

บทที่ 2

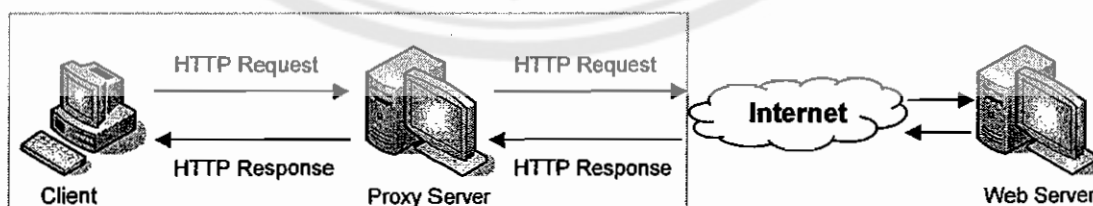
เอกสารและงานวิจัยที่เกี่ยวข้อง

ในการศึกษาเปรียบเทียบประสิทธิภาพระหว่างพร็อกซีเซิร์ฟเวอร์ที่จัดเก็บแคชแบบไฟล์เดียวและแบบหลายไฟล์นั้น มีแนวคิดและทฤษฎีที่เกี่ยวข้องที่ผู้วิจัยนำมาเป็นแนวทางประกอบ การดำเนินงานดังต่อไปนี้

หลักการทำงานพื้นฐานของพร็อกซีเซิร์ฟเวอร์ (Principle of Proxy Server)

พร็อกซีเซิร์ฟเวอร์(Proxy Server) หรือมีชื่อเรียกอีกอย่างหนึ่งว่าเว็บแคช (Web Cache) คือ เซิร์ฟเวอร์ที่ทำหน้าที่แทนเครื่องไคลเอนท์ (Client) ในการสื่อสารกับเว็บเซิร์ฟเวอร์ (Web Server) พร็อกซีเซิร์ฟเวอร์ (Proxy Server) จะมีแหล่งเก็บเว็บเพจ (Web Page) ล่าสุดที่ถูกร้องขอ โดยไคล-เอนท์ (Client) ซึ่งทุกๆ การร้องขอ (Request) จากเครื่องไคลเอนท์ (Client) จะถูกส่งไปยังพร็อกซีเซิร์ฟเวอร์ก่อน ซึ่งพร็อกซีเซิร์ฟเวอร์ก็จะทำหน้าที่แทนเว็บเบราว์เซอร์ (Web Browser) ในการค้นหา และทำการร้องขอ (Request) เว็บเพจที่ Client ต้องการแล้วส่งกลับไปที่ Client ทำการแสดงผลต่อไป (จตุชัย แพงจันทร์, อนุโชต วุฒิพรพงษ์. 2546 : 220)

พร็อกซีเซิร์ฟเวอร์ (Proxy Server) จะทำงานในระดับแอปพลิเคชันโปรโตคอล (Application Protocol) ซึ่งมีหลายรูปแบบด้วยกันเช่น FTP Proxy, Gopher Proxy, HTTP Proxy (Web Proxy) ซึ่ง HTTP Proxy หรือ Web Proxy เป็นโปรโตคอล (Protocol) ที่เราพบเห็นและมีการใช้งานกันอย่างแพร่หลาย เนื่องจากเป็นโปรโตคอล (Protocol) ที่ถูกนำมาใช้ในเป็นมาตรฐานของ Browser ทั่วๆ ไป พร็อกซีเซิร์ฟเวอร์ (Proxy Server) มีขั้นตอนการทำงานโดยทั่วไปดังภาพ 1



ภาพ 1 แสดงหลักการทำงานของ Proxy Server

1.1. เว็บเบราว์เซอร์ที่เครื่อง Client จะสร้างการเชื่อมต่อแบบ TCP กับพร็อกซีเซิร์ฟเวอร์ แล้วส่งการร้องขอไปยังพร็อกซีเซิร์ฟเวอร์

1.2. พร็อกซีเซิร์ฟเวอร์จะทำการตรวจสอบว่าไฟล์ที่ผู้ใช้ต้องการนั้น พร็อกซีได้เก็บข้อมูลไฟล์ไว้หรือไม่ ถ้าเจอพร็อกซีเซิร์ฟเวอร์ก็จะตอบกลับพร้อมกับไฟล์ที่ผู้ใช้ต้องการ

1.3. ถ้าในพร็อกซีเซิร์ฟเวอร์ไม่มีไฟล์ที่ผู้ใช้ต้องการ พร็อกซีก็จะทำการสร้างการเชื่อมต่อกับเว็บเซิร์ฟเวอร์จริง แล้วส่งการร้องขอไปยังเซิร์ฟเวอร์ดังกล่าว เมื่อเว็บเซิร์ฟเวอร์จริงได้รับการร้องขอก็จะส่งไฟล์กลับไปยังพร็อกซีเซิร์ฟเวอร์

1.4. เมื่อพร็อกซีได้รับการตอบกลับ (Reply) ก็จะมีการคัดลอก (Copy) ไฟล์ดังกล่าวเก็บไว้ในตัวพร็อกซี พร้อมกับส่งต่อไฟล์ดังกล่าวกลับไปยังเครื่องไคลเอนท์ (Client) เพื่อแสดงผลให้ผู้ใช้งานต่อไป

จากขั้นตอนการทำงานที่กล่าวมาข้างต้นนั้น พร็อกซีเซิร์ฟเวอร์ จะทำหน้าที่เป็นทั้งไคลเอนท์และเซิร์ฟเวอร์ในเวลาเดียวกัน โดยเมื่อไคลเอนท์ส่งการร้องขอมายังพร็อกซีนั่น พร็อกซีจะทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ แต่เมื่อพร็อกซีเซิร์ฟเวอร์ติดต่อกับเว็บเซิร์ฟเวอร์จริง ตัวพร็อกซีก็จะทำหน้าที่เป็นไคลเอนท์

พร็อกซีสามารถแบ่งออกตามลักษณะการใช้งานได้เป็น 3 ประเภทด้วยกันคือ

1. Transparent Proxy

Transparent Proxy จะมีการส่งค่า IP Address ของเครื่องที่ได้ทำการร้องขอ (Request) ข้อมูล ซึ่งจะทำให้ Server สามารถมองเห็น IP Address ของ Client ด้วย ดังนั้น Server สามารถที่ทราบได้ว่ากำลังติดต่อกับอยู่กับเครื่อง Proxy ตัวใด และ Proxy ส่งต่อข้อมูลไปให้กับ Client ตัวโดยยกตัวอย่างเช่นข้อมูลของ Header ทางฝั่ง Server มีลักษณะดังนี้

```
REMOTE_ADDR 202.183.45.254   IP Address ของ Proxy
REMOTE_PORT 12345           Port Number ของ Proxy Server
HTTP_VIA      proxy.NextProject.Net   ชื่อ Proxy Server
HTTP_X_FORWARDED_FOR 192.168.0.1   IP Address ของ Client
```

2. Anonymous Proxy

Anonymous Proxy จะทำการส่ง Header เฉพาะ HTTP_VIA และ REMOTE_ADDR เท่านั้น และ Proxy Server จะไม่ส่ง HTTP_X_FORWARDED_FOR ไปด้วย ดังนั้น Server จะรู้แต่เพียงว่ากำลังติดต่อกับ Proxy ตัวใดแต่จะ 모르 Proxy นั้นเอาข้อมูลนั้น ๆ ไปทำอะไร หรือให้ใครยกตัวอย่างเช่นข้อมูลของ Header ทางฝั่ง Server มีลักษณะดังนี้

REMOTE_ADDR 202.183.45.254 IP Address ของ Proxy
 REMOTE_PORT 12345 Port Number ของ Proxy Server
 HTTP_VIA proxy.NextProject.Net ชื่อ Proxy Server

3. High Anomity Proxy

High Anomity Proxy จะส่ง Header แค่ REMOTE_ADDR เท่านั้น ซึ่งเป็นของ Proxy Server ดังนั้น Server จะไม่รู้เลยว่า ข้อมูลถูกส่งหรือมาจากที่ใด เพราะจะเห็น Proxy Server เสมือนว่าเป็น Client ธรรมดาตัวหนึ่งเท่านั้น ยกตัวอย่างเช่นข้อมูลของ Header ทางฝั่ง Server มีลักษณะดังนี้

REMOTE_ADDR 202.183.45.254 IP Address ของ Proxy

พร็อกซีมีประโยชน์หลายอย่าง เช่น พร็อกซีจะช่วยลดเวลาของการตอบกลับ (Response Time) โดยเฉพาะในกรณีที่ลิงค์นั้นมีการเชื่อมต่อเข้ากับอินเทอร์เน็ตจะใช้เวลามากกว่า หากเทียบกับการลิงค์ระหว่างไคลเอนท์ (Client) กับ พร็อกซีเซิร์ฟเวอร์ (Proxy Server) เพราะเนื่องจากลิงค์ระหว่างไคลเอนท์ และองค์กรนั้นจะผ่านระบบเครือข่าย LAN ซึ่งความเร็วในปัจจุบันนั้นแบนด์วิดท์ (Bandwidth) จะอยู่ที่ 100 Mbps ส่วนลิงค์ที่เชื่อมต่อเข้ากับอินเทอร์เน็ตนั้น สำหรับองค์กรขนาดใหญ่อาจอยู่ที่ 1-2 Mbps แต่ถ้าเป็นองค์กรขนาดเล็กอาจใช้โมเด็มที่ความเร็ว 64 Kbps ซึ่งยังเป็นความเร็วที่ต่ำหากเทียบกับระบบเครือข่าย LAN จากลักษณะการทำงานของพร็อกซีเซิร์ฟเวอร์นั้น จะช่วยลดทราฟฟิกระหว่างองค์กรและอินเทอร์เน็ตได้ค่อนข้างมาก ทำให้องค์กรไม่ต้องอัปเดตลิงค์ไปยังอินเทอร์เน็ตมากเกินไปจนเกิดความจำเป็น เนื่องจากทราฟฟิกส่วนใหญ่จะเป็นเว็บ ดังนั้นถ้าสามารถลดทราฟฟิกของเว็บได้ทราฟฟิกโดยส่วนใหญ่ในระบบก็จะลดลงตามด้วยเช่นกัน ดังนั้นพร็อกซีจึงเป็นสิ่งจำเป็นสำหรับองค์กรในปัจจุบัน

หลักการเบื้องต้นของแคช (Principle of Cache)

ในการเรียกใช้ข้อมูลจากเวิร์ลไวด์เว็บ โปรแกรมเว็บไคลเอนต์ (Web Client) หรือเบราว์เซอร์ (Browser) จะทำการเรียกข้อมูลโดยระบุอาร์แอล (URL หรือ Uniform Resource Locator) ซึ่งประกอบด้วยชื่อเครื่องที่ต้องการติดต่อ ซึ่งผู้ใช้แต่ละคนสามารถเรียกข้อมูลจากแหล่งต่าง ๆ ได้โดยอิสระจากกันทำให้เกิดปัญหาการเรียกใช้ข้อมูลซ้ำกันหลายครั้งจากแหล่งข้อมูลเดียวกัน โดยเฉพาะแหล่งข้อมูลที่นิยมใช้กันมาก ก่อให้เกิดปัญหาความหนาแน่นของช่องสัญญาณการสื่อสาร ในที่นี้เราสามารถลดปัญหาการเรียกใช้ข้อมูลที่ซ้ำซ้อนกันเนื่องจากผู้ใช้แต่ละคนได้โดยอาศัยหลักการการทำงานของแคช หรืออาจเรียกว่าระบบพักข้อมูล

แคช คือ ฮาร์ดแวร์หรือซอฟต์แวร์ที่ถูกนำมาใช้เป็นที่พักข้อมูลที่มีการเรียกใช้ในระบบเครือข่ายอินเทอร์เน็ตโดยจะทำหน้าที่จัดเก็บเอกสารในอินเทอร์เน็ตที่ถูกเรียกโดยผู้ใช้งาน ซึ่งการจัดเก็บอาจจะจัดเก็บลงในหน่วยความจำหรือในฮาร์ดดิสก์ เมื่อมีการเรียกข้อมูลที่เคยเรียกมาแล้วก็ที่สามารถนำเอาข้อมูลที่มีอยู่ในแคชมาใช้ได้ทันที โดยไม่จำเป็นต้องนำมาจากแหล่งนั้นซ้ำอีก และการทำงานของแคชมักทำงานควบคู่ไปกับระบบพร็อกซี (Proxy) ซึ่งเป็นโปรแกรมที่เป็นตัวกลางรับ-ส่งข้อมูลระหว่างเบราว์เซอร์กับแหล่งข้อมูล โดยเว็บแคชสามารถแบ่งออกเป็นหลายรูปแบบดังนี้

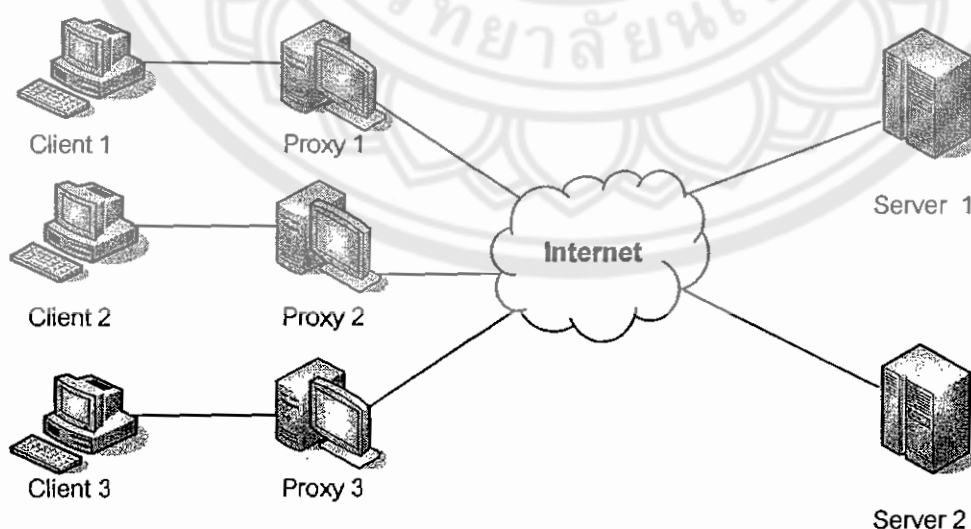
เบราว์เซอร์แคช (Browser Cache) ในปัจจุบันเบราว์เซอร์ส่วนใหญ่ทำงานในลักษณะนี้ได้ โดยแคชอาจจะใช้เนื้อที่ในหน่วยความจำ หรือในฮาร์ดดิสก์เป็นตัวเก็บข้อมูล ซึ่งช่วยลดเวลาในการเรียกข้อมูลซ้ำเดิม แต่จะมีข้อเสียตรงที่การจัดเก็บข้อมูลของเบราว์เซอร์ หรือเครื่องคอมพิวเตอร์นั้น อาจจะไปซ้ำซ้อนกับข้อมูลในแคชของเบราว์เซอร์หรือเครื่องคอมพิวเตอร์ของผู้ใช้เครื่องอื่นได้

พร็อกซีแคช (Proxy Cache) หรือเซิร์ฟเวอร์แคช (Server Cache) ส่วนใหญ่จะเป็นเครื่องที่มีสปีดค่อนข้างสูงหรือเป็นเครื่องคอมพิวเตอร์ขนาดใหญ่ โดยส่วนมากจะทำหน้าที่เป็นตัวเกตเวย์ (gateway) หรือไฟร์วอลล์ (firewall) ของหน่วยงานหรือบริษัทเพื่อให้เบราว์เซอร์ในเครื่องผู้ใช้งานสามารถ

สามารถเรียกข้อมูลจากภายนอกพร้อมกันได้ แคชในระดับนี้จะช่วยลดการเรียกใช้ข้อมูลและจัดเก็บ

ข้อมูลที่ซ้ำซ้อนกันของหน่วยงานหรือองค์กรขนาดใหญ่ได้ เนื่องจากการเรียกใช้ข้อมูลของพนักงาน

ซึ่งจะทำให้สามารถช่วยลดปริมาณข้อมูลที่ต้องเรียกผ่านช่องสัญญาณสื่อสารราคาแพงสู่ภายนอก



ภาพ 2 รูปแบบของเว็บแคช

ซึ่งแคชในแต่ละรูปแบบจะมีหลักการทำงานโดยรวมเหมือนกัน คือ ทำหน้าที่เก็บเอกสารในอินเทอร์เน็ตที่ถูกเรียกจากผู้ใช้อย่างน้อย ๆ หรือน่าจะถูกเรียกใช้ในอนาคตได้ ทั้งนี้เพื่อลดระยะเวลาในการรับ-ส่งข้อมูล รวมทั้งช่วยลดปริมาณการติดต่อสื่อสารในระบบเครือข่ายอีกด้วย โดยถ้าเป็นเบราเซอร์แคช หรือพร็อกซีแคชจะช่วยในการลดปริมาณการติดต่อสื่อสารที่ออกสู่อินเทอร์เน็ต ในขณะที่เซิร์ฟเวอร์แคชจะช่วยในการลดปริมาณการติดต่อสื่อสารที่เข้ามายังเซิร์ฟเวอร์ นอกจากนี้ในส่วนของพร็อกซีแคชนั้น สามารถทำการติดต่อสื่อสารเพื่อแลกเปลี่ยนข้อมูลกันได้อีกด้วย เพื่อช่วยเพิ่มโอกาสในการพบเอกสารให้มากขึ้น โดยมีรูปแบบการติดต่อสื่อสารแบบต่าง ๆ มากมาย ซึ่งแต่ละแบบจะมีข้อดีข้อเสียที่แตกต่างกันออกไป ทั้งนี้เพื่อช่วยในการติดต่อสื่อสารเป็นไปอย่างมีประสิทธิภาพมากขึ้น

การทำงานของโพรโตคอลเฮกซ์ทีทีพี (Hypertext Transfer Protocol หรือ HTTP)

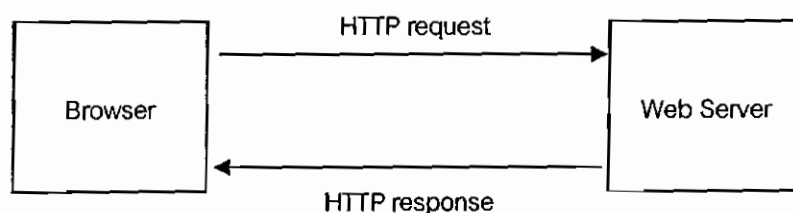
โพรโตคอลเฮกซ์ทีทีพี (Berners-Lee, 1996) เป็นโพรโตคอลในการสื่อสารข้อมูลระหว่างเว็บเซิร์ฟเวอร์ (Web Server) ซึ่งถือเป็นแหล่งข้อมูล กับ เว็บไคลเอนต์ (Web Client) หรือโปรแกรมเบราเซอร์ การเรียกใช้งานโพรโตคอลเฮกซ์ทีทีพีประกอบไปด้วย 3 ส่วนหลักคือ

1. วิธีการเรียกข้อมูล (Request method)
2. ตำแหน่งข้อมูล (Uniform Resource Locator หรือ URL)
3. เซตเดอร์เซตของวิธีการเรียกข้อมูล (Set of request header)

ข้อมูลที่เว็บเซิร์ฟเวอร์ตอบกลับมาจากโพรโตคอลเฮกซ์ทีทีพี ประกอบไปด้วย 3 ส่วนคือ

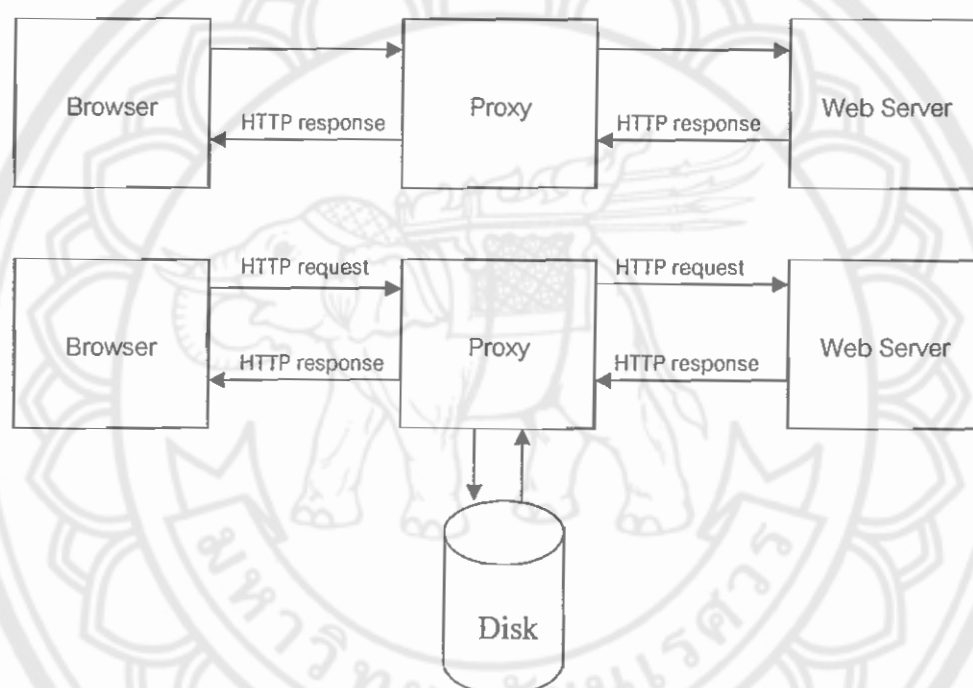
1. รหัสผลลัพธ์ (Numeric result code)
2. เซตคำตอบ (Set of reply header)
3. ตัวคำตอบเพิ่มเติม (Optional reply body)

การทำงานของโปรแกรมเบราเซอร์จะใช้วิธีส่งโดยการเรียกข้อมูลผ่านโพรโตคอลเฮกซ์ทีทีพีไปยังเว็บเซิร์ฟเวอร์ เมื่อเว็บเซิร์ฟเวอร์ได้รับการเรียกข้อมูลก็จะส่งคำตอบกลับมายังโปรแกรมเบราเซอร์ผ่านโพรโตคอลเฮกซ์ทีทีพี โดยมีการทำงานดังภาพ 3



ภาพ 3 แสดงการทำงานของโพรโตคอลเฮกซ์ทีทีพี

ในกรณีที่มีการเรียกข้อมูลผ่านพร็อกซีหรือแคชเซิร์ฟเวอร์ โดยพร็อกซีจะทำหน้าที่เป็นตัวกลางในการติดต่อ โดยเบราว์เซอร์จะติดต่อไปยังพร็อกซีโดยระบุอาร์ไอ (Uniform Resource Identifier) ของข้อมูลปลายทาง เมื่อพร็อกซีรับการติดต่อจากเบราว์เซอร์ก็จะส่งต่อไปยังเซิร์ฟเวอร์การใช้งาน ในกรณีนี้ส่วนมากจะใช้เพื่อจุดประสงค์ในด้านของความปลอดภัย โดยป้องกันการเรียกใช้งานจากผู้ใช้ไปยังแหล่งข้อมูลโดยตรง ส่วนการทำงานของแคชเซิร์ฟเวอร์จะคล้ายคลึงกับการทำงานของพร็อกซีแต่จะแตกต่างกันในแง่ เมื่อมีข้อมูลผ่านแคชเซิร์ฟเวอร์จะมีการพักข้อมูลนั้นภายในแคชเซิร์ฟเวอร์การทำงานสามารถแสดงได้ดังภาพ 4



ภาพ 4 แสดงการทำงานโพรโตคอลเอชทีทีพีกรณีเรียกใช้งานผ่านพร็อกซี/แคชเซิร์ฟเวอร์

ความทันสมัยของข้อมูลภายในแคช

เมื่อมีการเรียกใช้ข้อมูลจากแคชเซิร์ฟเวอร์ปัญหาที่ตามมา คือความทันสมัยของข้อมูลภายในแคช กล่าวคือเมื่อข้อมูลภายในเว็บเซิร์ฟเวอร์ที่เป็นแหล่งข้อมูลจริงมีการเปลี่ยนแปลงจะมีผลทำให้ข้อมูลภายในแคชไม่ตรงกับข้อมูลจริง ดังนั้นจำเป็นที่จะต้องมียุทธวิธีในการจัดการกับข้อมูลภายในแคช เพื่อให้ข้อมูลที่เรียกจากแคชมีความถูกต้องตรงกับข้อมูลที่มีอยู่จริงในเว็บเซิร์ฟเวอร์ปลายทาง โดยโปรแกรมแคชเซิร์ฟเวอร์มียุทธวิธีในการจัดการกับข้อมูลที่แตกต่างกันไป ซึ่งสามารถระบุได้ดังนี้

1. เมธอดเฮด (method head) วิธีการนี้แคชเซิร์ฟเวอร์จะตรวจสอบความถูกต้องของข้อมูลภายในแคช โดยการส่งเมธอดเฮดไปยังเว็บเซิร์ฟเวอร์ที่เป็นแหล่งข้อมูลต้นฉบับ ซึ่งคำตอบที่ได้รับจากเว็บเซิร์ฟเวอร์จะเป็นข้อมูลสรุป หลังจากนั้นแคชเซิร์ฟเวอร์จะทำการเปรียบเทียบข้อมูลว่ามีการเปลี่ยนแปลงหรือไม่ หากไม่มีการเปลี่ยนแปลงก็จะส่งข้อมูลนั้นไปยังเบราว์เซอร์ ถ้าข้อมูลมีการเปลี่ยนแปลงก็ใช้ เมธอดเก็ท (method get) เพื่อดึงข้อมูลใหม่เข้ามาเก็บภายในแคชก่อนส่งต่อไปยังเบราว์เซอร์ วิธีการนี้ต้องส่งข้อมูลไปยังเว็บเซิร์ฟเวอร์ถึง 2 ครั้ง ในกรณีที่ข้อมูลมีการเปลี่ยนแปลง

2. ไอเอ็มเอสเก็ท (If-Modified-Since (IMS) GET) วิธีการนี้เป็นการปรับปรุงการทำงานของเก็ทเมธอด โดยจะเพิ่มข้อมูลของเวลาในการเปลี่ยนแปลงครั้งสุดท้ายไว้ในส่วนหัวการเรียกข้อมูล (request header fields) ไปพร้อมกับเก็ทเมธอดไปยังเว็บเซิร์ฟเวอร์ที่เป็นแหล่งข้อมูลต้นฉบับ โดยก่อนที่จะทำการอ่านข้อมูล เก็ทเมธอดจะมีการเปรียบเทียบก่อนถ้าข้อมูลไม่มีการเปลี่ยนแปลงก็จะส่งผลลัพธ์กลับมาว่าไม่เปลี่ยนแปลง ทำให้สามารถเรียกข้อมูลจากแคชได้ทันที ถ้าข้อมูลมีการเปลี่ยนแปลงจะทำการอ่านข้อมูลนั้นกลับมา ซึ่งจะเห็นได้ว่าวิธีการนี้ทำการติดต่อระหว่างแคชเซิร์ฟเวอร์และเว็บเซิร์ฟเวอร์เพียงครั้งเดียวเท่านั้น

3. ทีทีแอล (Time To Live (TTL)) วิธีการนี้เป็นการกำหนดระยะเวลาของข้อมูลที่ใช้เมื่ออยู่ภายในแคช โดยหลังจากข้อมูลเข้ามาเก็บในแคช เมื่อมีการเรียก ถ้าหากข้อมูลในแคชเก็บอยู่เกินระยะเวลาที่กำหนด แคชเซิร์ฟเวอร์จะทำการเรียกข้อมูลใหม่จากแหล่งข้อมูลโดยตรง วิธีการนี้หากกำหนดค่าที่ทีทีแอลให้เหมาะสมกับชนิดของข้อมูลจะช่วยลดการติดต่อไปยังแหล่งข้อมูลต้นฉบับลงได้สูง หากข้อมูลเปลี่ยนแปลงช้าตามกำหนดเวลาที่สม่ำเสมอ

นอกจากวิธีการต่าง ๆ ที่ได้กล่าวมาข้างต้นแล้ว ปัจจุบันโพรโตคอลเอชทีทีพีเวอร์ชัน 1.1 (Berners-Lee et al., 1997) ซึ่งเป็นเวอร์ชันล่าสุดได้ทำการเพิ่มข้อมูลในส่วนหัวการเรียกข้อมูล ซึ่งโพรโตคอลเอชทีทีพี เวอร์ชัน 1.0 จะมีเฉพาะไอเอ็มเอสเท่านั้น โดยข้อมูลที่เพิ่มเข้ามาได้แก่

1. If Match
2. If None Match
3. If Range
4. If Unmodified Since
5. Last Modified

ลำดับชั้นของแคช (Hierarchical Cache)

การทำงานในลักษณะลำดับชั้นของแคช (Anawat et al., 1996) เป็นการเชื่อมต่อแคชเซิร์ฟเวอร์เข้าด้วยกัน เพื่อแลกเปลี่ยนข้อมูลระหว่างแคชเซิร์ฟเวอร์ ช่วยลดการใช้งานของสัญญาณออกไปภายนอกวงได้อีกชั้นหนึ่ง การทำงานเป็นดังนี้คือ เมื่อมีการเรียกข้อมูลที่ไม่อยู่ในแคชเซิร์ฟเวอร์เข้ามา แคชเซิร์ฟเวอร์จะสอบถามไปยังแคชเซิร์ฟเวอร์ตัวอื่น ในหน่วยงานเดียวกันที่ประสานการทำงานในลักษณะลำดับชั้น เพื่อดูว่ามีข้อมูลนั้นหรือไม่ ก่อนติดต่อไปยังแหล่งข้อมูลภายนอก

การทำงานของแคชในลักษณะลำดับชั้นสามารถแบ่งออกได้เป็น 2 ชนิดคือ

1. แคชแม่ (parent cache) การทำงานวิธีนี้ หากมีการเรียกข้อมูลที่ไม่อยู่ในแคชเซิร์ฟเวอร์จะถามไปยังแคชแม่ ซึ่งอยู่สูงขึ้นไปในลำดับชั้น เพื่อเรียกข้อมูลนั้น วิธีการนี้จะมีข้อจำกัดคือแคชแม่มีได้เพียงตัวเดียวเท่านั้น ทำให้เป็นคอขวดของระบบได้ และเป็นจุดอ่อนของระบบ หากแคชแม่ไม่สามารถทำงาน จะทำให้ทั้งระบบไม่สามารถทำงานได้
2. แคชแม่/พี่น้อง (parent/sibling cache) การทำงานวิธีนี้ ถ้าหากมีการเรียกข้อมูลที่ไม่อยู่ในแคชเซิร์ฟเวอร์จะส่งข้อมูลไปถามแคชพี่น้องในระดับเดียวกันก่อน ถ้าไม่มีจึงสอบถามไปยังแคชแม่ วิธีการนี้จะช่วยกระจายการทำงานของแคชเซิร์ฟเวอร์ และลดการทำงานของแคชแม่

ค่าที่ใช้ในการวัดประสิทธิภาพของแคช

ในการเปรียบเทียบผลการทดลองในแต่ละวิธีว่า การใช้แคชแบบใดจะมีประสิทธิภาพมากกว่ากันนั้น จะต้องมีการวัดค่าที่ใช้แสดงถึงประสิทธิภาพของแคช โดยค่าที่ใช้ในการวัดประสิทธิภาพของแคชนั้นมีอยู่หลายแบบด้วยกัน ซึ่งในวิทยานิพนธ์นี้จะทำการวัดประสิทธิภาพของแคชด้วยค่าเฉลี่ยของแอ็กเซสไทม์ (Average of Access time) คืออัตราส่วนระหว่างเวลาที่ใช้ไปตั้งแต่การเรียกหาข้อมูลจนได้รับข้อความตอบหรือได้รับข้อมูล (Total of elapsed time) ต่อจำนวนความต้องการข้อมูลทั้งหมด (Total number of accesses) ซึ่งสามารถคำนวณได้จากสูตร

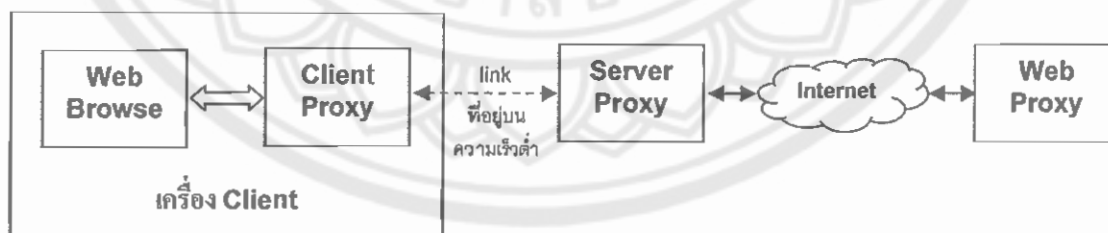
$$\text{Average of Access time} = \text{Total elapsed time} / \text{Total number of accesses}$$

กรณีที่คำนวณได้ค่าเฉลี่ยของแอ็กเซสไทม์ค่าต่ำกว่า จะเป็นตัวชี้ตัวหนึ่งที่จะพิจารณาว่าระบบแคชมีประสิทธิภาพดีกว่า ระบบแคชที่มีค่าเฉลี่ยของแอ็กเซสไทม์ค่าสูงกว่า เนื่องจากการที่ผู้ใช้ (User) ใช้เว็บไคลเอนต์ (Web client) จะเสียเวลาโดยเฉลี่ยน้อยกว่าสำหรับการได้รับข้อมูล ย่อมแสดงว่าระบบแคชนั้นมีประสิทธิภาพดีกว่า

แนวคิดและหลักการของเอ็กซ์พร็อกซี (Principle and Concept XProxy)

งานวิจัยเอ็กซ์พร็อกซี (xProxy) ได้ทำการออกแบบระบบการทำงานของ Web Proxy ออกเป็น 2 ส่วน คือ ส่วนของ Web Browser และ Web server โดยได้นำเอาวิธีการแบบเดลต้า (deltas) และการบีบอัดข้อมูล (Compression) มาใช้เพื่อลดจำนวนการจราจร (Traffic) บนระบบ WWW (World Wide Web) เนื่องจากในปัจจุบันยังมีผู้ใช้เป็นจำนวนมากมีการเชื่อมต่อแบบความเร็วต่ำ เช่น การใช้โมเด็ม (Modem) และการเชื่อมต่อระบบไร้สาย (Wireless) ซึ่งสถาปัตยกรรมของระบบได้ถูกแสดงไว้ในภาพ 8

จากภาพระบบ xProxy จะประกอบไปด้วย web caching 2 ประเภท คือ Client Proxy (CP) ซึ่งจะใช้วิธีการฝังตัวของโปรแกรม (Daemon) เข้าไปในเครื่องผู้ใช้งานปลายทาง เพื่อทำหน้าที่ในการให้บริการเกี่ยวกับการบีบอัด ซึ่ง CP จะเชื่อมต่ออยู่บนเครือข่ายที่มีความเร็วต่ำ และ Server Proxy (SP) จะทำหน้าที่เป็นตัวให้บริการงานต่าง ๆ บนระบบเครือข่าย เช่น การให้บริการ www, บริการ Download เป็นต้น โดย SP จะถูกเชื่อมต่อบนเครือข่ายอินเทอร์เน็ตความเร็วสูง เช่น การเชื่อมต่อกับศูนย์ให้บริการ หรือในมหาวิทยาลัย เป็นต้น โดยทั้ง CP และ SP จะทำหน้าที่เป็น HTTP Traffic Proxies ซึ่งเราสามารถอธิบายหน้าที่การทำงานของ CP และ SP ได้ดังนี้ คือ



ภาพ 5 ไดอะแกรมแสดงสถาปัตยกรรมของ xProxy โดยไดอะแกรมนี้จะแสดงให้เห็น
เข้าใจได้ง่ายขึ้นโดย Server Proxy (SP) สามารถที่จะมี Client Proxy (CP) ได้
หลายตัวหรือ Proxy Server (SP) ไม่จำเป็นจะต้องรู้ที่ตั้งของ Client Proxy (CP)

1. Client Proxy (CP)

ระบบ xProxy จะใช้ประโยชน์จาก Web Browser โดยจะต้องทำการคอนฟิกค่าบางอย่างหากผู้ใช้งานมีการร้องขอ (request) จาก Web Browser ก็จะต้องผ่านตัว CP ซึ่ง CP จะทำการตัดสินใจ โดยทำการค้นหาข้อมูลที่ต้องการจากสำเนาแคชของ CP เอง หาก CP ไม่ได้มีการทำสำเนาแคชข้อมูลนั้นไว้ CP ก็จะมีการร้องขอ (request) ไปยัง Server Proxy (SP) เมื่อได้รับการตอบกลับมันก็จะทำสำเนาแคชมายัง CP

2. Server Proxy (SP)

SP จะทำงานคล้าย ๆ กับ CP แต่จะแตกต่างกันตรงที่ CP จะทำการร้องขอ (request) จาก SP เพื่อทำสำเนาแคช ส่วน SP จะทำการร้องขอเว็บจากเครื่อง Server ที่เป็นแหล่งกำเนิด (คือจะทำสำเนาแคชจากเครื่อง Server ที่ให้บริการในระบบเครือข่ายอินเทอร์เน็ต) สิ่งที่มีความเหมือนกัน ทั้ง CP และ SP ก็คือการสำเนาแคช โดยเครื่อง Server จะเริ่มจากการทำการร้องขอ (request) ผ่านโปรโตคอล HTTP และเมื่อได้รับการตอบสนอง(response) Proxy ก็จะทำสำเนาแคชทันที แต่ถ้าหาก Proxy ไม่สามารถทำสำเนาแคชได้ เช่น อาจจะถูกเข้ารหัสแบบ MD5 ไว้ ไม่สามารถ copy ได้ ก็จะใช้วิธีการสร้างเดลต้า(deltas) เพื่อตอบสนองโดยจะบรรจุเอาความแตกต่างของเวอร์ชันปัจจุบันและเวอร์ชันเฉพาะของ object และถ้าหากมีการตอบสนองกลับมา Proxy ก็จะใช้ข้อมูลข่าวสารนี้สร้างเวอร์ชันใหม่ขึ้นมา และทำการสำเนาแคชไว้ และถ้าหากการตอบสนองของเดลต้าไม่สามารถทำได้ object ก็จะถูกบีบอัดก่อนที่จะถูกส่งออกไปยังผู้ใช้ส่วน object ที่ไม่ใช่ข้อความ เช่น รูปภาพ จะไม่ทำการบีบอัดเพราะว่า ไฟล์ชนิดนี้ได้ทำการบีบอัดไว้แล้ว

3. ลักษณะการทำงานของระบบ

โดยจะเริ่มจาก web browser ได้ทำการร้องขอ (request) ข้อมูลเอกสาร HTML และรูปภาพหลังจากที่ CP ได้รับการร้องขอ (request) มามันก็จะเริ่มทำการค้นหา object ที่ต้องการใน CP ก่อน หากไม่พบใน CP ก็จะมีการร้องขอ (request) ต่อไปยัง SP หาก SP ยังไม่ได้มีการทำสำเนาแคชไว้ มันก็จะทำการร้องขอ (request) ไปยัง server แหล่งกำเนิด (แหล่ง Web site ในอินเทอร์เน็ต) ต่อไปหลังจากที่ได้รับการตอบสนองจาก Server แหล่งกำเนิดแล้ว SP ก็จะมีการบันทึกสำเนาแคชของ object นี้เอาไว้ และทำการบีบอัด (compression) เอกสาร object HTML, รูปภาพ แล้วทำการตอบกลับ (reply) ไปยัง CP เมื่อ CP ได้รับแล้วมันก็จะทำการคลายการบีบอัด (decompresses) object HTML และทำการบันทึกเอกสาร HTML และรูปภาพ และทำสำเนาแคช object ทั้งหมดใน web browser ต่อไป แต่ถ้าสมมุติว่าเอกสาร HTML และรูปภาพที่ต้องการไม่ได้อยู่ในแคชของเครื่อง Server เมื่อผู้ใช้กลับไปยัง page นั้น การร้องขอข้อมูลภาพจะถูกให้บริการ

ในทันที โดย CP ในขณะที่เอกสาร HTML สามารถที่จะทำส่วนอื่น ๆ บน Server แหล่งกำเนิดได้ต่อไป และที่สำคัญเคล็ดลับสามารถที่จะทำการแทนที่เพิ่มข้อมูลที่ถูกบีบอัดไว้ โดยจะทำการถ่ายโอนข้อมูลเก็บไว้ใน SP และ CP ได้

4. เครื่องมือ(Implementation)

เครื่องมือที่นำมาใช้ในการทำสำเนาแคชของ Object คือ XDFS เป็นไฟล์ประเภท API ซึ่งเป็นส่วนหนึ่งของระบบ xProxy โดยจะมีความสามารถที่จะบันทึก object ได้หลาย ๆ เวอร์ชัน XDFS จะใช้อัลกอริทึมของ xDelta [Mac98] เพื่อคำนวณเดลต้า (deltas) และใช้ Berkeley DB[OBS99] ในการบันทึกข้อมูลข่าวสารบนดิสก์ ซึ่งการทำสำเนาแคชของ XDFS นี้จะคล้ายกับวิธีการของการทำ swap file ในระบบปฏิบัติการทั่ว ๆ ไป ซึ่งการบีบอัดเอ็กซ์พรีกซ์ ได้นำเอาเครื่องมือที่ชื่อว่า gzip [D96] มาใช้ในการบีบ-อัด เนื่องจากว่ามันสามารถที่จะนำไปใช้กับ IE และ Netscape ได้ เพราะ web browser ทั้งสองชนิดจะสนับสนุนเครื่องมือนี้อยู่แล้ว และ gzip เป็นเครื่องมือที่ได้รับการยอมรับว่า มีความสามารถในการบีบอัดได้ดี

5. สรุป

งานวิจัย xProxy ได้ทำการออกแบบระบบการทำงาน web proxy ออกเป็น 2 ส่วน คือ ส่วนของ Web Browser และ Web Server โดยได้นำวิธีการแบบเดลต้า (deltas) และการบีบอัดข้อมูลมาใช้เพื่อลดการจราจรบนระบบเครือข่ายและช่วยลดเวลาการมาถึงของข้อมูล ซึ่งการทำงานของระบบจะเป็นแบบแคชที่เปิดเผยมการทำงาน (Transparent Caching) โดยจะเจาะจงกับระบบที่มีการเชื่อมต่อที่มีความเร็วต่ำ เช่น การต่อกับโมเด็มหรือเครือข่ายแบบไร้สาย ซึ่งจากการทดลองพบว่าสามารถลดการจราจรบนเครือข่ายลงได้ มีความเร็วเพิ่มขึ้น และเวลาการตอบสนองดีขึ้น ประมาณ 53.2 %

Evaluating and Configuring WWW Cache Hierarchies

ในงานวิจัย Evaluating and Configuring WWW Cache Hierarchies ได้ทำการศึกษาว่า การใช้แคชหลายตัว ซึ่งมีรูปแบบความสัมพันธ์ที่ต่างกันจะทำให้เกิดผลกระทบอย่างไรบ้าง ทั้งนี้เพื่อหารูปแบบความสัมพันธ์ระหว่างแคชที่เหมาะสมต่อไป ซึ่งงานวิจัย Evaluating and Configuring WWW Cache Hierarchies ได้ทำการทดลอง โดยใช้แคชทั้งหมด 4 ตัว มีการทดสอบรูปแบบความสัมพันธ์ทั้งหมด 4 แบบ คือ แบบพี่น้องทั้งหมด แบบพี่น้อง 3 ตัว กับแม่ 1 ตัว แบบพี่น้องกับแม่อย่างละ 2 ตัว และแบบพี่น้องกับแม่อย่างละ 2 ตัว แต่มีการแยกเก็บโดเมนที่แม่โดยแม่ตัวแรกเก็บโดเมน .br เพียงอย่างเดียว ส่วนแม่อีกตัวเก็บโดเมนที่เหลือ โดยได้ทำการวัดประสิทธิภาพ

ภาพของแต่ละรูปแบบจากค่าฮิตเรโซ เวลาเฉลี่ยที่แคชใช้ในการตอบสนองผู้ใช้ เพอร์เซ็นต์ของช่วงเวลาทั้งหมดที่แคชใช้ในการทำงานแต่ละอย่าง (ทำงานกับเซิร์ฟเวอร์ ทำงานกับแม่ ทำงานกับผู้ใช้ และค่าส่วนเกิน (Overhead)) นอกจากนี้ยังมีการวัดค่าประสิทธิภาพของแคชอีกด้วย โดยสามารถคำนวณได้จากสูตรอย่างง่าย คือ ค่าประสิทธิภาพของแคชมีค่าเท่ากับ ช่วงเวลาที่แคชแบ่งเวลาให้ในการทำงานร่วมกับผู้ใช้ หารด้วย ช่วงเวลาทั้งหมดที่แคชใช้ในการทำงาน โดยจะนำค่าประสิทธิภาพของแคชที่ได้มาเขียนกราฟร่วมกับค่าฮิตเรโซ

จากผลการทดลองที่ได้ สรุปได้ว่า รูปแบบความสัมพันธ์แบบพี่น้องทั้งหมด จะให้ค่าฮิตเรโซที่สูงที่สุด ในขณะที่ความสัมพันธ์แบบพี่น้องกับแม่อย่างละ 2 ตัวและมีการแยกเก็บโดเมนที่เครื่องแม่จะให้ค่าประสิทธิภาพของแคชสูงที่สุด แต่จะให้ค่าฮิตเรโซที่ต่ำที่สุด ส่วนความสัมพันธ์แบบพี่น้อง 3 ตัวกับแม่ 1 ตัวให้ค่าประสิทธิภาพของแคชต่ำที่สุด

Utility of Co-operating Web Proxy Caches

ในงานวิจัย Utility of Co-operating Web Proxy Caches ได้ทำการศึกษาว่าการที่แคชแต่ละตัวสามารถติดต่อสื่อสารกันได้ก่อให้เกิดผลดีและผลเสียอย่างไรบ้าง เมื่อแคชแต่ละตัวมีขนาดและจำนวนและจำนวนของผู้ใช้ที่รับผิดชอบแตกต่างกันไป ซึ่งได้ทำการทดลองโดยใช้แคชทั้งหมด 3 ตัว แคชแต่ละตัวมีขนาดของเนื้อที่เก็บเอกสารไม่เท่ากัน รวมทั้งรับผิดชอบจำนวนของผู้ใช้ไม่เท่ากัน โดยได้ทำการทดสอบทั้งหมด 2 รูปแบบ คือรูปแบบที่แคชมีการติดต่อสื่อสารครบทุกตัวและรูปแบบที่แคชมีการติดต่อสื่อสารกับแคชที่มีขนาดใหญ่กว่าเท่านั้น โดยมีการวัดประสิทธิภาพของแต่ละรูปแบบจากค่าฮิตเรโซ ค่าไบตฮิตเรโซ ค่าฮิตเรโซที่เพิ่มขึ้นจากการติดต่อสื่อสารระหว่างแคช และเพอร์เซ็นต์ของจำนวนข้อความที่เพิ่มขึ้นเมื่อมีการติดต่อสื่อสารระหว่างแคช

จากผลการทดลองที่ได้ สรุปได้ว่า เมื่อแคชแต่ละตัวมีขนาดเนื้อที่เก็บเอกสารรวมทั้งรับผิดชอบจำนวนของผู้ใช้ที่แตกต่างกันมาก ทำให้แคชที่มีขนาดเล็กกว่าต้องรับภาระจากข้อความจำนวนมากที่ถูกส่งมาจากแคชที่มีขนาดใหญ่กว่า ในขณะที่โอกาสพบเอกสารในแคชที่มีขนาดเล็กกว่านั้นมีน้อยมาก ซึ่งถ้าหากมีการเปลี่ยนรูปแบบการติดต่อสื่อสารให้แคชติดต่อกับแคชที่มีขนาดใหญ่กว่านั้น ทำให้แคชที่มีขนาดเล็กกว่าไม่ต้องรองรับข้อความจำนวนมากซึ่งถูกส่งมาจากแคชที่มีขนาดใหญ่ รวมทั้งค่าฮิตเรโซที่ได้ไม่ตกลงมากเท่าใดนัก

Efficient Management of Data in Proxy Cache

งานวิจัยการจัดการประสิทธิภาพของข้อมูลในแคชพร็อกซี ได้พูดถึงการนำเอาเว็บแคชพร็อกซีมาเป็นสื่อกลางในการจัดการได้อย่างเหมาะสม โดยจะนำเสนอประสิทธิภาพในการจัดการกับ Framework ข้อมูลเว็บในแคชพร็อกซี squid ซึ่งข้อมูลที่อยู่ในแคชนั้นจะถูกจัดการทั้งส่วนที่ logical object และ logical page ซึ่งลักษณะของ logical page นั้นจะใช้วิธีการทำดัชนีโดยแบ่งเป็นเทอม และทำการสร้าง keyword และผู้ใช้สามารถที่จะทำการค้นหาข้อมูลได้โดยตรง เว็บอ็อบเจกต์โดยส่วนมากจะไม่นิยมทำดัชนี หากข้อมูลดังกล่าวเป็นชนิดรูปภาพ เสียง หรือภาพเคลื่อนไหว อีกทั้งงานวิจัยการจัดการประสิทธิภาพของข้อมูลในแคชพร็อกซี จะนำเสนอโมเดลของข้อมูลที่เป็นกลุ่มอ็อบเจกต์แบบ physical ภายในหน่วยของ logical โดยจะทำการเรียกใช้ในลักษณะที่เป็นแบบ logical page ในขณะที่การแทนที่ของข้อมูลในแคชยังคงอาศัยพื้นฐานของ physical object ลักษณะโครงสร้างของ physical page และ logical page จะมีลักษณะเป็นแบบลำดับขั้น

โครงสร้างลำดับขั้นของเว็บข้อมูลเป็นการแสดงให้เห็นถึงโครงสร้างแบบลำดับขั้นของเว็บข้อมูล ซึ่งเพจแบบ physical ที่จะจัดการกับเพจแบบ logical ได้นั้นจะต้องขึ้นอยู่กับโครงสร้างของเอกสารนั้น ๆ ด้วย ซึ่งเพจแบบ physical นั้นจะมีลักษณะเป็นไฟล์เดียวโดยจะทำการจัดเก็บกลุ่มของเอกสารเว็บที่มีความเป็นอิสระต่อกัน (2 ระดับด้านล่างในภาพ 9) ซึ่งนโยบายของเว็บแคชเกือบทั้งหมดได้ถูกพัฒนาให้เป็นแบบ physical page ส่วน logical page คือกลุ่มของ physical page กับอีกหนึ่งเพจหลักและลิงค์ของ physical page หลาย ๆ ลิงค์จะทำการเชื่อมโยงโดยตรง กลุ่มของ physical page จะเป็นสื่อประเภทแบบฝังตัวจาก logical page ซึ่งหน้าที่ของ logical page นี้จะเป็นหน่วยที่ใช้สำหรับการค้นหาและแยกประเภทข้อมูลข่าวสาร

ส่วนโครงสร้างการจัดเก็บไฟล์ของแคชบนพร็อกซี squid จะถูกจัดเก็บในลักษณะโครงสร้างแบบพิเศษ ซึ่งประกอบไปด้วย 3 ส่วนคือ (ดังภาพ 9) Meta Swap, Protocol HTTP และในส่วนของข้อมูล โดยการตอบสนองของ HTTP นี้ จะทำการบันทึกถัดจาก "Swap Meta" ซึ่งเป็นส่วนพิเศษที่ใช้สำหรับแคช ส่วนโปรโตคอล HTTP จะทำการเก็บข้อมูลข่าวสารต่าง ๆ เพื่อใช้ในการตอบสนอง ยกตัวอย่างเช่น เส้นแสดงสถานะภาพ, เวอร์ชันของ HTTP, URL, วันที่, ประเภทของสื่อ, แท็กระบุเอกลักษณ์เฉพาะตัว

"Swap meta" เป็นการอ้างอิงในส่วนของการจัดเก็บข้อมูลของเมต้า ซึ่งเป็นการเริ่มต้นของอ็อบเจกต์ที่ถูกจัดเก็บบนดิสก์ ข้อมูลเมต้าที่จัดเก็บนี้จะใช้รูปแบบของ type-length-value กล่าวคือกลุ่มข้อมูลข่าวสารของเมต้าแต่ละอันจะประกอบไปด้วยประเภทของเอกลักษณ์นั้น ๆ

โดยจะมีฟิลด์ความยาว และค่าความยาว โดยในส่วนนี้จะเริ่มต้นจาก 5 ไบต์แรกในไฟล์ ที่มีการบันทึกและสิ้นสุดด้วยค่าว่าง (0x00) ข้อมูลของเมตาด้านี้จะรวมข้อมูลข่าวสารเช่นเดียวกับค่าคีย์ของฮ็อบเบ็คแคช (MD5), URL, การใช้ฮ็อบเบ็ค ประเภทหลักๆ ที่ถูกจำกัด ได้แก่ STORE_META_KEY_MD5 ซึ่งจะแสดงให้เห็นว่าคีย์ของแคชที่ใช้บนพร็อกซี squid มีการเข้ารหัสด้วย MD5 เมื่อพร็อกซี squid มีการเปิดไฟล์บนดิสก์เพื่อทำการอ่าน มันสามารถที่จะทำการตรวจสอบได้ว่าการเข้ารหัสด้วย MD5 นี้ตรงกันกับการเข้ารหัส MD5 ที่ผู้ใช้ได้ทำการร้องขอมาหรือไม่ ถ้าไม่ตรงกันแสดงว่าผู้ใช้อาจป้อนรหัสไม่ถูกต้อง ส่วนฮ็อบเบ็ค STORE_META_URL เป็นฮ็อบเบ็คชนิด URL ซึ่งมันอาจจะถูกจับคู่กับการร้องขอของผู้ใช้ ส่วนการสร้างดัชนีเพื่อจัดเก็บบนพร็อกซี squid จะถูกสร้างขึ้นบนระบบการค้นหาหรือข้อความขนาดใหญ่ การทำดัชนีถือเป็นสิ่งที่ยากที่สุดสำหรับจัดการกับไฟล์ รวมถึงการสร้างดัชนีค่า ระยะเวลาในการทำดัชนี วันที่สร้างดัชนี การสร้างดัชนี URL และการสร้างดัชนีของฟิลด์เฉพาะ เช่น จาก, ถึง, หัวข้อ, รหัสข่าวสาร เป็นต้น สำหรับการ Search เมล์ และหัวข้อข่าว การสร้างดัชนีแบบอินเวอร์นี้จะใช้วิธีการจัดการ โดยทำการรายการในลักษณะกลับกัน ส่วนนโยบายของการแทนที่จะอาศัยพื้นฐานของการจัดการกับเว็บข้อมูลแบบลำดับขั้น โดยการแทนที่ฮ็อบเบ็คจะเริ่มจากการเลือกใช้วิธีแบบ logical page กับการให้สิทธิความสำคัญก่อนหลัง ลำดับความสำคัญของ logical page จะอยู่กับตัวแปร รวมถึงจำนวนของ physical page ภายใน logical page ความถี่ และช่วงเวลาในการเข้าถึง logical page ครั้งล่าสุด สมมติให้ C เป็น logical page ท่ามกลางการมีสิทธิก่อนจาก logical page ทั้งหมดในแคชนั้น ขั้นตอนต่อไปจะเป็นการเลือก physical page อย่างน้อยหนึ่งเพจหรือมากกว่า จากนั้นก็จะถูกแทนที่โดยเพจใหม่ที่ได้รับการเพิ่มเข้าไปในภายหลัง เนื่องจากแคชแบบเดิมยังคงใช้วิธีการแบบ physical page ขอบเขตของการแทนที่แคชนี้จะถูกจำกัดเป็นลักษณะ แบบวงกลมกล่าวคือ {D1{E1,E2},D2,D3{E3,E4,E5}} นโยบายของแคชสามารถได้ในกรณีนี้ได้รวมถึง LRU (least Recently Used), LFU(Least Frequently Used), ขนาดที่ปรับแต่งได้ของ LRU และอัลกอริทึมที่ดีที่นำมาใช้ในการพัฒนา งานวิจัยการจัดการประสิทธิภาพของข้อมูลในแคชพร็อกซีได้นำเอาอัลกอริทึมที่มีชื่อว่า LRU-SP มาใช้