

## บทที่ 4

### การพัฒนาส่วนประกอบสำหรับเอ็นเอสทูเพื่อจำลองการทำงานของเว็บแคช

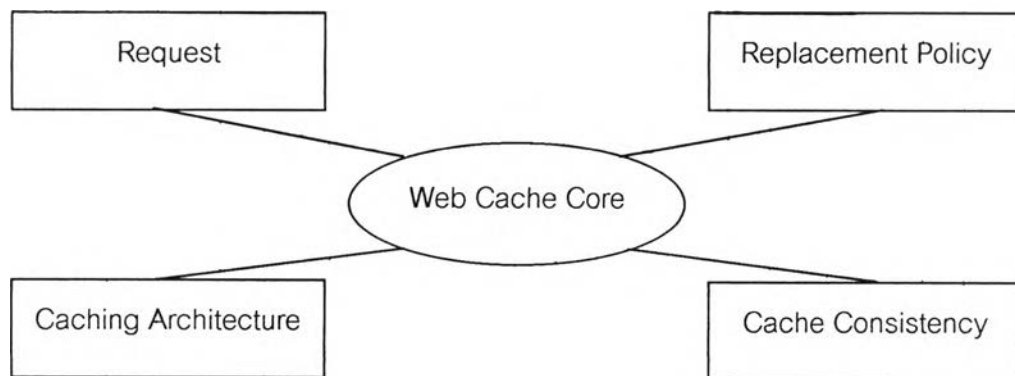
ในบทนี้จะนำเสนอแนวคิดที่ใช้ในการพัฒนาส่วนประกอบต่างๆ สำหรับการจำลองการทำงานของเว็บแคช และอธิบายถึงรายละเอียดของส่วนประกอบนั้นๆ พร้อมทั้งตัวอย่างวิธีการเรียกใช้งานส่วนประกอบต่างๆ เพื่อจำลองการทำงานของเว็บแคช รวมถึงแนวทางการเพิ่มเติมส่วนประกอบในอนาคตตั่งรายละเอียดต่อไปนี้

#### 4.1 แนวคิดในการพัฒนา

จากการที่เอ็นเอสทูถูกพัฒนาขึ้นมาจากภาษาซีพลัสพลัส และภาษาโอทีซีแอล ทำให้เอ็นเอสทูมีลักษณะเป็นส่วนประกอบต่างๆ ซึ่งแต่ละส่วนประกอบจะประกอบด้วยคลาส และเมทอดที่แตกต่างกันตามการทำงานของส่วนประกอบนั้นๆ สำหรับส่วนจำลองการทำงานของเว็บแคชที่มีอยู่ในเอ็นเอสทูเป็นการจำลองการทำงานของเว็บแคชในระดับโปรแกรมประยุกต์ มีคลาส Http (ในภาษาโอทีซีแอล) หรือคลาส HttpApp (ในภาษาซีพลัสพลัส) เป็นคลาสพื้นฐานสำหรับโปรโตคอล Http ซึ่งใช้ในไคลแอนท์ แคช และเซิร์ฟเวอร์ โดยเน้นการจำลองการทำงานที่ความต้องกันของเว็บแคชเท่านั้น ไม่มีขั้นตอนวิธีการแทนที่ และไม่สามารถจำลองการทำงานสถาปัตยกรรมของการแคชแบบพี่น้องได้ รวมทั้งส่วนประกอบเดิมสำหรับจำลองการทำงานของเว็บแคชจะมีลักษณะเป็นกลุ่มใหญ่ การเพิ่มเติมส่วนประกอบเป็นเรื่องยาก ดังนั้นงานวิทยานิพนธ์นี้จะมุ่งเน้นการพัฒนาส่วนประกอบต่างๆ ของการจำลองเว็บแคชเพิ่มเติมโดยเน้นที่ความยืดหยุ่นในการขยายระบบ โดยการแบ่งส่วนจำลองการทำงานของเว็บแคชออกเป็นส่วนประกอบหลัก 4 ส่วน คือ ส่วนจัดการการร้องขอ ส่วนจัดการความต้องกันของเว็บแคช ส่วนจัดการขั้นตอนวิธีการแทนที่ และส่วนจัดการสถาปัตยกรรมของการแคช โดยทำงานร่วมกับส่วนแกนการทำงานของเว็บแคชเดิมของเอ็นเอสทู ดังรูปที่ 4.1 ซึ่งแต่ละส่วนจะประกอบด้วยคลาส และเมทอดที่สร้างขึ้นมาเพื่อทำงานตามลักษณะการทำงานของส่วนประกอบนั้นๆ โดยมีการนำคลาส และเมทอดเดิมบางส่วนมาใช้ และทำการแก้ไขเพื่อให้สามารถทำงานร่วมกับส่วนประกอบที่ได้ทำการพัฒนาเพิ่มเติม และเพื่อให้ง่ายแก่การเพิ่มเติมส่วนประกอบสำหรับจำลองการทำงานของเว็บแคชในอนาคต โดยคลาสต่างๆ ที่เกี่ยวข้อง กับส่วนประกอบหลักมีสถานะดังตารางที่ 4.1

การจำลองการทำงานสามารถทำการเรียกใช้ส่วนประกอบต่างๆ เพื่อทำงานร่วมกัน ซึ่งไม่จำเป็นต้องเรียกใช้ส่วนประกอบทั้งหมด ทั้งนี้ขึ้นอยู่กับรูปแบบของการจำลองการทำงานของเว็บ

แคช และการพัฒนาส่วนประกอบเพิ่มเติมในอนาคตก็สามารถทำได้โดยทำเฉพาะในส่วนที่ต้องการเพิ่มเติมอย่างเช่น ขั้นตอนวิธีการแทนที่แบบใหม่ ซึ่งผู้ใช้งานยังคงสามารถเรียกใช้ส่วนประกอบเดิมได้ และยังสามารถเชื่อมต่อกับส่วนประกอบใหม่เพื่อทำงานร่วมกันได้เช่นกัน ดังนั้น การจำลองการทำงานของเว็บแคชเพื่อทดสอบประสิทธิภาพของวิธีการต่างๆ สามารถทำได้ง่าย และรวดเร็ว ซึ่งรายละเอียดของส่วนประกอบต่างๆ และแนวทางการเพิ่มเติมนส่วนประกอบในอนาคตจะกล่าวในหัวข้อถัดไป



รูปที่ 4.1 แสดงส่วนประกอบหลักสำหรับจำลองการทำงานของเว็บแคช

ตารางที่ 4.1 แสดงสถานะของคลาสที่ใช้ในส่วนประกอบสำหรับจำลองการทำงานของเว็บแคช

คลาส	แก้ไขของเดิม	พัฒนาใหม่
PagePool/Math	✓	
PagePool/Client	✓	
PagePool/ProxyTrace	✓	
Http/Cache	✓	
Http/Cache/TTL	✓	
Http/Cache/TTL/Plain	✓	
Http/Cache/FIFO		✓
Http/Cache/LRU		✓
Http/Cache/SIZE		✓
Http/Cache/ICP		✓

## 4.2 ส่วนประกอบสำหรับจำลองการทำงานของเว็บแคช

จากแนวคิดในการพัฒนาที่เน้นความยืดหยุ่นในการขยายระบบ ทำให้สามารถแบ่งส่วนประกอบสำหรับจำลองการทำงานของเว็บแคชได้เป็น 4 ส่วน ดังที่ได้กล่าวมาแล้ว โดยทำงานร่วมกับส่วนแกนการทำงานของเว็บแคชเดิมของเอ็นเอสทู ซึ่งแต่ละส่วนมีรายละเอียดดังต่อไปนี้

### 4.2.1 ส่วนแกนการทำงานของเว็บแคช

ส่วนแกนการทำงานของเว็บแคชทำหน้าที่รับการร้องขอจากไคลแอนท์ โดยข้อมูลการร้องขอได้มาจากส่วนจัดการการร้องขอ และทำการตรวจสอบว่าเอกสารที่ถูกร้องขอเก็บอยู่ภายในแคชหรือไม่ ถ้าเอกสารถูกพบอยู่ภายในแคช แคชจะตรวจสอบเอกสารนั้นว่าหมดอายุแล้วหรือยัง ซึ่งวิธีการตรวจสอบที่ใช้คือ ความต้องกันของเว็บแคชที่ได้กำหนดไว้ในส่วนจัดการความต้องกันของเว็บแคช ถ้าเอกสารที่เก็บอยู่ไม่หมดอายุ แคชจะทำการส่งเอกสารที่พบให้กับไคลแอนท์ แต่ถ้าเอกสารที่เก็บอยู่หมดอายุแล้ว แคชจะทำการส่งข้อความการร้องขอแบบไอเอ็มเอสไปยังเซิร์ฟเวอร์ปลายทางที่เก็บเอกสารนั้น เพื่อตรวจสอบความถูกต้องของเอกสาร ถ้าเอกสารที่อยู่ในแคชไม่ตรงกับเอกสารที่เก็บอยู่บนเซิร์ฟเวอร์ปลายทาง เซิร์ฟเวอร์ปลายทางจะทำการส่งเอกสารใหม่กลับมายังแคช และแคชจะเก็บเอกสารใหม่แทนที่เอกสารเดิม แล้วจัดส่งเอกสารให้กับไคลแอนท์ต่อไป ในกรณีที่แคชไม่พบเอกสารที่ถูกร้องขอ แคชจะทำการส่งต่อการร้องขอไปยังแคชตัวอื่นตามสถาปัตยกรรมของการแคชที่ได้กำหนดไว้ในส่วนจัดการสถาปัตยกรรมของการแคช และจะส่งต่อการร้องขอไปยังเซิร์ฟเวอร์ถ้าไม่พบเอกสารภายในแคชตัวใดเลย เมื่อแคชได้รับเอกสารที่ถูกร้องขอจากเซิร์ฟเวอร์ หรือจากแคชตัวอื่นแล้ว แคชจะทำการเก็บเอกสารที่ได้รับไว้ภายในแคช ถ้าแคชไม่มีที่ว่างพอในการเก็บเอกสาร แคชจะทำการลบเอกสารเดิมที่เก็บอยู่ภายในแคช โดยใช้ขั้นตอนวิธีการแทนที่ที่ได้กำหนดไว้ในส่วนจัดการขั้นตอนวิธีการแทนที่ ซึ่งคลาสของส่วนแกนการทำงานของเว็บแคชคือ `Http/Cache` โดยจะมีเมทอดหลัก เช่น `get-request`, `cache-hit` และ `cache-miss` สำหรับทำหน้าที่ต่างๆ ดังที่ได้กล่าวไป

### 4.2.2 ส่วนจัดการการร้องขอ

ส่วนจัดการการร้องขอทำหน้าที่อ่านข้อมูลการใช้เว็บจากล็อกไฟล์ หรือทำการสร้างข้อมูลการใช้เว็บขึ้นมาตามพารามิเตอร์ต่างๆ ที่ผู้ใช้งานกำหนด โดยส่วนนี้ถูกใช้โดยเซิร์ฟเวอร์เพื่อสร้างข้อมูลต่างๆ ของเอกสาร เช่น ชื่อ ขนาด เวลาที่เกิดการเปลี่ยนแปลง อายุของเอกสาร เป็นต้น ถูกใช้โดยแคชสำหรับการตรวจสอบว่ามีเอกสารอะไรบ้างที่เก็บอยู่ภายในแคช และถูกใช้โดยไคลแอนท์สำหรับสร้างกระแสข้อมูลการร้องขอ

สำหรับงานวิทยานิพนธ์นี้ มีคลาสที่เกี่ยวข้องในส่วนนี้อยู่ 3 คลาส คือ

1) PagePool/Math ใช้สำหรับสร้างเอกสารขึ้นมาตามพารามิเตอร์ที่กำหนด โดยมีโอทีซีแอลเมทอดหลัก คือ

- *gen-pageid* เมทอดจะส่งเลขประจำเอกสารซึ่งจะถูกเรียกขอในครั้งถัดไปกลับมา ซึ่งสร้างจากตัวแปรสุ่มที่ได้กำหนดไว้
- *gen-size* เมทอดจะส่งค่าขนาดของเอกสารกลับมา ซึ่งสร้างจากตัวแปรสุ่มที่ได้กำหนดไว้
- *gen-modtime {pageid} {mt}* เมทอดจะส่งค่าเวลาที่เอกสารเปลี่ยนแปลงในครั้งถัดไปกลับมา โดย *{mt}* จะให้ค่าการเปลี่ยนแปลงครั้งสุดท้ายของเอกสาร โดยค่าเวลาที่เอกสารเปลี่ยนแปลงในครั้งถัดไปจะสร้างจากตัวแปรสุ่มในการกำหนดอายุ
- *ranvar-age {rv}* ใช้กำหนดตัวแปรสุ่มสำหรับอายุของเอกสาร โดย *{rv}* จะให้ค่าที่ได้จากการสุ่มตามแบบจำลองทางคณิตศาสตร์ที่กำหนด เช่น *RandomVariable/Exponential*, *RandomVariable/Constant* เป็นต้น
- *ranvar-size {rv}* ใช้กำหนดตัวแปรสุ่มสำหรับขนาดของเอกสารจาก *{rv}*
- *ranvar-pid {rv}* ใช้กำหนดตัวแปรสุ่มสำหรับเลขประจำเอกสารจาก *{rv}*

ตัวอย่างในการเรียกใช้งาน PagePool/Math

```
# Use PagePool/Math
set pgp [new PagePool/Math]
set tmp [new RandomVariable/Exponential]
$tmp set avg_ 4096
$pgp ranvar-size $tmp
set tmp [new RandomVariable/Exponential]
$tmp set avg_ 14400
$pgp ranvar-age $tmp
set tmp [new RandomVariable/Exponential]
$tmp set avg_ 300
$pgp ranvar-pid $tmp
```

2) PagePool/Client คลาสนี้ช่วยให้แคชสามารถติดตามเอกสารต่างๆ ภายในแคช และใช้เก็บค่าต่างๆ ของเอกสารที่เกี่ยวข้องกับแคช โดยโอทีซีแอลเมทอดหลักที่ใช้

ในคลาสนี้จะเป็นเม็ท็อดที่ใ้ซ็อยู่ใ้ภายใน ไม่ใ้ได้นำมาใ้ในสคริปต์ใ้สำหรับจำลองการใ้ทำงานของเว็บแคช ซึ่งเม็ท็อดเหล่านี้ประกอบด้วย

- `get-page {pageid}` เม็ท็อดจะส่งค่าต่างๆ ของเอกสารกลับมา เช่น ขนาดอายุของเอกสาร เป็นต้น
- `enter-page {pageid} {args}` ทำการเก็บเอกสารลงในแคช พร้อมกับค่าต่างๆ ของเอกสารตามที่ได้กำหนดในอาร์กิวเมนต์
- `remove-page {pageid}` ทำการลบเอกสารออกจากแคชในกรณีทีแคชไม่มีที่ว่างพอสำหรับเก็บเอกสารที่ใ้เข้ามาใหม่

3) PagePool/ProxyTrace ใช้สำหรับการจำลองการทำงานของเว็บแคช โดยใ้ใ้ข้อมูลการใ้เว็บจริงจากพร็อกซีแคช เนื่องจากข้อมูลการใ้เว็บเหล่านี้มีรูปแบบที่แตกต่างกันหลายรูปแบบดังนั้นจึงต้องทำการแปลงใ้เป็นรูปแบบทีคลาสนี้สามารถอ่านได้ สำหรับตัวแปลงนี้สามารถแปลงรูปแบบของข้อมูลของพร็อกซีแคชได้ 4 รูปแบบ คือ DEC (1996) [12], UCB Home-IP, NLANR (SQUID) [17] และ EPA เมื่อแปลงข้อมูลเสร็จแล้ว จะใ้ได้ผลลัพธ์ออกมา 2 ไฟล์หลัก คือ `reqlog` และ `pglog` โดยมี 4 ไฟล์เสริม คือ `clog`, `slog`, `u1log` และ `u1log` สำหรับรูปแบบ NLANR และ DEC ใ้ใ้ในการตรวจสอบย้อนกลับ รูปแบบของทั้ง 6 ไฟล์มีลักษณะดังนี้

- `pglog` : [`<serverID>` `<URL_ID>` `<PageSize>` `<AccessCount>`] ดังตัวอย่างต่อไปนี้

```
82 0 431 1515
```

```
...
```

```
4173 253776 4551 1
```

- `reqlog` : [`<time>` `<clientID>` `<serverID>` `<URL_ID>`] และใ้ใ้บรทัดสุดท้ายของไฟล์จะมีรูปแบบคือ `i <Duration>` `<Number_of_URL>` ดังตัวอย่างต่อไปนี้

```
0.000000 806 2706 30282
```

```
...
```

```
i 29576.149000 253777
```

- `clog` : [`<client>` `<clientID>`] ดังตัวอย่างต่อไปนี้

```
3246508193 1
```

```
...
```

```
3093547169 1998
```

- slog : [<server> <serverID>] ดังตัวอย่างต่อไปนี้  
     www.music.th.com 1  
     www.garfield.com 2  
     ...  
     www.windowscentral.com 10388
- ulog : [<URL> <PageSize> -> <old\_URL\_ID>] ดังตัวอย่างต่อไปนี้  
     http://www.music.th.com/images/fastlane.gif 851 -> 1  
     http://www.garfield.com/images/promo\_catalog.swf 16865 -> 2  
     ...  
     http://www1.bluemountain.com/enggif/sps\_love/k57/k57.gif  
     25865 -> 253777
- u1log : [<old\_URL\_ID> <new\_URL\_ID> <PageSize>] ดังตัวอย่างต่อไปนี้  
     141 0 431  
     1020 1 525  
     ...  
     82559 253776 4551

โดย Pagepool/ProxyTrace จะใช้ไฟล์ reglog และ pglog เป็นข้อมูลนำเข้าของการจำลองการทำงานของเว็บแคช เนื่องจากข้อมูลการใช้งานเว็บโดยส่วนมากจะไม่มีข้อมูลการเปลี่ยนแปลงของเอกสาร ดังนั้นในคลาสนี้จะใช้แบบจำลองการเปลี่ยนแปลงของเอกสารแบบไบโมเดิล (Bimodal) ซึ่งผู้ใช้สามารถเลือกจำนวนเอกสารเป็นเปอร์เซ็นต์ของเอกสารทั้งหมดให้ใช้ค่าการเปลี่ยนแปลงของเอกสารแบบสุ่มจากตัวสร้างหนึ่ง และเอกสารที่เหลือใช้ค่าการเปลี่ยนแปลงของเอกสารแบบสุ่มจากอีกตัวสร้างหนึ่ง จากการใช้แบบจำลองนี้ทำให้สามารถกำหนดเปอร์เซ็นต์ของเอกสารให้เป็นเอกสารที่เปลี่ยนแปลงบ่อยครั้ง (Dynamic) และเอกสารที่เหลือให้เป็นเอกสารที่ไม่ค่อยมีการเปลี่ยนแปลง (Static) ได้

โพรพเพอร์ตี้แอตทริบิวต์หลักที่ใช้ในคลาสนี้ คือ

- *get-poolsize* เมท็อดจะส่งค่าจำนวนเอกสารทั้งหมดกลับมา
- *get-duration* เมท็อดจะส่งค่าเวลาความยาวนานของข้อมูลการใช้เว็บจาก reqlog กลับมา โดยดึงค่าจากบรรทัดสุดท้ายของ reqlog
- *bimodal-ratio* เมท็อดจะส่งค่าอัตราส่วนไบโมเดิลกลับมา

- `set-client-num {num}` ใช้กำหนดจำนวนผู้เรียกขอในการจำลองการทำงานของเว็บแคช
- `gen-request {ClientID}` ใช้สร้างการร้องขอครั้งถัดไปสำหรับผู้เรียกขอที่กำหนด
- `gen-size {PageID}` เมทีอดจะส่งค่าขนาดของเอกสารที่กำหนดกลับมา
- `bimodal-ratio {ratio}` ใช้กำหนดค่าของเอกสารที่มีการเปลี่ยนแปลงบ่อยในรูปแบบ  $\{ratio\} * 10$  เปอร์เซนต์
- `ranvar-dp {ranvar}` ใช้กำหนดช่วงเวลาการเปลี่ยนแปลงของเอกสารสำหรับเอกสารที่มีการเปลี่ยนแปลงบ่อย
- `set-reqfile {file}` ใช้กำหนดไฟล์ที่ใช้สำหรับการร้องขอ
- `set-pgfile {file}` ใช้กำหนดไฟล์ที่ใช้เป็นข้อมูลของเอกสาร
- `gen-modtime {PageID} {LastModTime}` ใช้สร้างเวลาในการเปลี่ยนแปลงเอกสารครั้งถัดไปสำหรับเอกสารที่กำหนด

ตัวอย่างในการเรียกใช้งาน PagePool/ProxyTrace

```
# Use PagePool/ProxyTrace
set pgp [new PagePool/ProxyTrace]
$pgp set-reqfile "reqlog"
$pgp set-pagefile "pglog"
$pgp set-client-num 1
$pgp bimodal-ratio 0.1
set tmp [new RandomVariable/Constant]
$tmp set val_ 100
$pgp ranvar-dp $tmp
set tmp [new RandomVariable/Constant]
$tmp set val_ 5000
$pgp ranvar-sp $tmp
```

#### 4.2.3 ส่วนจัดการความต้อกันของเว็บแคช

ส่วนจัดการความต้อกันของเว็บแคชทำหน้าที่ตรวจสอบความต้อกันของเอกสารที่เก็บอยู่ภายในแคช โดยทำการเปรียบเทียบระยะเวลาที่เก็บเอกสาร หรืออายุของเอกสารภายในแคชกับค่าขีดแบ่งที่กำหนดตามวิธีการทำงานของความต้อกันของเว็บแคชนั้นๆ ถ้าเอกสารที่เก็บภายในแคชไม่ต้อกันแคชจะทำการร้องขอเอกสารจากเซิร์ฟเวอร์อีกครั้งหนึ่ง

สำหรับส่วนจัดการความต้องกันของเว็บแคชใช้วิธีการที่มีอยู่แล้วในเอ็นเอสทู ทำการดัดแปลงโค้ดให้สามารถทำงานร่วมกับส่วนประกอบอื่นๆ ได้ โดยความต้องกันของเว็บแคชที่ใช้มี 2 แบบ คือ

- 1) Plain Old TTL (Fixed TTL) [13] วิธีการนี้ใช้ค่าขีดแบ่งที่กำหนดแน่นอน โดยค่าขีดแบ่งที่กำหนดคือ ระยะเวลาที่ใช้ตรวจสอบความถูกต้องของเอกสารที่เก็บอยู่ภายในแคช คือ เอกสารที่เก็บอยู่จะไม่ถูกต้องเมื่อระยะเวลาที่เก็บเอกสารเข้าไปในแคชถึงเวลาปัจจุบัน มีค่ามากกว่าค่าขีดแบ่งที่กำหนด แคชจะต้องทำการตรวจสอบกับเซิร์ฟเวอร์ว่าเอกสารมีการเปลี่ยนแปลงหรือไม่ โดยส่งไอเอ็มเอส (IMS) ไปถาม ถ้าเอกสารในเซิร์ฟเวอร์มีการเปลี่ยนแปลง แคชจะทำการเก็บเอกสารที่เปลี่ยนแปลงลงในแคช และส่งต่อให้กับไคลแอนท์ ในกรณีที่เอกสารในเซิร์ฟเวอร์ไม่มีการเปลี่ยนแปลงแคชจะส่งเอกสารที่อยู่ภายในแคชให้กับไคลแอนท์

ไอทีซีแอลเมท็อดที่ใช้ในการตรวจสอบความต้องกัน คือ

```
Http/Cache/TTL/Plain instproc is-expired { pageid } {
    $self instvar ns_ thresh_
    set cktime [expr [$ns_ now] - [$self get-cachetime $pageid]]
    if {$cktime < $thresh_} {
        return 0 ;# Not Expire
    }
    return 1
}

Http/Cache/TTL instproc is-consistent { cl type pageid } {
    return ![$self is-expired $pageid]
}
```

- 2) Adaptive TTL [13] วิธีการนี้อายุของเอกสารถูกกำหนดเป็นเปอร์เซ็นต์ของอายุเอกสารปัจจุบัน คือ เปอร์เซ็นต์ x (เวลาปัจจุบัน – เวลาที่ครั้งล่าสุดที่มีการเปลี่ยนแปลงของเอกสาร) ซึ่งค่าขีดแบ่งคือ จำนวนเปอร์เซ็นต์ที่กำหนด โดยเอกสารจะไม่ถูกต้องเมื่อระยะเวลาที่เก็บเอกสารเข้าไปในแคชถึงเวลาปัจจุบัน มีค่ามากกว่าเปอร์เซ็นต์ของอายุเอกสารปัจจุบัน

ไอทีซีแอลเมท็อดที่ใช้ในการตรวจสอบความต้องกัน คือ

```
Http/Cache/TTL instproc is-expired { pageid } {
    $self instvar thresh_ ns_
```



```

set cktime [expr [$ns_ now] - [$self get-cachetime $pageid]]
set age [expr (($ns_ now) - [$self get-modtime $pageid]) * \
$thresh_]
if {$cktime <= $age} {
    return 0 ;# Not expired
}
return 1
}

Http/Cache/TTL instproc is-consistent { cl type pageid } {
    return ![$self is-expired $pageid]
}

```

ตัวอย่างในการเรียกใช้ส่วนจัดการความต้อกันของเว็บแคช

- `set cache [new Http/Cache/TTL $ns $node] ;# Adaptive TTL`  
`$cache set-thresh 0.1`
- `set cache [new Http/Cache/TTL/Plain $ns $node] ;# Plain Old TTL`  
`$cache set-thresh 100`

#### 4.2.4 ส่วนจัดการขั้นตอนวิธีการแทนที่

ส่วนจัดการขั้นตอนวิธีการแทนที่ [11] ทำหน้าที่พิจารณาเลือกแทนที่เอกสารภายในแคชเมื่อมีเอกสารใหม่เข้ามาในขณะที่แคชมีที่ว่างไม่เพียงพอ

สำหรับขั้นตอนวิธีการแทนที่ของเว็บแคชที่ใช้ในงานวิทยานิพนธ์นี้มี 3 แบบ คือ

- 1) FIFO โดยแคชจะแทนที่เอกสารที่ถูกเก็บภายในแคชตามลำดับก่อนหลัง สำหรับคลาสที่ใช้สำหรับขั้นตอนวิธีการแทนที่นี้ คือ `Http/Cache/FIFO`
- 2) LRU โดยแคชจะแทนที่เอกสารที่ไม่ถูกเรียกใช้มานานที่สุดก่อน สำหรับคลาสที่ใช้สำหรับขั้นตอนวิธีการแทนที่นี้ คือ `Http/Cache/LRU`
- 3) SIZE โดยแคชจะแทนที่เอกสารที่มีขนาดใหญ่ที่สุดก่อน สำหรับคลาสที่ใช้สำหรับขั้นตอนวิธีการแทนที่นี้ คือ `Http/Cache/SIZE`

โอทีซีแอลเมท็อดหลักที่ใช้สำหรับส่วนจัดการขั้นตอนวิธีการแทนที่ทั้ง 3 แบบคือ

- 1) `move-hit {pageid}` ใช้ใน LRU สำหรับเปลี่ยนแปลงตำแหน่งของเอกสารที่ถูกพบภายในแคช และเตรียมไว้สำหรับขั้นตอนวิธีการแทนที่แบบอื่นที่สามารถเพิ่มเติมเข้ามาในเอ็นเอสทูในอนาคต

- 2) *set-pos {pageid pagesize}* มีในทุกขั้นตอนวิธีการแทนที่ แตกต่างกันตามขั้นตอนวิธีการแทนที่ที่ใช้ โดยจะทำการเก็บเลขประจำเอกสารและขนาดของเอกสารไว้ในแถวลำดับ (Array) เพื่อใช้สำหรับการแทนที่เอกสารตามตำแหน่งที่เก็บอยู่ในแถวลำดับ
- 3) *cache-repl {cache curr\_size}* มีในทุกขั้นตอนวิธีการแทนที่ แตกต่างกันตามขั้นตอนวิธีการแทนที่ที่ใช้ โดยจะทำการลบเอกสารออกจากแคชตามขั้นตอนวิธีการแทนที่ที่ใช้จนกว่าแคชจะมีที่ว่างพอสำหรับเอกสารใหม่ที่จะทำการเก็บลงในแคช
- 4) *attach-repl {algorithm}* เป็นเม็ท็อดที่ใช้กำหนดขั้นตอนวิธีการแทนที่ภายในแคช ตัวอย่างในการเรียกใช้ส่วนจัดการขั้นตอนวิธีการแทนที่
  - *set fifo [new Http/Cache/FIFO]*  
*\$cache attach-repl \$fifo*
  - *set lru [new Http/Cache/LRU]*  
*\$cache attach-repl \$lru*
  - *set fifo [new Http/Cache/SIZE]*  
*\$cache attach-repl \$size*

#### 4.2.5 ส่วนจัดการสถาปัตยกรรมของการแคช

ส่วนจัดการสถาปัตยกรรมของการแคชทำหน้าที่ควบคุมการทำงานร่วมกันของแคชที่มีมากกว่าหนึ่งตัว เพื่อให้แคชแต่ละตัวสามารถติดต่อสื่อสาร หรือส่งผ่านการร้องขอระหว่างกันได้ ซึ่งจะช่วยให้แคชไม่ต้องเรียกขอเอกสารจากเซิร์ฟเวอร์โดยตรงในกรณีที่มีเอกสารอยู่ในแคชที่ใกล้เคียงกัน

การจัดการสถาปัตยกรรมของการแคชในงานวิทยานิพนธ์นี้ทำได้ทั้งในแบบพอลูก(ลำดับชั้น) และแบบพี่น้อง(ระดับเดียวกัน) [11] โดยโอทีซีแอลเม็ท็อดหลักที่ใช้สำหรับจัดการสถาปัตยกรรมของการแคชคือ

- 1) *set-parent {cache}* ใช้สำหรับกำหนดว่าแคชตัวใดเป็นแคชในระดับที่สูงกว่าที่เชื่อมต่อกันอยู่ ซึ่งใช้ในการเชื่อมต่อแบบลำดับชั้น
- 2) *set-group {group}* ใช้สำหรับกำหนดกลุ่มของแคชในระดับเดียวกัน เพื่อใช้ในการติดต่อสื่อสารภายในกลุ่ม
- 3) *get-server {cache pageid}* ใช้สำหรับค้นหาเอกสารภายในแคชของกลุ่มแคชในระดับเดียวกัน โดยเม็ท็อดนี้อยู่ภายในคลาส *Http/Cache/ICP* ซึ่งแคชแต่ละตัวจะมีลิสต์ของแคชที่อยู่ในระดับเดียวกันอยู่จากการกำหนดโดยใช้เม็ท็อด *set-group*

ในกรณีที่มีเอกสารอยู่ภายในแคชภายในกลุ่มเดียวกัน แคชจะทำการร้องขอเอกสารจากแคชตัวนั้น หากไม่พบเอกสารในแคชตัวใดเลย แคชจะส่งการร้องขอไปยังแคชในระดับที่สูงกว่าต่อไป ถ้าหากไม่พบเอกสารในแคชใดๆ เลย แคชจะทำการดึงเอกสารจากเซิร์ฟเวอร์โดยตรง

- 4) *attach-arch {architecture}* เป็นเม็ท็อดที่ใช้กำหนดรูปแบบของสถาปัตยกรรมการแคชแบบพี่น้อง สำหรับสถาปัตยกรรมแบบพอลูกสามารถกำหนดได้โดยใช้เม็ท็อด *set-parent* ในส่วนของการเชื่อมต่อโหนดต่างๆ ภายในสคริปต์ ซึ่งสามารถดูตัวอย่างสคริปต์ได้ในภาคผนวกเพื่อความเข้าใจยิ่งขึ้น

ตัวอย่างในการเรียกใช้ส่วนจัดการสถาปัตยกรรมของการแคช

- `$cache1 set-parent $cache2`
- `set arch [new Http/Cache/ICP]`  
`$cache attach-arch $arch`

#### 4.3 แนวทางการเพิ่มเติมส่วนประกอบในอนาคต

เนื่องจากวิธีการต่างๆ สำหรับการจำลองการทำงานของเว็บแคชในงานวิทยานิพนธ์นี้ มีเพียงจำนวนน้อยเมื่อเทียบกับวิธีการต่างๆ ที่มีอยู่ทั้งหมดในปัจจุบัน ดังนั้นการเพิ่มเติมส่วนประกอบสำหรับวิธีการใหม่ๆ จึงเป็นสิ่งที่จำเป็นในการทดสอบประสิทธิภาพของวิธีการนั้นๆ โดยเมื่อทำการเพิ่มเติมส่วนประกอบแล้วผู้ใช้งานยังสามารถเรียกใช้ส่วนประกอบเดิมได้ และยังสามารถเชื่อมต่อกับส่วนประกอบใหม่ได้อีกด้วย ซึ่งมีแนวทางในการเพิ่มเติมส่วนประกอบสำหรับจำลองการทำงานของเว็บแคชในอนาคตตามลักษณะของส่วนประกอบต่างๆ ดังนี้

##### 4.3.1 ส่วนจัดการการร้องขอ

สำหรับการพัฒนาส่วนจัดการการร้องขอเพิ่มเติม จะเป็นการเพิ่มเติมในลักษณะของการแปลงรูปแบบของล็อกไฟล์แบบต่างๆ ให้เป็นรูปแบบที่เอ็นเอสทูสามารถทำงานได้ ซึ่งการแปลงรูปแบบของล็อกไฟล์สามารถทำได้โดยการใช้ภาษาต่างๆ หรือทำการแปลงให้เป็นรูปแบบของสควิดและใช้โปรแกรมแปลงรูปแบบที่ได้พัฒนาไว้แล้วแปลงให้เป็นรูปแบบของเอ็นเอสทูอีกครั้งหนึ่ง

##### 4.3.2 ส่วนจัดการความต้อกันของเว็บแคช

สำหรับการพัฒนาความต้อกันของเว็บแคชเพิ่มเติม สามารถทำได้โดยการสร้างคลาสใหม่ ซึ่งภายในคลาสจะมีเม็ท็อด *is-expired* ใช้สำหรับคำนวณว่าเอกสารที่เก็บอยู่ภายในเว็บแคชหมดอายุหรือไม่ โดยแต่ละรูปแบบของความต้อกันของเว็บแคชจะมีการคำนวณที่แตกต่างกัน ดัง

นั้นการเพิ่มเติมความต้องกันของเว็บแคชในลักษณะนี้จึงทำได้ง่าย โดยสามารถดูตัวอย่างของเม็ท็อด *is-expired* ได้ในหัวข้อ 4.2.3

#### 4.3.3 ส่วนจัดการขั้นตอนวิธีการแทนที่

สำหรับการพัฒนาขั้นตอนวิธีการแทนที่เพิ่มเติม สามารถทำได้โดยการสร้างคลาสใหม่ ซึ่งภายในคลาสนั้นจะต้องประกอบด้วยเม็ท็อดต่างๆ ดังรายละเอียดที่ได้กล่าวในหัวข้อ 4.2.4 โดยมีตัวอย่างเม็ท็อดภายในคลาสดังต่อไปนี้

```
Http/Cache/XXX instproc init args {
...
}
Http/Cache/XXX instproc move-hit {pageid} {
...
}
Http/Cache/XXX instproc set-pos {pageid pagesize} {
...
}
Http/Cache/XXX instproc cache-repl {cache curr_size} {
...
}
```

#### 4.3.4 ส่วนจัดการสถาปัตยกรรมของการแคช

สำหรับการพัฒนาสถาปัตยกรรมของการแคชเพิ่มเติม สามารถทำได้โดยการสร้างคลาสใหม่ ซึ่งภายในคลาสนั้นจะต้องมีเม็ท็อด *get-server* ใช้สำหรับค้นหาเอกสารภายในแคชของกลุ่มแคชตามลักษณะของสถาปัตยกรรมของการแคชแบบต่างๆ โดยอาจจะต้องมีการเพิ่มเติมเม็ท็อดเข้าไปในคลาสใหม่ที่สร้างขึ้น ในกรณีนี้ที่แคชจำเป็นต้องมีการแลกเปลี่ยนข้อมูลอื่นๆ ซึ่งขึ้นอยู่กับสถาปัตยกรรมของการแคชที่ใช้

จากแนวทางที่ได้กล่าวไปนั้น จะเห็นได้ว่าการเพิ่มเติมส่วนประกอบอื่นๆ เพื่อจำลองการทำงานของเว็บแคชสามารถทำได้ไม่ยาก ซึ่งเมื่อพัฒนาส่วนประกอบใหม่ต่างๆ ขึ้นมาแล้ว การทำให้เอ็นเอสทูสามารถเรียกใช้ส่วนประกอบใหม่เหล่านี้ได้จะต้องทำตามวิธีการซึ่งได้กล่าวมาแล้วในบทที่ผ่านมา