต่ปัญหาดังกล่าวในปัจจุบันน้อยลง หลังจากเข้าสู่ยุคของ Cluster หรือ Cloud เพราะจะมีเครื่องคอยทำงานแทนกันอยู่ตลอดเวลานั่นเอง



|  |
| --- |
| **ตัวอย่างการทำ HA ระดับองค์กร** |

 โดยคำว่า HA มักจะใช้คู่กับสิ่งที่เราจะต้องการจะสื่อว่าจะทำอย่างไรให้ระบบมี Uptime สูง ๆ เช่น Server High Availability ก็คือทำให้เครื่อง Server มีการให้สามารถทำงานได้โดยไม่ Hang ไม่ดับ ถ้าใช้กับคำว่า Web Server High Availability ก็หมายความว่า จะทำอย่างไรให้สามารถให้บริการ Web Site โดยไม่มีอาการล่ม ซึ่งบางครั้งเราจะมีการ ตัด Downtime ที่เป็นการวางแผนเอาไว้ก่อนแล้ว หรือเรียกสั้น ๆ ว่าการ Maintenance ระบบนั่นเอง

 การที่จะทำให้เกิดสิ่งที่เรียกว่า HA ก็มีอยู่ 3-4 แบบใหญ่ ๆ ที่สามารถนำมาใช้เพื่อเพิ่ม Uptime ได้ดังนี้

**การใช้เทคโนโลยีคลัชเตอร์ (Cluster)** : คือการนำเครื่องหลาย ๆ เครื่องมาตั้งค่าให้เหมือนกัน และทำงานได้เสมือนว่ามี Server แค่ตัวเดียว ในกรณที่เมื่อเครื่องใดเครื่องหนึ่งเสียหายจะไม่ส่งผลต่อการให้บริการ

**การทำรีดันแดนซ์ (Redundancy)** : คือการใช้อุปกรณ์จำนวนสองชุดขึ้นไปทำงานพร้อมกัน ทำให้ได้ประสิทธิภาพเต็มที่ นิยมใช้ในระบบเครือข่าย

**การทำโหลดบาลานซ์ (Load Balance)** : เราสามารถที่จะเอาเครื่อง Server หลาย ๆ เครื่องมาเชื่อมต่อกัน เพื่อให้ทำงานหลาย ๆ อย่างร่วมกัน

**การเรพพลีเคนและแบคอัพ (Replication & Backup)** : คือการคัดลอกข้อมูลจากระบบหลักไประบบรอง เมื่ออุปกรณ์หลักเกิดล่ม ก็จะมีการเปิดระบบสำรองใช้ได้ทันที

 โดยการทำ Load Balance ยังสามารถอธิบายเพิ่มเติมได้ 3 ลักษณะด้วยกันดังนี้ **Round-Robin**: คือการใช้วิธีหมุนวนไปเรื่อย ๆ ซึ่งอาจทำได้ทั้งแบบ Round-Robin ที่อุปกรณ์ Load Balance หรือการจด DNS แบบ Round Robin **Sticky**  : คือการยึดติดกับ Session ในกรณีที่เมื่อระบบมีการเลือก Server ไปแล้วก็จะให้ ผู้ใช้คนเดิมใช้ Server เดิมไปจนกว่าจะหมด Session (นึกถึงกรณีต้อง Login ก่อนใช้งาน ถ้าเปลี่ยน Server ไปเรื่อย ๆ คงไม่ดีแน่)
**Work Load** : คือการตรวจสอบประสิทธิภาพของ Server ก่อนที่จะเลือก โดยเลือกจากเครื่องที่ยังมีประสิทธิภาพดีกว่าก่อน ทำให้ระบบโดยรวมจะมีประสิทธิภาพดีกว่าแบบ Round-Robin เหมาะกับระบบที่มีประสิทธิภาพ Server แตกต่างกัน (เครื่องแรงปนเครื่องอ่อน)
 จากที่กล่าวไปเราสามารถใช้หลาย ๆ วิธีผสมกันก็ได้เพื่อจุดมุ่งหมายเดียวกันคือการเพิ่ม Uptime โดยเครื่องมือที่เป็น Opensource ที่ใช้วัด และที่มีมานานรวมทั้งเป็นยอมรับคือเครื่องมือตระกูล nagios และ cacti นั่นเอง แต่อย่าลืมไปละครับว่าเครื่องมือเหล่านั้นต้องรันอยู่บนเครื่องที่มี Uptime สูงด้วย ไม่ใช่ คนตรวจสอบก็ Down, อุปกรณ์ ก็ Down ไม่เช่นนั้นอาจจะได้ค่าหลอก ๆ ไม่ตรงกับความเป็นจริงก็เป็นได้

***“ถึงเวลาต้องทำ HA กันหรือยังเอ่ย ?”***

 **Reference :**
[1] http://en.wikipedia.org/wiki/High\_availability
[2] http://smart.tnis.com/th/articles/details/30/%E0%B8%9E%E0%B8%B1%E0%B8%92%E0%B8%99%E0%B8%B2%E0%B8%82%E0%B8%B5%E0%B8%94%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%AA%E0%B8%B2%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B8%96%E0%B9%83%E0%B8%AB%E0%B9%89%E0%B8%AD%E0%B8%87%E0%B8%84%E0%B9%8C%E0%B8%81%E0%B8%A3%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-HIGH-AVAILABILITY
[3] http://www.dctwo.com.au/virtual/virtual\_server\_hardware\_platform.html[4] http://bomber.exteen.com/20040813/cluster-load-balance

**Section 2: ติดตั้ง Apache2 + PHP5 บนเครื่อง LBS**

**Goal:**

1. สามารถติดตั้ง Apache บนเครื่อง Ubuntu 14.04

2. Config Apache ให้สามารถเปิดให้บริการ https

3. สามารถตรวจสอบ IP ผ่านคำสั่ง ifconfig

4. สามารถเปลี่ยน default port

5. สามารถติดตั้ง PHP5 เพื่อใช้งานกับ Apache

**Config Guide:**
1. เปิดเครื่อง LBS ขึ้นมา

2. ทำการติดตั้ง Apache2 Web Server

3. ทำการตั้งค่า SSL สำหรับ Apache

4. เรียนรู้วิธีการดู IP จากคำสั่ง ifconfig
5. เปิด Web Site หลังการติดตั้ง

6. ทำการเปลี่ยน Default Port เป็น port **8080**

7. ทำการติดตั้ง PHP5 เพื่อใช้งานกับ Apache2 Web Server

**Guide:**

http://sysadmin.psu.ac.th/2014/05/22/setting-apache2-php5-ubuntu-14-04/

**วิธีการติดตั้ง Apache2 Web Server**
1. ทำการติดตั้ง Apache ผ่าน apt-get ดังนี้ครับ

# sudo apt-get install -y apache2

2. ให้ทำการ Restart Apache ด้วยคำสั่ง

# sudo service apache2 restart

 จะเห็นได้ว่าจะมี Warning เกี่ยวกับ Domain Name ไม่ต้องตกใจครับ สามารถใช้งานได้ แต่เพราะยังไม่ได้จด Domain ให้กับ IP เครื่องที่ติดตั้ง ทุกครั้งที่ Restart ก็จะมีการฟ้องจนกว่าจะจดครับ

 ถ้าไม่ต้องการให้ขึ้น Warning ดังกล่าว ให้ทำการแก้ไขไฟล์ /etc/apache2/apache2.conf ดังนี้

- เปิดไฟล์ /etc/apache2/apache2.conf

sudo nano /etc/apache2/apache2.conf

- เพิ่มข้อความท้ายไฟล์ดังนี้ (วิธีการ save ให้กด Ctrl-X แล้วกด y แล้ว enter)

ServerName localhost

3. จากนั้นทำการ Restart Apache อีกครั้งจะไม่พบข้อความ Warning อีกแล้ว

# sudo service apache2 restart

4. ทำการทดสอบเข้าใช้งาน HTTP โดยพิมพ์คำสั่ง http://หมายเลขไอพีของเซิร์ฟเวอร์ เช่น

http://192.168.99.1

5. ในกรณีที่ต้องการติดตั้ง SSL ให้พิมพ์คำสั่งเพื่อเปิด site ดังนี้

sudo a2enmod ssl

sudo a2ensite default-ssl

6. จากนั้นให้สั่ง Restart Apache ตามปกติ

# sudo service apache2 restart

7. ทำการทดสอบเข้าใช้งาน HTTPS โดยพิมพ์คำสั่ง https://หมายเลขไอพีของเซิร์ฟเวอร์ เช่น

https://192.168.99.1

8. ในกรณีที่ต้องการยกเลิกการติดตั้ง SSL ให้พิมพ์คำสั่งเพื่อปิด site ดังนี้

sudo a2dismod ssl

sudo a2dissite default-ssl

9. จากนั้นให้สั่ง Restart Apache ตามปกติ

# sudo service apache2 restart

\*หมายเหตุ วิธีการดู IP ให้พิมพ์ดังนี้

ifconfig

ให้วิธีการดูให้สังเกตุตามรูปตัวอย่าง

- eth0 เป็น ip จาก nat interface
- eth1 เป็น ip จาก bridge interface
- lo เป็น ip จาก loopback interface

**วิธีการเปลี่ยน Apache Server Default Port**

\*ยกตัวอย่างการเปลี่ยนจาก port 80->8080
1. เปิดไฟล์ /etc/apache2/sites-enabled/000-default.conf

sudo nano /etc/apache2/sites-enabled/000-default.conf

2. แก้ไขข้อความในไฟล์ดังนี้ (วิธีการ save ให้กด Ctrl-X แล้วกด y แล้ว enter)

<VirtualHost \*:8080>

3. เปิดไฟล์ /etc/apache2/ports.conf

sudo nano /etc/apache2/ports.conf

2. แก้ไขข้อความในไฟล์ดังนี้ (วิธีการ save ให้กด Ctrl-X แล้วกด y แล้ว enter)

Listen 8080

3. จากนั้นให้สั่ง Restart Apache ตามปกติ

# sudo service apache2 restart

4. ทำการทดสอบเข้าใช้งาน โดยพิมพ์คำสั่ง http://หมายเลขไอพีของเซิร์ฟเวอร์:8080 เช่น

http://192.168.99.1:8080

**ตั้ง PHP5 เพื่อใช้งานกับ Apache2 Web Server**
1. ทำการติดตั้ง PHP5 ดังนี้

sudo apt-get install -y php5 libapache2-mod-php5

2. วิธีการสร้างไฟล์ตรวจสอบสามารถทำได้ดังนี้

- เปิดไฟล์ /var/www/html/info.php

sudo sh -c 'echo "<?php echo phpinfo(); ?>" > /var/www/html/info.php'

- จากนั้นทำการเปิด web http://หมายเลขไอพีของเซิร์ฟเวอร์/info.php ตัวอย่าง

http://192.168.99.1/info.php

**Section 3: ติดตั้ง Lighttpd + PHP5 บนเครื่อง LBS**

**Goal:**

1. สามารถติดตั้ง Lighttpd บนเครื่อง Ubuntu 14.04

2. Config Lighttpd ให้สามารถเปิดให้บริการ https

3. สามารถจัดการ SSL-Cer ที่จะนำไปใช้สำหรับเปิดบริการ https

4. สามารถเปลี่ยน default port

5. สามารถติดตั้ง PHP5 เพื่อใช้งานกับ Lighttpd

**Configuration Guide:**
1. ทำการติดตั้ง Lighttpd Web Server

2. ทำการตั้งค่า SSL สำหรับ Lighttpd

3. ทำการรวม Key จากโปรแกรม ssl-cert เพื่อนำไปใช้กับ Lighttpd

4. เปิด Web Site หลังการติดตั้ง

5. ทำการเปลี่ยน Default Port เป็น port **8081**

6. ทำการติดตั้ง PHP5 เพื่อใช้งานกับ Lighttpd Web Server

**Guide:**

http://sysadmin.psu.ac.th/2014/05/22/setting-lighttpd-php5-ubuntu-14-04/

**วิธีการติดตั้ง Lighttpd Web Server**
1. ทำการติดตั้ง Lighttpd ผ่าน apt-get ดังนี้ครับ

# sudo apt-get install -y lighttpd

2. ให้ทำการ Restart Lighttpd ด้วยคำสั่ง

# sudo service lighttpd restart

3. ทำการทดสอบเข้าใช้งาน HTTP โดยพิมพ์คำสั่ง http://หมายเลขไอพีของเซิร์ฟเวอร์ เช่น

http://192.168.99.1

4. ในกรณีที่ต้องการติดตั้ง SSL ให้พิมพ์คำสั่งเพื่อเปิด site ดังนี้

sudo lighttpd-enable-mod ssl

5. หลังจากนั้นทำการ สร้าง key จาก default cer (ถ้ามี key จริงสามารถนำมาแทนได้ในภายหลัง)

ทำการรวม Key โดยสร้างเป็นไฟล์ใหม่ เก็บไว้ที่ /etc/lighttpd/snakeoil.pem

sudo sh -c 'cat /etc/ssl/certs/ssl-cert-snakeoil.pem /etc/ssl/private/ssl-cert-snakeoil.key > /etc/lighttpd/snakeoil.pem'

6. จากนั้นทำการเข้าไปแก้ไฟล์ระบุ เพื่อระบุตำแหน่ง cer ให้ถูกต้อง

sudo nano /etc/lighttpd/conf-enabled/10-ssl.conf

7. ทำการแก้ไขข้อความในไฟล์ดังนี้

ssl.pemfile = "/etc/lighttpd/snakeoil.pem"

8. จากนั้นให้สั่ง Restart Lighttpd  ตามปกติ

# sudo service lighttpd restart

9. ทำการทดสอบเข้าใช้งาน HTTPS โดยพิมพ์คำสั่ง https://หมายเลขไอพีของเซิร์ฟเวอร์ เช่น

https://192.168.99.1

**วิธีการเปลี่ยน Lighttpd Server Default Port**

\*ยกตัวอย่างการเปลี่ยนจาก port 80->8081
- เปิดไฟล์ /etc/lighttpd/lighttpd.conf

sudo nano /etc/lighttpd/lighttpd.conf

- แก้ไขข้อความในไฟล์ดังนี้ (วิธีการ save ให้กด Ctrl-X แล้วกด y แล้ว enter)

server.port = 8081

**วิธีการติดตั้ง PHP5 เพื่อใช้งานกับ Lighttpd Web Server**
1. ทำการติดตั้ง PHP5 ดังนี้

sudo apt-get install php5-cgi

2. ทำการเปิดใช้งาน module เพื่อให้สามารถใช้งาน php บน lighttpd ได้

sudo lighty-enable-mod fastcgi

sudo lighty-enable-mod fastcgi-php

3. จากนั้นให้สั่ง Restart Lighttpd  ตามปกติ

# sudo service lighttpd restart

4. วิธีการสร้างไฟล์ตรวจสอบสามารถทำได้ดังนี้

- เปิดไฟล์ /var/www/info.php

sudo sh -c 'echo "<?php echo phpinfo(); ?>" > /var/www/info.php'

- จากนั้นทำการเปิด web http://หมายเลขไอพีของเซิร์ฟเวอร์/info.php ตัวอย่าง

http://192.168.99.1/info.php

**Section 4: การติดตั้งโปรแกรมตรวจสอบประสิทธิภาพ Server : Apache Jmeter บนเครื่อง Windows**

**Goal:**

1. สามารถติดตั้ง JAVA for Windows

2. สามารถติดตั้ง Apache Jmeter บนเครื่อง Windows Client ได้

3. เรียนรู้วิธีใช้งาน Apache Jmeter

4. สามารถนำไปใช้ทดสอบ Web Server ที่ต้องทดสอบได้

**Configuration Guide:**
1. ทำการติดตั้ง JAVA for Windows

2. ดาวน์โหลดโปรแกรม Apache Jmeter for Windows

3. ทำการ Extract File เพื่อทำการติดตั้ง

4. เลือก Execute File ที่อยู่ภายใน Folder apache-jmeter-x.y->bin->ApacheJmeter.jar

5. เรียนรู้วิธีการเพิ่ม Thread Group และการตั้งค่าเบื้องต้น

|  |
| --- |
| **Number of Threads (users)** : 1**Ramp-up Period (in seconds)**: 1**Loop Count**: 1**Server Name of IP** : sis.psu.ac.th**Implrementatoin** : HttpClient4**Protocol** : http**Method** : GET**Path** : / |

6. ทำการเพิ่ม Results Tree และ Graph Results

7. ทดสอบยิงและทำความเข้าใจเกี่ยว Results Tree และ Graph Results

**Guide:**

http://sysadmin.psu.ac.th/2014/05/23/monitor-server-apache-jmeter-windows/

# การติดตั้งโปรแกรมตรวจสอบประสิทธิภาพ Server : Apache Jmeter บนเครื่อง Windows

 โปรแกรม Jmeter เป็นโปรแกรมสารพัดประโยชน์ใช้สำหรับทดสอบประสิทธิภาพการรองรับโหลดจำนวนมาก โดยในเบื้องต้นผมยังจะไม่อธิบายรายละเอียดโปรแกรมมากนะครับ จะนำเสนอแค่วิธีติดตั้ง และการ Test ในส่วนที่เป็น Web Server แบบไม่มี Script อะไรพิเศษเป็นหลัก แล้ววันหลังค่อยมาเปล่าในส่วนอื่น ๆ เช่นการ Test Load Database, LDAP, FTP, WebService ฯลฯ ถ้ามีโอกาสครับ

**ความต้องการพื้นฐานสำหรับโปรแกรม Jmeter**
1. Java 6 ขึ้นไป
2. OS ซึ่งรองรับทั้ง Unix, Windows

**วิธีการติดตั้งบน Windows**
Test OS : Windows 8.1
1. ติดตั้ง JAVA สามารถทำการ Download และติดตั้งได้จาก

http://www.java.com/en/download/index.jsp

2. ดาวน์โหลดโปรแกรม Jmeter ได้จาก

http://jmeter.apache.org/download\_jmeter.cgi

**3. Save file .zip ดังรูป**

****

**4. ทำการแตกไฟล์ออกมา และทำการรันไฟล์ ApacheMeter.jar ดังรูป (จะปรากฎเป็น icon java สามารถ double click รันได้เลย)**

****

**ตัวอย่างการสร้าง Test Plan เพื่อทดสอบ Web Server**

1. ทำการสร้าง Thread Group ภายใต้ Test Plan ดังรูป

2. หลังจากนั้นจะปรากฎหน้าต่างให้ตั้งค่าได้ดังนี้



**Number of Threads (users)** : จำนวนผู้ใช้งานที่ต้องการ ณ เวลานั้น ๆ (concurrent users)
**Ramp-up Period (in seconds)**: เวลาหน่วงในการเพิ่มผู้ใช้งานจนถึงจำนวน Number of Threads ที่ตั้งไว้ เช่น Number of Threads = 300, Ramp-up Period = 60 แสดงว่า ภายใน 60 วินาทีจะมี จำนวนผู้ใช้เข้าสู่ระบบรวม 300 คน (ค่อย ๆ เพิ่มเข้าไปจนครบ 300 ใน 60 วินาที)
**Loop Count**: คือจำนวนรอบที่ทำ ถ้าติก Forever ก็แสดงว่าทำตลอดไป

\*หมายเหตุ ระวังอย่างตั้งค่าเว่อร์จนเกินไป เครื่องผู้ที่สั่ง Test อาจจะทำงานไม่ไหว Hang ได้

3. จากนั้นทำการสร้าง Sampler ที่เป็น HTTP Request เพื่อกำหนดค่า config เบื้องต้นที่จำเป็นในการทดสอบ ดังรูป

****

4. จะปรากฎให้ตั้งค่า หลัก ๆ ถ้าต้องการยิงทดสอบธรรมดา ๆ ให้ใส่เฉพาะ Server Name หรือ IP และ Path เช่น /regist.php ถ้าต้องการยิงหน้าแรกก็ใส่ / เฉย ๆ ดังรูป

5. หลังจากนั้นทำการเพิ่มในส่วนของรายงานผล โดยแนะนำ 2 ตัวคือ View Results Tree และ Graph Results

6. ตัวอย่างผลลัพธ์จาก Results Tree (วิธีการรัน Test ให้กดที่ปุ่ม Play สีเขียว และปุ่ม Stop สีแดงเพื่อหยุด)



7. ตัวอย่างผลลัพธ์ในรูปแบบ Graph Results

**Reference :**
[1] http://www.narisa.com/forums/index.php?app=blog&blogid=9&showentry=3019
[2] http://www.narisa.com/forums/index.php?app=blog&blogid=9&showentry=2873

**Section 5: การทดสอบประสิทธิภาพ Web Server**

**ประเภท Static Page : Apache2 vs Lighttpd บนเครื่อง LBS**

**Goal:**

1. สามารถใช้งาน Apache Jmeter ในการทดสอบประสิทธิภาพ

2. เรียนรู้วิธีใช้งานโปรแกรม htop

**Configuration Guide:**
1. ทำการตั้งค่า Apache Port และ Lighttpd Port ดังนี้

|  |
| --- |
| Apache Port : 8080 |
| Lighttpd Port : 8081 |

2. เรียนรู้วิธีการเพิ่ม Thread Group, HTTP Request, View Results Tree, Graph Result

3. ติดตั้งโปรแกรม htop เพื่อเฝ้าดูการ Performance Server

4. ตั้งค่าการทดสอบและทดสอบดังตารางนี้

|  |  |  |  |
| --- | --- | --- | --- |
| Number of Threads (users) | **100** | **200** | **400** |
| **Apache2**  |  |  |  |
| **Lighttpd** |  |  |  |

**Guide:**
http://sysadmin.psu.ac.th/2014/05/23/performance-webserver-static-apache-lighttpd/

# การทดสอบประสิทธิภาพ Web Server ประเภท Static Page : Apache2 vs Lighttpd บนเครื่อง Ubuntu 14.04 LTS

 **เครื่องมือในการทดสอบ**
Web Server OS : Ubuntu 14.04 LTS
Client OS : Windows 8.1
Software : Apache Jmeter 2.11

**Environment :**
***Web Server***
Ubuntu Server (Oracle VM VirtualBox)
Intel Haswell 1.6GHz (2.30GHz) 4 core RAM 512G
**Client**
Windows 8.1 (Physical Notebook)
Intel Haswell 1.6GHz (2.30GHz) Intel Haswell 4 core RAM 8G
- รันอยู่บนเครื่องเดียวกัน- ใช้ค่า Default ไม่มีการ Tuning เพิ่มเติม

**วิธีการทดสอบ**

1. เปิดโปรแกรม Apache Jmeter ที่อยู่บน Desktop เครื่อง LBS ดังนี้

/home/testlab/Desktop/apache-jmeter-2.11/bin/jmeter

2. เลือก Execute

3. เปิดโปรแกรม Jmeter ทำการ Add Thread Group, HTTP Request, View Results Tree และ Graph Results


4. ทำการทดสอบโดยชี้ไปที่ Web Site ของ Web Server ที่ต้องการ โดยมีตัวอย่างดังรูป
(จะสังเกตุว่ามีการตั้งค่า Timeout = 1500 ms เพื่อป้องกันโปรแกรมค้าง เนื่องจาก Web Server ไม่ยอมตอบ)



**\*หมายเหตุ** ระหว่างรัน Test สามารถตรวจสอบการทำงานของเครื่อง Web Server โดยติดตั้งโปรแกรมชื่อ htop ดังนี้

sudo apt-get install -y htop

พิมคำสั่ง htop จะปรากฎหน้าจอดังรูป

****

**การตั้งค่าทดสอบ**

|  |  |
| --- | --- |
| **Number of Threads (users) :** | 100,200,400 |
| **Ramp-Up Period (in seconds) :** | 1 |
| **Loop Count :** | 1 |

**ตัวอย่างการทดสอบ**

จำนวน Users ที่เปิด Page ไม่สำเร็จ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Number of Threads (users)** | **500** | **1000** | **2000** | **2500** | **3000** |
| ***Apache2*** | 0 | 28 | 424 | 2232 | Jmeter Hang |
| ***Lighttpd*** | 0 | 0 | 3 | 1418 | Jmeter Hang |

 ซึ่งการทดสอบแบบไม่มีการ Tuning อาจจะไม่ Fair สำหรับแต่ละ Web Server (แม้ผลจะออกมาชัดว่า Lighttpd เหนือกว่า) เพราะยังไม่ได้มีการดึงประสิทธิภาพหรือตั้งค่า Limit Connection ให้รองรับได้สูงสุด และยังมีในส่วนของ Cache หลาย ๆ แบบที่จะช่วยให้ไม่ต้องประมวลผล Page เดิม ๆ ซ้ำบ่อย ๆ แต่ถึงกระนั้นวิธีนี้เป็นวิธีการหนึ่งที่ใช้ในการทดสอบก่อนจะมีการใช้งานจริง เพื่อทำให้เราเตรียมการได้ทันท่วงที เช่นทดสอบแล้วมีบ้าง SQL ที่ใช้เวลานานกว่าปกติ หรือ database connection limit

 สำหรับ Jmeter นั้นสามารถทำอะไรได้อีกมากมาย สามารถเขียน Script ในลักษณะของ Batch Job เพื่อดู Flow การทำงานของทั้ง Web Database และอื่น ๆ ถ้าใครอยากศึกษาเพิ่มเติมสามารถหาอ่านได้จาก Google ทั่ว ๆ ไปครับ

Reference :
[1] http://www.cyberciti.biz/tips/howto-performance-benchmarks-a-web-server.html

**Section 6: การตรวจสอบสถานะการให้บริการ Web Server ด้วย WGET และสร้าง Shell Script เพื่อตรวจสอบอัตโนมัติ**

**Goal:**

1. สามารถใช้งานโปรแกรม wget สำหรับโหลด html Code

2. สามารถสร้าง Script เพื่อใช้ตรวจสอบความผิดปกติของ Web Server ได้

3. เรียนรู้และเข้าใจ Shell Script ที่ใช้เรียก wget เพื่อตรวจสอบแบบอัตโนมัติ

**Configuration Guide:**
1. เปิดเครื่อง Webserver1 และ WebServer2

2. ทดสอบเรียก WebSite : http://webserver1.testlab, http://webserver2.testlab จากเครื่อง LBS

3. ทำการทดสอบ wget Website ในข้อ 2

4. ทดสอบการใช้งาน wget ในกรณีเรียก https

5. ทดสอบ grep ข้อมูลในไฟล์ html code ที่ได้จากข้อ 2

6. เขียน Shell Script เพื่อทำการตรวจสอบอัตโนมัติ

7. ทำการเปลี่ยน Permission ให้สามารถ execute ได้

8. ทดสอบรันโปรแกรม และตรวจสอบผลลัพธ์

**Guide:**
http://sysadmin.psu.ac.th/2014/06/19/monitor-web-server-wget-shell-script-daemon/

# การตรวจสอบสถานะการให้บริการ Web Server ด้วย WGET และสร้าง Shell Script เพื่อตรวจสอบอัตโนมัติ

 **การตรวจสอบสถานะการให้บริการด้วย WGET**

 **วิธีติดตั้ง wget สำหรับ ubuntu 14.04**

สามารถติดตั้งโดยใช้คำสั่ง (โดยปกติจะลงมาให้อยู่แล้ว)

# sudo apt-get install -y wget

**วิธีติดตั้ง wget สำหรับ windows**

สามารถ Download ได้ที่

http://downloads.sourceforge.net/project/gnuwin32/wget/1.11.4-1/wget-1.11.4-1-setup.exe

สามารถเปิด command prompt รันได้ที่ path นี้ (ยกตัวอย่างจาก windows 64 bit)

C:\Users\xxx>cd C:\Program Files (x86)\GnuWin32\bin
C:\Program Files (x86)\GnuWin32\bin>wget
SYSTEM\_WGETRC = c:/progra~1/wget/etc/wgetrc
syswgetrc = C:\Program Files (x86)\GnuWin32/etc/wgetrc
wget: missing URL
Usage: wget [OPTION]... [URL]...

Try `wget --help' for more options.

ตัวอย่างคำสั่งที่ใช้สำหรับโหลด html URL ที่ต้องการตรวจสอบ

wget -nv --timeout 2 --connect-timeout=10 -t 1 http://webserver1.testlab -O /tmp/webserver1-tmp

*-nv : ใช้สำหรับปิดการแสดงผล แต่ยังแสดงผล error และข้อมูลเบื้องต้น*

*--timeout [sec] : ใช้สำหรับกำหนดค่า timeout ในการเชื่อมต่อไปยัง web server*

*--connect-timeout [sec] : ใช้กำหนดเวลาในกรณีที่การเชื่อมต่อไม่สิ้นสุดในเวลาที่*

 *กำหนด*

*-t [sec] : จำนวนครั้งในการ retry connection*

*-O : บันทึกผลลัพธ์ไปยังไฟล์*

ในกรณีที่ต้องการโหลด html URL ประเภท https ให้ทำการข้ามการตรวจสอบ certificate ด้วยตัวอย่างคำสั่งดังนี้

wget -nv --timeout 2 --connect-timeout=10 -t 1 **--no-check-certificate** https://webserver1.testlab -O /tmp/webserver1-https-tmp

หลังจากได้ไฟล์ผลลัพธ์ของหน้า html website เป้าหมายแล้ว สามารถใช้โปรแกรม grep เพื่อตรวจสอบว่าปรากฎข้อความที่แสดงว่าหน้า web ทำงานปกติหรือไม่ เช่น

grep WEBSERVER /tmp/webserver1-tmp

หลังจากนั้นเราสามารถประยุกต์เพื่อเขียน Shell Script ง่าย ๆ ที่สามารถตรวจสอบ Web Server เพื่อดูว่า Web ปกติหรือไม่ในขั้นตอนถัดไป

**วิธีการเขียน Shell Script เพื่อตรวจสอบการใช้งานแบบอัตโนมัติ**

**Shell Script สำหรับตรวจสอบ Web Server**

- ทำการสร้างไฟล์ /home/[user]/checkweb.sh

nano /home/testlab/checkweb.sh

- สร้าง script ข้อความดังนี้

**\*ตัวอย่างรูปแบบ sh shell**

#!/bin/sh
url1='webserver1.testlab'
url2='webserver2.testlab'
searchtxt='WEBSERVER'
#URL PATH
path='/'
while(true);
do
 cp /dev/null /tmp/server-status
 for i in $(seq 1 2)
 do
 eval wget -q --timeout 2 --connect-timeout=10 -t 1 http://\${url${i}}$path -O /tmp/\${url${i}}-tmp
 eval echo "Get URL : http://\${url${i}}$path"
 status=$(eval grep $searchtxt /tmp/\${url${i}}-tmp | wc -l)

 if [ $status -ge "1" ]; then
 eval echo \${url${i}}" : STATUS = UP"
 eval echo \${url${i}}" UP" >> /tmp/server-status
 else
 eval echo \${url${i}}" : STATUS = DOWN"
 eval echo \${url${i}}" DOWN" >> /tmp/server-status
 fi
 done
 echo "Sleep 15 secs....."
 sleep 15
 /usr/bin/clear
done

**\*ตัวอย่างรูปแบบ bash shell**

#!/bin/bash
url[1]='webserver1.testlab'
url[2]='webserver2.testlab'
searchtxt='WEBSERVER'
#URL PATH
path='/'
while(true);
do
 cp /dev/null /tmp/server-status
 for i in {1..2}

 do
 wget -q --timeout 2 --connect-timeout=10 -t 1 http://${url[$i]}$path -O /tmp/${url[$i]}-tmp
 echo "Get URL : http://${url[$i]}$path"
 status=$(grep $searchtxt /tmp/${url[$i]}-tmp | wc -l)
 if [ $status -ge "1" ]; then
 echo ${url[$i]}" : STATUS = UP"
 echo ${url[$i]}" UP" >> /tmp/server-status
 else
 echo ${url[$i]}" : STATUS = DOWN"
 echo ${url[$i]}" DOWN" >> /tmp/server-status
 fi
 done
 echo "Sleep 15 secs....."
 sleep 15
 /usr/bin/clear
done

- ทำการเปลี่ยน permission ให้สามารถ execute ได้

chmod 755 checkweb.sh

- ทดสอบ execute script จะได้ข้อความดังนี้

testlab@lbs:~$ ./checkweb.sh
Get URL : http://webserver1.testlab/
webserver1.testlab : STATUS = UP
Get URL : http://webserver2.testlab/
webserver2.testlab : STATUS = UP
Sleep 15 secs.....

สามารถนำ script ดังกล่าวไปปรับเปลี่ยนเพื่อใช้ในการรันเป็น daemon ได้ตามสะดวกครับ

**Section 7: การสร้างระบบ Load Balance Web Server ด้วยวิธีการ URL Redirect**

**Goal:**

1. สามารถสร้าง Code ที่ทำ Load Balance ด้วยภาษา PHP ได้

2. เข้าใจถึงกระบวนการทำงานของระบบ Load Balance แบบ URL Redirect

**Configuration Guide:**
1. เรียนรู้การเขียน PHP Code ง่าย ๆ เพื่อ ดึงไฟล์, Random, Redirect

2. เรียนรู้การเขียน PHP Code เพื่อสร้างหน้ารอในกรณีไม่มีเครื่องใช้งาน

3. เขียน PHP Code ที่ทำงานได้ตามข้อ 1 และ ข้อ 2 ตั้งชื่อ index.php

4. ทดสอบใช้งานจริง โดยนำ Code ไปวางใน /var/www/html บนเครื่อง LBS

5. ทดลองเรียก URL : http://lbs.psu.ac.th

**Guide:**
http://sysadmin.psu.ac.th/2014/06/19/load-balance-web-server-url-redirect/

# การสร้างระบบ Load Balance Web Server ด้วยวิธีการ URL Redirect

 เราสามารถสร้างระบบ Load Balance Web Server ด้วยวิธีการ URL Redirect
โดยสามารถสร้างด้วยภาษาใดก็ได้ แต่จะขอยกตัวอย่างด้วยภาษา PHP ได้ง่าย ๆ ดังนี้

**ตัวอย่างไฟล์ server-status ที่ได้ทำการ monitor web server มาเรียบร้อยแล้ว**

webserver1.testlab UP
webserver2.testlab DOWN

- เราสามารถใช้คำสั่งดึงไฟล์เพื่อคัดเฉพาะ Web Server ที่สามารถใช้งานได้จริง ๆ ดังนี้

exec("cat /tmp/server-status | grep UP",$redirect);

- จากนั้นจะทำการ นำรายการ server ทั้งหมดซึ่งอยู่ในตัวแปร $redirect มาทำการสุ่ม ดังนี้

$random = rand(0,(count($redirect)-1);

- ก็จะทำเอาเลข server ที่ random ได้ไปเข้ากระบวนการ Redirect ดังนี้

$server = explode(" ",$redirect[$random]);
header('HTTP/1.1 301 Moved Permanently');
header('Cache-Control: no-store, no-cache, must-revalidate');
header('Expires: Thu, 01 Jan 1970 00:00:00 GMT');
header('Location: https://'.$server[0].'/');
unset($server);

ในกรณีที่ต้องการสร้างหน้ารอให้ทำการตรวจสอบว่าในกรณีที่ไม่มีเครื่องให้บริการให้แสดงข้อความ ตัวอย่างดังนี้

if(count($redirect)==0){
 echo "Server Unavailable or Maintenance Period";
}

ดังที่กล่าวมาทั้งหมดข้างต้นสามารถรวมเป็น script ไฟล์ PHP ได้ดังนี้

<?php
exec("cat /tmp/server-status | grep UP",$redirect);
if(count($redirect)==0){
 echo "Server Unavailable or Maintenance Period";
}else{
 $random = rand(0,(count($redirect)-1));
 $server = explode(" ",$redirect[$random]);
 header('HTTP/1.1 301 Moved Permanently');
 header('Cache-Control: no-store, no-cache, must-revalidate');
 header('Expires: Thu, 01 Jan 1970 00:00:00 GMT');
 header('Location: http://'.$server[0].'/');
 unset($server);
}
unset($redirect);
?>