กลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล สำหรับเอกสาร RSS

Determining Optimal Retrieval Points Mechanism for RSS Documents

เชาวนันท์ ขุนดำ

Chaowanan Khundam

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Prince of Songkla University
2554

ชื่อวิทยานิพนธ์ กลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล
สำหรับเอกสาร RSS
ผู้เขียน นายเชาวนันท์ ขุนดำ
สาขาวิชา วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

…….....
(ผู้ช่วยศาสตราจารย์ ดร.ลัดดา ปรีชาวีรกุล)

คณะกรรมการสอบ

……...ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.กฤษณะ ชินสาร)

…….....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ศิริรัตน์ วณิชโยบล)

…….....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.ลัดดา ปรีชาวีรกุล)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

……..……
(ศาสตราจารย์ ดร.อมรรัตน์ พงศ์ดารา)
คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ กลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS
ผู้เขียน นายเชาวนันท์ ขุนดำ
สาขาวิชา วิทยาการคอมพิวเตอร์
ปีการศึกษา 2553

บทคัดย่อ

เทคโนโลยี RSS (Really Simple Syndication) ถูกพัฒนาขึ้นเพื่อช่วยกระจายข้อมูลที่มีการเปลี่ยนแปลงบ่อยๆ เช่น ข้อมูลข่าวสารต่างๆ เว็บล็อก ไปยังผู้รับบริการ โดยผู้รับบริการไม่ต้องเสียเวลาเข้าชมแต่ละเว็บไซต์ อย่างไรก็ตามในการปรับปรุงข้อมูลข่าวสารต่างๆ ของ RSS จะมีตัวรวบรวมข่าวสารที่ทำการดึงข้อมูลจากแหล่งข้อมูลต่างๆ อยู่เป็นระยะ การทำงานของตัวรวบรวมข่าวสารโดยทั่วไปจะทำงานโดยตั้งเวลาในการดึงข้อมูลเป็นช่วงเวลาที่เท่าๆ กัน เช่น ตั้งเวลาให้ดึงข้อมูลทุกๆ 2 ชั่วโมง เป็นต้น จากการตั้งเวลาดังกล่าว ทำให้การดึงข้อมูลในบางครั้งอาจไม่มีการเปลี่ยนแปลงข้อมูลเลย หรือมีโอกาสได้ข่าวในเวลาที่ช้ากว่าความเป็นจริง วิทยานิพนธ์นี้จึงนำเสนอกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS (Determining Optimal Retrieval Points Mechanism for RSS Documents: DORPM) ทำให้ตัวรวบรวมข่าวสารสามารถดึงข้อมูลจากแหล่งข้อมูลต่างๆ ได้ในเวลาที่เหมาะสม ผลการศึกษาแสดงให้เห็นว่ากลไกดังกล่าวสามารถลดความล่าช้าเฉลี่ยในการดึงข้อมูล ช่วยให้ผู้ใช้บริการได้รับข่าวสารที่มีความทันสมัยมากยิ่งขึ้น ทั้งยังช่วยให้ตัวรวบรวมข่าวสารจัดสรรทรัพยากรในการดึงข้อมูลได้อย่างมีประสิทธิภาพอีกด้วย

Thesis Title	Determining Optimal Retrieval Points Mechanism for RSS Documents
Author	Mr. Chaowanan Khundam
Major Program	Computer Science
Academic Year	2010

ABSTRACT

Really Simple Syndication or RSS Technology is developed to help users receive updated contents from various publishers such as news web sites, weblog without visiting each site individually using an aggregator. The aggregator is scheduled at time intervals to feed automatically. However, setting time intervals may cause a delay between the publication of new contents at a publisher site and the appearance at the aggregator, or no updated contents may occur. Then, this thesis proposes a mechanism to determine an optimal retrieval points for RSS documents (Determining Optimal Retrieval Points Mechanism for RSS Documents: DORPM). With this mechanism, an aggregator can retrieve RSS documents in appropriate time. The study results show that the retrieval points can reduce the average delay in retrieving contents, enable the users to receive more updated contents and make the aggregator work more efficiently.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความช่วยเหลือและสนับสนุนจากบุคคลหลายฝ่าย ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง คือ

ผู้ช่วยศาสตราจารย์ ดร.ลัดดา ปรีชาวีรกุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่กรุณาให้คำปรึกษาแนะนำ และช่วยเหลือในการแก้ปัญหาต่างๆ ให้แก่ผู้วิจัยเสมอมา พร้อมทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้แก่ผู้วิจัย

ผู้ช่วยศาสตราจารย์ ดร.ศิริรัตน์ วณิชโยบล กรรมการในการสอบวิทยานิพนธ์ ที่กรุณาให้ข้อเสนอแนะในการทำวิจัย รวมทั้งตรวจทานแก้ไขวิทยานิพนธ์

อาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ทุกท่านที่ให้ความรู้ทางด้านวิชาการ ซึ่งสามารถนำมาใช้ในการทำวิทยานิพนธ์ได้อย่างดียิ่ง

ผู้ช่วยศาสตราจารย์ ดร.กฤษณะ ชินสาร ประธานกรรมการสอบวิทยานิพนธ์ ที่กรุณาช่วยตรวจทานและแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์

เจ้าหน้าที่ภาควิชาวิทยาการคอมพิวเตอร์ และเจ้าหน้าที่บัณฑิตวิทยาลัยทุกท่านที่ให้ความช่วยเหลือ และอำนวยความสะดวกเกี่ยวกับเอกสารต่างๆ

ทุนผู้ช่วยวิจัยคณะวิทยาศาสตร์ (RA) มหาวิทยาลัยสงขลานครินทร์ ที่มอบทุนสนับสนุนการศึกษาแก่ข้าพเจ้า

เพื่อนๆ พี่ๆ และน้องๆ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ ที่ให้คำปรึกษา และช่วยเหลือในการทำวิทยานิพนธ์

คุณพ่อ คุณแม่ และน้องชาย ที่ให้การสนับสนุนคอยเป็นห่วงสุขภาพและให้กำลังใจแก่ผู้วิจัยมาโดยตลอด

ผู้วิจัยขอขอบคุณทุกท่านเป็นอย่างสูงมา ณ โอกาสนี้

เชาวนันท์ ขุนดำ

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญ (ต่อ)

สารบัญ (ต่อ)

หน้า

รายการตาราง

รายการตาราง (ต่อ)

ตาราง **หน้า**

รายการภาพประกอบ

รายการภาพประกอบ (ต่อ)

รายการภาพประกอบ (ต่อ)

ภาพประกอบ **หน้า**

บทที่ 1

บทนำ

ในยุคที่มีข้อมูลสารสนเทศมากมายดังเช่นในปัจจุบัน อินเตอร์เน็ตเข้ามามีบทบาทสำคัญสำหรับเผยแพร่ข้อมูลสารสนเทศของแหล่งข้อมูลต่างๆ ไปยังผู้รับบริการ ทำให้ผู้ใช้บริการสามารถรับข้อมูลสารสนเทศได้อย่างสะดวกและง่ายยิ่งขึ้น แต่เนื่องจากจำนวนเว็บไซต์ที่เพิ่มมากขึ้นทำให้ข้อมูลข่าวสารต่างๆ เพิ่มมากขึ้นตามไปด้วย จึงเป็นเรื่องยากที่ผู้รับบริการจะติดตามข้อมูลข่าวสารต่างๆ จากแต่ละเว็บไซต์ว่ามีข้อมูลอะไรใหม่หรือไม่ ด้วยเหตุนี้เทคโนโลยี RSS (Really Simple Syndication) จึงถูกออกแบบขึ้นเพื่อช่วยกระจายข้อมูลที่มีการเปลี่ยนแปลงบ่อยๆ เช่น ข้อมูลข่าวสารต่างๆ เว็บล็อก ไปยังผู้รับบริการ โดยที่ผู้รับบริการไม่ต้องเสียเวลาเข้าชมแต่ละเว็บไซต์เหมือนเช่นในอดีต

RSS ซึ่งเป็นรูปแบบย่อยอย่างหนึ่งของภาษา XML (Extensible Markup Language) นั้น ถูกนำมาใช้เพื่อรวบรวมและเผยแพร่ข้อมูลข่าวสารที่ทันสมัยให้กับผู้รับบริการ โดยมีโครงสร้างที่เป็นมาตรฐานคือมีหัวข้อ วันที่ และเวลา พร้อมทั้งเนื้อหาบางส่วน อีกทั้งยังสามารถเชื่อมโยงไปยังรายละเอียดของเนื้อหานั้นๆ ได้อีกด้วย ประโยชน์ของ RSS ก็คือ การรวบรวมเนื้อหาทั้งหมดจากแหล่งข้อมูลหลายๆ แหล่งบนอินเตอร์เน็ตมาไว้ในที่เดียวกัน โดยที่ผู้รับบริการไม่จำเป็นต้องเยี่ยมชมเว็บไซต์ต่างๆ ซึ่ง RSS จะมีการจัดส่งสรุปเนื้อหาให้แก่ผู้รับบริการ หากผู้รับบริการต้องการอ่านเนื้อหาอย่างละเอียดก็สามารถทำได้เพียงแค่คลิกเชื่อมโยงไปยังเนื้อหาจากแหล่งข้อมูลนั้น

อย่างไรก็ตามในการปรับปรุงข้อมูลข่าวสารต่างๆ ของ RSS จะมีตัวรวบรวมข่าวสารที่ทำการดึงข้อมูลจากแหล่งข้อมูลต่างๆ อยู่เป็นระยะๆ การทำงานของตัวรวบรวมข่าวสารโดยทั่วไปจะทำงานโดยตั้งเวลาในการดึงข้อมูลเป็นช่วงเวลาที่เท่าๆ กัน เช่น ตั้งเวลาให้ดึงข้อมูลทุกๆ 2 ชั่วโมง เป็นต้น จากการตั้งเวลาดังกล่าว ทำให้การดึงข้อมูลในบางครั้งอาจไม่มีการเปลี่ยนแปลงข้อมูลเลย หรือมีโอกาสได้ข่าวในเวลาที่ช้ากว่าความเป็นจริง ดังนั้นการมีกลไกในการทำนายช่วงเวลาการดึงข้อมูลที่ดีจะช่วยเพิ่มประสิทธิภาพการทำงานของตัวรวบรวมข่าวสาร ทำให้สามารถดึงข้อมูลได้ในเวลาที่เหมาะสม

โดยทั่วไปตัวรวบรวมข่าวสารจะมีรูปแบบในการดึงข้อมูลอยู่ 2 รูปแบบ คือ

1. Resource allocation เป็นรูปแบบที่คำนึงถึงการจัดสรรทรัพยากรในการดึงข้อมูลของตัวรวบรวมข่าวสาร โดยแต่ละแหล่งข้อมูลจะได้รับจำนวนครั้งในการดึงข้อมูลที่แตกต่างกันขึ้นอยู่กับความสำคัญและอัตราการแสดงข้อมูลของแหล่งข้อมูลนั้น

2. Retrieval scheduling เป็นรูปแบบที่คำนึงถึงการกำหนดตำแหน่งเวลาในการดึงข้อมูล ซึ่งแต่ละแหล่งข้อมูลจะมีตำแหน่งเวลาในการดึงข้อมูลที่แตกต่างกันขึ้นอยู่กับว่าเวลาใดมีการแสดงข้อมูลมากน้อยอย่างไร โดยตัวรวบรวมข่าวสารจะทราบจำนวนครั้งในการดึงข้อมูลที่แน่นอนจากนั้นจึงทำการหาว่าตำแหน่งเวลาใดที่จะทำการดึงข้อมูล

ในปัจจุบันการพัฒนารูปแบบการดึงข้อมูลของตัวรวบรวมข่าวสารจะทำการรวบรวมข้อมูลระยะหนึ่งแล้วนำข้อมูลนั้นมาสร้างแบบจำลองการแสดงข้อมูล โดยแบบจำลองที่ได้นั้นจะมีการแสดงข้อมูลที่สอดคล้องกับแบบจำลองปัวส์ซอง จึงใช้แบบจำลองนั้นมากำหนดจำนวนครั้งและตำแหน่งในการดึงข้อมูลของแต่ละแหล่งข้อมูลเป็นไปตามรูปแบบ Resource allocation และ Retrieval scheduling ตามลำดับ โดยประสิทธิภาพของตัวรวบรวมข่าวสารจะขึ้นอยู่กับสิ่งที่ต้องการวัด เช่น ความล่าช้าในการดึงข้อมูลจะคำนึงถึงตำแหน่งเวลาในการดึงข้อมูลให้มีความใกล้เคียงกับเวลาในการแสดงข้อมูลมากที่สุด การหายไปของข้อมูลจะคำนึงจำนวนครั้งในการดึงข้อมูลให้มีความเหมาะสมไม่ให้น้อยจนเกินไปจนทำให้บางข้อมูลหายไป เป็นต้น จะเห็นได้ว่าประสิทธิภาพของตัวรวบรวมข่าวสารจะเกี่ยวข้องกับรูปแบบการดึงข้อมูลด้วยเช่นกัน

วิทยานิพนธ์นี้ได้นำเสนอการพัฒนารูปแบบการดึงข้อมูลแบบ Retrieval scheduling ซึ่งเป็นรูปแบบที่คำนึงถึงการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล โดยใช้ความล่าช้าในการดึงข้อมูลเป็นตัววัดประสิทธิภาพ เพื่อให้ได้ตำแหน่งเวลาในการดึงข้อมูลที่มีความล่าช้าที่น้อยลง แต่เนื่องจากการใช้แบบจำลองปัวส์ซองแบบเดิมนั้นจะสร้างแบบจำลองโดยใช้เวลาเรียนรู้ข้อมูลเพียงแค่ 2 สัปดาห์เท่านั้น จึงทำให้มีโอกาสที่แบบจำลองมีโอกาสที่ผิดพลาดได้สูง จึงนำเสนอการใช้เวลาจริงของการแสดงข้อมูลในอดีตแทนการใช้แบบจำลอง ซึ่งจะใช้ระยะเวลาการเรียนรู้ข้อมูลที่มากขึ้น ทำให้ความล่าช้าในการดึงข้อมูลลดลง และได้ตำแหน่งเวลาในการดึงข้อมูลที่ดียิ่งขึ้น

1.1 การตรวจเอกสารและงานวิจัยที่เกี่ยวข้อง

ในการปรับปรุงข้อมูลข่าวสารของ RSS ให้มีความทันสมัย จะมีตัวรวบรวมข่าวสารทำหน้าที่ดึงข้อมูลจากแหล่งข้อมูลต่างๆ เป็นระยะ โดยงานที่เกี่ยวข้องทางด้านนี้จะได้แก่ตัวรวบรวมข่าวสารของ RSS เกี่ยวโยงไปถึง Web crawler ซึ่งจะมีความคล้ายคลึงกันในเรื่องของการดึงข้อมูลซ้ำ และเพื่อให้การดึงข้อมูลมีประสิทธิภาพจึงมีประเด็นที่ต้องคำนึงถึงหลายประเด็นซึ่งจะส่งผลให้มีการดึงข้อมูลที่ต่างกันออกไป

1.1.1 เทคโนโลยี RSS (Really Simple Syndication)

เทคโนโลยี RSS เป็นข้อมูลรูปแบบหนึ่งที่อยู่ในรูปของภาษา XML ถูกพัฒนาขึ้นตามแนวคิดของเทคโนโลยี Push (Umbach, 1997) ถูกนำมาใช้เพื่อรวบรวมและเผยแพร่เนื้อหาหรือข่าวสารของเว็บไซต์ที่มีการเปลี่ยนแปลงบ่อย ส่วนใหญ่จะเป็นเนื้อหาที่มีลักษณะเป็นข่าว เว็บบอร์ด หรือบล็อก ช่วยให้ผู้ใช้ได้รับข่าวสารที่ทันสมัยได้โดยไม่ต้องเสียเวลาเยี่ยมชมเว็บไซต์บ่อยๆ

โดยผู้ใช้สามารถรับข่าวสารจากเว็บไซต์ที่ให้บริการ RSS ได้ด้วยการใช้โปรแกรมรวบรวมข่าวสารที่เรียกว่า Reader หรือ Aggregator ซึ่งมีหลักการทำงานคล้ายกับโปรแกรมรับอีเมล์ เพียงแต่ผู้รับบริการต้องลงทะเบียนรับข่าวสารที่สนใจจากตัวรวบรวมข่าวสาร โดยตัวรวบรวมข่าวสารจะรวบรวมเอกสาร RSS จากเว็บไซต์ต่างๆ และตรวจสอบการเปลี่ยนแปลงข้อมูล แล้วแสดงผลข้อมูลล่าสุดให้อัตโนมัติ ซึ่งข้อมูลที่ได้จะเป็นเพียงหัวข้อข่าวหรือรายละเอียดโดยย่อเท่านั้น ส่วนเนื้อหา หรือข้อความหลักของข่าวนั้นจะมีลิงค์เชื่อมโยงให้อีกทีหนึ่ง

1.1.2 แบบจำลองการแสดงข้อมูล (Posting Generation Model)

วิธีหนึ่งที่จะช่วยให้ทราบถึงลักษณะการแสดงข้อมูลของแหล่งข้อมูลแต่ละแหล่งคือการสร้างแบบจำลองขึ้นมา เพื่อนำมาใช้กำหนดจำนวนครั้งและตำแหน่งเวลาในการดึงข้อมูลสำหรับแหล่งข้อมูลนั้นๆ เดิมทีการนำแบบจำลองมาใช้นั้นมาจากการหาระยะเวลาในการดึงข้อมูลซ้ำของ Web crawler โดยแบบจำลองที่ใช้นั้นจะสร้างขึ้นโดยใช้ข้อมูลของการเปลี่ยนแปลงหน้าเว็บเพจในอดีต ซึ่งลักษณะของข้อมูลที่ได้นั้นสอดคล้องกับการแจกแจงแบบปัวส์ซอง จึงมีการนำเสนอการนำแบบจำลองปัวส์ซองมาใช้ในการพิจารณาหาระยะเวลาในการดึงข้อมูลซ้ำ (Cho and Molina, 2000) อย่างไรก็ตามมีงานวิจัยที่ไม่เห็นด้วยกับการนำแบบจำลองปัวส์ซองมาใช้ เนื่องจากข้อมูลการเปลี่ยนแปลงหน้าเว็บเพจที่อ้างว่าสอดคล้องกับการแจกแจงแบบปัวส์ซองนั้นมาจากข้อมูลของหน้าเว็บเพจในประเทศสหรัฐอเมริกาเพียงอย่างเดียว จึงนำเสนอการใช้

ข้อมูลโดยตรงแทนการใช้แบบจำลอง (Edwards *et al*., 2000) แต่ก็ยังมีการปรับปรุงการใช้แบบจำลองปัวส์ซองให้มีความถูกต้องแม่นยำมากยิ่งขึ้นในงานวิจัยต่อมา (Cho and Molina, 2003) นอกจากนี้ยังมีการนำเสนอการนำอัตราการดาวน์โหลดและการปรับปรุงข้อมูลแทนการใช้แบบจำลองปัวส์ซองอีกด้วย (Kim and Lee, 2007)

ในการดึงข้อมูลของตัวรวบรวมข่าวสารนั้นมีความคล้ายคลึงกับการหาระยะเวลาในการดึงข้อมูลซ้ำของ Web crawler งานวิจัยทางด้านนี้จึงอ้างอิงงานวิจัยทางด้าน Web crawler แบบจำลองเป็นสิ่งหนึ่งที่ได้มีการพัฒนานำมาใช้ต่อโดยได้ทำการปรับปรุงให้มีความเหมาะสม ระยะเวลาหรือช่วงเวลาที่พิจารณาของการเปลี่ยนแปลงข้อมูลเป็นสิ่งหนึ่งที่แตกต่างกัน เนื่องจากข้อมูลส่วนมากที่ให้บริการ RSS นั้นเป็นข้อมูลที่มีการเปลี่ยนแปลงบ่อยต่างจากเว็บเพจที่จะใช้เวลาที่นานกว่าในการเปลี่ยนแปลงข้อมูลแต่ละครั้ง ดังนั้นระยะเวลาที่นำมาใช้พิจารณาในแบบจำลองปัวส์ซองจึงแตกต่างกัน (Sia and Cho, 2005) โดยแบบจำลองที่ตัวรวบรวมข่าวสารของ RSS ใช้นั้นจะมีระยะเวลาที่สั้นกว่า อีกทั้งเวลายังส่งผลต่อการแสดงข้อมูลอีกด้วย เช่น เมื่อพิจารณาในระยะเวลา 1 วัน เวลาช่วงกลางวันมีอัตราการแสดงข้อมูลที่มากกว่าช่วงเวลากลางคืนในทุกๆ วัน เป็นต้น แบบจำลองที่ใช้จึงเป็นแบบจำลองปัวส์ซองแบบไม่เป็นเอกพันธ์ (Non – homogeneous Poisson Model) แทนการใช้แบบจำลองปัวส์ซองแบบเอกพันธ์ (Homogeneous Poisson Model) ที่ใช้ในงานวิจัยทางด้าน Web crawler

1.1.3 รูปแบบการดึงข้อมูล (Retrieval Polices)

ในการดึงข้อมูลไม่ว่าจะเป็นของ Web crawler หรือตัวรวบรวมข่าวสารของ RSS มีรูปแบบในการดึงข้อมูลที่แตกต่างกันออกไป แต่สามารถแบ่งออกได้เป็น 2 รูปแบบใหญ่ๆ คือ การดึงข้อมูลโดยพิจารณาจำนวนครั้ง และการดึงข้อมูลโดยพิจารณาตำแหน่งเวลา (Sia and Cho, 2005) โดยการดึงข้อมูลแบบพิจารณาจำนวนครั้งหรือ เรียกว่า “Resource allocation” เป็นรูปแบบที่คำนึงถึงการจัดสรรทรัพยากรในการดึงข้อมูล โดยแต่ละแหล่งข้อมูลจะได้รับจำนวนครั้งในการดึงข้อมูลที่แตกต่างกันขึ้นอยู่กับความสำคัญและอัตราการแสดงข้อมูลของแหล่งข้อมูลนั้น งานวิจัยทางด้าน Web crawler มักจะพิจารณารูปแบบการดึงข้อมูลนี้ ซึ่งจะทำการหาจำนวนครั้งในการดึงข้อมูลหน้าเว็บเพจซ้ำในช่วงระยะเวลาที่กำหนด ส่วนงานวิจัยทางด้านตัวรวบรวมข่าวสารของ RSS ได้มีการกำหนดจำนวนครั้งในการดึงข้อมูลที่คำนึงถึงการหายไปของข้อมูลโดยดูจากจำนวนข้อมูลที่แหล่งข้อมูลจะแสดงได้ (Han *et al*., 2008)

สำหรับการดึงข้อมูลแบบพิจารณาตำแหน่งเวลาหรือ เรียกว่า “Retrieval scheduling” เป็นรูปแบบที่คำนึงถึงการกำหนดตำแหน่งเวลาในการดึงข้อมูล ซึ่งแต่ละแหล่งข้อมูลจะมีตำแหน่งเวลาในการดึงข้อมูลที่แตกต่างกันขึ้นอยู่กับว่าเวลาใดมีการแสดงข้อมูลมากน้อยอย่างไร งานวิจัยทางด้าน Web crawler จะสนใจรูปแบบนี้น้อยกว่าแบบ Resource allocation อย่างไรก็ตามหลายงานวิจัยก็ทำควบคู่กันทั้ง 2 รูปแบบ สำหรับตัวรวบรวมข่าวสาร

ของ RSS ได้มีการนำเอาประวัติการเข้าใช้งานของผู้ใช้มาพิจารณาหาตำแหน่งเวลาในการดึงข้อมูลเพื่อให้ผู้ใช้ได้รับข้อมูลที่มีความทันสมัยตรงกับการใช้งาน (Sia *et al*., 2007) ซึ่งงานวิจัยด้านนี้การหาตำแหน่งเวลาในการดึงข้อมูลจะมีความสำคัญเนื่องจากตำแหน่งเวลาในการดึงข้อมูลจะส่งผลต่อความทันสมัยของข้อมูลที่มีการเปลี่ยนแปลงอย่างรวดเร็วอย่างมาก

จากการตรวจสอบเอกสารและงานวิจัยที่เกี่ยวข้องสามารถสรุปได้ว่า ในการดึงข้อมูลไม่ว่าจะเป็นของ Web crawler หรือตัวรวบรวมข่าวสารของ RSS จะมีการกำหนดช่วงเวลาในการดึงข้อมูลซ้ำ ซึ่งจะใช้การแสดงข้อมูลในอดีตมาเป็นตัวกำหนดว่าควรดึงข้อมูลซ้ำเมื่อเวลาใด โดยมีลักษณะการนำข้อมูลมาใช้อยู่ 2 แบบ คือ แบบแรกจะเสนอการใช้แบบจำลองซึ่งแบบจำลองที่ได้สอดคล้องกับแบบจำลองปัวส์ซอง แบบที่สองจะเสนอการใช้ข้อมูลโดยตรง เนื่องจากเห็นว่าการใช้แบบจำลองที่สอดคล้องกับแบบจำลองปัวส์ซองนั้นได้มาจากการเก็บข้อมูลที่ยังไม่ครอบคลุม ทั้งนี้การนำข้อมูลมาใช้ทั้งสองแบบจะมีข้อดีและข้อเสียแตกต่างกัน ข้อดีของการใช้แบบจำลองคือจะทำให้สามารถนำข้อมูลจากแบบจำลองไปใช้ได้ง่าย สะดวก และประหยัดเวลาในการแปลงข้อมูล แต่ก็จะมีข้อเสียตรงที่แบบจำลองมีโอกาสผิดพลาดได้สูง เนื่องจากแบบจำลองที่ได้จะถือเป็นตัวแทนของข้อมูลทั้งหมด ส่วนการใช้ข้อมูลโดยตรงนั้นจะทำให้ได้ข้อมูลที่ถูกต้องและมีความแม่นยำสูงเนื่องจากนำข้อมูลมาใช้โดยตรง แต่มีข้อเสียคือจะมีความยุ่งยากและใช้เวลามากในการแปลงข้อมูลให้อยู่ในรูปที่เหมาะสมที่จะนำมาใช้ได้ และในประเด็นการเลือกรูปแบบการดึงข้อมูลนั้นจะขึ้นอยู่กับลักษณะของการใช้งานเป็นหลัก เช่น รูปแบบ “Resource allocation” หรือการดึงข้อมูลโดยพิจารณาจำนวนครั้ง จะเป็นรูปแบบที่คำนึงถึงการจัดสรรทรัพยากรให้มีความเหมาะสม โดยจะให้ความสำคัญกับแหล่งข้อมูลที่การแสดงข้อมูลมากหรือมีการเปลี่ยนแปลงข้อมูลบ่อยๆ ซึ่งรูปแบบนี้จะเหมาะสำหรับ Web crawler ส่วนรูปแบบ “Retrieval scheduling” หรือการดึงข้อมูลโดยพิจารณาตำแหน่งเวลา จะเป็นรูปแบบที่คำนึงการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล รูปแบบนี้จึงเหมาะสำหรับการดึงข้อมูลที่มีการเปลี่ยนแปลงที่รวดเร็วและต้องการตำแหน่งเวลาที่แม่นยำในการดึงข้อมูล ดังเช่นตัวรวบรวมข่าวสารของ RSS อย่างไรก็ตามทั้งสองรูปแบบสามารถนำมาใช้ร่วมกันได้เพื่อทำให้การดึงข้อมูลนั้นมีประสิทธิภาพมากยิ่งขึ้น

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อวิเคราะห์และออกแบบกลไกกำหนดตำแหน่งเวลาในการดึงข้อมูลสำหรับตัวรวบรวมข่าวสารให้สามารถดึงข้อมูลได้ในเวลาที่เหมาะสม

1.2.2 เพื่อพัฒนากลไกกำหนดตำแหน่งเวลาในการดึงข้อมูลสำหรับตัวรวบรวมข่าวสารให้สามารถดึงข้อมูลได้ในเวลาที่เหมาะสม

1.3 ขอบเขตการดำเนินงาน

1.3.1 วิเคราะห์และออกแบบกลไกการทำงานสำหรับตัวรวบรวมข่าวสารให้สามารถดึงข้อมูลได้ในเวลาที่เหมาะสม ซึ่งใช้หลักการดังต่อไปนี้

1) เก็บรวบรวมแหล่งข่าวสารที่ให้บริการ RSS เพื่อนำมาใช้เป็นข้อมูลในการวิเคราะห์รูปแบบในการแสดงข้อมูลของแต่ละแหล่งข้อมูล

2) สกัดข้อมูลโดยนำแท็กเวลาในการแสดงข่าว มาสกัดเอาข้อมูลวันและเวลาเก็บเอาไว้ในฐานข้อมูล

3) แปลงข้อมูลวันและเวลาจัดให้อยู่ในรูปแบบที่เหมาะสม โดยรวบรวมจำนวนข่าวเป็นช่วงๆ ละ 1 ชม. เท่าๆ กัน เพื่อให้ง่ายต่อการนำข้อมูลไปวิเคราะห์ในขั้นตอนต่อไป

4) กำหนดปัจจัยที่ส่งผลต่อความล่าช้าในการดึงข้อมูล ซึ่งจะประกอบด้วย ตำแหน่งเวลาในการแสดงข่าว จำนวนข่าวที่แสดงในช่วงเวลานั้น และตำแหน่งในการดึงข้อมูล

5) คำนวณความล่าช้าที่เกิดขึ้น โดยนำปัจจัยดังกล่าวมาคำนวณความล่าช้าที่ตำแหน่งเวลาในการดึงข้อมูลต่างๆ

6) กำหนดตำแหน่งเวลาในการดึงข้อมูลที่ทำให้เกิดความล่าช้าน้อยที่สุด

1.3.2 พัฒนาและทดสอบกลไกการทำงานสำหรับตัวรวบรวมข่าวสารให้สามารถดึงข้อมูลได้ในเวลาที่เหมาะสม ตามที่ได้ออกแบบไว้

1.4 ขั้นตอนและระยะเวลาการดำเนินงาน

1.4.1 ขั้นตอนการดำเนินงาน

1) ศึกษางานวิจัยและเอกสารที่เกี่ยวข้อง ดังนี้
   1.1) เทคโนโลยี RSS (Really Simple Syndication)
   1.2) โมเดลทางสถิติที่ใช้จำลองการแสดงข้อมูล
   1.3) เทคโนโลยีอื่นๆ ที่เกี่ยวข้อง
2) ศึกษาเทคโนโลยีและเครื่องมือสำหรับงานวิจัย
3) กำหนดขอบเขตของปัญหาในการทำวิจัย
4) วิเคราะห์และออกแบบกลไกการทำงาน
5) พัฒนาและทดสอบกลไกการทำงานตามที่ได้ออกแบบไว้

6) เขียนบทความวิจัย

7) จัดทำเอกสารวิทยานิพนธ์

1.4.2 ระยะเวลาการดำเนินงาน

มกราคม 2553 – เมษายน 2554

1.4.3 แผนการดำเนินการวิจัย

ระยะเวลาการดำเนินการวิจัยแสดงดังตารางที่ 1.1

ตารางที่ 1.1 ระยะเวลาการดำเนินการวิจัย

กิจกรรม/ขั้นตอนการดำเนินงาน	เดือน															
	2553												2554			
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4
1. ศึกษางานวิจัยและเอกสารที่เกี่ยวข้อง	←											→				
2. ศึกษาเทคโนโลยีและเครื่องมือสำหรับงานวิจัย			←						→							
3. กำหนดขอบเขตของปัญหาในการทำวิจัย					←				→							
4. วิเคราะห์และออกแบบกลไกการทำงาน					←				→							
5. พัฒนาและทดสอบกลไกการทำงาน					←							→				
6. เขียนบทความวิจัย					←							→				
7. จัดทำเอกสารวิทยานิพนธ์												←				→

1.5 สถานที่และเครื่องมือที่ใช้

1.5.1 สถานที่

ห้องปฏิบัติการวิจัยเทคโนโลยีระบบสารสนเทศและการประยุกต์ (CS207) ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์

1.5.2 เครื่องมือที่ใช้

1) ด้านฮาร์ดแวร์

เครื่องคอมพิวเตอร์ส่วนบุคคล หน่วยความจำขนาด 2 GB และฮาร์ดดิสก์ความจุ 300 GB จำนวน 2 เครื่อง สำหรับพัฒนาและทดสอบระบบ

2) ด้านซอฟต์แวร์

2.1) ระบบปฏิบัติการ Microsoft Windows XP

2.2) โปรแกรมเว็บเซิร์ฟเวอร์ Apache2.2

2.3) ระบบจัดการฐานข้อมูล MySQL

2.4) ภาษา PHP Java และ C

1.6 ประโยชน์ที่คาดว่าจะได้รับ

นำหลักการที่ได้ไปประยุกต์ใช้เพื่อให้ได้กลไกกำหนดตำแหน่งเวลาในการดึงข้อมูลสำหรับตัวรวบรวมข่าวสารที่สามารถดึงข้อมูลได้ในเวลาที่เหมาะสม ซึ่งจะช่วยลดความล่าช้าในการดึงข้อมูล ทำให้ผู้ใช้ได้รับข่าวสารที่มีความทันสมัย

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีต่างๆ ที่ใช้ในการออกแบบและพัฒนากลไกกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับตัวรวบรวมข่าวสาร ประกอบด้วย การติดต่อสื่อสารผ่านเครือข่าย ภาษา XML (Extensible Markup Language) เทคโนโลยี Push และ Pull (Push and Pull Technology) เทคโนโลยี RSS (Really Simple Syndication) และการปรับปรุงข้อมูล

2.1 การติดต่อสื่อสารผ่านเครือข่าย

เครือข่ายคอมพิวเตอร์เป็นการนำเอาเครื่องคอมพิวเตอร์ต่างๆ มาเชื่อมต่อกันเป็นเครือข่ายให้บริการทางด้านการติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันระหว่างเครือข่าย โดยแบ่งการทำงานออกเป็น 2 ฝั่ง คือ ฝั่งเครื่องคอมพิวเตอร์ที่เป็นผู้รับบริการหรือไคลเอนต์ (Client) และฝั่งเครื่องคอมพิวเตอร์ที่เป็นผู้ให้บริการหรือเซิร์ฟเวอร์ (Server)

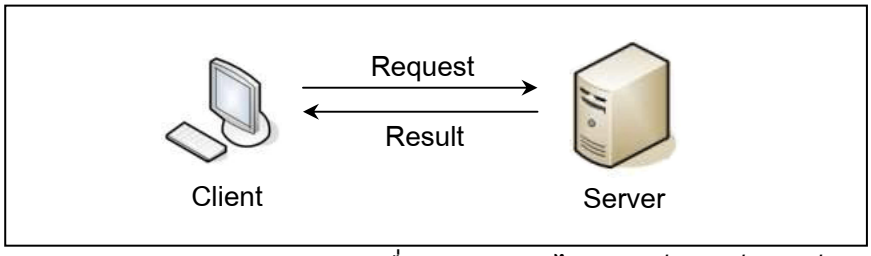
2.1.1 TCP/IP

การที่จะให้คอมพิวเตอร์สามารถติดต่อสื่อสารกันได้อย่างเข้าใจนั้น จำเป็นต้องมีภาษาในการสื่อสารโดยเฉพาะ สำหรับภาษาของการสื่อสารในคอมพิวเตอร์เรียกว่า โปรโตคอล (Protocol) เป็นระเบียบวิธีที่กำหนดขึ้นสำหรับการสื่อสาร ให้สามารถติดต่อสื่อสารกันหรือรับส่งข้อมูลระหว่างต้นทางกับปลายทางได้อย่างถูกต้องไม่ผิดพลาด TCP/IP (Transmission Control Protocol/Internet Protocol) (Steven, 1994) เป็นโปรโตคอลหนึ่งที่ใช้กันในเครือข่ายอินเตอร์เน็ตในปัจจุบัน เพื่อกำหนดกฎเกณฑ์ รูปแบบ การเชื่อมต่อเครื่องคอมพิวเตอร์ในเครือข่าย การโอนย้ายข้อมูล การแสดงสถานะที่ใช้ในการเชื่อมต่อข้อมูลระหว่างเครื่องต้นทางและเครื่องปลายทางที่มีความแตกต่างกัน ให้สามารถติดต่อสื่อสารทำงานร่วมกันได้อย่างถูกต้องและมีประสิทธิภาพ

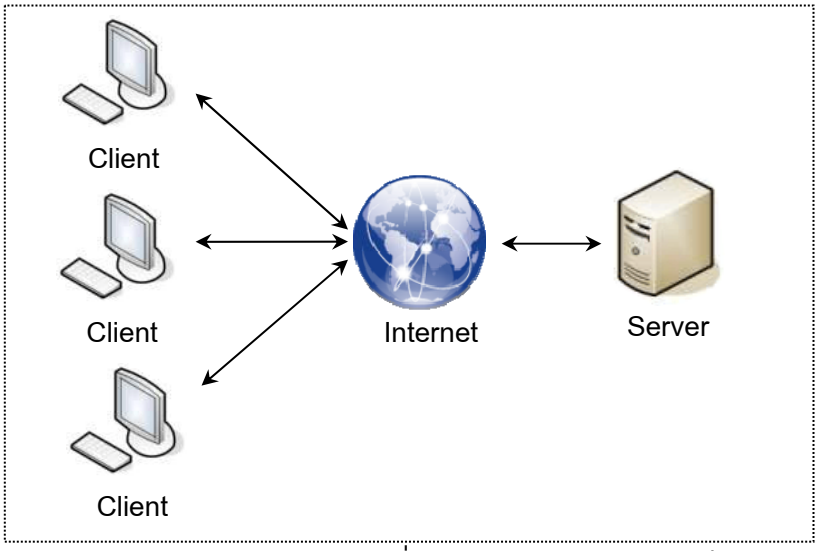
2.1.2 สถาปัตยกรรมไคลเอนต์ – เซิร์ฟเวอร์ (Client – Server)

ไคลเอนต์ – เซิร์ฟเวอร์ (Client – Server) (Gallaugher, 1996) เป็นสถาปัตยกรรมหนึ่งทางด้านคอมพิวเตอร์ที่ใช้อย่างแพร่หลายในเครือข่ายอินเตอร์เน็ตในปัจจุบัน โดยมีเซิร์ฟเวอร์ทำหน้าที่หลักในการจัดการข้อมูลและทรัพยากรต่างๆ ให้กับไคลเอนต์ เพื่อให้เกิดการใช้ข้อมูลและทรัพยากรร่วมกันระหว่างเครื่องคอมพิวเตอร์ในระบบเครือข่าย ซึ่งเซิร์ฟเวอร์สามารถรองรับการใช้งานของหลายๆ ไคลเอนต์พร้อมกันได้ในเวลาเดียวกัน

การทำงานของไคลเอนต์ – เซิร์ฟเวอร์จะเริ่มต้นขึ้นเมื่อผู้รับบริการหรือฝั่งไคลเอนต์ต้องการข้อมูลหรือผลลัพธ์จากการคำนวณ ผู้รับบริการจะส่งคำร้องขอไปยังผู้ให้บริการหรือฝั่งเซิร์ฟเวอร์ เมื่อผู้ให้บริการได้รับคำร้องก็จะดำเนินการตามคำร้องขอของผู้รับบริการเพื่อหาผลลัพธ์ที่ต้องการ จากนั้นผู้ให้บริการจะส่งผลลัพธ์ที่ได้กลับไปยังผู้รับบริการ ซึ่งจะทำงานรับส่งข้อมูลกันในลักษณะเช่นนี้บนเครือข่าย การติดต่อสื่อสารระหว่างไคลเอนต์ – เซิร์ฟเวอร์และการติดต่อสื่อสารบนเครือข่ายอินเตอร์เน็ตแสดงดังภาพประกอบ 2.1 และ 2.2 ตามลำดับ



ภาพประกอบ 2.1 การติดต่อสื่อสารระหว่างไคลเอนต์ – เซิร์ฟเวอร์



ภาพประกอบ 2.2 การติดต่อสื่อสารบนเครือข่ายอินเตอร์เน็ต

2.2 เทคโนโลยี Push และ Pull (Push – Pull Technology)

เทคโนโลยี Push และ Pull เป็นรูปแบบการติดต่อสื่อสารกันผ่านระบบอินเตอร์เน็ต โดยแบ่งการทำงานออกเป็น 2 ฝั่ง คือ ฝั่งเครื่องคอมพิวเตอร์ที่เป็นผู้รับบริการหรือไคลเอนต์ (Client) และฝั่งเครื่องคอมพิวเตอร์ที่เป็นผู้ให้บริการหรือเซิร์ฟเวอร์ (Server) เทคโนโลยี Push และ Pull จะมีลักษณะการส่งข้อมูลที่แตกต่างกันดังนี้

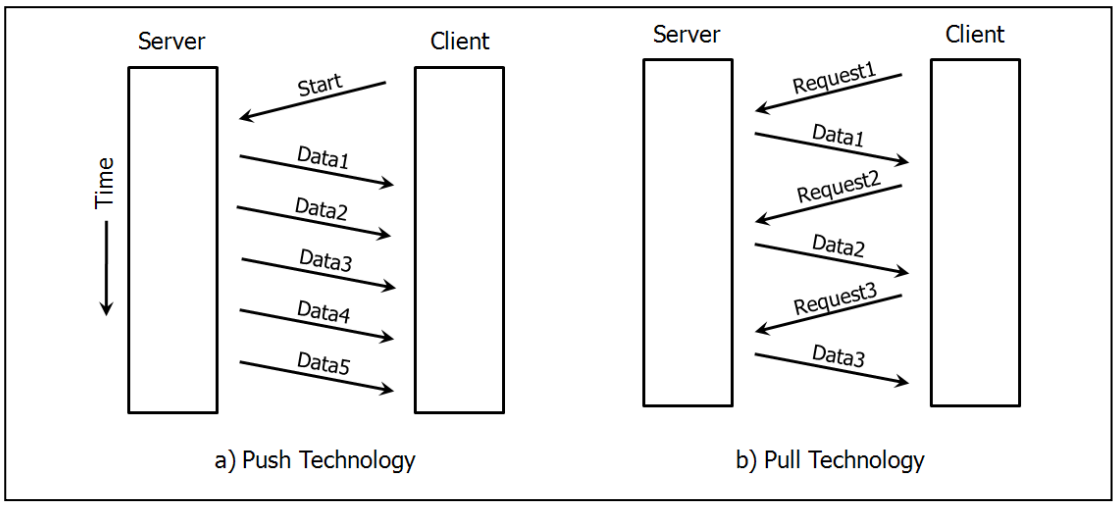
2.2.1 เทคโนโลยี Push (Push Technology)

เป็นเทคโนโลยีที่ใช้ส่งข้อมูลและสารสนเทศไปบนอินเตอร์เน็ตอย่างอัตโนมัติ โดยที่ผู้รับบริการไม่จำเป็นต้องค้นหาหรือดาวน์โหลดข้อมูล (Umbach, 1997) โดยหลักการทำงานผู้รับบริการจะต้องทำการสมัครสมาชิกกับแหล่งข้อมูลที่ต้องการรับข้อมูลข่าวสารก่อน จากนั้นเมื่อแหล่งข้อมูลมีข้อมูลใหม่เซิร์ฟเวอร์จะทำการส่งข้อมูลให้กับผู้รับบริการอัตโนมัติ ส่วนใหญ่จะเป็นเว็บไซต์ประเภทข่าวสาร เช่น BBC, CNN, REUTERS เป็นต้น โดยอีเมล์ถือเป็นเทคโนโลยี Push รูปแบบหนึ่งเช่นกัน เนื่องจากผู้รับบริการจะต้องสมัครสมาชิกก่อนเพื่อรับอีเมล์แอดเดรส จากนั้นเมื่อมีจดหมายเข้ามาเซิร์ฟเวอร์จะทำการส่งจดหมายให้กับผู้รับบริการอัตโนมัติโดยผู้รับบริการไม่ต้องร้องขอ

2.2.2 เทคโนโลยี Pull (Pull Technology)

เป็นเทคโนโลยีที่ดึงข้อมูลโดยผู้รับบริการ เป็นการติดต่อสื่อสารกันผ่านระบบเครือข่ายมีลักษณะการสื่อสารที่ผู้รับบริการร้องขอข้อมูลและเครื่องผู้ให้บริการจะทำการส่งข้อมูลให้ (Miryam, 1997) ซึ่งถูกใช้อย่างกว้างขวางในอินเตอร์เน็ตสำหรับการร้องขอหน้า HTTP จากเว็บไซต์โดยผู้รับบริการ เว็บฟีดส่วนใหญ่ เช่น RSS จึงเป็นเทคโนโลยี Pull รูปแบบหนึ่งซึ่งดึงข้อมูลโดยผู้รับบริการผ่าน RSS Reader แต่เนื่องจาก RSS Reader จะทำการร้องขอข้อมูลจากเซิร์ฟเวอร์เป็นระยะเพื่อตรวจสอบข้อมูลใหม่ จึงทำให้การร้องขอข้อมูลเป็นระยะของ RSS Reader ไม่มีประสิทธิภาพเพราะมีปัญหาของการใช้แบนด์วิดธ์ (Bandwidth) มากเกินไป

ลักษณะการส่งข้อมูลของเทคโนโลยี Push และ Pull แสดงดังภาพประกอบ 2.3



ภาพประกอบ 2.3 ลักษณะการส่งข้อมูลของเทคโนโลยี Push และ Pull

2.3 ภาษา XML (Extensible Markup Language)

ภาษา XML (W3C, 2009) เป็นส่วนหนึ่งของภาษา HTML เป็นภาษามาร์กอัปสำหรับใช้งานทั่วไป ภาษา XML ได้ถูกกำหนดให้เป็นมาตรฐานโดย W3C (World Wide Web Consortium) ซึ่งใช้ในการแลกเปลี่ยนข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่แตกต่างกัน และเน้นการแลกเปลี่ยนข้อมูลผ่านอินเตอร์เน็ต เป็นภาษาที่ใช้เน้นส่วนที่เป็นข้อมูล โดยสามารถกำหนดชื่อแท็ก (Tag name) และชื่อแอตทริบิวต์ (Attribute name) ได้ตามความต้องการของผู้สร้างเอกสาร XML จึงเป็นแค่ไฟล์ข้อความ (Text file) ชนิดหนึ่ง ที่มีแท็กเปิดและแท็กปิดครอบข้อมูลไว้ตรงกลางเท่านั้น ทำให้เอกสาร XML ถูกใช้อย่างแพร่หลายเนื่องจากความง่ายในการสร้างเอกสาร

2.3.1 องค์ประกอบของภาษา XML

ส่วนสำคัญของภาษา XML ประกอบไปด้วยแท็ก (Tag) อิลิเมนต์ (Element) และแอททริบิวต์ (Attribute) ซึ่งผู้ใช้สามารถกำหนดขึ้นมาเองและให้รายละเอียดต่างๆของข้อมูลได้ ทำให้ง่ายละสะดวกในการใช้งาน เป็นการปรับปรุงมาจากภาษา HTML ที่ไม่สามารถกำหนดแท็กขึ้นมาใช้เองได้

1) แท็ก(Tag) และอิลิเมนต์ (Element)

สำหรับในภาษา XML แล้วแท็กมีความหมายในลักษณะเดียวกับแท็กที่ใช้ในภาษา HTML เป็นข้อความที่อยู่ระหว่างสัญลักษณ์ “<” และ “>” มี 2 แบบคือ แท็กเริ่มต้น(Start Tag) จะอยู่ภายในเครื่องหมาย “<” และ “>” เช่น <item> และแท็กปิด (End Tag) จะกำหนดชื่อ

ของแท็กอยู่ภายในเครื่องหมาย “</” และ “>” เช่น </item> โดยจะมีเครื่องหมาย “/” แทรกอยู่ด้านหน้า และตั้งแต่แท็กเริ่มต้นไปจนถึงแท็กสิ้นสุดจะถูกเรียกว่า “อิลิเมนต์ (Element)” หมายถึงส่วนของข้อมูลที่ประกอบด้วยแท็ก

```
<item>Example</item>
```

ภาพประกอบ 2.4 ตัวอย่างอิลิเมนต์ของ XML

จากภาพประกอบ 2.4

<item>	คือ แท็กเริ่มต้น
</item>	คือ แท็กสิ้นสุด
<item>Example</item>	คือ อิลิเมนต์

2) แอททริบิวต์ (Attribute)

แอททริบิวต์ คือ การระบุคุณสมบัติให้กับอิลิเมนต์ ใช้เพื่ออธิบายส่วนเพิ่มเติมให้กับแต่ละอิลิเมนต์ โดยมีรูปแบบการกำหนดค่าดังนี้

```
<guid isPermaLink = “false”>http://www.bbb.co.uk/news/business-12935228</guid>
หรือ
<guid isPermaLink = ‘false’>http://www.bbb.co.uk/news/business-12935228</guid>
```

ภาพประกอบ 2.5 ตัวอย่างการกำหนดค่าแอททริบิวต์

จากภาพประกอบ 2.5 จะเห็นว่ามี isPermaLink = “false” เพิ่มขึ้นมา ทั้งหมดนี้เรียกว่าแอททริบิวต์ โดย isPermaLink นั้นเป็นชื่อของแอททริบิวต์ ส่วน “false” เรียกว่าค่าของแอททริบิวต์ (Attribute value) นั้นๆ โดยค่าของแอททริบิวต์จะต้องเขียนอยู่ภายในอัญประกาศคู่ (Double quotes: “ ”) หรืออัญประกาศเดี่ยว (Single quotes: ‘ ’) เสมอ

2.3.2 การตรวจสอบความถูกต้องของภาษา XML (Well – Formed XML)

Well – Formed XML เป็นการตรวจสอบเอกสาร XML ว่าเขียนได้ถูกหรือไม่ โดยเอกสารที่ Well – Formed แล้วจะสามารถนำไปใช้ได้ เอกสารที่เป็น Well – Formed มีลักษณะดังนี้

1) เอกสาร XML จะต้องมีอิลิเมนต์ราก (Root Element) และมีได้เพียงหนึ่งรากเท่านั้น โดยเป็นแท็กที่อยู่บนสุดตามหลังส่วนของการประกาศ XML

2) ทุกอิลิเมนต์ของ XML จะต้องประกอบด้วยแท็กเริ่มต้นและแท็กสิ้นสุด โดยทั้งสองแท็กจะต้องมีชื่อเหมือนกัน เช่น <item>…</item>

3) การกำหนดชื่อแท็กจะคำนึงถึง Case Sensitive คือ ตัวอักษรพิมพ์ใหญ่และพิมพ์เล็กมีความหมายแตกต่างกัน เช่น <item> กับ <Item> ถือเป็นแท็กที่ต่างกัน

4) อิลิเมนต์ของ XML ต้องซ้อนกันอย่างเป็นลำดับโดยไม่สามารถสลับตำแหน่งของแท็กปิดได้ เช่น <item><title>…</title></item> เป็นต้น ห้ามมีการซ้อนแท็กกัน เช่น <item><title>…</item></title> ถือเป็นการซ้อนแท็กอย่างไม่เป็นลำดับ

5) แท็กที่ไม่มีข้อมูลหรือแท็กว่างสามารถเขียนได้ 2 ลักษณะ คือ <item></item> หรือ <item/>

6) การตั้งชื่ออิลิเมนต์ของเอกสาร XML สามารถใช้อักขระ ตัวเลขและอักขระพิเศษได้ ยกเว้นเครื่องหมาย “&” และไม่สามารถใช้ตัวเลขหรือตัวอักขระพิเศษนำหน้าชื่อของอิลิเมนต์ นอกจากนี้ยังห้ามเว้นช่องว่างระหว่างชื่ออิลิเมนต์อีกด้วย เอกสารที่จำเป็นต้องใช้ตัวอักขระพิเศษจะใช้แทนด้วย

&lt;	แทน	<
&amp;	แทน	&
&gt;	แทน	>
&quot;	แทน	“
&apos;	แทน	‘

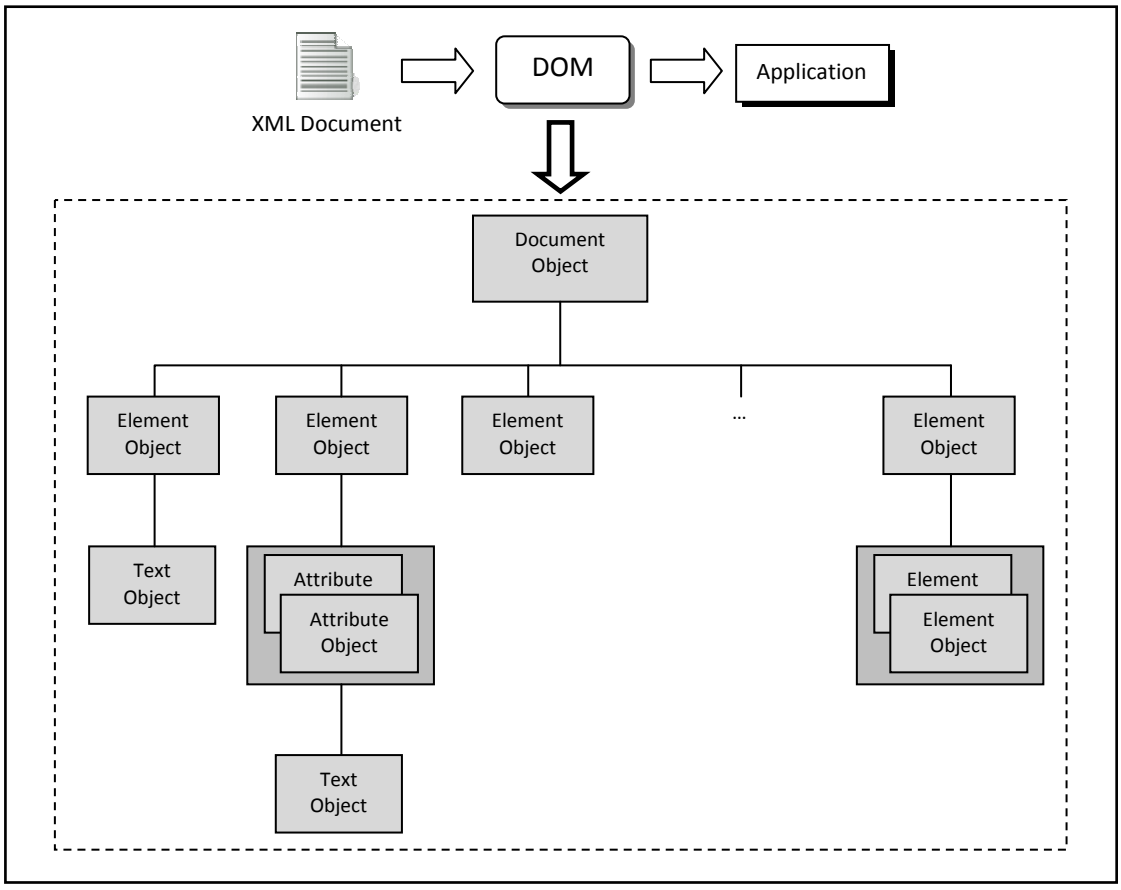
2.3.3 ตัวแปลภาษา XML (XML Parser)

ตัวแปลภาษา XML ทำหน้าที่ในการแปลความหมาย ตรวจสอบความถูกต้องของเอกสาร รวมถึงวิเคราะห์โครงสร้างของภาษา XML ด้วย ในการเข้าถึงเอกสาร XML นั้นตัวแปลภาษา XML จะทำหน้าที่เป็นตัวกลางระหว่างเอกสาร XML และโปรแกรมประยุกต์ที่ต้องการใช้งานข้อมูลของ XML โดยตัวแปลภาษา XML ที่จำแนกตามวิธีการสำรวจเนื้อหาเอกสาร จะแบ่งออกได้เป็น 2 รูปแบบ คือ Tree – based Parser และ Event – based Parser

1) DOM (Document Object Model)

DOM ได้รับการรับรองเป็นมาตรฐานโดย W3C เป็นการแปลงเอกสาร XML ให้อยู่ในรูปของโครงสร้างต้นไม้ วิธีการคือจะอ่านข้อมูลจากแฟ้ม XML ขึ้นมาทั้งหมด แล้วจัดองค์ประกอบต่าง ๆให้อยู่ในรูปแบบต้นไม้ เก็บไว้ในหน่วยความจำของคอมพิวเตอร์ที่ประกอบด้วยอิลิเมนต์และแอททริบิวต์ต่างๆ โดย DOM สามารถอ่าน และจัดการเพิ่ม ลบ หรือแก้ไขข้อมูลในเอกสาร XML ได้ ในการเข้าถึงข้อมูลจะใช้วิธีการเดินเข้าถึงต้นไม้ (Traverse)

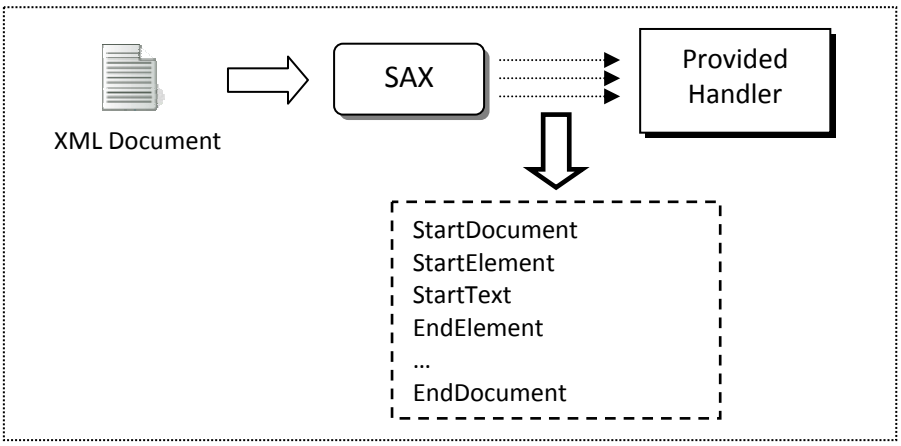
โดยเริ่มต้นเข้าถึงจากโหนดรากของโครงสร้างต้นไม้และจะมองเอกสารเป็นอ็อบเจ็กต์ ถือได้ว่า DOM เป็น Tree – based Parser รูปแบบหนึ่ง การทำงานของ DOM แสดงดังภาพประกอบ 2.6



ภาพประกอบ 2.6 การทำงานของ DOM

2) SAX (Simple API for XML)

SAX เป็นอีกรูปแบบหนึ่งของกระบวนการดึงข้อมูล ซึ่งจะอ่านเอกสาร XML และตอบสนองต่อสิ่งที่อ่านได้โดยถือเป็นเสมือนเหตุการณ์ หลักการทำงานของ SAX จะต่างกับ DOM คือจะไม่โหลดข้อมูลทั้งหมดในเอกสาร XML เข้ามาในหน่วยความจำ แต่จะแปลความหมายของเหตุการณ์ที่เกิดขึ้นเป็นหลัก ขึ้นอยู่กับเหตุการณ์ต่าง ๆที่เกิดขึ้น เมื่อ Parser อ่านข้อมูลจากเอกสาร XML ในแต่ละครั้งจะจดจำโครงสร้างไวยากรณ์ของเอกสาร XML ไว้ และตอบสนองต่อเหตุการณ์ที่อ่านได้ด้วยวิธีการที่กำหนด ถือได้ว่า SAX เป็น Event – based Parser รูปแบบหนึ่ง การทำงานของ SAX แสดงดังภาพประกอบ 2.7



ภาพประกอบ 2.7 การทำงานของ SAX

ตารางที่ 2.1 เปรียบเทียบการทำงานระหว่าง DOM และ SAX

การทำงาน	DOM	SAX
วิธีการสำรวจข้อมูล	Tree-based Parser	Event-based Parser
วิธีการอ่านข้อมูล	อ่านทั้งหมดเพียงครั้งเดียว เก็บไว้ในหน่วยความจำ ทำให้สามารถใช้งานได้ตลอด โดยไม่ต้องอ่านข้อมูลซ้ำ เหมาะกับการใช้งานในลักษณะที่ต้องใช้ข้อมูลบ่อยๆ	อ่านข้อมูลทีละชุด เมื่อต้องการข้อมูลชุดใหม่ทำให้ต้องอ่านข้อมูลซ้ำอีก
วิธีการดึงข้อมูล	ต้องอ่านเอกสารทั้งหมดก่อน ถึงจะดึงข้อมูลได้	สามารถดึงข้อมูลเฉพาะที่ต้องการได้
วิธีการเข้าถึงข้อมูล	เข้าถึงแบบสุ่ม (Random Access)	เข้าถึงแบบ Sequential เท่านั้น
การใช้หน่วยความจำ	ใช้หน่วยความจำค่อนข้างมากเพราะต้องอ่านเอกสารทั้งหมดเก็บไว้เป็นโครงสร้างต้นไม้ในหน่วยความจำ	ไม่มีการโหลดข้อมูลทั้งหมดเข้าในหน่วยความจำ
การจัดการข้อมูล	สามารถแก้ไขและเปลี่ยนแปลงโครงสร้างเอกสาร XML ได้	อ่านได้อย่างเดียว
ความเร็วในการเข้าถึงข้อมูล	การเรียกใช้งานครั้งแรกจะช้า หลังจากนั้นการเข้าถึงจุดต่างๆ จะเร็วขึ้นเพราะข้อมูลถูกเก็บอยู่ในหน่วยความจำไว้แล้ว	ทำงานเร็วกว่า DOM ในการเข้าถึงชุดข้อมูลที่ต้องการ
โครงสร้างเอกสาร XML	ทราบรายละเอียดและโครงสร้างของเอกสาร XML ทั้งหมด	ไม่ทราบรายละเอียดของโครงสร้างเอกสาร XML ทั้งหมด

2.4 เทคโนโลยี RSS (Really Simple Syndication)

RSS (Finkelstein, 2005) เป็นข้อมูลที่อยู่ในรูปของภาษา XML พัฒนาขึ้นเพื่อรวบรวมและเผยแพร่เนื้อหาหรือข่าวสารของเว็บไซต์ที่มีการเปลี่ยนแปลงบ่อย โดยผู้รับบริการสามารถรับข่าวสารจากเว็บไซต์ที่ให้บริการ RSS ได้ด้วยการใช้โปรแกรมรวบรวมข่าวสารที่เรียกว่า Reader หรือ Aggregator ซึ่งมีหลักการทำงานคล้ายกับโปรแกรมรับอีเมล์ เพียงแต่ผู้รับบริการต้องลงทะเบียนรับข่าวสารที่สนใจจากตัวรวบรวมข่าวสาร โดยตัวรวบรวมข่าวสารจะรวบรวมเอกสาร RSS จากเว็บไซต์ต่างๆ และตรวจสอบการเปลี่ยนแปลงข้อมูล แล้วแสดงผลข้อมูลล่าสุดให้อัตโนมัติ ซึ่งข้อมูลที่ได้จะเป็นเพียงหัวข้อข่าว หรือรายละเอียดโดยย่อเท่านั้น ส่วนเนื้อหา หรือข้อความหลักของข่าวนั้นจะมีลิงค์เชื่อมโยงให้อีกทีหนึ่ง

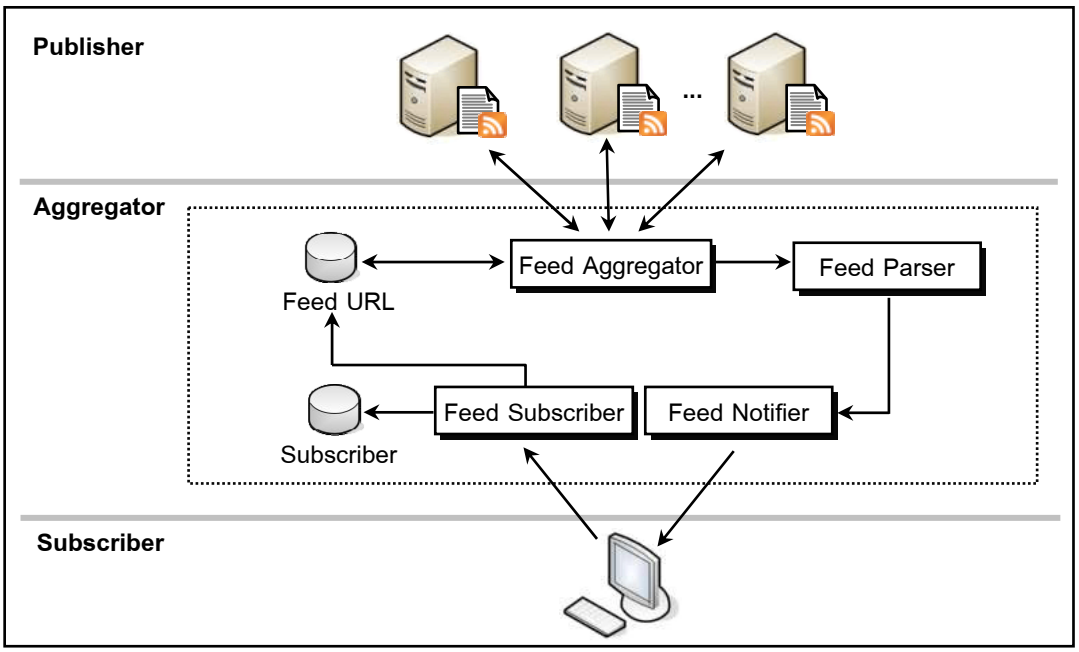
2.4.1 โครงสร้างการทำงานของ RSS

โครงสร้างการทำงานของ RSS แสดงดังภาพประกอบ 2.8 ประกอบด้วย 3 ส่วน

1. ผู้เผยแพร่ข้อมูลข่าวสาร (Publisher) คือ เว็บไซต์ที่ให้บริการข้อมูลข่าวสารในรูปแบบเอกสาร RSS

2. ตัวรวบรวมข่าวสาร (Aggregator) ทำหน้าที่เป็นตัวแทนรวบรวมข้อมูลข่าวสารจากเว็บไซต์ต่างๆ

3. ผู้รับบริการข้อมูลข่าวสาร (Subscriber) คือ ผู้รับบริการข้อมูลข่าวสารจากเว็บไซต์ต่างๆ



ภาพประกอบ 2.8 โครงสร้างการทำงานของ RSS

จากภาพประกอบ 2.8 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

1) ผู้รับบริการ (Subscriber) จะลงทะเบียนรับข้อมูลข่าวสารไปยัง Feed Subscriber

2) ข้อมูลของผู้รับบริการจะถูกบันทึกลงฐานข้อมูล Subscriber และเก็บข้อมูล URL ของเอกสาร RSS ที่ผู้รับบริการต้องการเก็บไว้ในฐานข้อมูล Feed URL ซึ่งตัวรวบรวมข่าวสาร (Aggregator) จะเก็บข้อมูลของ URL เอาไว้

3) เมื่อผู้รับบริการเปิดอ่านข้อมูลข่าวสาร Feed Aggregator จะทำการรวบรวมเอกสาร RSS จากเว็บไซต์ ต่างๆ ตาม URL ที่ผู้รับบริการได้ระบุไว้ และส่งไปยัง Feed Parser

4) Feed Parser จะทำการวิเคราะห์โครงสร้างของเอกสาร RSS และจัดรูปแบบผลลัพธ์ ส่งไปยัง Feed Notifier

5) Feed Notifier แสดงผลลัพธ์ที่ได้ไปยังผู้รับบริการ

2.4.2 การสร้างเอกสาร RSS

ในการสร้างเอกสาร RSS ผู้เผยแพร่ข่าวสารต้องระบุเนื้อหาของเว็บไซต์ ที่ต้องการเผยแพร่ให้บุคคลทั่วไปทราบโดยมากจะเป็นเนื้อหาที่มีการเปลี่ยนแปลงบ่อยๆ สำหรับเอกสาร RSS ส่วนมากจะแสดงเนื้อหาเพียงแค่บางส่วน แต่จะระบุลิงค์เชื่อมโยงไปยังเนื้อหาของข้อมูลข่าวสารทั้งหมดเอาไว้ เพื่อให้ผู้ที่สนใจในรายละเอียดของข้อมูลนั้นๆ เข้าไปอ่าน โดยรูปแบบของเอกสาร RSS แสดงดังภาพประกอบ 2.9

```
<?xml version = “1.0”  ?>
<rss version = “2.0” >
    <channel>
        <title>…</title>
        <link>…</link>
        <description>…</description>
        <item>
            <title>…</title>
            <link>…</link>
            <description>…</description>
            <pubDate>…</pubDate>
            ...
        </item>
        …
    </channel>
</rss
```

ภาพประกอบ 2.9 รูปแบบเอกสาร RSS

จากภาพประกอบ 2.9 แสดงรูปแบบของเอกสาร RSS 2.0 โดยมีแท็ก <rss> บอกจุดเริ่มต้น ตามด้วยแท็ก <channel> เก็บข้อมูลต่างๆ ของ RSS ไว้ และมีแท็ก <item> เป็นส่วนสำคัญทำหน้าที่เก็บรายการข้อมูล ซึ่งประกอบด้วยแท็กต่างๆ ที่อยู่ภายใน เพื่อบอกรายละเอียดข้อมูลแต่ละรายการ โดยตัวอย่างและคำอธิบายแท็กภายในแท็ก <channel> และแท็ก <item> แสดงดังตารางที่ 2.1 และตารางที่ 2.2 นอกจากนี้ในแต่ละแท็กยังสามารถกำหนดแอททริบิวต์ (Attribute) เพื่ออธิบายข้อมูลเพิ่มเติมได้อีกด้วย โดยการสร้างเอกสาร RSS มีขั้นตอนดังต่อไปนี้

1) ประกาศการกำหนดเป็นเอกสาร XML ซึ่งจะเป็นบรรทัดแรกสุดของเอกสาร RSS

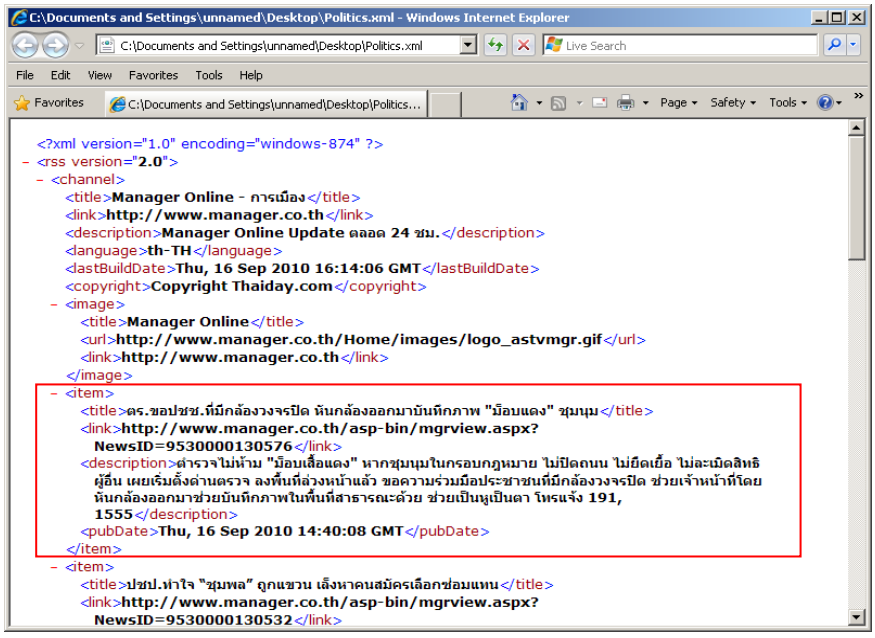
2) ขั้นตอนต่อไปจะทำการเปิดแท็ก <rss> และแท็ก <channel> เพื่อจะใส่ข้อมูล RSS ฟีดลงไป

3) ในขั้นตอนนี้จะใส่ข้อมูลต่างๆ ลงในแท็ก ซึ่งเนื้อหาในส่วนนี้จะอยู่ในแท็ก <channel> เช่น หัวข้อ, ลิงค์เชื่อมโยง, คำอธิบาย ซึ่งจะอยู่ในแท็ก <title>, <link>, <description> เป็นต้น

4) ขั้นตอนต่อไปจะแจกแจงรายละเอียดของแต่ละเอกสาร RSS ซึ่งจะอยู่ภายใต้แท็ก <item> ส่วนที่สำคัญจะประกอบไปด้วย หัวข้อ, ลิงค์เชื่อมโยง, คำอธิบาย และวันเวลาที่แสดงข้อมูล ซึ่งจะอยู่ในแท็ก <title>, <link>, <description>, <pubDate> เป็นต้น

5) ปิดแท็ก </channel> และ </rss>

6) ตรวจสอบความถูกต้องของข้อมูล



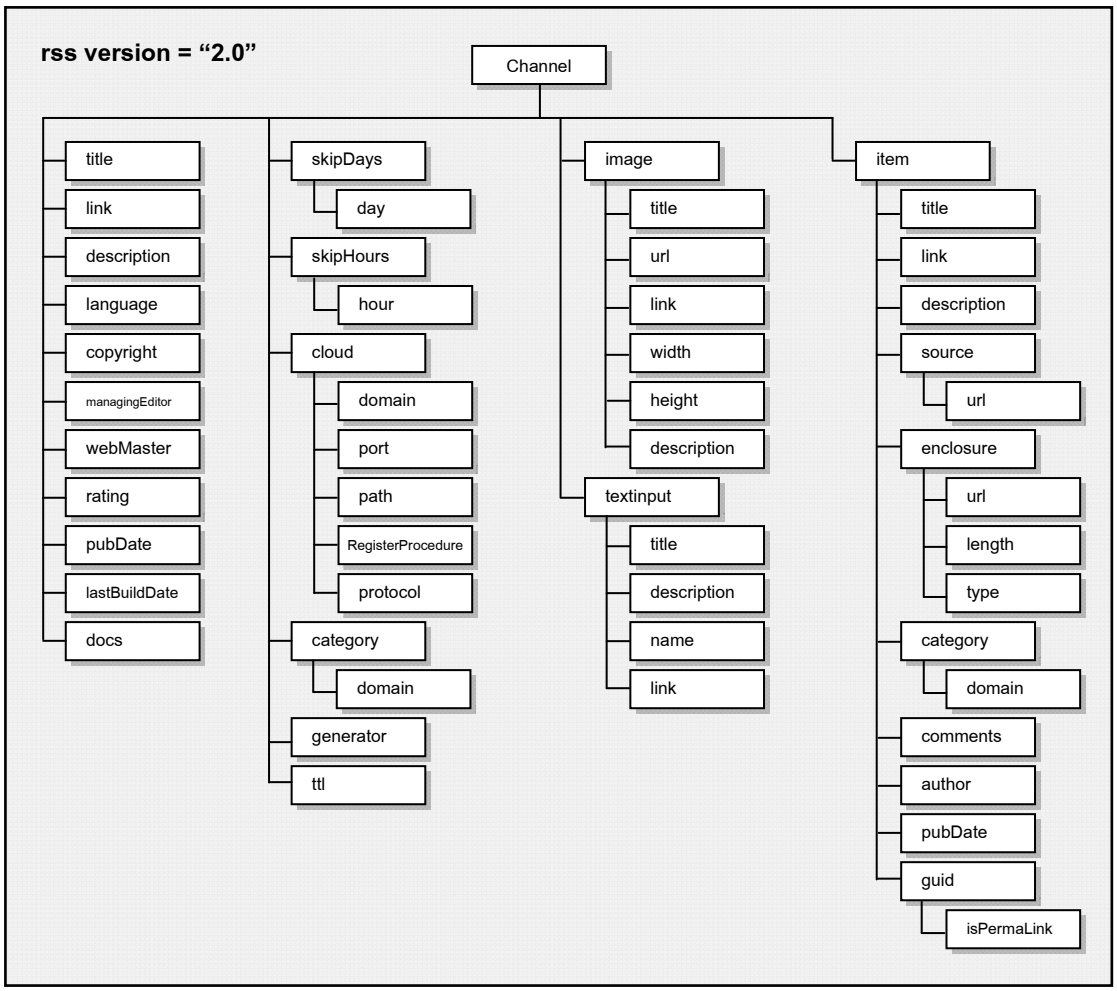
ภาพประกอบ 2.10 ตัวอย่างเอกสาร RSS

ตารางที่ 2.2 แท็กย่อยภายในแท็ก <channel> ของเอกสาร RSS

แท็ก	คำอธิบาย
<title>	หัวเรื่องของ Channel
<link>	ลิงค์เชื่อมโยงไปยังข้อมูลหลัก
<description>	คำอธิบายโดยย่อของเอกสาร RSS
<language>	ภาษาของข้อมูลภายในเอกสาร RSS
<copyright>	ข้อมูลลิขสิทธิ์
<managingEditor>	ที่อยู่อีเมล์ไปยังบรรณาธิการผู้ดูแลเนื้อหาของเอกสาร RSS
<webMaster>	ที่อยู่ของผู้พัฒนาเว็บไซต์ที่เผยแพร่ข้อมูลข่าวสาร
<rating>	ระบุการจัดลำดับความนิยมของข้อมูลในเอกสาร RSS
<pubDate>	วันเวลาที่ทำการเผยแพร่ข้อมูล
<lastBuilDate>	วันเวลาล่าสุดที่ทำการปรับปรุงข้อมูลภายในเอกสาร RSS
<docs>	ระบุ URL ของเอกสารที่ใช้กำหนดรูปแบบเอกสาร RSS
<skipDays>	ระบุวันเพื่อ RSS Reader จะได้ไม่ต้องตรวจสอบการอัพเดทเนื้อหา
<skipHours>	ระบุเวลาที่ RSS Reader ไม่ต้องตรวจสอบการอัพเดทเนื้อหา
<cloud>	ขั้นตอนการลงทะเบียนเมื่อผู้เผยแพร่มีการอัพเดทข้อมูลใหม่
<category>	กำหนดหมวดหมู่ให้กับข้อมูลภายในเอกสาร RSS
<generator>	ระบุโปรแกรมที่ใช้แสดงเอกสาร RSS
<ttl>	ช่วงระยะเวลาที่ข้อมูลยังคงใช้ได้ ก่อนจะมีการเปลี่ยนแปลงใหม่
<image>	รูปภาพอธิบายเอกสาร RSS
<textinput>	แถบข้อความเข้าที่แสดงในเอกสาร RSS
<item>	รายการข้อมูล

ตารางที่ 2.3 แท็กย่อยภายในแท็ก <item> ของเอกสาร RSS

แท็ก	คำอธิบาย
<title>	หัวเรื่องรายการข้อมูล
<link>	ลิงค์เชื่อมโยงไปยังข้อมูลหลัก
<description>	รายละเอียดข้อมูลโดยย่อ
<source>	แหล่งที่มาของรายการข้อมูล
<enclosure>	ไฟล์มัลติมีเดียที่แนบมาพร้อมกับรายการข้อมูล
<category>	ประเภทของข้อมูล
<comments>	ลิงค์ไปยังคำวิจารณ์เกี่ยวกับรายการข้อมูล
<author>	ข้อมูลผู้ประกาศ
<pubDate>	วันเวลาที่เผยแพร่ข้อมูล
<guid>	globally unique identifier ลิงค์สำหรับระบุแต่ละรายการข้อมูล



ภาพประกอบ 2.11 โครงสร้างแท็กต่างๆ ของเอกสาร RSS 2.0

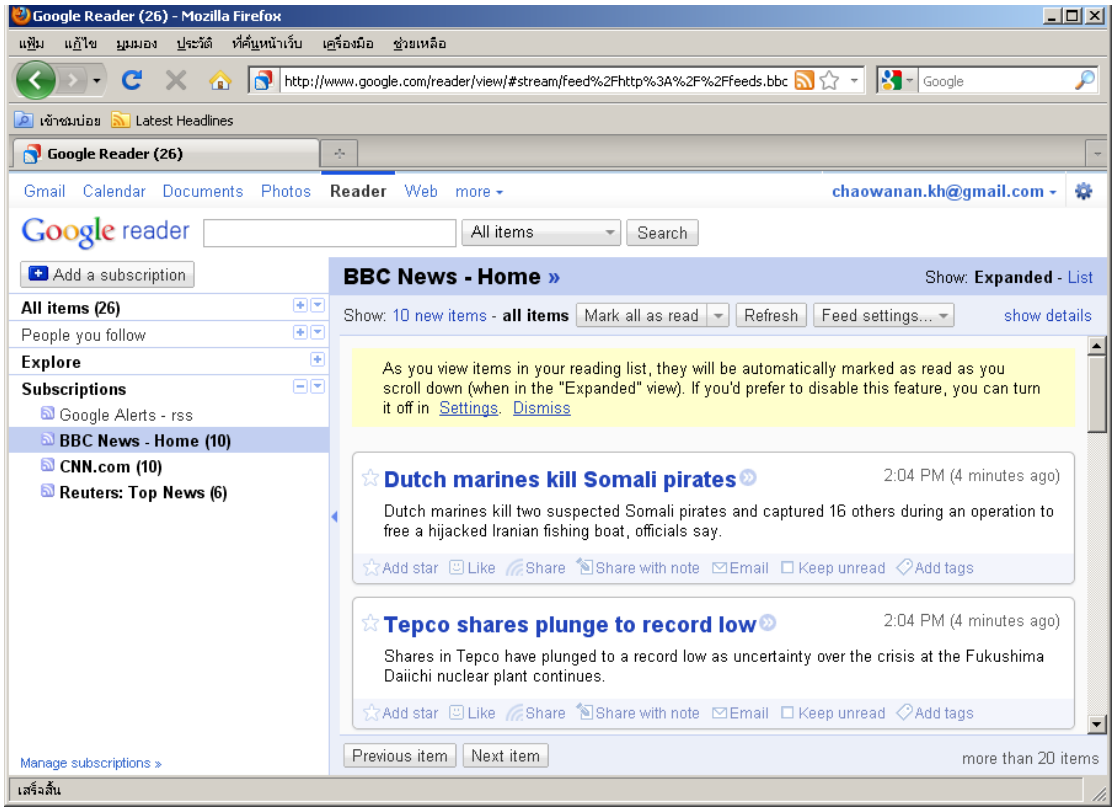
2.4.3 การรับเอกสาร RSS

ในการรับเอกสาร RSS นั้นผู้รับบริการจะต้องทราบแหล่งข้อมูลที่ให้บริการเอกสาร RSS โดยสังเกตได้จากสัญลักษณ์ดังแสดงในภาพประกอบ 2.12 ซึ่งส่วนมากจะเป็นแหล่งข้อมูลที่ข้อมูลจะมีการเปลี่ยนแปลงบ่อยๆ ได้แก่ เว็บไซต์ที่ให้บริการประเภทข่าวสารต่างๆ เว็บล็อก หรือเว็บบอร์ดต่างๆ เป็นต้น

ภาพประกอบ 2.12 สัญลักษณ์ RSS ที่ปรากฏในหน้าเว็บไซต์ต่างๆ

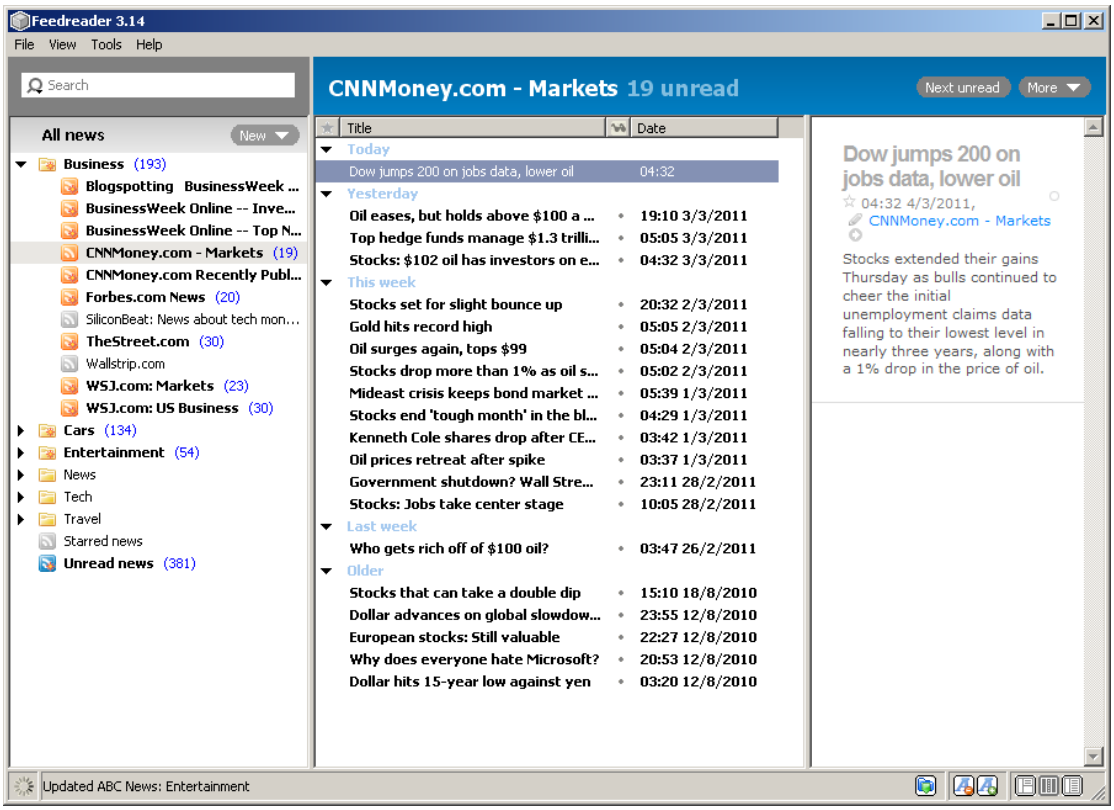
เมื่อได้แหล่งข้อมูลที่จะรับเอกสาร RSS แล้ว ขั้นตอนต่อไปจะเป็นวิธีการรับเอกสาร RSS ซึ่งผู้รับบริการสามารถรับข้อมูลได้โดยใช้ Feed Reader หรือเรียกอีกอย่างว่าตัวรวบรวมข่าวสาร (Aggregator) คือ สิ่งที่ใช้ในการอ่านเอกสาร RSS สามารถจำแนกออกได้เป็น 2 ประเภทใหญ่ๆ คือ

1) Web – based reader เป็น Feed Reader ที่ไม่จำเป็นต้องติดตั้งโปรแกรมเพียงแค่ใช้เบราเซอร์เปิดไปที่เว็บไซต์ที่ให้บริการซึ่งข้อดีของแบบ Web – based reader คือสะดวกในการใช้งานโดยผู้รับบริการไม่จำเป็นต้องติดตั้งโปรแกรมเพิ่มเติมอีก แต่มีข้อเสียคือในการเข้าใช้งานในครั้งแรกจำเป็นต้องสมัครสมาชิกก่อน จากนั้นในการใช้งานครั้งต่อไปต้องล็อกอินเข้าใช้งานทุกครั้ง การใช้งาน Feed Reader แบบ Web – based reader แสดงดังภาพประกอบ 2.13



ภาพประกอบ 2.13 การใช้งาน Feed Reader แบบ Web – based reader

2) Software reader เป็น Feed Reader ที่จำเป็นจะต้องติดตั้งโปรแกรมลงในฝั่งของผู้รับบริการก่อนจึงจะสามารถใช้งานได้ ข้อดีของแบบ Software reader คือ ในบางโปรแกรมนั้นสามารถอ่านเอกสาร RSS แบบออฟไลน์ได้ การใช้งาน Feed Reader แบบ Software reader แสดงดังภาพประกอบ 2.14



ภาพประกอบ 2.14 การใช้งาน Feed Reader แบบ Software reader

ในการรับเอกสาร RSS ผู้รับบริการจะต้องลงทะเบียนรับข้อมูลข่าวสารก่อน โดยใส่ Feed URL ที่อยู่ของเอกสาร RSS ที่ต้องการลงใน RSS Reader เมื่อ Feed URL ที่ต้องการสามารถใส่ลงใน RSS Reader ได้เรียบร้อยแล้ว จะเห็นข้อมูลข่าวสารของ Feed URL นั้นปรากฏใน RSS Reader

2.4.4 ประโยชน์ของ RSS

1) สามารถรับข้อมูลข่าวสารที่สนใจได้จากหลายๆ เว็บไซต์ด้วยการเข้าใช้งานที่ใดที่หนึ่งเท่านั้น ทำให้ลดเวลาในการเข้าถึงข้อมูลข่าวสาร

2) สามารถรับข้อมูลข่าวสารได้เมื่อต้องการ เพียงเข้าไปยัง RSS reader ทำให้ไม่ต้องเสียเวลารอให้มีการส่งข้อมูลข่าวสารมายังผู้รับบริการ

3) สามารถรับข้อมูลข่าวสารเฉพาะเรื่องที่สนใจได้ เพราะ RSS แสดงหัวข้อของข้อมูลและรายละเอียดย่อๆ ทำให้ผู้ใช้สามารถเลือกอ่านเฉพาะเรื่องที่สนใจได้

4) ผู้รับบริการสามารถใช้ RSS เพื่อรวบรวมข้อมูลข่าวสารจากเว็บไซต์ต่างๆ แล้วนำมาเผยแพร่ในเว็บไซต์ที่สร้างขึ้น โดยไม่เกิดปัญหาในเรื่องการละเมิดลิขสิทธิ์แต่อย่างใด

2.5 การปรับปรุงข้อมูล

การปรับปรุงข้อมูลเป็นการเปลี่ยนแปลงข้อมูลที่มีอยู่เดิมเป็นข้อมูลใหม่ อันเนื่องมาจากข้อมูลมีการเปลี่ยนแปลงทำให้ข้อมูลเดิมที่มีอยู่ในฐานข้อมูลนั้นเกิดความล่าช้าไม่ทันต่อเหตุการณ์ โดยฐานข้อมูลที่จำเป็นต้องมีการปรับปรุงข้อมูลจะเป็นฐานข้อมูลที่มีการเก็บข้อมูลมาไว้และนำข้อมูลที่เก็บมาไปใช้ต่อ ซึ่งข้อมูลในลักษณะนี้จะเป็นข้อมูลที่มีการเปลี่ยนแปลงไปตามระยะเวลาขึ้นอยู่กับแหล่งข้อมูล จึงจำเป็นต้องมีการเก็บข้อมูลใหม่เพื่อให้ได้ข้อมูลที่ทันสมัยอยู่เสมอ ฐานข้อมูลประเภทนี้จึงได้แก่ ฐานข้อมูลของโปรแกรมประเภท Web crawler ตัวรวมรวมข่าวสารของ RSS เป็นต้น ซึ่งการปรับปรุงข้อมูลจะมีสิ่งที่ต้องพิจารณาประกอบด้วย แบบจำลองการแสดงข้อมูล รูปแบบการดึงข้อมูล และการวัดประสิทธิภาพ

2.5.1 แบบจำลองการแสดงข้อมูล (Posting Generation Model)

แหล่งข้อมูลแต่ละแหล่งอาจมีการแสดงข้อมูลที่แตกต่างกัน แต่สิ่งหนึ่งที่มีลักษณะคล้ายกันคือรูปแบบในการแสดงข้อมูล กล่าวคือถ้าเราพิจารณาการแสดงข้อมูล ณ ช่วงเวลาหนึ่งๆ ของแหล่งข้อมูล บางแหล่งข้อมูลอาจมีการแสดงข้อมูลมากและบางแหล่งข้อมูลอาจมีการแสดงข้อมูลน้อยในช่วงเวลานั้นขึ้นอยู่กับแหล่งข้อมูล แต่พบว่าส่วนมากไม่ว่าแหล่งข้อมูลใดก็ตาม ถ้าในช่วงเวลาใดมีการแสดงข้อมูลมากช่วงเวลานั้นจะมีการแสดงข้อมูลที่มากเช่นนั้นเสมอ และในกรณีเดียวกันถ้าในช่วงเวลาใดมีการแสดงข้อมูลน้อยช่วงเวลานั้นจะมีการแสดงข้อมูลที่น้อยเช่นนั้นเสมออีกเช่นกัน ดังนั้นจึงมีการสร้างแบบจำลองการแสดงข้อมูลโดยใช้ข้อมูลการแสดงข้อมูลของแหล่งข้อมูล เพื่อช่วยให้ทราบว่าเวลาใดมีการแสดงข้อมูลมากน้อยอย่างไร ซึ่งช่วยให้สามารถกำหนดระยะเวลาที่เหมาะสมในการปรับปรุงข้อมูลได้

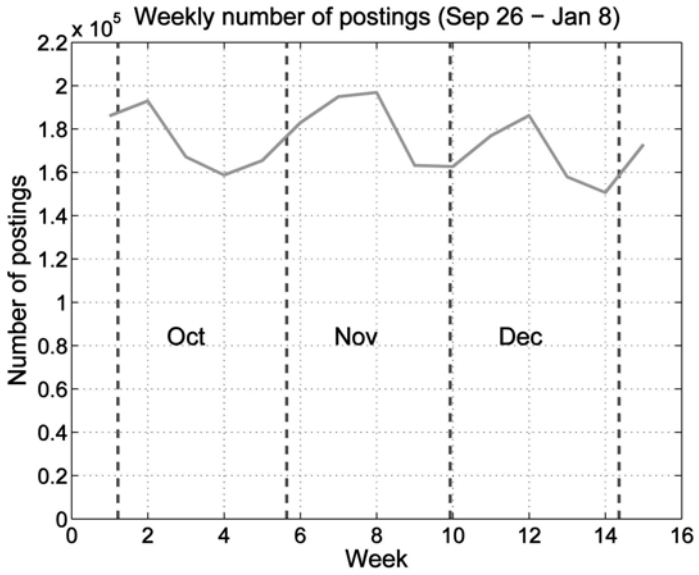
แบบจำลองที่ใช้นั้นจะสร้างขึ้นโดยใช้ข้อมูลของการเปลี่ยนแปลงข้อมูลในอดีต ซึ่งลักษณะของข้อมูลที่ได้นั้นสอดคล้องกับการแจกแจงแบบปัวส์ซอง จึงมีการนำเสนอการนำแบบจำลองปัวส์ซองมาใช้ในการพิจารณาหาระยะเวลาในการปรับปรุงข้อมูล โดยแบ่งออกได้เป็น 2 รูปแบบ (Lipschuts and Schiller, 1995) คือ

1) แบบจำลองปัวส์ซองแบบเอกพันธ์ (Homogeneous Poisson Model)

แบบจำลองปัวส์ซองแบบเอกพันธ์เป็นแบบจำลองที่การเปลี่ยนแปลงของข้อมูลไม่ขึ้นอยู่กับเวลา เป็นกระบวนการที่มีค่า λ เป็นพารามิเตอร์หรือที่เรียกว่าค่าความหนาแน่น ซึ่งหมายถึงจำนวนเหตุการณ์ที่เกิดขึ้นในช่วงเวลา $(t, t+\tau]$ เป็นไปตามการแจกแจงแบบปัวส์ซองที่สอดคล้องกับค่า $\lambda\tau$ ดังนี้

$$P(N(t+\tau)-N(t)=k)=e^{-\lambda\tau}\frac{(\lambda\tau)^k}{k!}$$ โดยที่ $k=0,1,2,...$

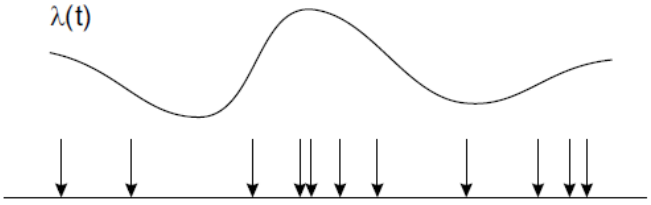
และมี $N(t+\tau)-N(t)$ เป็นจำนวนเหตุการณ์ที่เกิดขึ้นในช่วงเวลา $(t,t+\tau]$ โดยกระบวนการปัวส์ซองแบบเอกพันธ์มี λ เป็นพารามิเตอร์เช่นเดียวกันกับการแจกแจงตัวแปรสุ่มแบบปัวส์ซองที่มี λ เป็นพารามิเตอร์ ซึ่งคือค่าคาดหวังของจำนวนเหตุการณ์ที่เกิดขึ้นต่อช่วงเวลา



ภาพประกอบ 2.15 ตัวอย่างการแสดงข้อมูลเมื่อพิจารณาช่วงเวลาเป็นเดือน

จากภาพประกอบ 2.15 เป็นการแสดงข้อมูลจากแหล่งข้อมูล จะเห็นว่าการแสดงข้อมูลเปลี่ยนแปลงโดยไม่ขึ้นอยู่กับเวลาเมื่อพิจารณาช่วงเวลาเป็นเดือนตามเส้นประ ซึ่งเป็นไปตามแบบจำลองปัวส์ซองแบบเอกพันธ์

2) แบบจำลองปัวส์ซองแบบไม่เป็นเอกพันธ์ (Non – homogeneous Poisson Model)



ภาพประกอบ 2.16 จำนวนเหตุการณ์ที่เปลี่ยนไปตามเวลา

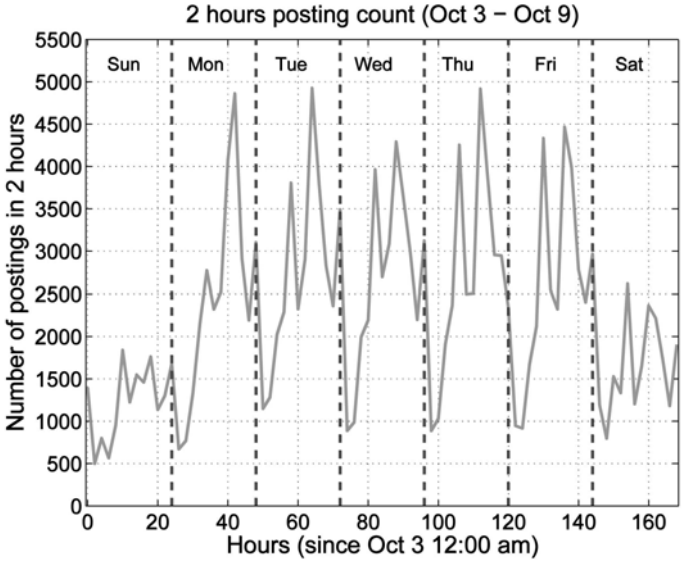
แบบจำลองปัวส์ซองแบบเอกพันธ์เป็นแบบจำลองที่การเปลี่ยนแปลงของข้อมูลขึ้นอยู่กับเวลาเมื่อพิจารณาด้วยช่วงเวลาที่เท่ากัน โดยทั่วไปเมื่อเวลาเปลี่ยนไปค่าพารามิเตอร์อาจมีการเปลี่ยนแปลง จำนวนเหตุการณ์ที่เปลี่ยนไปตามเวลาแสดงดังภาพประกอบ 2.16 กระบวนการดังกล่าวเรียกว่ากระบวนการปัวส์ซองแบบไม่เป็นเอกพันธ์ ในกรณีนี้จะทำการเปลี่ยนค่าพารามิเตอร์ใหม่เป็น $\lambda(t)$ เพื่อให้สอดคล้องกับเวลาที่เปลี่ยนไป ดังนั้นค่าคาดหวังของจำนวนเหตุการณ์ที่เกิดขึ้น ณ เวลา a ถึงเวลา b คือ

$$\lambda_{a,b} = \int_a^b \lambda(t)dt$$

ดังนั้นจำนวนเหตุการณ์ที่เกิดขึ้นในช่วงเวลา $(a,b]$ ให้เป็น $N(b)-N(a)$ ซึ่งเป็นไปตามการแจกแจงแบบปัวส์ซองที่มี $\lambda_{a,b}$ เป็นค่าพารามิเตอร์ จะได้ว่า

$$P(N(b)-N(a)=k) = e^{-\lambda_{a,b}} \frac{(\lambda_{a,b})^k}{k!} \text{ โดยที่ } k = 0,1,2,...$$

โดยกระบวนการปัวส์ซองแบบเอกพันธ์อาจเป็นกรณีหนึ่งของแบบไม่เป็นเอกพันธ์เมื่อ $\lambda(t)=\lambda$ ซึ่งเป็นค่าคงที่



ภาพประกอบ 2.17 ตัวอย่างการแสดงข้อมูลเมื่อพิจารณาช่วงเวลาเป็นวัน

จากภาพประกอบ 2.17 จะเห็นว่าเมื่อพิจารณาช่วงเวลาที่สั้นลงเป็นวันลักษณะการแสดงข้อมูลจะเปลี่ยนแปลงโดยขึ้นอยู่กับเวลา คือช่วงแรกจะน้อยและจะค่อยๆ เพิ่มมากขึ้นเช่นนี้ทุกวัน จึงเป็นไปตามแบบจำลองปัวส์ซองแบบไม่เป็นเอกพันธ์

2.5.2 รูปแบบการดึงข้อมูล (Retrieval Polices)

รูปแบบการดึงข้อมูลจะส่งผลต่อการปรับปรุงข้อมูลด้วยเช่นกัน เนื่องจากรูปแบบการดึงข้อมูลที่ต่างกันจะส่งผลให้ระยะเวลาในการดึงข้อมูลต่างกัน โดยรูปแบบการดึงข้อมูลจะแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือ

1) การดึงข้อมูลโดยพิจารณาจำนวนครั้ง (Resource allocation)

เป็นรูปแบบที่คำนึงถึงการจัดสรรทรัพยากรในการดึงข้อมูลของตัวรวบรวมข่าวสาร โดยแต่ละแหล่งข้อมูลจะได้รับจำนวนครั้งในการดึงข้อมูลที่แตกต่างกันขึ้นอยู่กับความสำคัญและอัตราการแสดงข้อมูลของแหล่งข้อมูลนั้น

2) การดึงข้อมูลโดยพิจารณาตำแหน่งเวลา (Retrieval scheduling)

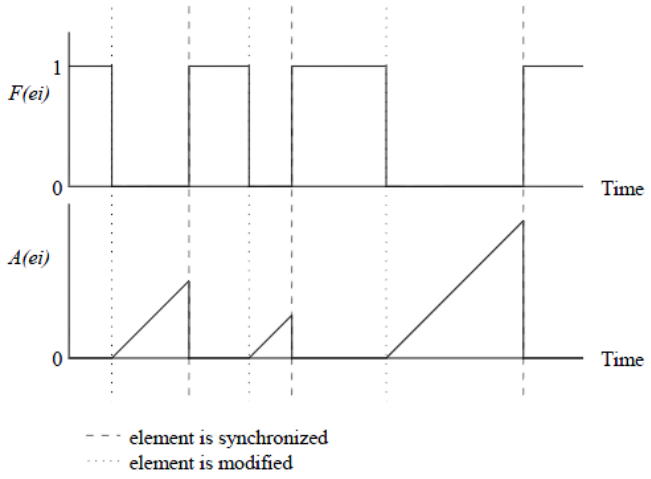
เป็นรูปแบบที่คำนึงถึงการกำหนดตำแหน่งเวลาในการดึงข้อมูล ซึ่งแต่ละแหล่งข้อมูลจะมีตำแหน่งเวลาในการดึงข้อมูลที่แตกต่างกันขึ้นอยู่กับว่าเวลาใดมีการแสดงข้อมูลมากน้อยอย่างไร โดยตัวรวบรวมข่าวสารจะทราบจำนวนครั้งในการดึงข้อมูลที่แน่นอนจากนั้นจึงทำการหาว่าตำแหน่งเวลาใดที่จะทำการดึงข้อมูล

2.5.3 การวัดประสิทธิภาพ (Efficiency)

ประเด็นสำคัญที่ต้องคำนึงถึงในการปรับปรุงข้อมูลอีกประเด็นหนึ่งคือประสิทธิภาพในการปรับปรุงข้อมูล การวัดประสิทธิภาพนั้นจะวัดจากข้อมูลเดิมที่มีอยู่กับข้อมูลที่ทำการปรับปรุงแล้วซึ่งจะขึ้นอยู่กับสิ่งที่พิจารณาด้วย โดยประสิทธิภาพของการปรับปรุงข้อมูลจะเป็นตัวกำหนดระยะเวลาในการปรับปรุงข้อมูล ซึ่งจะกำหนดโดยเลือกจากระยะเวลาที่มีประสิทธิภาพมากที่สุด สิ่งที่ใช้วัดประสิทธิภาพมีหลายลักษณะดังต่อไปนี้

1) ความใหม่และอายุของข้อมูล (Freshness and Age)

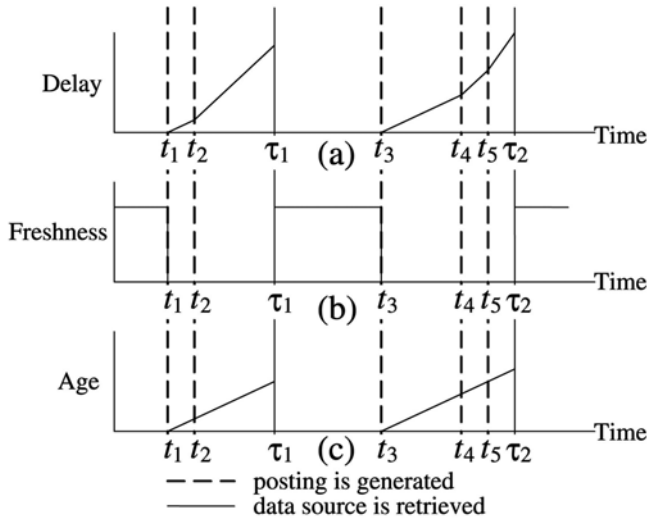
ความใหม่และอายุของของข้อมูล (Cho and Molina, 2000) เป็นการวัดประสิทธิภาพของ Web crawler ในการปรับปรุงข้อมูลของฐานข้อมูล โดยการกำหนดให้ความใหม่ของข้อมูลที่มีอยู่ในฐานข้อมูลนั้นมีค่าเท่ากับ 1 หลังจากปรับปรุงข้อมูลจนกว่าจะมีการแสดงข้อมูลใหม่ และมีค่าเท่ากับ 0 หลังจากมีการแสดงข้อมูลใหม่ ส่วนอายุของข้อมูลเมื่อมีการปรับปรุงข้อมูลอายุของข้อมูลจะถูกกำหนดให้มีค่าเท่ากับ 0 จากนั้นจะเพิ่มขึ้นเรื่อยๆ จนกว่าจะมีการปรับปรุงข้อมูลอีกครั้งหนึ่ง การวัดประสิทธิภาพความใหม่และอายุของข้อมูลแสดงดังภาพประกอบ 2.18



ภาพประกอบ 2.18 ความใหม่และอายุของข้อมูล

2) ความล่าช้าในการดึงข้อมูล (Delay)

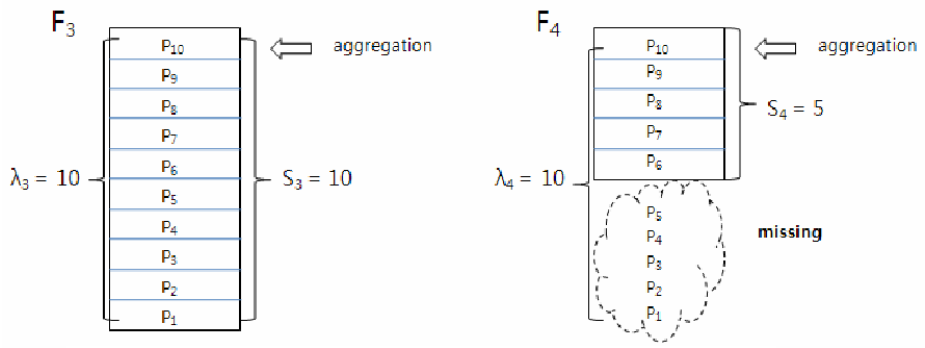
ความล้าช้าในการดึงข้อมูล (Sia and Cho, 2007)เป็นสิ่งที่ใช้วัดประสิทธิภาพของเวลาในการปรับปรุงข้อมูล โดยหาผลต่างของเวลาในการปรับปรุงข้อมูลกับเวลาจริงที่ข้อมูลถูกแสดง ถ้าผลต่างมีค่าน้อยจะได้ว่าเวลาในการปรับปรุงข้อมูลมีประสิทธิภาพ กล่าวคือทำการปรับปรุงข้อมูลได้อย่างทันถ่วงที แต่หากผลต่างมีค่ามากแสดงว่าทำการปรับปรุงข้อมูลช้ากว่าความเป็นจริงมาก ข้อดีอีกอย่างหนึ่งของการวัดประสิทธิภาพชนิดนี้คือข้อมูลทุกข้อมูลจะมีผลต่อความล่าช้าต่างจากความใหม่และอายุของข้อมูลที่ใช้เฉพาะข้อมูลแรกที่ถูกแสดงเท่านั้น ทำให้ช่วงเวลาที่มีข้อมูลแสดงมากๆ ควรปรับปรุงข้อมูลบ่อยๆ การวัดประสิทธิภาพความล่าช้าในการดึงข้อมูลเทียบกับความใหม่และอายุของข้อมูลแสดงดังภาพประกอบ 2.19



ภาพประกอบ 2.19 การวัดประสิทธิภาพความล่าช้าในการดึงข้อมูล
เทียบกับความใหม่และอายุของข้อมูล

3) การหายไปของข้อมูล (Missing posting)

การหายไปของข้อมูล (Han *et al*, 2000) เป็นการวัดประสิทธิภาพโดยคำนึงถึงการเก็บข้อมูลให้ครบถ้วนและไม่ขาดหายไปเป็นสำคัญ เนื่องจากแหล่งข้อมูลมีพื้นที่ในการแสดงข้อมูลที่จำกัด เมื่อมีข้อมูลใหม่เข้ามาข้อมูลเก่าจะหายไป รายละเอียดอธิบายจากภาพประกอบ 2.20 วิธีการนี้จึงเหมาะสำหรับฐานข้อมูลที่ต้องการเก็บข้อมูล และทำการปรับปรุงข้อมูลเพื่อไม่ให้ข้อมูลขาดหายไปโดยไม่คำนึงถึงความล่าช้าในการปรับปรุงข้อมูล



ภาพประกอบ 2.20 การหายไปของข้อมูล

4) การวัดประสิทธิภาพแบบอื่น ๆ

นอกจากสิ่งที่ใช้วัดประสิทธิภาพที่กล่าวมาข้างต้นแล้วยังมีสิ่งที่ใช้วัดประสิทธิภาพอีกหลากหลายรูปแบบ เช่น การเปลี่ยนแปลงของแหล่งข้อมูล อัตราการดาวน์โหลดและปรับปรุงแหล่งข้อมูล (Kim and Lee, 2007) ขึ้นอยู่กับว่าพิจารณาอะไร อย่างไรก็ตามในการเลือกว่าจะใช้อะไรในการวัดประสิทธิภาพจะต้องสอดคล้องกับสิ่งที่ใช้วัดประสิทธิภาพด้วย เช่น การวัดประสิทธิภาพของการปรับปรุงข้อมูลที่ได้ข้อมูลครบถ้วน สิ่งที่ใช้วัดคือการหายไปของข้อมูล หากมีการหายไปของข้อมูลแสดงว่าเป็นการปรับปรุงข้อมูลที่มีประสิทธิภาพน้อย และหากมีการหายไปของข้อมูลน้อยหรือไม่มีเลยแสดงว่าเป็นการปรับปรุงข้อมูลที่มีประสิทธิภาพมาก เป็นต้น

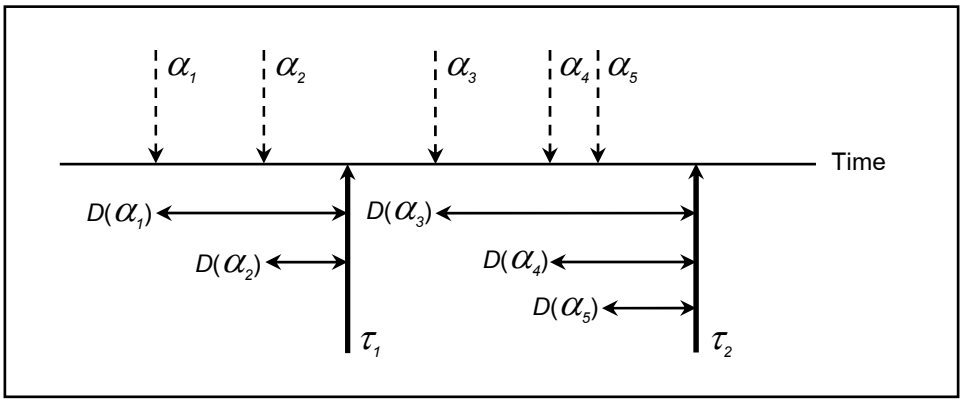
บทที่ 3

บทนิยามและทฤษฎีบท
สำหรับกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

ในบทนี้จะกล่าวถึงการกำหนดบทนิยามและทฤษฎีบทที่ใช้สำหรับกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล โดยจะกล่าวถึงความล่าช้าในการดึงข้อมูลเป็นอันดับแรกเพื่ออธิบายถึงปัจจัยต่างๆ ที่ส่งผลต่อการหาค่าความล่าช้าในการดึงข้อมูล จากนั้นจะนิยามปัจจัยต่างๆ เหล่านั้นเพื่อกำหนดให้เป็นตัวแปรและสามารถแทนค่าตัวแปรนั้นได้ รายละเอียดประกอบด้วย ความล่าช้าในการดึงข้อมูล การกำหนดตำแหน่งเวลาในการดึงข้อมูล บทนิยามปัจจัยที่ส่งผลต่อความล่าช้าในการดึงข้อมูล การคำนวณความล่าช้าในการดึงข้อมูล การคำนวณความล่าช้าในการดึงข้อมูลเมื่อปัดเศษนาทีและวินาที และทฤษฎีบทของตำแหน่งเวลาในการดึงข้อมูล

3.1 ความล่าช้าในการดึงข้อมูล

ในบทที่ 2 ได้กล่าวถึงการวัดประสิทธิภาพของการดึงข้อมูลด้วยรูปแบบต่างๆ ไปแล้ว ในส่วนนี้จะกล่าวถึงการวัดประสิทธิภาพที่เหมาะสมสำหรับการดึงข้อมูลเอกสาร RSS เนื่องจากเอกสาร RSS เป็นข้อมูลที่มีการเปลี่ยนแปลงข้อมูลอย่างรวดเร็ว ดังนั้นตัวรวบรวมข่าวสารที่ทำหน้าที่ดึงข้อมูลจากแหล่งข้อมูลที่ให้บริการเอกสาร RSS จึงจำเป็นที่จะต้องทำการดึงข้อมูลจากแหล่งข้อมูลอย่างรวดเร็วที่สุดเมื่อมีการแสดงข้อมูลใหม่ การวัดประสิทธิภาพในการดึงข้อมูลของตัวรวบรวมข่าวสารจึงต้องให้ความสำคัญกับประเด็นของความรวดเร็วในการดึงข้อมูลเป็นอันดับแรก ซึ่งการวัดประสิทธิภาพที่สอดคล้องกับความรวดเร็วในการดึงข้อมูลคือความล่าช้าในการดึงข้อมูลนั่นเอง ความล่าช้าในการดึงข้อมูลแสดงดังภาพประกอบ 3.1



ภาพประกอบ 3.1 ความล่าช้าในการดึงข้อมูล ณ ตำแหน่งเวลาต่างๆ

จากภาพประกอบ 3.1 แสดงความล่าช้าในการดึงข้อมูล โดยมีข้อมูลที่แสดง ณ เวลาต่างๆ 5 ข้อมูล คือ $\alpha_1,\alpha_2,\ldots,\alpha_5$ ซึ่งถูกแสดงด้วยเส้นประ และมีการดึงข้อมูล 2 ครั้ง คือ τ_1 และ τ_2 ซึ่งถูกแสดงด้วยเส้นทึบ จะเห็นว่าข้อมูล α_1 และ α_2 ถูกดึงข้อมูลโดย τ_1 และข้อมูล α_3, α_4 และ α_5 ถูกดึงข้อมูลโดย τ_2 ตามลำดับ โดยแต่ละข้อมูลที่แสดง ณ เวลาต่างๆ มีความล่าช้าในการดึงข้อมูล คือ $D(\alpha_1),D(\alpha_2),\ldots,D(\alpha_5)$ ความล่าช้าในการดึงข้อมูลจะมาจากผลต่างของระยะเวลาที่ข้อมูลแสดงกับตำแหน่งเวลาในการดึงข้อมูลที่ใกล้ที่สุดหลังจากแสดงข้อมูลไปแล้ว ซึ่งแสดงให้เห็นดังภาพประกอบ 3.1

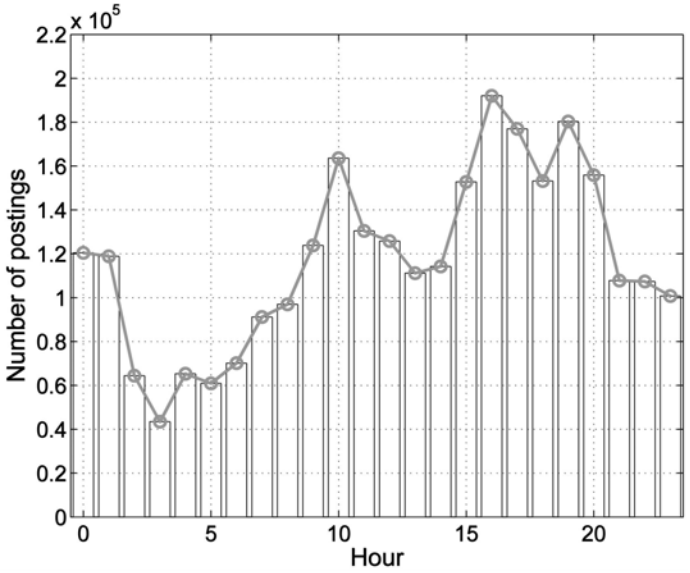
ดังนั้นตำแหน่งเวลาในการดึงข้อมูลจะมีผลอย่างมากต่อการดึงข้อมูลในแต่ละครั้ง เนื่องจากหากมีข้อมูลแสดงหลังจากทำการดึงข้อมูลไปแล้วข้อมูลนั้นจะต้องรอจนกว่าจะมีการดึงข้อมูลในครั้งต่อไปซึ่งจะส่งผลต่อความล่าช้าที่มากขึ้นจากระยะเวลาในการดึงข้อมูล ด้วยเหตุนี้เวลาในการแสดงข้อมูลทุกข้อมูลจึงมีผลต่อความล่าช้า เพราะเวลาในการแสดงข้อมูลมีผลต่อผลต่างของระยะเวลาที่ข้อมูลแสดงกับตำแหน่งเวลาในการดึงข้อมูล ทำให้ความล่าช้าในการดึงข้อมูลเป็นการวัดประสิทธิภาพของตัวรวบรวมข่าวสารที่ดี เนื่องจากให้ความสำคัญต่อการแสดงข้อมูลทุกข้อมูล

3.2 การกำหนดตำแหน่งเวลาในการดึงข้อมูล

ในการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล จะทำการเลือกตำแหน่งเวลาในการดึงข้อมูลจากตำแหน่งเวลาที่มีความล่าช้าในการดึงข้อมูลที่น้อยที่สุด โดยสามารถจำแนกออกเป็น 2 วิธี คือ การสร้างแบบจำลองการแสดงข้อมูล และการใช้ข้อมูลโดยตรง

3.2.1 การสร้างแบบจำลองการแสดงข้อมูล

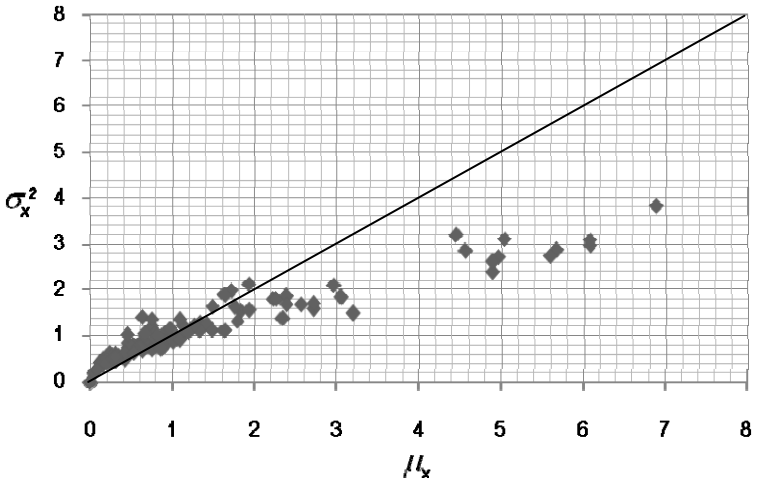
วิธีการหนึ่งที่ช่วยทำให้ทราบถึงลักษณะการแสดงข้อมูลของแหล่งข้อมูล คือ การสร้างแบบจำลองการแสดงข้อมูล งานวิจัยแรกที่เริ่มใช้แบบจำลองการแสดงข้อมูลได้นำเสนอแบบจำลองปัวส์ซอง (Sia and Molina, 2000) เนื่องจากเห็นว่าการแจกแจงของข้อมูลนั้นสอดคล้องกับการแจกแจงแบบปัวส์ซอง โดยในงานวิจัยนี้ใช้แบบจำลองปัวส์ซองแบบเอกพันธ์ (Homogeneous Poisson Model) เป็นแบบจำลองที่การเปลี่ยนแปลงของข้อมูลไม่ขึ้นอยู่กับเวลา ต่อมาได้มีการนำแบบจำลองแบบปัวส์ซองมาประยุกต์ใช้กับการดึงข้อมูลของตัวรวบรวมข่าวสาร โดยการแสดงข้อมูลจากแหล่งข้อมูลมาสร้างเป็นแบบจำลอง อย่างไรก็ตามแบบจำลองที่ใช้จะเป็นแบบจำลองปัวส์ซองแบบไม่เป็นเอกพันธ์ (Non – homogeneous Poisson Model) ซึ่งเป็นแบบจำลองที่การเปลี่ยนแปลงของข้อมูลขึ้นอยู่กับเวลา เนื่องจากการแสดงข้อมูลจากแหล่งข้อมูลในแต่ละวันจะขึ้นอยู่กับช่วงเวลาในแต่ละวัน การสร้างแบบจำลองแสดงดังภาพประกอบ 3.2



ภาพประกอบ 3.2 การสร้างแบบจำลองจากลักษณะการแสดงข้อมูล

จากภาพประกอบ 3.2 แสดงการสร้างแบบจำลองโดยใช้แบบจำลองปัวส์ซองแบบไม่เป็นเอกพันธ์ โดยในแต่ละวันจะมีลักษณะการแสดงข้อมูลที่คล้ายกันดังนั้นช่วงเวลาที่พิจารณาคือเวลา 1 วัน ซึ่งแบ่งข้อมูลออกเป็นช่วงๆ ละ 1 ชั่วโมง แล้วนำจำนวนข้อมูลที่แสดงในแต่ละชั่วโมงของในแต่ละวันมานับรวมกัน จะเห็นได้ว่าในแต่ละชั่วโมงมีจำนวนการแสดงข้อมูลที่ต่างกัน จากจุดยอดของจำนวนข้อมูลในแต่ละชั่วโมงนำมาแสดงในรูปกราฟและกำหนดตำแหน่งเวลาในการดึงข้อมูลจากลักษณะของกราฟ

จากการนำลักษณะการแสดงข้อมูลมาสร้างเป็นแบบจำลองโดยใช้แบบจำลองปัวส์ซอง พบว่าในการใช้แบบจำลองปัวส์ซองนั้นจะได้ลักษณะของแบบจำลองที่ดีก็ต่อเมื่อค่าเฉลี่ยกับความแปรปรวนต้องมีค่าใกล้เคียงกัน ($\mu_x \approx \sigma_x^2$) (Gardner and Mulvey, 1995) และจากการนำเวลาของการแสดงข้อมูลจากแหล่งข้อมูล BBC ใน 1 เดือน มาสร้างกราฟกระจายระหว่างค่าเฉลี่ยกับความแปรปรวน ดังภาพประกอบ 3.3 พบว่าค่า μ_x กับ σ_x^2 มีแนวโน้มที่จะเท่ากัน แต่เมื่อพิจารณาถึงค่าของ $\frac{\left|\mu_x - \sigma_x^2\right|}{\mu_x}$ และ $\frac{\left|\mu_x - \sigma_x^2\right|}{\sigma_x^2}$ พบว่า เมื่อเทียบเป็นเปอร์เซ็นต์แล้วจะมีความแตกต่างกันมาก เช่น ค่าเฉลี่ย (μ_x) = 0.3 ความแปรปรวน (σ_x^2) = 0.5 ค่าของ $\frac{\left|\mu_x - \sigma_x^2\right|}{\mu_x} = \frac{0.2}{0.3}$ = 0.66 ซึ่งเท่ากับ 66% และ $\frac{\left|\mu_x - \sigma_x^2\right|}{\sigma_x^2} = \frac{0.2}{0.5}$ = 0.4 ซึ่งเท่ากับ 40% ดังนั้นการสร้างแบบจำลองโดยใช้แบบจำลองปัวส์ซองจึงมีโอกาสเกิดความคลาดเคลื่อนจากความเป็นจริงได้มาก ซึ่งแบบจำลองดังกล่าวข้างต้นถือเป็นตัวแบบการแสดงข้อมูลที่เหมือนกันทุกวัน ดังนั้นหากได้แบบจำลองที่ไม่ดี อาจส่งผลต่อการกำหนดตำแหน่งเวลาในการดึงข้อมูลเป็นอย่างมาก



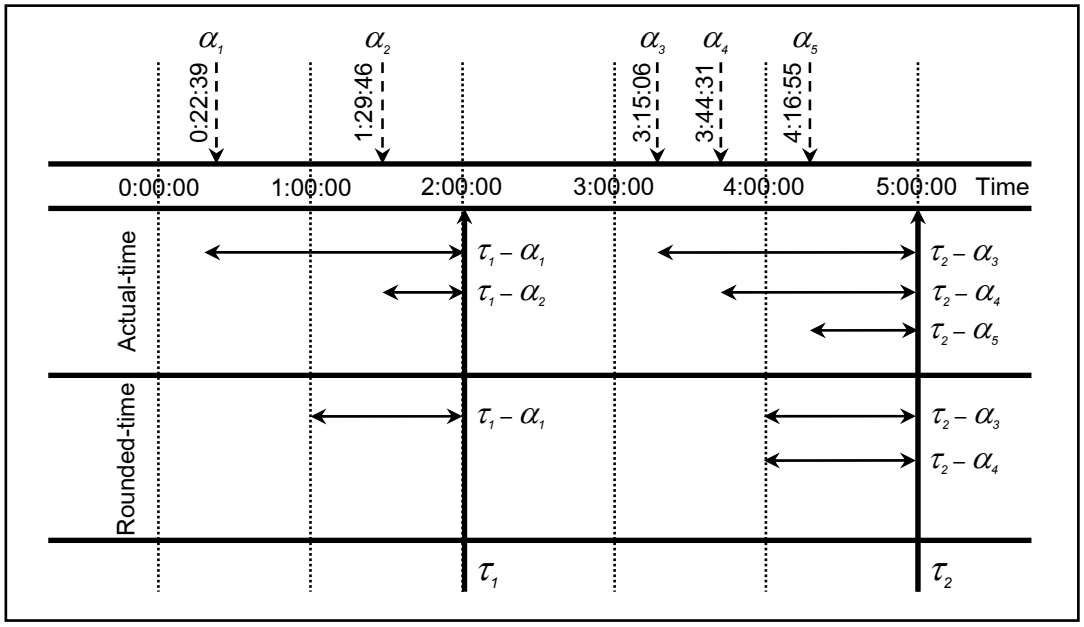
ภาพประกอบ 3.3 กราฟการกระจายระหว่างค่าเฉลี่ยกับความแปรปรวนของข้อมูลจากแหล่งข้อมูล BBC ในเดือนเมษายน 2553

3.2.2 การใช้ข้อมูลโดยตรง

เพื่อหลีกเลี่ยงการใช้แบบจำลองเนื่องจากการสร้างแบบจำลองการแสดงข้อมูลมีโอกาสผิดพลาดได้ ดังนั้นวิธีการหนึ่งที่จะกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลคือการนำข้อมูลมาใช้โดยตรง แต่เนื่องจากการหาค่าของความล่าช้าคำนวณมาจากผลต่างของระยะเวลาที่ข้อมูลแสดงกับตำแหน่งเวลาในการดึงข้อมูล ซึ่งเวลาที่ข้อมูลแสดงนั้นจะอยู่ในรูป

ชั่วโมง นาที และวินาที เช่นเดียวกับตำแหน่งเวลาในการดึงข้อมูลที่สามารถกำหนดเป็นเวลาใด ๆ ในแต่ละวันทำให้การคำนวณผลต่างของระยะเวลานั้นทำได้ยากอีกทั้งยังเสียเวลามาก เนื่องจากต้องคำนวณทีละข้อมูลอีกด้วย

ดังนั้นการนำข้อมูลมาใช้โดยตรงจำเป็นต้องมีการแปลงข้อมูลให้อยู่ในรูปแบบที่ง่ายต่อการคำนวณความล่าช้าก่อน โดยการปัดเศษนาทีและวินาทีขึ้นเหลือเพียงเวลาที่เป็นชั่วโมงเท่านั้น ซึ่งทำให้เวลาในการแสดงข้อมูลในแต่ละวันมีเพียง 24 ค่า จาก 1 – 24 ตามเวลาที่คิดเป็นชั่วโมง และเช่นเดียวกันตำแหน่งเวลาในการดึงข้อมูลจะถูกปัดเศษนาทีและวินาทีทิ้งไป ทำให้ตำแหน่งเวลาในการดึงข้อมูลคือตำแหน่งเวลาที่เป็นชั่วโมงและมีเพียง 24 ค่า จาก 1 – 24 เช่นกัน ความล่าช้าในการดึงข้อมูลโดยการปัดเศษนาทีและวินาทีแสดงดังภาพประกอบ 3.4



ภาพประกอบ 3.4 ความล่าช้าในการดึงข้อมูลโดยการปัดเศษนาทีและวินาที

จากภาพประกอบ 3.4 แสดงความแตกต่างระหว่างการหาความล่าช้าแบบใช้เวลาจริงกับแบบปัดเศษนาทีและวินาที โดยมีข้อมูลที่แสดง 5 ข้อมูล คือ $\alpha_1, \alpha_2, \ldots, \alpha_5$ แสดงข้อมูลในเวลาที่ต่างกันดังภาพ มีการดึงข้อมูล 2 ครั้ง คือ τ_1 และ τ_2 ดึงข้อมูลในเวลา 2.00 น. และ 5.00 น. ตามลำดับ ซึ่งเป็นตำแหน่งเวลาที่ปัดเศษนาทีและวินาที สำหรับเวลาในการแสดงข้อมูลเมื่อปัดเศษนาทีและวินาทีขึ้น จะได้ว่าเวลาที่เป็นชั่วโมงในการแสดงข้อมูล คือ α_1 = 1 , α_2 = 2 , α_3 = 4 , α_4 = 4 , α_5 = 5 ตามลำดับ และตำแหน่งในการดึงข้อมูล τ_1 = 2 และ τ_2 = 5

จะได้ความล่าช้าในการดึงข้อมูลที่ใช้เวลาจริง (Actual – time) แสดงดังรูปส่วนบน โดยคำนวณจากผลต่างของระยะเวลาที่ข้อมูลแสดงกับตำแหน่งเวลาในการดึงข้อมูล และความล่าช้าในการดึงข้อมูลที่ใช้เวลาแบบปัดเศษนาทีและวินาที (Rounded – time) แสดงดัง

รูปส่วนล่าง โดยสามารถคำนวณความล่าช้าในการดึงข้อมูลของแต่ละข้อมูลให้ออกมาในรูปจำนวนเต็มได้ดังนี้

ความล่าช้าของการดึงข้อมูลที่ 1 คือ $\tau_1 - \alpha_1 = 2 - 1 = 1$

ความล่าช้าของการดึงข้อมูลที่ 2 คือ $\tau_1 - \alpha_2 = 2 - 2 = 0$

ความล่าช้าของการดึงข้อมูลที่ 3 คือ $\tau_2 - \alpha_3 = 5 - 4 = 1$

ความล่าช้าของการดึงข้อมูลที่ 4 คือ $\tau_2 - \alpha_4 = 5 - 4 = 1$

ความล่าช้าของการดึงข้อมูลที่ 5 คือ $\tau_2 - \alpha_5 = 5 - 5 = 0$

โดยค่าความล่าช้าที่ได้จะมีหน่วยเป็นชั่วโมงและไม่มีเศษนาทีและวินาที ซึ่งทำให้ง่ายต่อการนำไปกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

3.3 บทนิยามปัจจัยที่ส่งผลต่อความล่าช้าในการดึงข้อมูล

เพื่อให้เข้าใจได้ง่ายและสะดวกในการใช้จึงจำเป็นต้องกำหนดบทนิยามของตัวแปรและคำศัพท์ต่างๆ ที่ใช้ในการออกแบบกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS ประกอบไปด้วยบทนิยามต่างๆ ดังต่อไปนี้

<u>บทนิยามที่ 1</u> กำหนดให้ T คือ เซตของตำแหน่งเวลาในแต่ละวัน โดยที่ $T = \{1,2,\ldots,24\}$ จะได้ว่า 1 คือ ตำแหน่งเวลา 1.00 น., 2 คือ ตำแหน่งเวลา 2.00 น., … , 24 คือ ตำแหน่งเวลา 0.00 น.

<u>บทนิยามที่ 2</u> กำหนดให้ช่วงเวลา i คือ ช่วงเวลาระหว่าง $[\, i - 1 \, , \, i \,)$ โดยที่ $i \in T$ จะได้ว่า $i = 1$ คือ ช่วงเวลาระหว่าง 0.00 น. ถึง 0.59 น., $i = 2$ คือ ช่วงเวลาระหว่าง 1.00 น. ถึง 1.59 น. , … , $i = 24$ คือ ช่วงเวลาระหว่าง 23.00 น. ถึง 23.59 น.

<u>บทนิยามที่ 3</u> กำหนดให้ τ คือ ตำแหน่งเวลาในการดึงข้อมูล โดยที่ $\tau \in T$ จะได้ว่า $\tau = 1$ คือ การดึงข้อมูล ณ เวลา 1.00 น., $\tau = 2$ คือ การดึงข้อมูล ณ เวลา 2.00 น., … , $\tau = 24$ คือ การดึงข้อมูล ณ เวลา 0.00 น.

<u>บทนิยามที่ 4</u> กำหนดให้ช่วงเวลา $i \in T$ และให้ η_i คือ จำนวนข้อมูลที่แสดงในช่วงเวลา i จะได้ว่า η_1 คือ จำนวนข้อมูลที่แสดงในช่วงเวลา 0.00 น. ถึง 0.59 น., η_2 คือ จำนวนข้อมูลที่แสดงในช่วงเวลา 1.00 น. ถึง 1.59 น., ... , η_{24} คือ จำนวนข้อมูลที่แสดงในช่วงเวลา 23.00 น. ถึง 23.59 น.

<u>บทนิยามที่ 5</u> กำหนดให้ $\alpha_{i,j}$ คือ เวลาในการแสดงข้อมูล โดยที่ i เป็นตัวบอกช่วงเวลาในการแสดงข้อมูล และ j คือลำดับที่ของการแสดงข้อมูล

<u>บทนิยามที่ 6</u> กำหนดให้ช่วงเวลา $i \in T$ และให้ τ_k คือ ตำแหน่งเวลาในการดึงข้อมูลครั้งที่ k โดยที่ $\tau_k \in T$ และ จะได้ว่าความล่าช้าของการดึงข้อมูลในช่วง i คือ

$$\sum_{j=1}^{\eta_i}\left(\tau_k - \alpha_{i,j}\right)$$

โดยที่ τ_k เป็นเวลาที่ใกล้ที่สุดที่ $i \leq \tau_k$

3.4 การคำนวณผลรวมความล่าช้าในการดึงข้อมูล

จากบทนิยามที่ 6 จะสามารถหาความล่าช้าในแต่ละชั่วโมงได้ บทตั้งที่ 1 ถึง บทตั้งที่ 3 เป็นการอธิบายถึงการคำนวณผลรวมความล่าช้าในการดึงข้อมูลของระยะเวลาต่างๆ โดยจะแบ่งออกเป็นระยะเวลาที่แตกต่างกันเพื่อให้เห็นถึงความแตกต่างของแต่ละระยะเวลา รวมถึงการคำนวณที่ต่างกันด้วย ซึ่งจะนำการคำนวณความล่าช้าจากบทตั้งไปใช้ในทฤษฎีบทต่อไป

<u>บทตั้งที่ 1</u> กำหนดให้ช่วงเวลา $i \in T$ และ $\alpha_{i,j}$ คือ เวลาในการแสดงข้อมูล จะได้ว่าผลรวมความล่าช้าระหว่างตำแหน่งเวลาในการดึงข้อมูล τ_k ถึง τ_{k+1} คือ

$$\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_i}\left(\tau_{k+1} - \alpha_{i,j}\right)$$

<u>บทตั้งที่ 2</u> กำหนดให้ $\alpha_{i,j}$ และ η_i คือ เวลาในการแสดงข้อมูล และจำนวนข้อมูลที่แสดงในช่วง i ตามลำดับ และกำหนดให้ในแต่ละวันดึงข้อมูล m ครั้ง ณ ตำแหน่งเวลา $\tau_1, \tau_2, \ldots, \tau_m$ ตำแหน่งเดียวกันทุกวัน จะได้ว่าผลรวมความล่าช้าของการดึงข้อมูลใน 1 วัน คือ

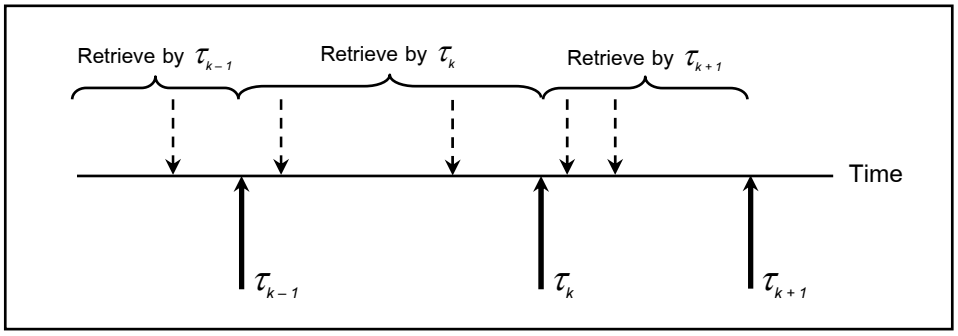
$$\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1}-\alpha_{i',j}\right)$$

โดยที่ $\tau_{m+1} = \tau_1 + 24$ และ $i' \equiv (i-1)(\text{mod } 24)+1$

<u>บทตั้งที่ 3</u> กำหนดให้ดึงข้อมูลจากแหล่งข้อมูล n วัน โดยที่ $\alpha_{i,j}(\ell)$ และ $\eta_i(\ell)$ เป็นเวลาในการแสดงข้อมูลและจำนวนข้อมูลที่แสดงของวันที่ ℓ ตามลำดับ และกำหนดให้ในแต่ละวันดึงข้อมูล m ครั้ง ณ ตำแหน่งเวลา $\tau_1, \tau_2, \ldots, \tau_m$ ตำแหน่งเดียวกันทุกวัน จะได้ว่าผลรวมความล่าช้าของการดึงข้อมูล n วัน คือ

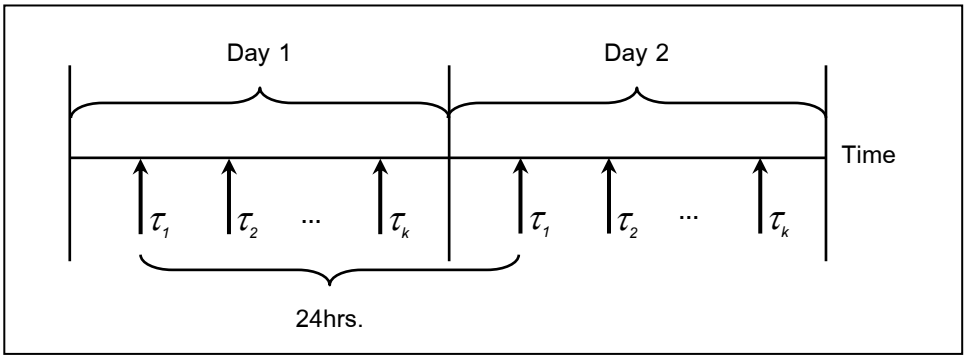
$$\sum_{\ell=1}^{n}\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}(\ell)}\left(\tau_{k+1}-\alpha_{i',j}(\ell)\right)$$

โดยที่ $\tau_{m+1} = \tau_1 + 24$ และ $i' \equiv (i-1)(\text{mod } 24)+1$



ภาพประกอบ 3.5 การดึงข้อมูลของแต่ละตำแหน่งเวลา

บทตั้งที่ 1 อธิบายผลรวมความล่าช้าในการดึงข้อมูลระหว่างแต่ละตำแหน่งเวลาในการดึงข้อมูล จากภาพประกอบ 3.5 แสดงให้เห็นว่าข้อมูลที่แสดงระหว่างตำแหน่งเวลา τ_k ถึง τ_{k+1} จะถูกดึงข้อมูลด้วยตำแหน่งเวลา τ_{k+1} หรือตำแหน่งที่ใกล้ที่สุดหลังจากแสดงข้อมูลไปแล้วนั่นเอง



ภาพประกอบ 3.6 ตำแหน่งเวลาในการดึงข้อมูลในวันถัดไป

บทตั้งที่ 2 อธิบายผลรวมความล่าช้าในการดึงข้อมูลในแต่ละวัน โดยนำค่าของความล่าช้าในแต่ละช่วงจากบทตั้งที่ 1 มารวมกันซึ่งจะมี m ช่วงจากจำนวนการดึงข้อมูลใน 1 วัน อย่างไรก็ตามจะเห็นว่าค่า τ_{m+1} เป็นเวลาที่เกินระยะเวลา 1 วัน เนื่องจากตำแหน่งเวลาในการดึงข้อมูลครั้งสุดท้ายคือ τ_k จึงต้องกำหนดให้ $\tau_{m+1} = \tau_1 + 24$ และ $i' \equiv (i-1)(\text{mod } 24)+1$ เนื่องจากมีบางค่าของ i เกิน 24

และบทตั้งที่ 3 แสดงการหาผลรวมความล่าช้าในการดึงข้อมูลโดยนำผลรวมความล่าช้าในการดึงข้อมูลของแต่ละวันจากบทตั้งที่ 2 มารวมกัน

3.5 การคำนวณผลรวมความล่าช้าในการดึงข้อมูลเมื่อปัดเศษนาทีและวินาที

จากบทนิยามที่ 6 ซึ่งเป็นการหาความล่าช้าในการดึงข้อมูลของแต่ละชั่วโมงแบบใช้เวลาจริง เมื่อปัดเศษนาทีและวินาทีการหาความล่าช้าในการดึงข้อมูลของแต่ละชั่วโมงจึงเปลี่ยนไปจากเดิม และส่งผลต่อการหาผลรวมความล่าช้าในการดึงข้อมูล จึงกำหนดบทนิยามของความล่าช้าในการดึงข้อมูลแบบปัดเศษนาทีและวินาที และกำหนดบทตั้งที่อธิบายถึงการคำนวณผลรวมความล่าช้าในการดึงข้อมูลแบบปัดเศษนาทีและวินาทีของระยะเวลาต่างๆ ขึ้นมาใหม่ด้วย

<u>บทนิยามที่ 7</u> กำหนดให้ช่วงเวลา $i \in T$ และให้ τ_k คือ ตำแหน่งเวลาในการดึงข้อมูลครั้งที่ k โดยที่ $\tau_k \in T$ และ จะได้ว่าความล่าช้าของการดึงข้อมูลแบบปัดเศษนาทีและวินาทีในช่วง i คือ

$$\eta_i(\tau_k - i)$$

โดยที่ τ_k เป็นเวลาที่ใกล้ที่สุดที่ $i \leq \tau_k$

<u>บทตั้งที่ 4</u> กำหนดให้ช่วงเวลา $i \in T$ และ α_{ij} คือ เวลาในการแสดงข้อมูล จะได้ว่าผลรวมความล่าช้าแบบปัดเศษนาทีระหว่างตำแหน่งเวลาในการดึงข้อมูล τ_k ถึง τ_{k+1} คือ

$$\sum_{i=\tau_k+1}^{\tau_{k+1}} \eta_i \left(\tau_{k+1} - i\right)$$

<u>บทตั้งที่ 5</u> กำหนดให้ η_i คือ จำนวนข้อมูลที่แสดงในช่วง i และกำหนดให้ในแต่ละวันดึงข้อมูล m ครั้ง ณ ตำแหน่งเวลา $\tau_1, \tau_2, \ldots, \tau_m$ ตำแหน่งเดียวกันทุกวัน จะได้ว่าผลรวมความล่าช้าแบบปัดเศษนาทีและวินาทีของการดึงข้อมูลใน 1 วัน คือ

$$\sum_{k=1}^{m} \sum_{i=\tau_k+1}^{\tau_{k+1}} \eta_{i'} \left(\tau_{k+1} - i\right)$$

โดยที่ $\tau_{m+1} = \tau_1 + 24$ และ $i' \equiv (i-1)(\text{mod } 24)+1$

<u>บทตั้งที่ 6</u> กำหนดให้ดึงข้อมูลจากแหล่งข้อมูล n วัน โดยที่ $\eta(\ell)_i$ เป็นจำนวนข้อมูลที่แสดงของวันที่ ℓ และกำหนดให้ในแต่ละวันดึงข้อมูล m ครั้ง ณ ตำแหน่งเวลา $\tau_1, \tau_2, \ldots, \tau_m$ ตำแหน่งเดียวกันทุกวัน จะได้ว่าผลรวมความล่าช้าแบบปัดเศษนาทีและวินาทีของการดึงข้อมูล n วัน คือ

$$\sum_{\ell=1}^{n} \sum_{k=1}^{m} \sum_{i=\tau_k+1}^{\tau_{k+1}} \eta_{i'}(\ell)\left(\tau_{k+1} - i\right)$$

โดยที่ $\tau_{m+1} = \tau_1 + 24$ และ $i' \equiv (i-1)(\text{mod } 24)+1$

3.6 ทฤษฎีบทของตำแหน่งเวลาในการดึงข้อมูล

ทฤษฎีบทต่อไปนี้แสดงให้เห็นว่าตำแหน่งเวลาในการดึงข้อมูลที่มีความล่าช้าในการดึงข้อมูลน้อยที่สุดสำหรับข้อมูลแบบปัดเศษนาทีและวินาที เป็นตำแหน่งเวลาในการดึงข้อมูลที่มีความล่าช้าในการดึงข้อมูลที่น้อยที่สุดเช่นกันสำหรับข้อมูลที่ใช้เวลาจริง

<u>ทฤษฎีบทที่ 1</u> กำหนดให้ในแต่ละวันดึงข้อมูล m ครั้ง ณ ตำแหน่งเวลา $\tau_1, \tau_2, \ldots, \tau_m$ ตำแหน่งเดียวกันทุกวัน ถ้า $\tau_1, \tau_2, \ldots, \tau_m$ เป็นตำแหน่งเวลาในการดึงข้อมูลที่มีความล่าช้าในการดึงข้อมูลน้อยที่สุดสำหรับข้อมูลแบบปัดเศษนาที แล้ว $\tau_1, \tau_2, \ldots, \tau_m$ จะเป็นตำแหน่งเวลาในการดึงข้อมูลที่มีความล่าช้าในการดึงข้อมูลที่น้อยที่สุดสำหรับข้อมูลที่ใช้เวลาจริงด้วย

<u>พิสูจน์</u> สมมติให้ในแต่ละวันดึงข้อมูลดึงข้อมูล m ครั้ง ณ ตำแหน่งเวลา $\tau_1, \tau_2, \ldots, \tau_m$ ตำแหน่งเดียวกันทุกวัน

จากบทตั้งที่ 5 จะได้ว่า $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}} \eta_{i'}\left(\tau_{k+1}-i\right)$ คือผลรวมความล่าช้าในการดึงข้อมูลแบบปัดเศษนาทีและวินาทีในหนึ่งวัน และ

จากบทตั้งที่ 2 จะได้ว่า $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1}-\alpha_{i',j}\right)$ คือผลรวมความล่าช้าในการดึงข้อมูลที่ใช้เวลาจริงในหนึ่งวัน

สมมติให้ $\tau_1, \tau_2, \ldots, \tau_m$ เป็นตำแหน่งเวลาในการดึงข้อมูลที่ทำให้ผลรวมความล่าช้าในการดึงข้อมูลแบบปัดเศษนาทีและวินาทีในหนึ่งวันหรือ $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}} \eta_{i'}\left(\tau_{k+1}-i\right)$ มีค่าน้อยที่สุด

จะพิสูจน์ว่า $\tau_1, \tau_2, \ldots, \tau_m$ เป็นตำแหน่งเวลาในการดึงข้อมูลที่ทำให้ผลรวมความล่าช้าในการดึงข้อมูลที่ใช้เวลาจริงในหนึ่งวันหรือ $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1}-\alpha_{i',j}\right)$ มีค่าน้อยที่สุด

จากผลรวมความล่าช้าในการดึงข้อมูลที่ใช้เวลาจริงในหนึ่งวันจะได้ว่า

$$\begin{aligned}
\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1}-\alpha_{i',j}\right) &= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1}-i+i-\alpha_{i',j}\right) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\left(\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1}-i\right)+\sum_{j=1}^{\eta_{i'}}\left(i-\alpha_{i',j}\right)\right) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\left(\eta_{i'}\left(\tau_{k+1}-i\right)+\sum_{j=1}^{\eta_{i'}}\left(i-\alpha_{i',j}\right)\right) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}\left(\tau_{k+1}-i\right)+\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(i-\alpha_{i',j}\right) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}\left(\tau_{k+1}-i\right)+\sum_{i=1}^{24}\sum_{j=1}^{\eta_i}\left(i-\alpha_{i,j}\right) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}\left(\tau_{k+1}-i\right)+c
\end{aligned}$$

โดยที่ $c=\sum_{i=1}^{24}\sum_{j=1}^{\eta_i}\left(i-\alpha_{i,j}\right)$ หรือผลรวมความล่าช้าในการดึงข้อมูลของเศษนาทีและวินาทีที่ถูกปัดทิ้งไป ซึ่งตำแหน่งเวลาในการดึงข้อมูลที่เปลี่ยนไปไม่มีผลต่อความล่าช้าในส่วนนี้ นั่นคือ c เป็นค่าคงที่

จาก $\tau_1,\tau_2,\ldots,\tau_m$ เป็นตำแหน่งเวลาในการดึงข้อมูลที่ทำให้ $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}\left(\tau_{k+1}-i\right)$ มีค่าน้อยที่สุด

ดังนั้น $\tau_1,\tau_2,\ldots,\tau_m$ เป็นตำแหน่งเวลาในการดึงข้อมูลที่ทำให้ $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}\left(\tau_{k+1}-i\right)+c$ มีค่าน้อยที่สุดด้วย

นั่นคือ $\tau_1,\tau_2,\ldots,\tau_m$ เป็นตำแหน่งเวลาในการดึงข้อมูลที่ทำให้ผลรวมความล่าช้าในการดึงข้อมูลที่ใช้เวลาจริงในหนึ่งวันหรือ $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1}-\alpha_{i',j}\right)$ มีค่าน้อยที่สุด ❖

<u>บทแทรกที่ 1</u> กำหนดให้ดึงข้อมูลจากแหล่งข้อมูล *n* วัน แต่ละวันดึงข้อมูลดึงข้อมูล *m* ครั้ง ณ ตำแหน่งเวลา $\tau_1,\tau_2,\ldots,\tau_m$ ตำแหน่งเดียวกันทุกวัน ถ้า $\tau_1,\tau_2,\ldots,\tau_m$ เป็นตำแหน่งเวลาในการดึงข้อมูล *n* วันที่มีความล่าช้าในการดึงข้อมูลน้อยที่สุดสำหรับข้อมูลแบบปัดเศษนาที แล้ว $\tau_1,\tau_2,\ldots,\tau_m$ จะเป็นตำแหน่งเวลาในการดึงข้อมูลที่มีความล่าช้าในการดึงข้อมูล *n* วันที่น้อยที่สุดสำหรับข้อมูลที่ใช้เวลาจริงด้วย

<u>พิสูจน์</u> พิสูจน์ทำนองเดียวกันกับทฤษฎีบทที่ 1

<u>ทฤษฎีบทที่ 2</u> กำหนดให้ดึงข้อมูลจากแหล่งข้อมูล *n* วัน แต่ละวันดึงข้อมูล *m* ครั้ง ณ ตำแหน่งเวลา $\tau_1,\tau_2,\ldots,\tau_m$ ตำแหน่งเดียวกันทุกวัน จะได้ว่าความล่าช้าในการดึงข้อมูลแบบปัดเศษนาทีและวินาทีในแต่ละวันรวมกันมีค่าเท่ากับความล่าช้าแบบปัดเศษนาทีและวินาทีของผลรวมของจำนวนการแสดงข้อมูลในแต่ละชั่วโมงของทุกวันรวมกัน

<u>พิสูจน์</u> สมมติให้ดึงข้อมูลจากแหล่งข้อมูล *n* วัน โดยในแต่ละวันดึงข้อมูลดึงข้อมูล *m* ครั้ง ณ ตำแหน่งเวลา $\tau_1,\tau_2,\ldots,\tau_m$ ตำแหน่งเดียวกันทุกวัน

จากบทตั้งที่ 6 จะได้ว่า $\sum_{\ell=1}^{n}\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}(\ell)\left(\tau_{k+1}-i'\right)$ คือผลรวมความล่าช้าในการดึงข้อมูล *n* วัน แบบปัดเศษนาทีและวินาที และจะได้ว่า

$$\sum_{\ell=1}^{n}\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}(\ell)\left(\tau_{k+1}-i'\right)=\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{\ell=1}^{n}\eta_{i'}(\ell)\left(\tau_{k+1}-i'\right)$$

$$=\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\left[\left(\sum_{\ell=1}^{n}\eta_{i'}(\ell)\right)\cdot\left(\tau_{k+1}-i'\right)\right]$$

โดย $\sum_{\ell=1}^{n}\eta_{i'}(\ell)$ คือ ผลรวมของจำนวนการแสดงข้อมูลในแต่ละชั่วโมง

นั่นคือความล่าช้าในการดึงข้อมูลแบบปัดเศษนาทีและวินาทีในแต่ละวันรวมกันมีค่าเท่ากับความล่าช้าแบบปัดเศษนาทีและวินาทีของผลรวมของจำนวนการแสดงข้อมูลในแต่ละชั่วโมงรวมกัน ❖

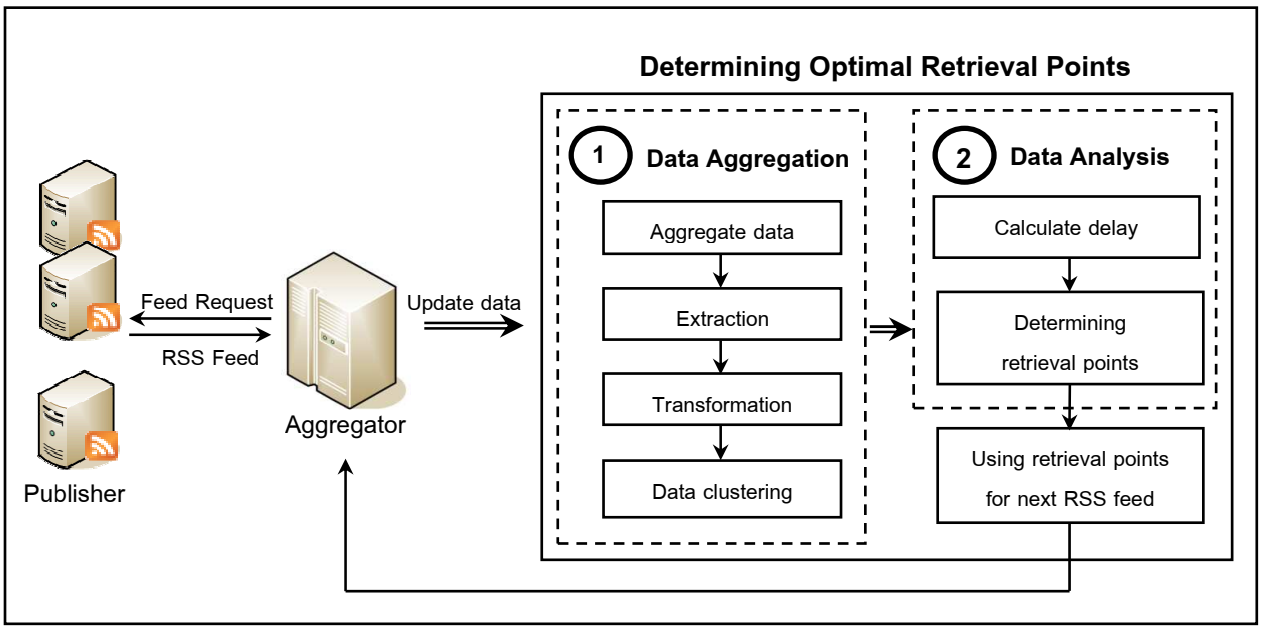
จากทฤษฎีบทและบทแทรกที่ได้ทำให้ทราบว่าตำแหน่งเวลาในการดึงข้อมูลแบบปัดเศษนาทีและวินาทีนั้นเป็นตำแหน่งเดียวกันกับแบบใช้เวลาจริง ซึ่งทำให้การหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลนั้นทำได้สะดวกยิ่งขึ้น และสามารถกำหนดตำแหน่งเวลาโดยการนำข้อมูลมาใช้โดยตรงได้

บทที่ 4

กลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล สำหรับเอกสาร RSS

โดยปกติแล้วตัวรวบรวมข่าวสารของ RSS จะถูกตั้งเวลาในการดึงข้อมูลจากแหล่งข้อมูลเป็นช่วงๆ โดยแต่ละช่วงจะมีระยะเวลาเท่าๆ กัน เช่น กำหนดให้ดึงข้อมูลทุกๆ 2 ชั่วโมง กำหนดให้ดึงข้อมูลวันละ 1 ครั้ง เป็นต้น ซึ่งในบางครั้งอาจดึงข้อมูลช้ากว่าความเป็นจริง เนื่องจากมีการแสดงข้อมูลใหม่แล้วแต่ต้องรอให้ถึงเวลาที่กำหนดจึงจะดึงข้อมูล หรือบางครั้งอาจดึงข้อมูลโดยไม่จำเป็นเนื่องจากยังไม่มีการแสดงข้อมูลใหม่แต่ถึงเวลาที่กำหนดให้ดึงข้อมูลแล้ว วิทยานิพนธ์นี้จึงออกแบบกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล เพื่อช่วยให้ตัวรวบรวมข่าวสารสามารถดึงข้อมูลจากแหล่งข้อมูลต่างๆ ได้อย่างมีประสิทธิภาพ อีกทั้งจัดสรรทรัพยากรของตัวรวบรวมข่าวเพื่อให้นำไปใช้ประโยชน์ได้อย่างคุ้มค่า

สถาปัตยกรรมการกำหนดตำแหน่งเวลาที่เหมาะสม ในการดึงข้อมูล (Determining Optimal Retrieval Points Architecture: DORPA) มีวิธีการทำงานแสดงดังภาพประกอบ 4.1



ภาพประกอบ 4.1 สถาปัตยกรรมการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

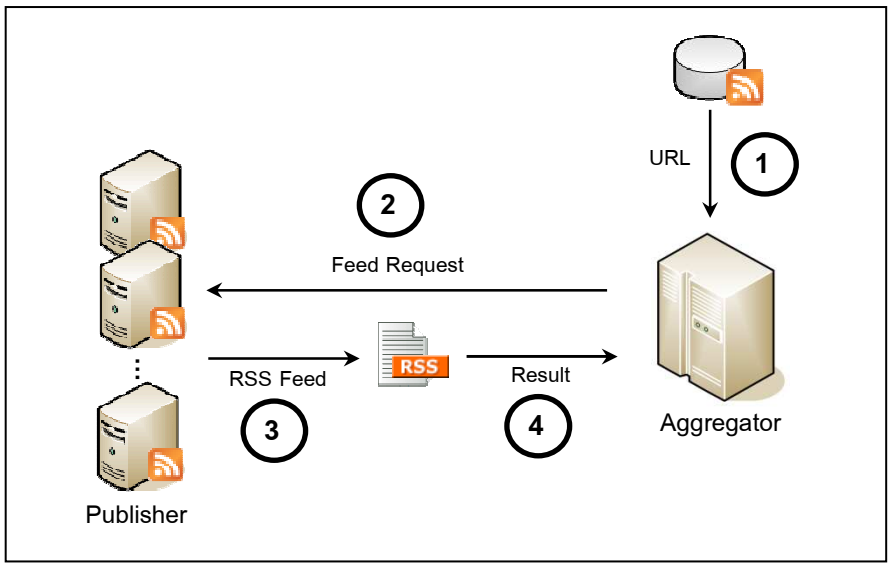
การทำงานของสถาปัตยกรรมการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล ประกอบด้วย 2 ส่วนสำคัญ คือ ส่วนรวบรวมข้อมูล และส่วนวิเคราะห์ข้อมูล โดยส่วนรวบรวมข้อมูลจะเก็บรวบรวมข้อมูลจากแหล่งข้อมูลมาและแปลงข้อมูลเพื่อนำไปใช้ต่อในขั้นตอนต่อไป และส่วนวิเคราะห์ข้อมูลจะนำข้อมูลที่ได้จากส่วนแรกมาวิเคราะห์หาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล ซึ่งแต่ละส่วนมีกลไกการทำงานภายในดังนี้

4.1 ส่วนรวบรวมข้อมูล (Data Aggregation)

ในส่วนนี้มีหน้าที่เตรียมข้อมูลให้อยู่ในรูปแบบที่สามารถนำไปวิเคราะห์หาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลในขั้นตอนต่อไป การทำงานในส่วนนี้ประกอบไปด้วย การรวบรวมข้อมูลจากแหล่งข้อมูล (Aggregate Data) การสกัดข้อมูล (Extraction) การแปลงข้อมูล (Transformation) และการแบ่งกลุ่มข้อมูล (Data Clustering) ซึ่งมีรายละเอียดดังต่อไปนี้

4.1.1 การรวบรวมข้อมูลจากแหล่งข้อมูล (Aggregate Data)

ขั้นตอนนี้เป็นขั้นตอนแรกของการรวบรวมข้อมูล ตัวรวบรวมข่าวสารจะทำการรวบรวมข้อมูลจากแหล่งข้อมูลต่างๆ ที่ให้บริการเอกสาร RSS โดยจะมีที่อยู่ของแหล่งข้อมูล (URL) ของแต่ละแหล่งข้อมูล และจะดึงข้อมูลจากแหล่งข้อมูลที่ได้กำหนดไว้เป็นช่วงเวลาที่เท่าๆ กัน ซึ่งจะดึงข้อมูลจากแหล่งข้อมูลเดิมซ้ำอีกครั้งเมื่อเวลาผ่านไประยะหนึ่งเป็นระยะเวลาที่เท่าๆ กัน อย่างไรก็ตามหากระยะเวลาในการดึงข้อมูลซ้ำช้าเกินไปอาจทำให้มีการหายไปของข้อมูลเกิดขึ้นได้ เนื่องจากแต่ละแหล่งข้อมูลมีจำนวนการแสดงข้อมูลที่จำกัด เมื่อมีการแสดงข้อมูลใหม่จึงทำให้ข้อมูลที่แสดงอยู่ก่อนหน้านั้นอาจหายไปได้ เช่น แหล่งข้อมูลมีจำนวนข้อมูลที่ถูกแสดงครั้งละ 10 ข้อมูล เมื่อเวลาผ่านไประยะหนึ่งสมมติว่ามีการแสดงข้อมูลใหม่จำนวน 5 ข้อมูล ข้อมูลที่แสดงอยู่ก่อนหน้านั้น 5 ข้อมูลจะหายไปเพื่อให้ข้อมูลที่แสดงใหม่แทนที่ ดังนั้นเพื่อให้ได้ข้อมูลครบถ้วนควรกำหนดเวลาในการดึงข้อมูลซ้ำที่เหมาะสมไม่นานจนเกินไป กระบวนการรวบรวมข้อมูลจากแหล่งข้อมูลแสดงดังภาพประกอบ 4.2



ภาพประกอบ 4.2 กระบวนการรวบรวมข้อมูลจากแหล่งข้อมูลต่างๆ

จากภาพประกอบ 4.2 อธิบายกระบวนการทำงานได้ดังต่อไปนี้

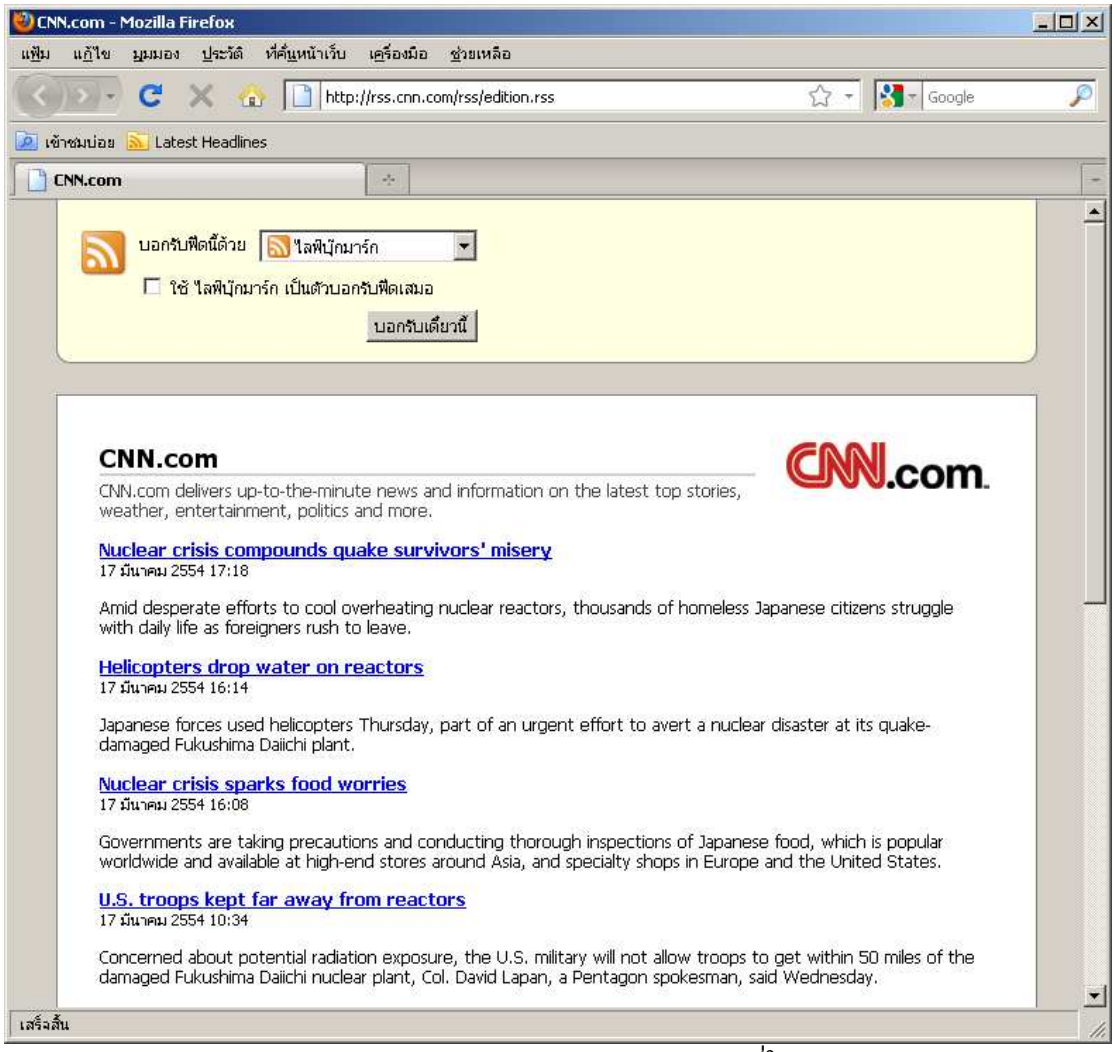
1) ตัวรวบรวมข่าวสารจะรับที่อยู่ของแหล่งข้อมูล (URL) ที่ได้เก็บรวบรวมมาเพื่อนำไปใช้ในการดึงข้อมูลจากแหล่งข้อมูลที่ต้องการ

2) เมื่อตัวรวบรวมข่าวสารได้รับที่อยู่ของแหล่งข้อมูลแล้วจะส่งคำขอไปยังแหล่งข้อมูลต่างๆ ตาม URL ที่ได้ระบุไว้

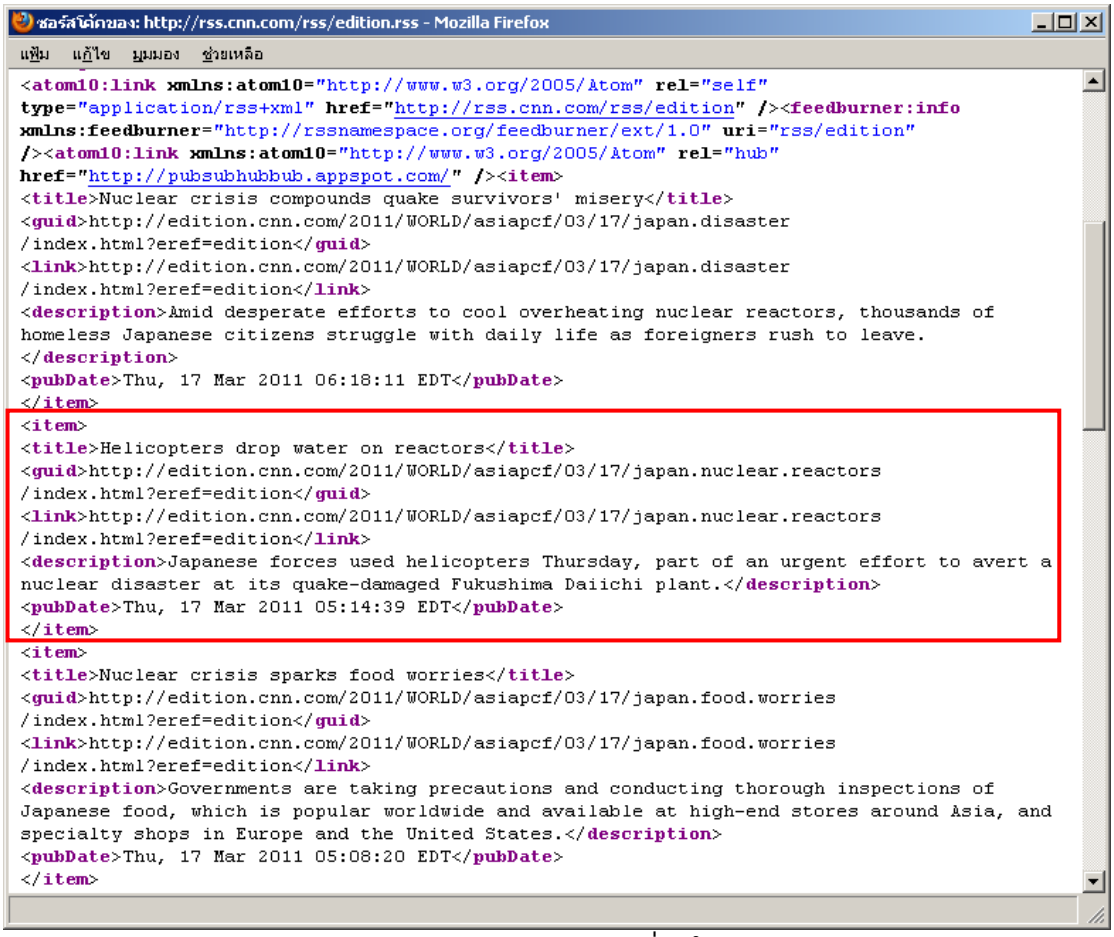
3) แหล่งข้อมูลส่งผลลัพธ์กลับไปโดยผลลัพธ์ที่ได้จะอยู่ในรูปเอกสาร RSS

4) ผลลัพธ์ที่ได้จะถูกส่งให้กับตัวรวบรวมข่าวสารและจะทำการดึงข้อมูลซ้ำเมื่อถึงเวลาที่กำหนด

ตัวอย่างข้อมูลที่ได้แสดงดังภาพประกอบ 4.3 และลักษณะของข้อมูลที่อยู่ในรูปเอกสาร RSS แสดงดังภาพประกอบ 4.4



ภาพประกอบ 4.3 ตัวอย่างข้อมูลจากแหล่งข้อมูลที่ให้บริการ RSS



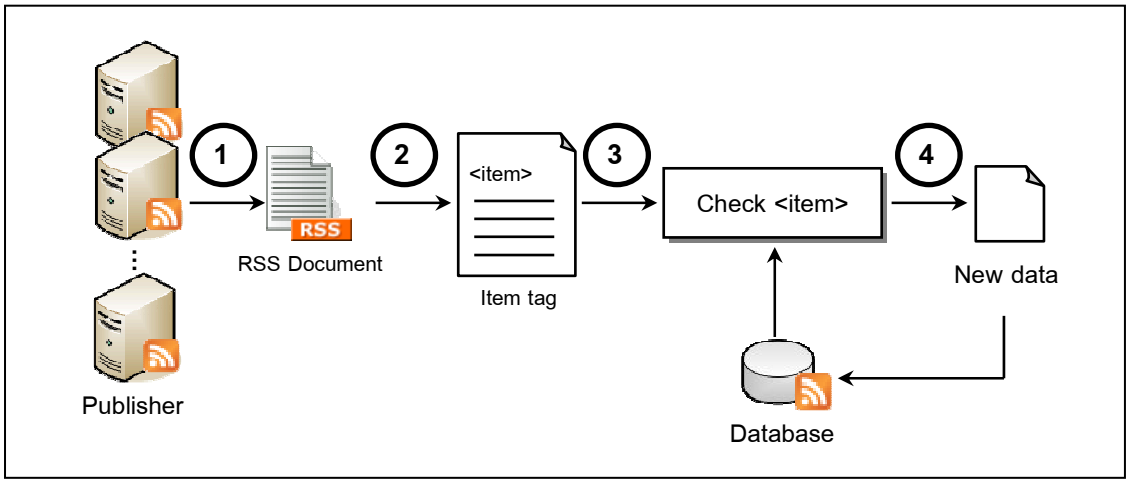
ภาพประกอบ 4.4 ลักษณะของข้อมูลที่อยู่ในรูปเอกสาร RSS

จากภาพประกอบ 4.4 จะเห็นว่าข้อมูลที่อยู่ในรูปเอกสาร RSS จะประกอบด้วยแท็กต่างๆ โดยภายในแท็ก <item> จะมีแท็กต่างๆ เพื่อบอกรายละเอียดของข้อมูลแต่ละข้อมูลที่ถูกส่งกลับมายังตัวรวบรวมข่าวสาร ซึ่งโดยทั่วไปแล้วข้อมูลที่ได้จะมีบางข้อมูลที่ซ้ำกับข้อมูลที่ถูกดึงมาก่อนหน้านี้ ดังนั้นข้อมูลที่ได้รับมาจำเป็นต้องทำการตรวจสอบข้อมูลว่าซ้ำกับข้อมูลเดิมหรือไม่ เพื่อให้ตัวรวบรวมข่าวสารเก็บข้อมูลเฉพาะข้อมูลที่ถูกแสดงเข้ามาใหม่เท่านั้น โดยวิธีการตรวจสอบนั้นจะถูกอธิบายในขั้นตอนต่อไป

4.1.2 การสกัดข้อมูล (Extraction)

เมื่อตัวรวบรวมข่าวสารได้รับข้อมูลที่แหล่งข้อมูลส่งมาให้แล้ว ขั้นตอนต่อไปจะทำการตรวจสอบข้อมูลที่ถูกส่งมาว่ามีข้อมูลใดเป็นข้อมูลใหม่บ้าง วิธีการตรวจสอบข้อมูลนั้นจะใช้ข้อมูลในแท็ก <title> ซึ่งเป็นหัวเรื่องของข้อมูลมาเปรียบเทียบกับหัวเรื่องของข้อมูลเดิมซึ่งถูกเก็บเอาไว้ในฐานข้อมูล ถ้าหัวเรื่องของข้อมูลที่ถูกส่งมาไม่ตรงกับหัวเรื่องข้อมูลเดิมที่มีอยู่แสดงว่าข้อมูลนี้เป็นข้อมูลใหม่ แต่ถ้าหัวเรื่องของข้อมูลที่ถูกส่งมาตรงกับหัวเรื่องของข้อมูลเดิมแสดง

ว่าเป็นข้อมูลที่ซ้ำกัน ด้วยวิธีการนี้จะทำให้ทราบว่าข้อมูลใดเป็นข้อมูลใหม่และข้อมูลใดเป็นข้อมูลเก่าที่ซ้ำกับของเดิมที่มีอยู่ จึงทำให้ตัวรวบรวมข่าวสารสามารถเก็บรวบรวมข้อมูลได้อย่างมีประสิทธิภาพไม่มีการเก็บข้อมูลซ้ำกับข้อมูลเดิม กระบวนการในการตรวจสอบข้อมูลแสดงดังภาพประกอบที่ 4.5 และขั้นตอนวิธีในการตรวจสอบข้อมูลแสดงดังภาพประกอบที่ 4.6



ภาพประกอบ 4.5 กระบวนการในการตรวจสอบข้อมูลใหม่

จากภาพประกอบ 4.5 อธิบายกระบวนการทำงานได้ดังต่อไปนี้

1) แหล่งข้อมูลส่งเอกสาร RSS ไปให้ตัวรวบรวมข่าวสารตามคำขอ

2) เอกสาร RSS ที่ได้จะถูกสกัดให้เหลือเพียงแต่แท็ก <item> โดยภายในแท็กจะประกอบไปด้วยข้อมูลต่างๆของข้อมูลที่ถูกส่งมา

3) แต่ละแท็ก <item> จะถูกตรวจสอบว่าเป็นข้อมูลที่แสดงเข้ามาใหม่หรือไม่ โดยการนำเอาข้อมูลภายในแท็ก <title> หรือหัวเรื่องของข้อมูลมาเปรียบเทียบกับหัวเรื่องของข้อมูลเดิมที่มีอยู่ในฐานข้อมูล

4) เมื่อตรวจสอบแล้วหากพบว่าเป็นหัวเรื่องที่ไม่ซ้ำกับข้อมูลเดิมถือว่าเป็นข้อมูลที่ถูกแสดงใหม่และจะถูกจัดเก็บในฐานข้อมูล

```
1   Method Feed (URL)
2       for each URL
3         retrieve RSS feed from URL
4         for each <item> in RSS feed
5           get data in <title>, <description>, <link>, <pubDate>
6           if data in <title> not the same as data in database
7             store data in database
8           end if
9         end for
10      end for
11  end method
```

ภาพประกอบ 4.6 ขั้นตอนวิธีในการตรวจสอบข้อมูลใหม่

จากภาพประกอบ 4.6 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

บรรทัดที่ 2 – 3 คือ การดึงข้อมูลจาก URL ที่กำหนดไว้ ซึ่ง URL จะชี้ไปยังแหล่งข้อมูลและจะส่งเอกสาร RSS กลับมายังตัวรวบรวมข่าวสาร

บรรทัดที่ 4 – 9 คือ การอ่านข้อมูลภายในแท็ก <item> ซึ่งแต่ละแท็กจะมีข้อมูลที่แตกต่างกันประกอบด้วยข้อมูลจากแท็ก <title>, <description>, <link> และแท็ก <pubDate>

บรรทัดที่ 6 – 8 คือ การตรวจสอบว่าข้อมูลภายในแท็ก <title> ของข้อมูลที่ดึงมาซ้ำกับข้อมูลเดิมหรือไม่ หากไม่ซ้ำแสดงว่าเป็นข้อมูลใหม่และเก็บข้อมูลลงในฐานข้อมูล

หลังจากตรวจสอบข้อมูลแล้วว่าเป็นข้อมูลใหม่ ขั้นตอนต่อไปคือการสกัดเลือกเอาข้อมูลเฉพาะแท็กที่จำเป็นต้องใช้เก็บไว้ในฐานข้อมูลของตัวรวบรวมข่าวสาร ซึ่งได้แก่แท็ก <title>, <description>, <link> และ <pubDate> โดยข้อมูลในแท็ก <title> บอกถึงหัวเรื่องของข้อมูล และจะเก็บเอาไว้เพื่อใช้ในการตรวจสอบกับข้อมูลใหม่ที่ดึงมาว่าเป็นข้อมูลเดียวกันหรือไม่ แท็ก <description> บอกให้ผู้ใช้ทราบถึงรายละเอียดของข้อมูลอย่างคร่าวๆ แท็ก <link> เชื่อมโยงไปยังแหล่งของข้อมูลหากต้องการรายละเอียดของข้อมูลนั้นเพิ่มเติม ส่วนข้อมูลในแท็ก <pubDate> จะเก็บไว้เพื่อให้ทราบว่าข้อมูลถูกแสดง ณ เวลาใด ตัวอย่างของข้อมูลจากแหล่งข้อมูล CNN ที่ถูกจัดเก็บลงในฐานข้อมูลแสดงดังตาราง 4.1

ตารางที่ 4.1 ตัวอย่างข้อมูลที่ถูกจัดเก็บในฐานข้อมูล (CNN.com, 2011: Online)

<title>	<description>	<link>	<pubDate>
Nuclear crisis compounds misery	Amid desperate efforts to cool overheating nuclear reactors, thousands of homeless Japanese citizens struggle with daily life as foreigners rush to leave.	http://edition.cnn.com/2011/WORLD/asiapcf/03/17/japan.disaster/index.html?eref=edition	Thu, 17 Mar 2011 10:25:49 EDT
Japan turns to helicopters, trucks to cool reactors	Cannon-equipped police trucks spray water inside reactor No. 3 at Fukushima Daiichi nuclear plant, defense officials say, after helicopter drops do little to lower radiation levels.	http://edition.cnn.com/2011/WORLD/asiapcf/03/17/japan.nuclear.reactors/index.html?eref=edition	Thu, 17 Mar 2011 08:58:27 EDT
Scramble for salt in China amid scare	Chinese shoppers in Beijing and Shanghai cleared salt from supermarkets shelves on Thursday morning amid fears of a potential radiation crisis from Japan's Fukushima Daiichi nuclear plant.	http://edition.cnn.com/2011/WORLD/asiapcf/03/17/china.salt.scramble/index.html?eref=edition	Thu, 17 Mar 2011 07:54:05 EDT
Nuclear crisis sparks food worries	Governments are taking precautions and conducting thorough inspections of Japanese food, which is popular worldwide and available at high-end stores around Asia, and specialty shops in Europe and the United States.	http://edition.cnn.com/2011/WORLD/asiapcf/03/17/japan.food.worries/index.html?eref=edition	Thu, 17 Mar 2011 07:15:48 EDT

จากตารางที่ 4.1 แสดงความสัมพันธ์ของแต่ละข้อมูลโดยข้อมูลแต่ละแท็ก <title> จะมีข้อมูลในแท็ก <description>, <link> และแท็ก <pubDate> ที่สัมพันธ์กัน ซึ่งเป็นข้อมูลที่มาจากแท็ก <item> แท็กเดียวกัน

4.1.3 การแปลงข้อมูล (Transformation)

เมื่อเก็บข้อมูลได้ระยะหนึ่งจะนำข้อมูลที่ได้เก็บรวบรวมมาวิเคราะห์เพื่อกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล โดยนำข้อมูลของแท็ก <pubDate> ซึ่งเป็นเวลาในการ

แสดงข้อมูลของแต่ละข้อมูลมาวิเคราะห์ตำแหน่งเวลาที่แสดง ซึ่งข้อมูลที่ได้นั้นจะอยู่ในรูปแบบของการแสดงเวลาดังแสดงในภาพประกอบ 4.7

```
Thu,  01  Apr 2010  11:59:18 GMT
Thu,  01  Apr 2010  15:09:49 GMT
Fri,  02  Apr 2010  08:04:15 GMT
Fri,  02  Apr 2010  08:15:16 GMT
Fri,  02  Apr 2010  10:33:55 GMT
Fri,  02  Apr 2010  19:38:11 GMT
Fri,  02  Apr 2010  23:02:26 GMT
Sat,  03  Apr 2010  09:35:45 GMT
Sat,  03  Apr 2010  12:06:02 GMT
Sun,  04  Apr 2010  14:15:54 GMT
```

ภาพประกอบ 4.7 ข้อมูลเวลาในการแสดงข้อมูลภายในแท็ก <pubDate>

การนำเอาข้อมูลในส่วนนี้มาใช้นั้นจึงจำเป็นจะต้องแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม โดยจะแปลงข้อมูลจากเวลาที่แสดงให้อยู่ในรูปแบบจำนวนเต็มเพื่อให้ง่ายต่อการนำไปคำนวณ ซึ่งจะปัดเศษนาทีและวินาทีของเวลาในการแสดงข้อมูลดังนั้นเวลาในการแสดงข้อมูลจะเป็นจำนวนเต็มในช่วง 1 – 24 โดยในการปัดเศษเวลาจะปัดเศษขึ้นทั้งหมด เมื่อทำการแปลงข้อมูลแล้วข้อมูลที่ได้จะอยู่ในรูปวัน เดือนและชั่วโมงที่แสดงข้อมูล ขั้นตอนวิธีในการแปลงข้อมูลแสดงดังภาพประกอบ 4.8

```
1   Method Transformation (Publisher)
2       for each Publisher
3          get data from <pubDate> in database
4          for each data from <pubDate>
5             select date, month and time
6             transform into integer format
7             store data in database
8          end for
9       end for
10   end method
```

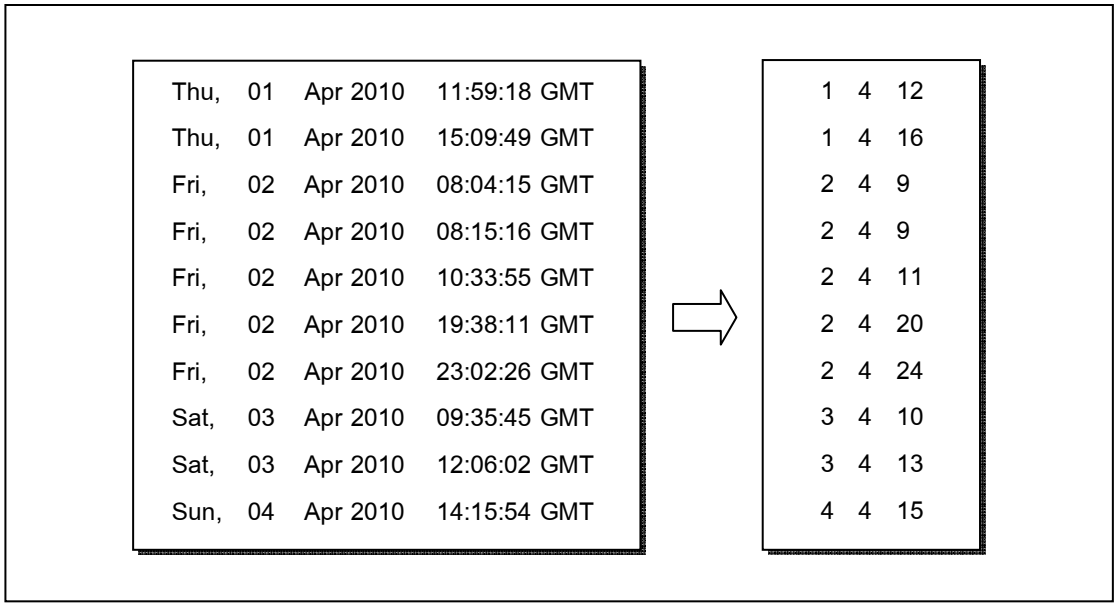
ภาพประกอบ 4.8 ขั้นตอนวิธีในการแปลงข้อมูลให้อยู่ในรูปจำนวนเต็ม

จากภาพประกอบ 4.8 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

บรรทัดที่ 2 – 3 คือ การนำเอาข้อมูลในแท็ก <pubDate> จากแต่ละแหล่งข้อมูลมาทำการแปลงข้อมูล โดยแต่ละแหล่งข้อมูลจะแยกข้อมูลออกจากกัน

บรรทัดที่ 4 – 8 คือ การอ่านข้อมูลภายในแท็ก <pubDate> ที่ได้เก็บไว้ในฐานข้อมูล และเลือกเอาเฉพาะข้อมูลวัน เดือน และเวลาในการแสดงข้อมูล มาทำการแปลงให้อยู่ในรูปแบบจำนวนเต็มและนำข้อมูลที่ได้แปลงข้อมูลเรียบร้อยแล้วเก็บไว้ในฐานข้อมูล

ตัวอย่างการแปลงข้อมูลโดยนำวัน เดือน และเวลาที่แสดงข้อมูลมาแปลงให้อยู่ในรูปจำนวนเต็มแสดงดังภาพประกอบ 4.9

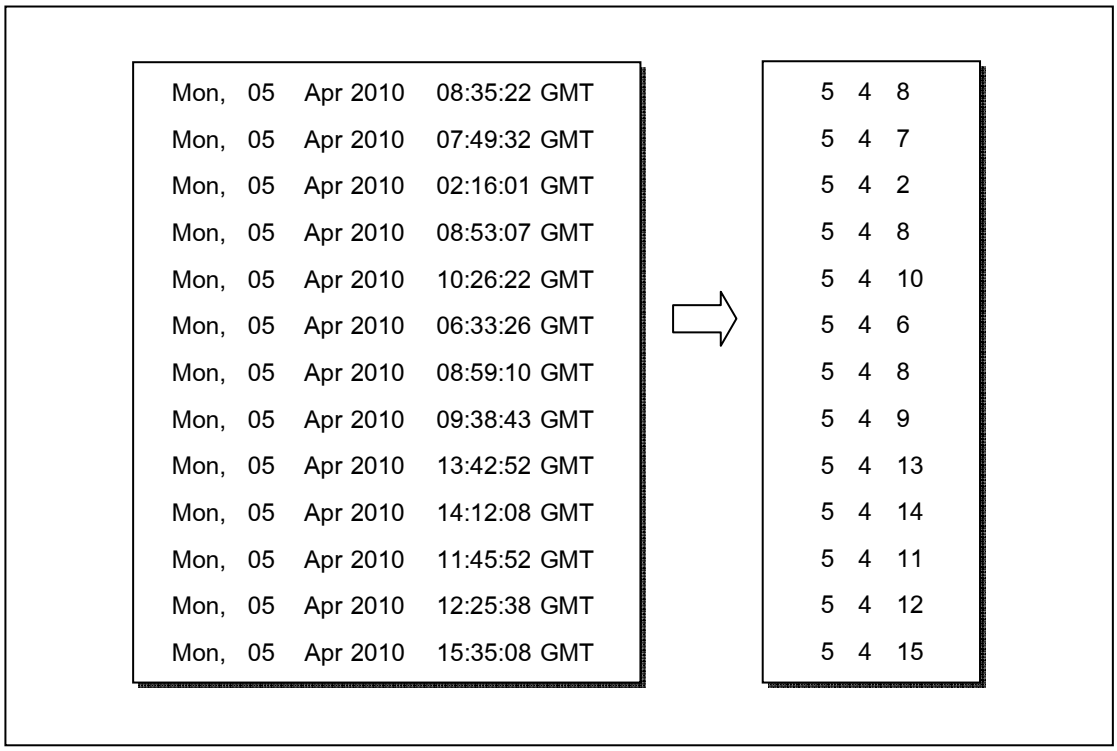


ภาพประกอบ 4.9 การแปลงข้อมูลให้ในอยู่ในรูปแบบจำนวนเต็ม

จากภาพประกอบ 4.9 จะเห็นว่าวันที่แสดงข้อมูลจะถูกแปลงให้อยู่ในรูปแบบจำนวนเต็ม 1 – 31 ซึ่งถูกแปลงให้เหลือแค่วันที่แสดงข้อมูลเท่านั้น เช่น Thu, 01 ถูกแปลงให้เป็น 1 เป็นต้น ส่วนเดือนและปีที่แสดงข้อมูลจะถูกแปลงให้อยู่ในรูปแบบจำนวนเต็ม 1 – 12 ซึ่งถูกแปลงให้เหลือแค่เดือนที่แสดงข้อมูลเท่านั้น เช่น Apr 2010 ถูกแปลงให้เป็น 4 เป็นต้น และส่วนสุดท้ายเป็นเวลาในการแสดงข้อมูลและ Time zone จะถูกแปลงให้อยู่ในรูปแบบจำนวนเต็ม 1 – 24 ซึ่งแปลงให้เหลือแค่ชั่วโมงที่แสดงข้อมูลเท่านั้น ทั้งนี้ข้อมูล Time zone จะเป็นตัวกำหนดให้ทราบว่าจะทำการบวกลบจำนวนชั่วโมงเท่าไรจึงจะได้เวลาที่แท้จริง เช่น GMT ทำให้ทราบว่าจะต้องบวกจำนวนชั่วโมงเพิ่มอีก 7 ชั่วโมงหากต้องการแปลงให้เป็นเวลาในประเทศไทย ซึ่งเป็นขั้นตอนสุดท้ายหลังจากแปลงข้อมูลเรียบร้อยแล้ว

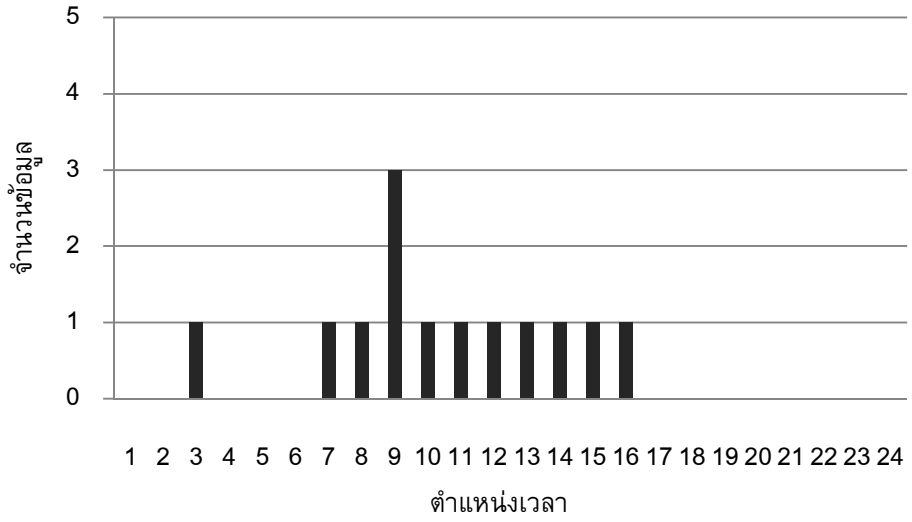
4.1.4 การแบ่งกลุ่มข้อมูล (Data Clustering)

เมื่อแปลงเวลาในการแสดงข้อมูลในอยู่ในรูปจำนวนเต็มแล้ว ในขั้นตอนต่อไปจะเป็นการรวมข้อมูลในแต่ละวันเข้าด้วยกัน โดยนับจำนวนการแสดงข้อมูลในแต่ละชั่วโมงว่ามีจำนวนการแสดงข้อมูลเท่าไรใน 1 วัน ซึ่งจากการปัดเศษนาทีและวินาทีของการแสดงข้อมูลทิ้งไปทำให้ใน 1 วัน แบ่งออกได้เป็น 24 ช่วง ช่วงละ 1 ชั่วโมง จาก 1 – 24 ตัวอย่างการแสดงข้อมูลใน 1 วันแสดงดังภาพประกอบ 4.10



ภาพประกอบ 4.10 การแปลงข้อมูลในวันเดียวกัน

จากภาพประกอบ 4.10 เป็นการแสดงข้อมูลในวันจันทร์ที่ 5 เมษายน พ.ศ. 2553 โดยแต่ละข้อมูลมีเวลาในการแสดงข้อมูลที่แตกต่างกันออกไป เมื่อทำการแปลงเวลาในการแสดงข้อมูลให้อยู่ในรูปจำนวนเต็มแล้วจะเห็นว่าข้อมูลวันที่จะถูกแปลงเป็นจำนวนเต็ม 5 เหมือนกันทุกข้อมูลและเดือนเมษายนถูกแปลงเป็น 4 เหมือนกันทุกข้อมูลเช่นกัน แตกต่างกันที่เวลาในการแสดงข้อมูล เมื่อนำข้อมูลที่แปลงเป็นจำนวนเต็มแล้วมานับจำนวนการแสดงข้อมูลในแต่ละช่วงเวลาจะได้กราฟลักษณะการแสดงข้อมูลดังภาพประกอบ 4.11



ภาพประกอบ 4.11 ลักษณะการแสดงข้อมูลในวันเดียวกัน

จากภาพประกอบ 4.11 แสดงให้เห็นว่าลักษณะการแสดงข้อมูลที่ยกตัวอย่างในวันเดียวกันมีลักษณะที่กระจุกตัวอยู่ในช่วงหนึ่ง ไม่ได้กระจายตัวในแต่ละช่วงที่เท่าๆ กัน ซึ่งจากกราฟจะเห็นว่ามีการแสดงข้อมูลในช่วงเวลา 6.00 น. ถึง 15.00 น. และมีแค่เพียงข้อมูลเดียวที่แสดงอยู่ในช่วง 2.00 น. จากลักษณะการแสดงข้อมูลเช่นนี้ทำให้ทราบว่าแหล่งข้อมูลมีการแสดงข้อมูล ณ ช่วงเวลาใดบ้างซึ่งเป็นประโยชน์อย่างมากในกำหนดตำแหน่งเวลาในการดึงข้อมูล แต่เนื่องจากในการกำหนดตำแหน่งเวลาในการดึงข้อมูลนั้นกำหนดเป็นตำแหน่งเวลาเดียวกันทุกวัน ดังนั้นลักษณะการแสดงข้อมูลที่ใช้จึงจำเป็นต้องนำข้อมูลมาพิจารณาหลายๆ วัน ตารางที่ 4.2 จะแสดงให้เห็นถึงจำนวนข้อมูลที่แสดงในแต่ละช่วงเวลาในแต่ละวันเป็นเวลา 2 สัปดาห์ โดยใช้ข้อมูลข่าวเศรษฐกิจจากแหล่งข้อมูล BBC

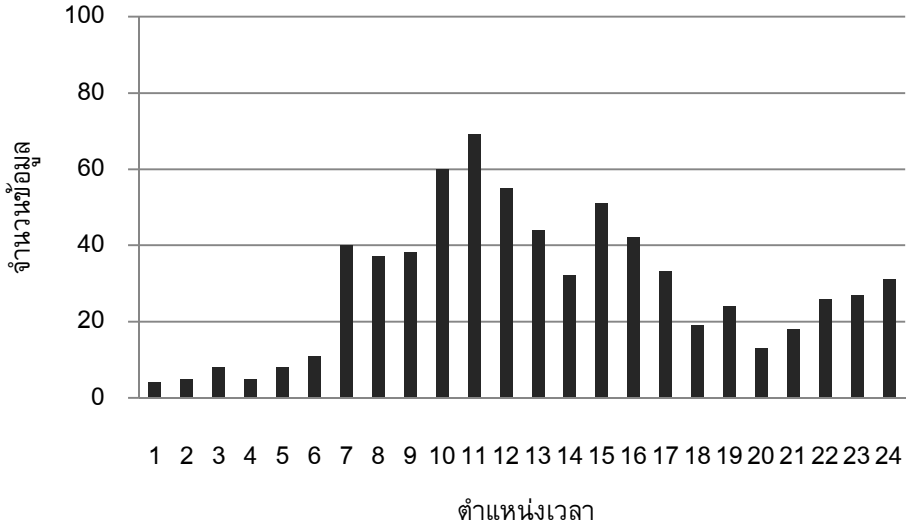
ตารางที่ 4.2 ตัวอย่างการแสดงข้อมูลในแต่ละวันระหว่างวันที่ 5 – 18 เมษายน พ.ศ.2553

วัน	จ.	อ.	พ.	พฤ.	ศ.	ส.	อา.	จ.	อ.	พ.	พฤ.	ศ.	ส.	อา.
เวลา	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	0	0	0	1	0	0	0	0	0	0	0	0	1
2	0	0	0	1	0	0	0	1	0	1	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	2	0	0	0
4	0	0	0	0	0	0	0	1	0	0	1	0	0	0
5	0	0	0	0	0	0	0	1	0	0	0	0	0	0
6	0	0	1	0	0	0	1	0	0	1	1	0	0	0
7	1	2	2	2	1	0	0	3	3	1	1	2	0	0
8	1	0	0	1	1	0	0	1	3	2	3	1	0	0
9	3	4	1	5	2	0	0	2	0	0	1	3	0	0
10	1	1	3	1	0	0	0	1	1	3	5	5	0	0
11	1	1	4	7	3	1	0	3	10	3	1	3	0	0
12	1	0	4	6	2	0	1	2	1	2	2	2	0	0
13	1	1	1	2	3	0	0	4	1	2	2	1	0	0
14	1	0	1	3	3	0	0	1	2	4	0	0	0	1
15	1	0	4	0	3	2	2	1	4	2	5	2	0	0
16	1	1	4	3	2	0	2	0	2	2	2	3	0	0
17	0	1	1	0	2	0	2	2	1	1	2	1	0	1
18	0	0	0	0	1	0	1	1	2	0	2	0	0	2
19	0	3	2	0	0	0	0	2	0	0	1	1	0	0
20	0	0	0	2	2	0	0	2	0	1	1	0	0	0
21	0	1	2	0	0	0	0	0	0	0	1	3	0	1
22	0	2	0	1	1	0	1	2	1	0	1	0	0	0
23	0	2	0	0	1	0	0	2	2	1	1	3	0	1
24	0	1	5	2	3	0	0	2	2	0	2	1	0	0
รวม	13	20	35	36	31	3	10	34	35	26	37	31	0	7

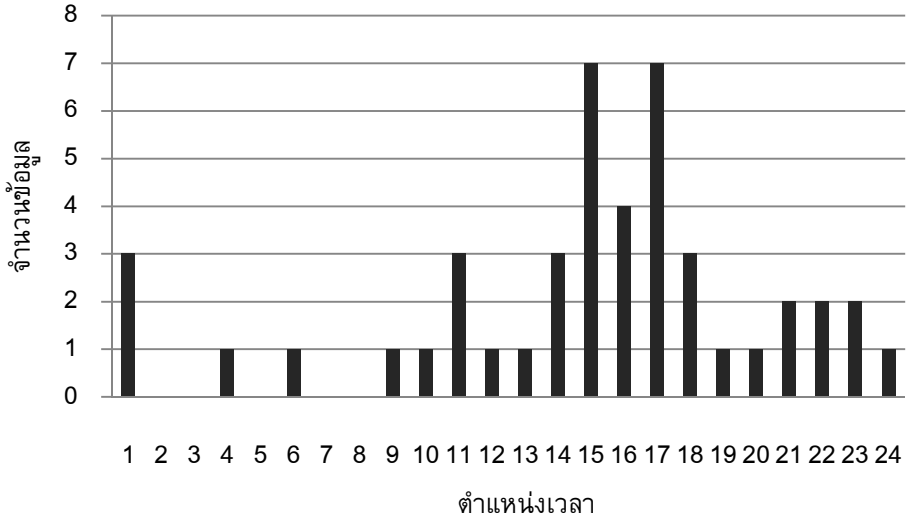
จากตารางที่ 4.2 แสดงให้เห็นว่าในแต่ละวันมีช่วงเวลาในการแสดงข้อมูลที่ใกล้เคียงกัน โดยในช่วงเวลา 1.00 น. – 6.00 น. มีการแสดงข้อมูลน้อยมากหรือไม่มีการแสดงข้อมูลเลย แต่ในช่วงเวลา 7.00 น. – 24.00 น. มีการแสดงข้อมูลในปริมาณที่มากกว่า อย่างไรก็ตามลักษณะการแสดงข้อมูลอาจไม่ได้เป็นไปตามลักษณะดังกล่าวทุกวัน แต่โดยภาพรวมแล้วจะมีลักษณะเช่นนี้ ทำให้ทราบได้ว่าการเวลาในการแสดงข้อมูลขึ้นอยู่กับช่วงเวลา

ทั้งนี้หากดูจำนวนการแสดงข้อมูลในแต่ละวันจะพบว่าจำนวนข้อมูลที่แสดงในแต่ละวันนั้นแตกต่างกัน จากตารางจะเห็นได้ว่าจำนวนข้อมูลที่แสดงในวันเสาร์และวันอาทิตย์จะมีจำนวนน้อยกว่าปกติ โดยวันจันทร์ถึงวันศุกร์จะมีจำนวนข้อมูลที่แสดงอยู่ในช่วง 13 – 31 ข้อมูล แต่วันเสาร์และวันอาทิตย์จะมีจำนวนข้อมูลที่แสดงเพียง 0 – 10 ข้อมูลเท่านั้น ซึ่ง

แตกต่างกันค่อนข้างชัดเจน ดังนั้นจำนวนข้อมูลที่แสดงจะขึ้นอยู่กับวันที่แสดงข้อมูลด้วย ตัวอย่างการแสดงข้อมูลที่ต่างกันในวันจันทร์ถึงวันศุกร์ กับวันเสาร์และวันอาทิตย์ โดยใช้ข้อมูลของข่าวเศรษฐกิจจาก BBC ในเดือนเมษายน แสดงดังภาพประกอบ 4.12 และ 4.13



ภาพประกอบ 4.12 ลักษณะการแสดงข้อมูลในวันจันทร์ถึงวันศุกร์



ภาพประกอบ 4.13 ลักษณะการแสดงข้อมูลในวันเสาร์และวันอาทิตย์

จากภาพประกอบ 4.12 และ 4.13 จะเห็นว่าเมื่อนำผลรวมของจำนวนข้อมูลที่แสดงในแต่ละช่วงมาเปรียบเทียบกัน จะพบว่ามีลักษณะการแสดงข้อมูลที่ต่างกัน โดยการแสดงข้อมูลวันจันทร์ถึงวันศุกร์จากภาพประกอบ 3.12 ช่วงเวลา 1.00 น. ถึง 6.00 น. มีการแสดงข้อมูลน้อย และช่วงเวลา 7.00 น. ถึง 24.00 น. มีการแสดงข้อมูลที่มากกว่าดังเช่นที่แสดงในกราฟ เหมือนกันกับที่แสดงในตารางที่ 4.2 แต่เมื่อแยกพิจารณาเฉพาะวันเสาร์และวันอาทิตย์จากภาพประกอบ 4.13 การแสดงข้อมูลจะมากในช่วง 14.00 น. ถึง 16.00 น. ส่วนช่วงเวลาอื่นจะเฉลี่ยกัน แตกต่างจากการแสดงข้อมูลวันจันทร์ถึงวันศุกร์อย่างเห็นได้ชัด

ด้วยเหตุผลที่ว่าเวลาในการแสดงข้อมูลขึ้นอยู่กับช่วงเวลา และจำนวนข้อมูลที่แสดงขึ้นอยู่กับวันที่แสดงข้อมูล จึงจำเป็นที่จะต้องมีการแบ่งกลุ่มของข้อมูลที่มีเวลาในการแสดงข้อมูลที่ต่างกันออกจากกัน เพื่อให้สามารถกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลตามลักษณะการแสดงข้อมูลของกลุ่มข้อมูลนั้น โดยวิธีการในการแบ่งกลุ่มของข้อมูลจะดูจากจำนวนข้อมูลที่แสดงในแต่ละวัน จำนวนข้อมูลที่มีจำนวนใกล้เคียงกันจะถูกจัดรวมให้อยู่ในกลุ่มข้อมูลเดียวกัน ตัวอย่างการแบ่งกลุ่มของข้อมูลแสดงดังตารางที่ 4.3

ตารางที่ 4.3 ตัวอย่างการแบ่งกลุ่มของแต่ละข้อมูลระหว่างวันที่ 1 – 21 เมษายน พ.ศ. 2553

วัน	พฤ.	ศ.	ส.	อา.	จ.	อ.	พ.	พฤ.	ศ.	ส.	อา.	จ.	อ.	พ.	พฤ.	ศ.	ส.	อา.	จ.	อ.	พ.
แหล่งข้อมูล	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**
Business, BBC	30	4	2	2	13	20	35	36	31	3	10	34	35	26	37	31	0	7	43	45	43
Entertainment, BBC	47	14	2	2	11	14	52	39	31	4	4	49	51	50	41	57	0	3	51	53	51
Top, BBC	31	27	6	11	29	3	59	47	53	9	20	71	66	65	75	71	11	29	69	70	64
US, BBC	8	6	2	1	6	12	18	10	8	1	6	12	14	13	16	9	1	5	17	17	22
World, BBC	11	16	2	1	3	6	10	13	17	1	5	16	20	14	13	15	1	2	11	18	14
Business, CNN	14	11	2	6	6	4	25	17	18	3	8	22	19	21	27	26	0	6	32	30	20
Entertainment, CNN	4	9	2	4	0	1	9	8	6	3	6	6	9	5	6	6	2	4	7	8	10
Top, CNN	37	35	19	23	38	5	53	33	37	30	37	46	44	52	44	50	27	31	43	47	51
US, CNN	36	49	22	24	42	6	47	45	55	46	52	58	64	53	67	69	50	50	70	58	62
World, CNN	26	25	18	5	25	19	23	32	16	7	9	24	41	25	20	18	4	7	22	31	19
Business, Reuters	27	39	5	13	23	6	32	28	32	5	14	29	28	36	40	32	3	2	16	29	38
Entertainment, Reuters	25	30	1	7	18	8	34	26	32	15	7	27	21	15	33	15	3	2	22	30	26
Top, Reuters	11	14	9	10	10	0	14	12	16	9	6	13	11	13	7	11	12	8	10	12	8
US, Reuters	23	27	13	18	16	10	24	26	23	18	30	38	34	28	35	26	11	20	29	25	24
World, Reuters	35	46	22	46	44	5	51	49	62	50	56	56	56	53	75	68	55	54	61	57	58

จากตารางที่ 4.3 ข้อมูลในตารางจะแทนจำนวนของข้อมูลที่แสดงในแต่ละวันของแต่ละแหล่งข้อมูล จะเห็นได้ว่าแหล่งข้อมูล BBC ทุกแหล่งข้อมูล แหล่งข้อมูล CNN ประเภทข่าวเศรษฐกิจ และข่าวรอบโลก และแหล่งข้อมูล Reuters ประเภทข่าวเศรษฐกิจและข่าวบันเทิง

มีการแบ่งกลุ่มของข้อมูลอย่างชัดเจน คือ ทุกวันเสาร์และวันอาทิตย์จำนวนการแสดงข้อมูลจะน้อยกว่าปกติ จึงแบ่งกลุ่มข้อมูลออกเป็น 2 กลุ่ม คือวันเสาร์อาทิตย์ และวันจันทร์ถึงวันศุกร์เช่นเดียวกัน

แต่ในส่วนของแหล่งข้อมูล CNN ประเภทข่าวบันเทิง ข่าวเด่น และข่าวของสหรัฐอเมริกา และแหล่งข้อมูล Reuters ประเภทข่าวเด่น ข่าวของสหรัฐอเมริกา และข่าวรอบโลก ไม่สามารถแบ่งเป็นกลุ่มที่ชัดเจนได้ จึงถือว่ามีลักษณะการแสดงข้อมูลที่เหมือนกันทุกวัน

4.2 ส่วนวิเคราะห์ข้อมูล (Data Analysis)

ในส่วนนี้มีหน้าที่วิเคราะห์ข้อมูลที่ได้จากส่วนแรกซึ่งทำการแปลงข้อมูลเรียบร้อยแล้ว โดยจะใช้ลักษณะการแสดงข้อมูลของแต่ละแหล่งข้อมูลเพื่อกำหนดตำแหน่งเวลาในการดึงข้อมูล ซึ่งจะเลือกตำแหน่งที่มีความล่าช้าในการดึงข้อมูลน้อยที่สุด การทำงานในส่วนนี้ประกอบไปด้วย การคำนวณความล่าช้าในการดึงข้อมูล และการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล โดยจะยกตัวอย่างการคำนวณความล่าช้าและการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลในตอนท้าย ซึ่งรายละเอียดมีดังต่อไปนี้

4.2.1 การคำนวณความล่าช้าในการดึงข้อมูล

ในขั้นตอนนี้จะคำนวณความล่าช้าในการดึงข้อมูลของแต่ละตำแหน่งเวลาในการดึงข้อมูล โดยใช้ลักษณะการแสดงข้อมูลที่ได้จากในส่วนแรกมาคำนวณหาผลต่างของระยะเวลาที่ข้อมูลแสดงกับตำแหน่งเวลาในการดึงข้อมูล ซึ่งตำแหน่งเวลาในการดึงข้อมูลที่ต่างกันก็จะมีความล่าช้าในการดึงข้อมูลที่ต่างกันไปด้วย สำหรับวิธีการในการคำนวณความล่าช้าประกอบด้วย 3 ขั้นตอน คือ

1) การกำหนดจำนวนครั้งในการดึงข้อมูล

เนื่องจากในการดึงข้อมูลจากแหล่งข้อมูลในแต่ละวันนั้นจะมีตำแหน่งเวลาในการดึงข้อมูลที่เหมือนกันทุกวัน ทำให้สิ่งแรกที่ต้องพิจารณาคือจำนวนครั้งในการดึงข้อมูล ซึ่งจำนวนครั้งจะเป็นตัวกำหนดตำแหน่งเวลาในการดึงข้อมูลว่าควรเป็นตำแหน่งเวลาใด โดยจำนวนครั้งในการดึงข้อมูลที่แตกต่างกันจะส่งผลให้ตำแหน่งเวลาในการดึงข้อมูลนั้นแตกต่างกันไปด้วย และเนื่องจากแต่ละตำแหน่งเวลาในการดึงข้อมูลมีผลต่อความล่าช้า ดังนั้นเพื่อให้สามารถคำนวณความล่าช้าในการดึงข้อมูลของแต่ละตำแหน่งเวลาในการดึงข้อมูลได้ จึงจำเป็นต้องกำหนดจำนวนครั้งในการดึงข้อมูลที่แน่นอนก่อนที่จะหาตำแหน่งเวลาในการดึงข้อมูล

2) การกำหนดจำนวนข้อมูลที่นำมาใช้

หลังจากกำหนดจำนวนครั้งในการดึงข้อมูลแล้ว ก่อนที่จะหาตำแหน่งเวลาในการดึงข้อมูลจะต้องกำหนดขอบเขตของข้อมูลที่จะนำมาใช้ว่าจะใช้จำนวนข้อมูลกี่วัน เนื่องจากจำนวนข้อมูลที่ต่างกันก็จะส่งผลให้ได้ตำแหน่งเวลาในการดึงข้อมูลที่แตกต่างกัน จึงต้องกำหนดจำนวนวันของข้อมูลที่นำมาใช้ก่อนนำไปคำนวณความล่าช้าเพื่อหาตำแหน่งที่เหมาะสมในการดึงข้อมูล เมื่อกำหนดจำนวนข้อมูลที่จะนำมาใช้แล้วจากนั้นจะนำข้อมูลทั้งหมดในแต่ละวันมารวมกัน โดยนำจำนวนการแสดงข้อมูลของแต่ละวันในชั่วโมงเดียวกันมารวมกันจะได้ข้อมูลที่ออกมาในรูปของผลรวมของจำนวนการแสดงข้อมูลในแต่ละชั่วโมง โดยขั้นตอนวิธีในการหาผลรวมจำนวนการแสดงข้อมูลแต่ละชั่วโมงแสดงดังภาพประกอบ 4.14 ตารางที่ 4.4 แสดงผลรวมจำนวนการแสดงข้อมูลของจำนวนวันที่เพิ่มขึ้นของข่าวเศรษฐกิจจากแหล่งข่าว BBC และลักษณะการแสดงข้อมูลของวันจันทร์ถึงวันศุกร์แสดงดังภาพประกอบ 4.15

```
1   Method Data_Combination (Data from each Publisher)
2       for each Publisher
3           get data η_i(ℓ) from database
4           for each hour i
5               for each day ℓ
6                   η_i = η_i + η_i(ℓ)
7               end for
8           end for
9       end for
10  end method
```

ภาพประกอบ 4.14 ขั้นตอนวิธีการหาผลรวมจำนวนการแสดงข้อมูลแต่ละชั่วโมง

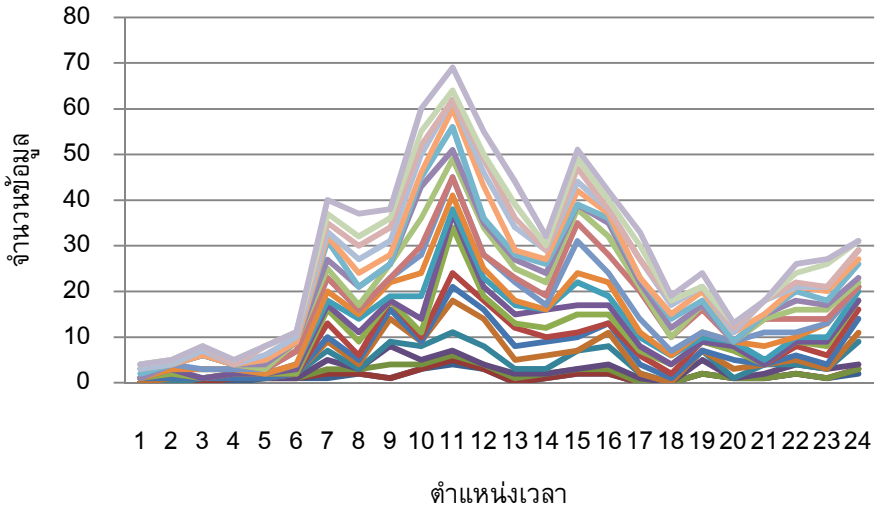
จากภาพประกอบ 4.14 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

บรรทัดที่ 2 – 4 คือ การนำเอาข้อมูลที่ได้จากการแปลงข้อมูลให้อยู่ในรูปจำนวนแล้วจากแต่ละแหล่งข้อมูล มาจัดให้อยู่ในรูปแบบวันและชั่วโมง และจัดเก็บในรูปอาร์เรย์ โดยจะเก็บค่าจำนวนข้อมูลที่แสดงของวันที่ ℓ ในชั่วโมงที่ *i*

บรรทัดที่ 5 – 9 คือ การรวมจำนวนการแสดงข้อมูลในชั่วโมงที่ *i* ของแต่ละวัน และเก็บไว้ในตัวแปร η_i จะได้ค่าของตัวแปร η_i เป็นผลรวมของจำนวนการแสดงข้อมูลของแต่ละชั่วโมง *i*

ตารางที่ 4.4 ผลรวมจำนวนการแสดงข้อมูลของจำนวนวันที่เพิ่มขึ้นของข่าวเศรษฐกิจจาก แหล่งข่าว BBC ในเดือนเมษายน พ.ศ. 2553 (เฉพาะวันจันทร์ถึงวันศุกร์)

เวลา / จำนวนวัน	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	1	1	1	2	1	3	4	3	0	1	2	2	0	0	2	1	1	2	1	2
2	0	0	0	1	1	1	2	2	1	3	5	3	0	1	2	2	0	0	2	1	1	2	1	3
3	0	0	1	1	1	1	3	3	4	4	6	4	1	2	3	3	0	0	2	1	1	2	1	3
4	0	0	1	1	1	1	5	3	8	5	7	4	2	2	3	4	1	0	5	1	2	4	3	4
5	0	0	1	1	1	2	7	3	9	8	11	8	3	3	7	8	2	0	7	1	4	4	3	9
6	0	1	1	1	1	2	9	4	14	9	18	14	5	6	7	11	2	0	7	3	4	5	3	11
7	1	1	1	1	1	2	10	5	16	9	21	16	8	9	10	13	4	1	7	5	4	6	4	14
8	1	2	1	2	2	2	13	6	18	10	24	18	12	10	11	13	6	2	9	7	4	8	6	16
9	1	2	1	2	2	2	16	9	18	11	34	19	13	12	15	15	7	4	9	7	4	9	8	18
10	1	3	1	2	2	3	17	11	18	14	37	21	15	16	17	17	8	4	9	8	4	9	9	18
11	1	3	3	3	2	4	18	14	19	19	38	23	17	16	22	19	10	6	10	9	5	10	10	20
12	1	3	3	3	2	4	20	15	22	24	41	25	18	16	24	22	11	6	11	9	8	10	13	21
13	1	4	3	3	3	7	23	16	23	28	45	28	22	17	31	24	14	7	11	9	11	11	13	21
14	1	4	6	4	3	7	23	16	23	30	45	28	23	19	35	28	20	10	16	9	14	14	14	21
15	1	4	6	4	3	9	25	17	26	36	49	34	25	22	38	32	21	10	17	9	14	16	16	22
16	1	4	6	4	4	9	27	21	26	43	51	35	27	24	39	35	22	12	17	9	15	18	17	23
17	2	4	6	4	5	9	31	21	26	45	56	36	28	26	39	36	23	14	18	9	15	20	18	26
18	3	4	6	4	5	9	32	24	28	46	60	43	29	27	42	37	23	15	20	11	15	21	20	27
19	3	4	7	4	6	10	33	27	31	50	62	46	34	29	44	38	27	17	21	11	18	21	21	29
20	4	5	8	4	8	11	35	30	34	52	62	49	36	29	47	38	27	18	21	12	18	22	21	29
21	4	5	8	5	8	11	37	32	36	55	64	50	39	30	49	40	30	18	21	13	18	24	26	31
22	4	5	8	5	8	11	40	37	38	60	69	55	44	32	51	42	33	19	24	13	18	26	27	31



ภาพประกอบ 4.15 ลักษณะการแสดงข้อมูลแต่ละชั่วโมงของจำนวนวันที่เพิ่มขึ้น

จากตารางที่ 4.4 แสดงผลรวมของจำนวนการแสดงข้อมูลในแต่ละชั่วโมง โดยเพิ่มจำนวนวันขึ้นเรื่อยๆ และภาพประกอบ 4.15 เป็นกราฟแสดงผลรวมจำนวนการแสดงข้อมูลจากตารางที่ 4.4 ซึ่งจะเห็นได้ว่าจำนวนวันที่เพิ่มมากขึ้น จะยิ่งเห็นได้ชัดว่าการแสดงข้อมูลในแต่ละชั่วโมงนั้นมีความแตกต่างกัน โดยช่วงใดที่มีการแสดงข้อมูลมากกราฟจะแสดงให้เห็นถึงจำนวนการแสดงข้อมูลที่เพิ่มขึ้นอย่างรวดเร็ว เนื่องจากเมื่อเพิ่มจำนวนวันมากขึ้นจำนวนข้อมูลในช่วงนั้นจะมากขึ้นตามไปด้วย และเพราะส่วนใหญ่ลักษณะการแสดงข้อมูลเป็นลักษณะเดียวกันทุกวัน จึงทำให้ช่วงเวลาที่มีการแสดงข้อมูลมากๆ เป็นช่วงเวลาเดียวกัน

3) การคำนวณความล่าช้าในการดึงข้อมูล

เมื่อกำหนดจำนวนครั้งและระยะเวลาของข้อมูลที่ใช้แล้ว ขั้นตอนต่อไปคือการคำนวณหาค่าความล่าช้าในการดึงข้อมูลของตำแหน่งเวลาในการดึงข้อมูล โดยจะหาความล่าช้าในการดึงข้อมูลจากตำแหน่งเวลาในการดึงข้อมูลที่เป็นไปได้ทั้งหมด ซึ่งตำแหน่งเวลาในการดึงข้อมูลที่เป็นไปได้ทั้งหมดจะขึ้นอยู่กับจำนวนครั้งในการดึงข้อมูล ดังนั้นก่อนการหาตำแหน่งเวลาในการดึงข้อมูลที่เป็นไปได้ทั้งหมดจะทำการกำหนดจำนวนครั้งในการดึงข้อมูลที่ต้องการก่อน แล้วจึงหาความล่าช้าในการดึงข้อมูลของแต่ละตำแหน่งเวลาในการดึงข้อมูล โดยขั้นตอนวิธีในการคำนวณความล่าช้าในการดึงข้อมูลแสดงดังภาพประกอบ 4.16

```
1   Method Retrieval_Points (Number of Retrieval m)
2       for each Publisher
3           get η_i
4           for τ_1 from 1 to 24 – m + 1
5               for τ_2 from τ_1 + 1 to 24 – m + 2
6                   ⋮
7                   for τ_m from τ_{m-1} + 1 to 24
8                       delay ← Calculate_Delay(τ_1, τ_2,..., τ_m)
9                       if min > delay
10                          τ ← τ_1, τ_2,..., τ_m
11                      end if
12                  end for
13                  ⋮
14              end for
15          end for
16      end for
17  end method

Procedure Calculate_Delay (τ_1, τ_2,..., τ_m)
18      for k from 1 to m
19          for i from τ_k + 1 to τ_{k+1}
20              if k = m
21                  τ_{k+1} ← τ_1 + 24
22                  i ← (i – 1)mod24 + 1
23              else
24                  delay ← delay + η_i(τ_{k+1} – i)
25              end if
26          end for
27      end for
```

ภาพประกอบ 4.16 ขั้นตอนวิธีในการคำนวณความล่าช้าในการดึงข้อมูล

จากภาพประกอบ 4.16 สามารถอธิบายขั้นตอนการทำงานได้ดังนี้

บรรทัดที่ 1 แสดงขั้นตอนวิธีการกำหนดตำแหน่งเวลาในการดึงข้อมูล โดยรับพารามิเตอร์ m เพื่อกำหนดจำนวนครั้งในการดึงข้อมูลซึ่งเป็นจำนวนเต็มบวกมีค่าอยู่ระหว่าง 1 – 24 เมื่อระบุจำนวนครั้งในการดึงข้อมูลแล้วจึงจะสามารถคำนวณตำแหน่งเวลาในการดึงข้อมูลที่เหมาะสมได้

บรรทัดที่ 2 – 3 เป็นการคำนวณความล่าช้าในการดึงข้อมูลของแต่ละแหล่งข้อมูล ซึ่งแหล่งข้อมูลที่ต่างกันจะมีการแสดงข้อมูลที่ต่างกันทำให้ตำแหน่งเวลาในการดึงข้อมูลมีความล่าช้าในการดึงข้อมูลที่ต่างกันด้วย

บรรทัดที่ 4 – 7 คือ การหาตำแหน่งเวลาในการดึงข้อมูลที่เป็นไปได้ทั้งหมด เพื่อนำตำแหน่งเวลาในการดึงข้อมูลที่เป็นไปได้เหล่านั้นมาคำนวณความล่าช้าในการดึงข้อมูล

บรรทัดที่ 8 คือ การนำตำแหน่งเวลาในการดึงข้อมูลที่ได้ไปคำนวณความล่าช้าในการดึงข้อมูล โดยส่งไปยัง Procedure Calculate_Delay และรับค่าความล่าช้าในการดึงข้อมูลกลับมา

บรรทัดที่ 9 – 11 เป็นการตรวจสอบความล่าช้าในการดึงข้อมูลของแต่ละตำแหน่งเวลาในการดึงข้อมูลว่าตำแหน่งเวลาในการดึงข้อมูลตำแหน่งใดที่ให้ความล่าช้าในการดึงข้อมูลน้อยที่สุด และเก็บตำแหน่งเวลาในการดึงข้อมูลเอาไว้

บรรทัดที่ 18 เป็นการกำหนดตำแหน่งเวลาในการดึงข้อมูลจากตำแหน่งเวลาทั้งหมด เพื่อนำไปใช้ในการคำนวณความล่าช้าในแต่ละช่วง

บรรทัดที่ 19 เป็นการกำหนดตำแหน่งแต่ละชั่วโมงในแต่ละช่วง $\tau_k + 1$ ถึง τ_{k+1} ของการดึงข้อมูล เนื่องจากข้อมูลในแต่ละชั่วโมงมีตำแหน่งเวลาในการดึงข้อมูลต่างกัน

บรรทัดที่ 20 – 22 เป็นการตรวจสอบว่าตำแหน่งในการดึงข้อมูลเป็นตำแหน่งสุดท้ายหรือไม่ เนื่องจากข้อมูลที่อยู่หลังจากตำแหน่งสุดท้ายของการดึงข้อมูลจะถูกดึงข้อมูลอีกครั้งในวันถัดไปซึ่งเป็นตำแหน่ง $\tau_1 + 24$ ดังนั้นเมื่อถึงตำแหน่งสุดท้ายจึงกำหนดให้ τ_{k+1} เป็นตำแหน่ง $\tau_1 + 24$

บรรทัดที่ 24 เป็นการคำนวณความล่าช้าในการดึงข้อมูล ซึ่งเป็นผลคูณระหว่างผลรวมของจำนวนข้อมูลที่แสดงในชั่วโมงนั้นกับความล่าช้าในการดึงข้อมูล นำผลลัพธ์ที่ได้ของแต่ละชั่วโมงมารวมกันจะได้ผลรวมความล่าช้าในการดึงข้อมูลของตำแหน่งเวลา $\tau_1, \tau_2, \ldots, \tau_m$

4.2.2 การกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

เมื่อคำนวณความล่าช้าของแต่ละตำแหน่งเวลาในการดึงข้อมูลที่เป็นไปได้ทั้งหมดแล้ว จะเห็นได้ว่าแต่ละตำแหน่งเวลาจะมีความล่าช้าในการดึงข้อมูลที่แตกต่างกัน ซึ่งตำแหน่งเวลาในการดึงข้อมูลที่ดีควรเป็นตำแหน่งเวลาในการดึงข้อมูลที่มีความล่าช้าในการดึงข้อมูลน้อยที่สุด

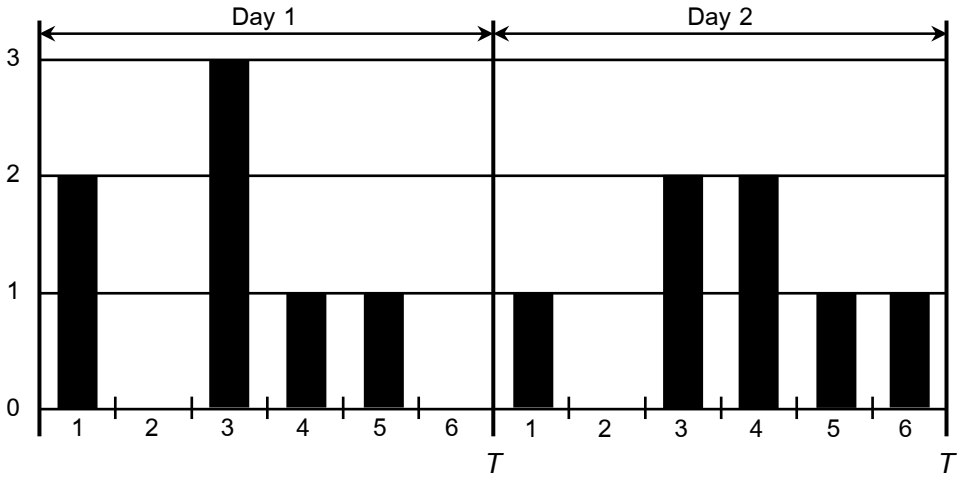
อย่างไรก็ตามหากพิจารณาถึงความล่าช้าในการดึงข้อมูลที่ได้จากขั้นตอนวิธีในการคำนวณความล่าช้าในการดึงข้อมูลจากภาพประกอบ 4.16 จะพบว่าข้อมูลที่นำมาใช้เป็นผลรวมของจำนวนการแสดงข้อมูลในแต่ละชั่วโมง ดังนั้นข้อมูลที่ได้จึงเป็นข้อมูลของทุกวันรวมกัน ทำให้ตำแหน่งเวลาในการดึงข้อมูลที่มีความล่าช้าในการดึงข้อมูลน้อยที่สุดอาจไม่ใช่ตำแหน่งเวลาในการดึงข้อมูลที่ดีที่สุดเมื่อพิจารณาทีละวัน เนื่องจากข้อมูลที่นำมาใช้เป็นข้อมูลที่รวมกันความล่าช้าที่ได้จึงเสมือนเป็นความล่าช้าโดยเฉลี่ย

และเนื่องจากในการกำหนดตำแหน่งเวลาในการดึงข้อมูลจะเป็นการกำหนดให้เป็นตำแหน่งเวลาเดียวกันทุกวัน ดังนั้นตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลจึงควรเป็นตำแหน่งเวลาที่มีความล่าช้ารวมทุกวันน้อยที่สุดและจะใช้ตำแหน่งเวลานี้ในการดึงข้อมูลทุกวัน

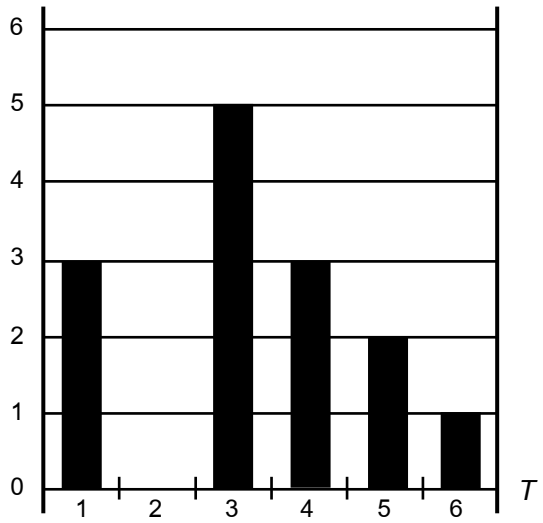
4.2.3 ตัวอย่างการคำนวณความล่าช้าและการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

เพื่อให้เข้าใจการคำนวณความล่าช้าและการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลได้ดียิ่งขึ้น จะอธิบายโดยนำเสนอตัวอย่างประกอบด้วย ตัวอย่างที่ 1 แสดงการหาผลรวมของจำนวนการแสดงข้อมูลในแต่ละช่วง ตัวอย่างที่ 2 แสดงการคำนวณความล่าช้า และตัวอย่างที่ 3 แสดงการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

ตัวอย่างที่ 1 กำหนดให้จำนวนวันของข้อมูลที่ใช้มี 2 วัน โดยในแต่ละวันแบ่งออกเป็น 6 ช่วง และมีลักษณะการแสดงข้อมูลดังภาพประกอบ 4.17 การหาผลรวมของจำนวนการแสดงข้อมูลในแต่ละช่วงจะแสดงดังภาพประกอบ 4.18



ภาพประกอบ 4.17 ลักษณะการแสดงข้อมูลในแต่ละวัน



ภาพประกอบ 4.18 ผลรวมของจำนวนการแสดงข้อมูลในแต่ละช่วง

จากภาพประกอบ 4.18 แสดงผลรวมของจำนวนการแสดงข้อมูล โดยในแต่ละช่วงจะนำการแสดงข้อมูลของวันที่ 1 และวันที่ 2 มารวมกัน ดังนี้

ช่วงที่ 1 จะได้ผลรวมของจำนวนการแสดงข้อมูล 2 + 1 = 3

ช่วงที่ 2 จะได้ผลรวมของจำนวนการแสดงข้อมูล 0 + 0 = 0

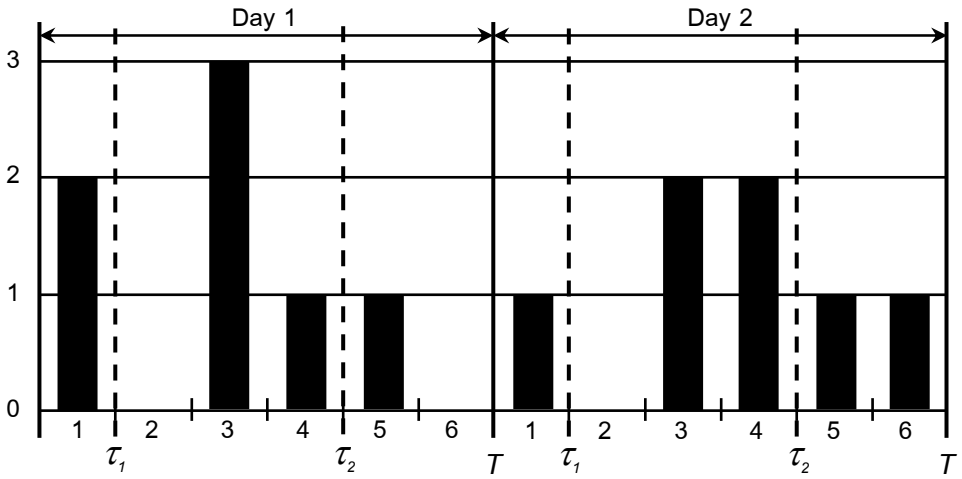
ช่วงที่ 3 จะได้ผลรวมของจำนวนการแสดงข้อมูล 3 + 2 = 5

ช่วงที่ 4 จะได้ผลรวมของจำนวนการแสดงข้อมูล 1 + 2 = 3

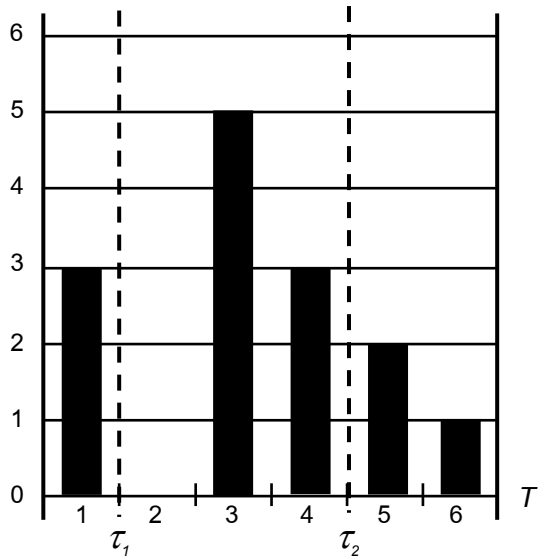
ช่วงที่ 5 จะได้ผลรวมของจำนวนการแสดงข้อมูล 1 + 1 = 2

ช่วงที่ 6 จะได้ผลรวมของจำนวนการแสดงข้อมูล 0 + 1 = 1

ตัวอย่างที่ 2 สมมติให้เก็บรวบรวมข้อมูลจากแหล่งข้อมูล 2 วัน โดยมีลักษณะการแสดงข้อมูลดังภาพประกอบ 4.17 และกำหนดให้ตัวรวบรวมข่าวสารดึงข้อมูลจากแหล่งข้อมูล 2 ครั้ง คือ τ_1 = 1 และ τ_2 = 4



ภาพประกอบ 4.19 ตำแหน่งเวลาในการดึงข้อมูลในแต่ละวัน



ภาพประกอบ 4.20 ตำแหน่งเวลาในการดึงข้อมูลเมื่อนำข้อมูลมารวมกัน

จะได้ว่า ตำแหน่งเวลาต่างๆ คือ $i \in T$ = {1,2,3,4,5,6}

จำนวนการแสดงข้อมูลในแต่ละช่วงของวันที่ 1 คือ

$\eta_1(1), \eta_2(1), \eta_3(1), \eta_4(1), \eta_5(1), \eta_6(1)$ = 2,0,3,1,1,0 ตามลำดับ

จำนวนการแสดงข้อมูลในแต่ละช่วงของวันที่ 2 คือ

$\eta_1(2), \eta_2(2), \eta_3(2), \eta_4(2), \eta_5(2), \eta_6(2)$ = 1,0,2,2,1,1 ตามลำดับ

ผลรวมของจำนวนการแสดงข้อมูลในแต่ละช่วง คือ

$\eta_1, \eta_2, \eta_3, \eta_4, \eta_5, \eta_6$ = 3,0,5,3,2,1 ตามลำดับ

ความล่าช้าในการดึงข้อมูล 2 ครั้ง ณ ตำแหน่งเวลา τ_1 = 1 และ τ_2 = 4 ในแต่ละช่วงเวลาคำนวณได้ดังต่อไปนี้

$D_1 = \eta_1(\tau_1 - 1) = 3(1 - 1) = 0$

$D_2 = \eta_2(\tau_2 - 2) = 0(4 - 2) = 0$

$D_3 = \eta_3(\tau_2 - 3) = 5(4 - 3) = 5$

$D_4 = \eta_4(\tau_2 - 4) = 3(4 - 4) = 0$

$D_5 = \eta_5(\tau_1 + 6 - 5) = 2(1 + 6 - 5) = 4$

$D_6 = \eta_6(\tau_1 + 6 - 6) = 1(1 + 6 - 6) = 1$

โดย D_i คือ ความล่าช้าในการดึงข้อมูลของข้อมูลในช่วง i

ดังนั้นความล่าช้าในการดึงข้อมูล ณ ตำแหน่งเวลา τ_1 = 1 และ τ_2 = 4 คือ

$D_1 + D_2 + D_3 + D_4 + D_5 + D_6 = 0 + 0 + 5 + 0 + 4 + 1 = 10$

ตัวอย่างที่ 3 สมมติให้เก็บรวบรวมข้อมูลจากแหล่งข้อมูล 2 วัน โดยมีลักษณะการแสดงข้อมูลดังภาพประกอบ 4.17 และกำหนดให้ตัวรวบรวมข่าวสารดึงข้อมูลจากแหล่งข้อมูล 2 ครั้ง จะทำการหาว่าตำแหน่งเวลาในการดึงข้อมูลตำแหน่งใดที่มีความล่าช้าในการดึงข้อมูลน้อยที่สุด

จากที่กำหนดให้ตัวรวบรวมข่าวสารดึงข้อมูลจากแหล่งข้อมูล 2 ครั้ง จะได้ว่าตำแหน่งเวลาในการดึงข้อมูลที่เป็นไปได้ทั้งหมดมี $\binom{6}{2}$ = 15 ตำแหน่ง ดังต่อไปนี้

(τ_1, τ_2) = (1,2), (1,3), (1,4), (1,5), (1,6), (2,3), (2,4), (2,5), (2,6), (3,4), (3,5), (3,6), (4,5), (4,6), (5,6)

ทั้งนี้จะไม่นับตำแหน่งเวลาที่ซ้ำกันและสลับกัน เช่น (1,1), (1,2), (2,1) เพราะถือว่าเป็นตำแหน่งเดียวกัน โดยลำดับมีความสำคัญและจะกำหนดให้ τ_1 และ τ_2 เป็นตำแหน่งเวลาก่อนและหลังตามลำดับ

นำแต่ละตำแหน่งเวลาไปคำนวณความล่าช้าดังตัวอย่างที่ 2 จะได้ว่า

τ_1 = 1 และ τ_2 = 2 มีความล่าช้าในการดึงข้อมูลเท่ากับ 34

τ_1 = 1 และ τ_2 = 3 มีความล่าช้าในการดึงข้อมูลเท่ากับ 14

τ_1 = 1 และ τ_2 = 4 มีความล่าช้าในการดึงข้อมูลเท่ากับ 10

τ_1 = 1 และ τ_2 = 5 มีความล่าช้าในการดึงข้อมูลเท่ากับ 14

τ_1 = 1 และ τ_2 = 6 มีความล่าช้าในการดึงข้อมูลเท่ากับ 23

τ_1 = 2 และ τ_2 = 3 มีความล่าช้าในการดึงข้อมูลเท่ากับ 23

τ_1 = 2 และ τ_2 = 4 มีความล่าช้าในการดึงข้อมูลเท่ากับ 16

τ_1 = 2 และ τ_2 = 5 มีความล่าช้าในการดึงข้อมูลเท่ากับ 18

τ_1 = 2 และ τ_2 = 6 มีความล่าช้าในการดึงข้อมูลเท่ากับ 26

τ_1 = 3 และ τ_2 = 4 มีความล่าช้าในการดึงข้อมูลเท่ากับ 17

τ_1 = 3 และ τ_2 = 5 มีความล่าช้าในการดึงข้อมูลเท่ากับ 12

τ_1 = 3 และ τ_2 = 6 มีความล่าช้าในการดึงข้อมูลเท่ากับ 14

τ_1 = 4 และ τ_2 = 5 มีความล่าช้าในการดึงข้อมูลเท่ากับ 18

τ_1 = 4 และ τ_2 = 6 มีความล่าช้าในการดึงข้อมูลเท่ากับ 16

τ_1 = 5 และ τ_2 = 6 มีความล่าช้าในการดึงข้อมูลเท่ากับ 25

จาก τ_1 = 1 และ τ_2 = 4 มีความล่าช้าในการดึงข้อมูลน้อยที่สุด ดังนั้นจึงเลือก τ_1 = 1 และ τ_2 = 4 เป็นตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล ณ ตำแหน่งเวลาเดียวกันทุกวัน

บทที่ 5

ประสิทธิภาพของกลไกและผลการศึกษา

ในบทนี้จะนำเสนอประสิทธิภาพและผลการศึกษาของกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS โดยรายละเอียดจะประกอบด้วย ข้อมูลที่ใช้ในการทดลองและระยะเวลาในการเรียนรู้ข้อมูลที่เหมาะสม ประสิทธิภาพของกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS จำนวนครั้งที่เหมาะสมในการดึงข้อมูล และระยะเวลาในการปรับปรุงข้อมูลที่เหมาะสม

5.1 ข้อมูลที่ใช้ในการทดลองและระยะเวลาในการเรียนรู้ข้อมูลที่เหมาะสม

5.1.1 ข้อมูลที่ใช้ในการทดลอง

ข้อมูลที่ใช้ในการพัฒนากลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS เป็นข้อมูลที่ได้เก็บรวบรวมจากเว็บไซต์ข่าวสารที่ให้บริการ RSS ได้แก่ BBC, CNN, และ REUTERS โดยแบ่งประเภทของข่าวแบ่งออกเป็น 5 ประเภท คือ ข่าวเศรษฐกิจ ข่าวบันเทิง ข่าวเด่น ข่าวสหรัฐอเมริกา และข่าวรอบโลก และเก็บข้อมูล 3 เดือน ตั้งแต่เดือนเมษายน พ.ศ.2553 ถึงเดือนมิถุนายน พ.ศ.2553 มีจำนวนข่าวทั้งหมด 42,375 ข่าว แบ่งออกเป็นข่าวของ BBC รวม 17,599 ข่าว CNN รวม 10,638 ข่าว และ REUTERS รวม 14,138 ข่าว จำนวนข่าวประเภทต่างๆ ของแต่ละแหล่งข้อมูลแสดงดังตาราง 5.1

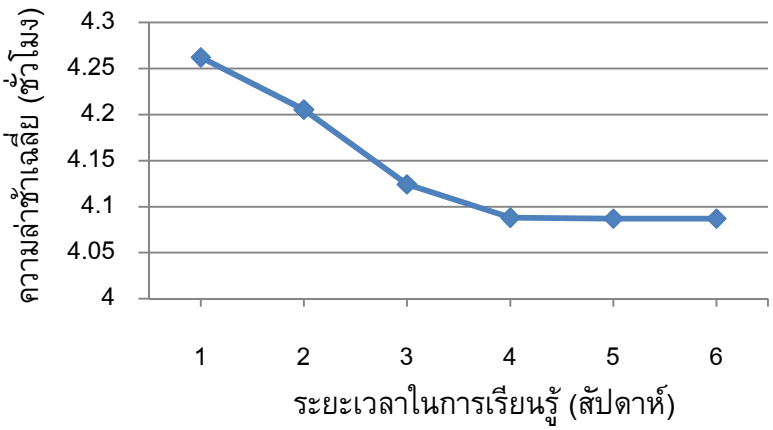
ตารางที่ 5.1 จำนวนข่าวประเภทต่างๆ ของแต่ละแหล่งข้อมูล

ประเภท / แหล่งข้อมูล	Business	Entertainment	Top news	US	World	รวม
BBC	2,713	1,175	10,429	2,299	983	17,599
CNN	2,818	864	3,044	2,032	1,880	10,638
REUTERS	3,270	1,258	3,854	1,987	3,769	14,138

5.1.2 ระยะเวลาในการเรียนรู้ข้อมูลที่เหมาะสม

เนื่องจากในการเรียนรู้การกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลจะต้องเรียนรู้ข้อมูลในระยะเวลาหนึ่งก่อนจึงจะสามารถกำหนดตำแหน่งเวลาที่เหมาะสมได้ ซึ่งระยะเวลาในการเรียนรู้ข้อมูลที่แตกต่างกันจะส่งผลให้ตำแหน่งเวลาในการดึงข้อมูลที่ได้นั้นมีความแตกต่างกันด้วย

เพื่อหาระยะเวลาในการเรียนรู้ข้อมูลที่เหมาะสมจึงทดลองเรียนรู้ข้อมูลด้วยระยะเวลาที่แตกต่างกัน แต่ละระยะเวลาก็จะมีตำแหน่งเวลาในการดึงข้อมูลที่ต่างกันแล้วนำตำแหน่งเวลาที่ได้นั้นไปใช้ดึงข้อมูลของข้อมูลในส่วนที่เหลือ และคำนวณความล่าช้าในการดึงข้อมูลของแต่ละตำแหน่งเวลา แล้วนำระยะเวลาในการเรียนรู้กับความล่าช้าในการดึงข้อมูลมาวาดกราฟแสดงความสัมพันธ์ จะได้กราฟดังภาพประกอบ 5.1



ภาพประกอบ 5.1 ความสัมพันธ์ระหว่างระยะเวลาในการเรียนรู้กับความล่าช้าในการดึงข้อมูล

จากภาพประกอบ 5.1 จะเห็นได้ว่าเมื่อระยะเวลาในการเรียนรู้ข้อมูลเพิ่มมากขึ้นความล่าช้าในการดึงข้อมูลจะมีแนวโน้มที่ลดลงเรื่อยๆ เนื่องจากยิ่งระยะเวลาในการเรียนรู้ยิ่งเพิ่มมากขึ้นข้อมูลในการเรียนรู้ก็จะมีมากขึ้นทำให้ตำแหน่งเวลาในการดึงข้อมูลที่ได้มีความล่าช้าในการดึงข้อมูลที่ลดลงเมื่อนำไปใช้

อย่างไรก็ตามจะพบว่าเมื่อระยะเวลาผ่านไประยะหนึ่งความล่าช้าในการดึงข้อมูลที่ได้จะมีค่าคงที่ เนื่องจากข้อมูลที่เพิ่มเข้ามาเป็นข้อมูลที่มีลักษณะคล้ายกับข้อมูลเดิม ดังนั้นเมื่อถึงระยะเวลาหนึ่งตำแหน่งในการดึงข้อมูลจึงเป็นตำแหน่งเดิม ทำให้ความล่าช้าในการดึงข้อมูลมีค่าเท่าเดิม ในการกำหนดระยะเวลาในการเรียนรู้ข้อมูลที่เหมาะสมจึงใช้วิธีกำหนดโดยหาระยะเวลาในการเรียนรู้ข้อมูลที่ความล่าช้าในการดึงข้อมูลไม่ลดลงมากไปกว่าเดิมแล้ว ซึ่งจากกราฟความสัมพันธ์จะแสดงให้เห็นว่าที่ระยะเวลาประมาณ 4 สัปดาห์ หรือ 1 เดือน เหมาะสมที่

จะใช้เป็นระยะเวลาในการเรียนรู้ข้อมูลเนื่องจากหากใช้ระยะเวลาเรียนรู้ที่มากกว่านี้ก็ไม่ทำให้ความล่าช้าในการดึงข้อมูลลดลง หรือตำแหน่งเวลาในการดึงข้อมูลที่ได้จะยังคงเป็นตำแหน่งเวลาเดิมนั่นเอง

5.2 ประสิทธิภาพของกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS

5.2.1 การออกแบบการทดลอง

เนื่องจากข้อมูลที่จะใช้ในการทดลองได้จากการเก็บข้อมูลในเดือนเมษายน พ.ศ. 2553 ถึงเดือนมิถุนายน พ.ศ.2553 ซึ่งมีระยะเวลาทั้งหมด 3 เดือน และจากระยะเวลาในการเรียนรู้ข้อมูลที่เหมาะสมคือระยะเวลาประมาณ 4 สัปดาห์ หรือ 1 เดือน จึงนำข้อมูล 1 เดือนแรกมาเรียนรู้เพื่อหาตำแหน่งเวลาในการดึงข้อมูล และอีก 2 เดือนที่เหลือเพื่อคำนวณความล่าช้าในการดึงข้อมูลที่ได้จากตำแหน่งเวลาในการดึงข้อมูลในเดือนแรก

ในการทดลองเพื่อศึกษาประสิทธิภาพของกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS ได้ทดลองโดยเปรียบเทียบประสิทธิภาพความล่าช้าในการดึงข้อมูลระหว่างรูปแบบในการดึงข้อมูล 2 รูปแบบ คือ

1) การดึงข้อมูลที่กำหนดตำแหน่งเวลาในการดึงข้อมูลจากการหาความสัมพันธ์ระหว่างจำนวนข้อมูลกับตำแหน่งเวลาในการดึงข้อมูลหรือเรียกว่า Retrieval – scheduling (Sia and Cho, 2007)

2) กลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS (Determining Optimal Retrieval Points Mechanism for RSS Documents : DORPM) เป็นรูปแบบการดึงข้อมูลโดยการกำหนดตำแหน่งเวลาในการดึงข้อมูลจากตำแหน่งที่มีความล่าช้าในการดึงข้อมูลน้อยที่สุด ซึ่งเป็นรูปแบบที่ได้นำเสนอ

5.2.2 ผลการศึกษา

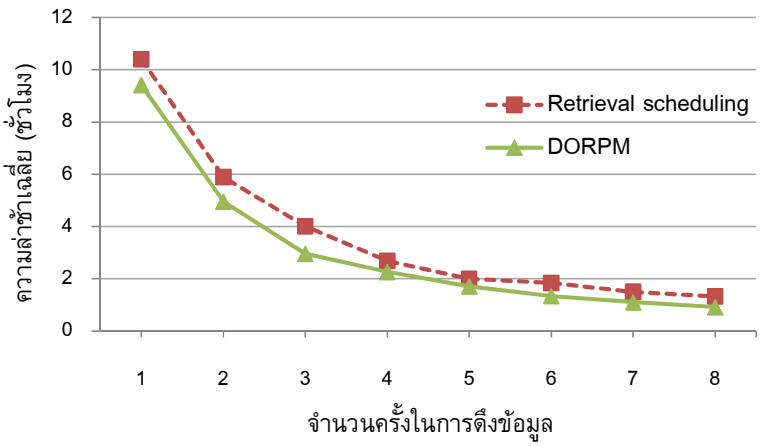
เนื่องจากข้อมูลที่ได้จากการเก็บข้อมูลมาจาก 3 แหล่งข้อมูลคือ BBC, CNN และ REUTERS จึงทดลองกับข้อมูลทีละแหล่งข้อมูลซึ่งแต่ละแหล่งข้อมูลจะมีประเภทของข่าวแบ่งออกเป็น 5 ประเภท แต่ละประเภทจะมีลักษณะในการแสดงข้อมูลที่ต่างกัน ดังนั้นชุดข้อมูลที่ใช้ในการทดลองจึงมีทั้งหมด 15 ชุด ซึ่งจะเปรียบเทียบประสิทธิภาพความล่าช้าในการดึงข้อมูลของแต่ละรูปแบบทั้ง 15 ชุดข้อมูล โดยทดลองทีละ 5 ชุด จากข่าวแต่ละประเภทที่มาจากแหล่งข้อมูลเดียวกัน ได้ผลการศึกษาดังต่อไปนี้

1) ข้อมูลจากแหล่งข้อมูล BBC

ความล่าช้าในการดึงข้อมูลแบบ Retrieval scheduling และแบบ DORPM ของแหล่งข้อมูล BBC แสดงดังตาราง 5.2 และกราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ยแสดงดังภาพประกอบ 5.2

ตารางที่ 5.2 ความล่าช้าในการดึงข้อมูลแต่ละรูปแบบของแหล่งข้อมูล BBC

รูปแบบ	จำนวนครั้งในการดึงข้อมูล	ผลรวมความล่าช้าในการดึงข้อมูล (ชั่วโมง)					ความล่าช้าเฉลี่ย (ชั่วโมง)
		Business	Entertainment	Top news	US	World	
Retrieval scheduling	1	20,819	11,879	66,657	17,707	7,387	10.399
	2	9,496	11,031	36,709	9,353	3,900	5.890
	3	7,881	10,195	21,065	6,179	2,678	4.011
	4	4,895	4,013	16,431	4,085	2,704	2.685
	5	3,294	4,007	12,197	3,153	1,249	1.997
	6	2,902	4,430	10,177	2,902	1,624	1.841
	7	2,844	3,710	8,450	2,201	746	1.500
	8	2,829	3,047	7,283	1,694	1,033	1.327
DORPM	1	17,021	5,817	64,573	18,207	7,057	9.415
	2	9,145	3,068	35,009	8,765	3,295	4.954
	3	5,090	2,012	20,783	5,351	2,154	2.957
	4	3,740	1,592	16,242	3,987	1,485	2.260
	5	2,906	1,128	12,373	3,031	1,060	1.713
	6	2,218	842	9,695	2,414	827	1.337
	7	1,920	654	7,868	1,961	757	1.100
	8	1,617	553	6,505	1,655	630	0.916



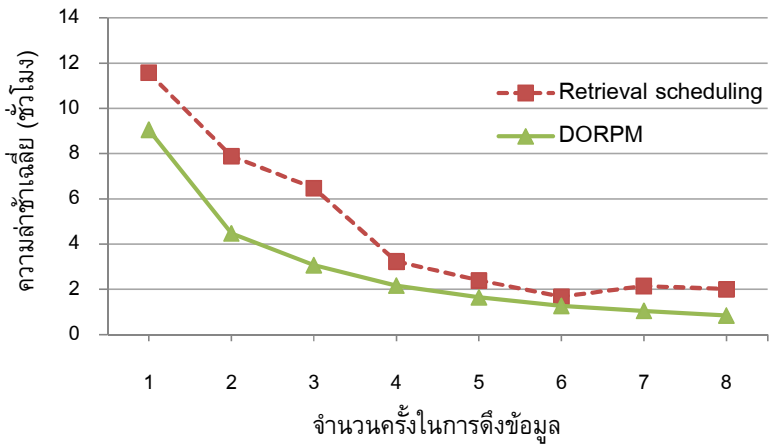
ภาพประกอบ 5.2 ความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ย โดยใช้ข้อมูลจากแหล่งข้อมูล BBC

2) ข้อมูลจากแหล่งข้อมูล CNN

ความล่าช้าในการดึงข้อมูลแบบ Retrieval scheduling และแบบ DORPM ของแหล่งข้อมูล CNN แสดงดังตาราง 5.3 และกราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ยแสดงดังภาพประกอบ 5.3

ตารางที่ 5.3 ความล่าช้าในการดึงข้อมูลแต่ละรูปแบบของแหล่งข้อมูล CNN

รูปแบบ	จำนวนครั้งในการดึงข้อมูล	ผลรวมความล่าช้าในการดึงข้อมูล (ชั่วโมง)					ความล่าช้าเฉลี่ย (ชั่วโมง)
		Business	Entertainment	Top news	US	World	
Retrieval scheduling	1	23,654	8,419	19,917	12,328	13,441	11.566
	2	21,867	6,907	10,467	7,162	6,600	7.884
	3	21,867	6,366	6,688	4,502	4,075	6.470
	4	4,439	5,833	5,119	3,311	3,017	3.231
	5	2,729	5,305	3,439	2,582	2,020	2.391
	6	2,021	2,449	2,821	2,215	1,713	1.669
	7	6,941	2,399	2,183	1,500	1,412	2.147
	8	5,657	2,397	1,923	2,253	1,204	1.998
DORPM	1	13,089	3,703	19,938	12,563	11,575	9.054
	2	6,573	1,976	9,586	6,183	5,715	4.467
	3	4,601	1,293	6,439	4,223	4,036	3.063
	4	3,204	996	4,542	2,907	2,861	2.158
	5	2,433	804	3,449	2,100	2,271	1.645
	6	1,834	603	2,696	1,701	1,660	1.263
	7	1,467	504	2,193	1,471	1,354	1.040
	8	1,163	380	1,814	1,181	1,098	0.838



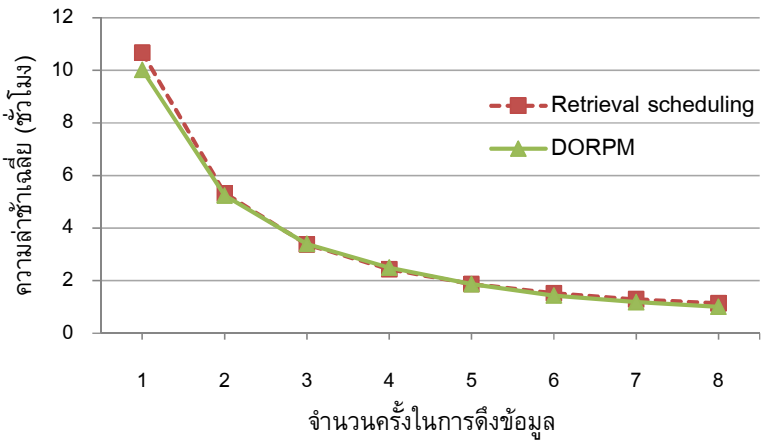
ภาพประกอบ 5.3 ความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ย โดยใช้ข้อมูลจากแหล่งข้อมูล CNN

3) ข้อมูลจากแหล่งข้อมูล REUTERS

ความล่าช้าในการดึงข้อมูลแบบ Retrieval scheduling และแบบ DORPM ของแหล่งข้อมูล REUTERS แสดงดังตาราง 5.4 และกราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ยแสดงดังภาพประกอบ 5.4

ตารางที่ 5.4 ความล่าช้าในการดึงข้อมูลแต่ละรูปแบบของแหล่งข้อมูล REUTERS

รูปแบบ	จำนวนครั้งในการดึงข้อมูล	ผลรวมความล่าช้าในการดึงข้อมูล (ชั่วโมง)					ความล่าช้าเฉลี่ย (ชั่วโมง)
		Business	Entertainment	Top news	US	World	
Retrieval scheduling	1	19,989	7,756	25,728	16,986	21,849	10.670
	2	11,059	3,993	12,722	7,166	11,133	5.326
	3	6,512	2,484	7,843	4,747	7,584	3.372
	4	4,686	1,650	6,056	3,317	5,338	2.433
	5	3,772	1,462	4,479	2,378	4,074	1.869
	6	2,851	991	3,855	2,188	3,291	1.523
	7	2,409	1,274	2,827	2,084	2,550	1.288
	8	1,920	808	2,546	2,057	2,506	1.137
DORPM	1	19,902	7,737	24,920	12,720	21,319	10.010
	2	10,444	4,085	12,528	7,166	11,105	5.240
	3	6,516	2,417	8,634	4,606	7,243	3.400
	4	4,837	1,650	6,131	3,663	5,345	2.500
	5	3,712	1,242	4,534	2,485	4,205	1.870
	6	2,937	1,052	3,391	1,928	3,145	1.439
	7	2,328	884	2,857	1,603	2,591	1.186
	8	1,991	764	2,441	1,463	2,097	1.012



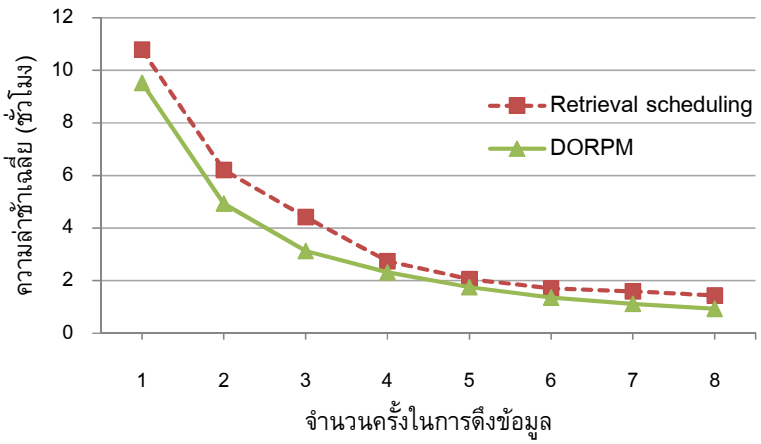
ภาพประกอบ 5.4 ความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ย โดยใช้ข้อมูลจากแหล่งข้อมูล REUTERS

4) ข้อมูลจากทั้ง 3 แหล่งข้อมูลรวมกัน

ความล่าช้าในการดึงข้อมูลแบบ Retrieval scheduling และแบบ DORPM ของแหล่งข้อมูลทั้ง 3 รวมกันแสดงดังตาราง 5.5 และกราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ยแสดงดังภาพประกอบ 5.5

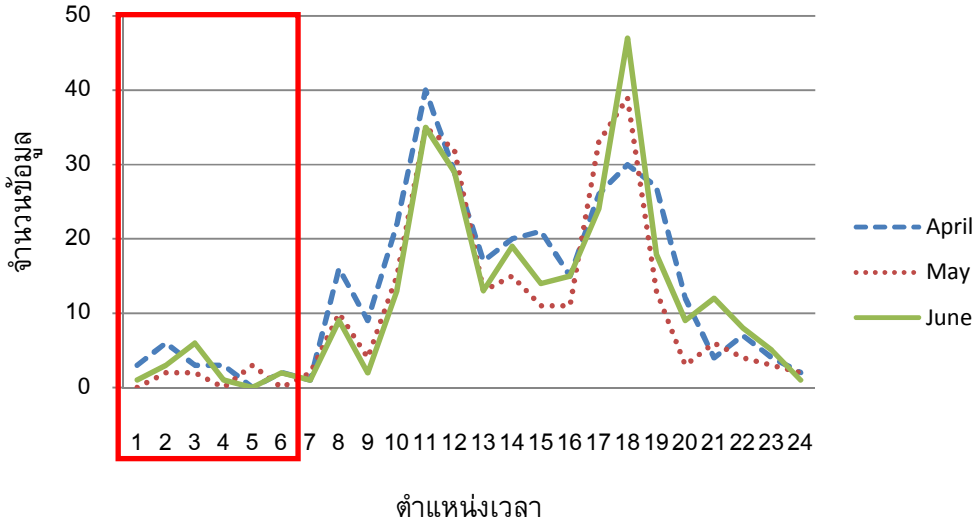
ตารางที่ 5.5 ความล่าช้าในการดึงข้อมูลแต่ละรูปแบบของแหล่งข้อมูลทั้ง 3 รวมกัน

รูปแบบ	จำนวนครั้งในการดึงข้อมูล	ผลรวมความล่าช้าในการดึงข้อมูล (ชั่วโมง)					ความล่าช้าเฉลี่ย (ชั่วโมง)
		Business	Entertainment	Top news	US	World	
Retrieval scheduling	1	64,462	28,054	112,302	47,021	42,677	10.772
	2	42,422	21,931	59,898	23,681	21,633	6.202
	3	36,260	19,045	35,596	15,428	14,337	4.413
	4	14,020	11,496	27,606	10,713	11,059	2.739
	5	9,795	10,774	20,115	8,113	7,343	2.053
	6	7,774	7,870	16,853	7,305	6,628	1.698
	7	12,194	7,383	13,460	5,785	4,708	1.592
	8	10,406	6,252	11,752	6,004	4,743	1.432
DORPM	1	50,012	17,257	109,431	43,490	39,951	9.515
	2	26,162	9,129	57,123	22,114	20,115	4.925
	3	16,207	5,722	35,856	14,180	13,433	3.123
	4	11,781	4,238	26,915	10,557	9,691	2.311
	5	9,051	3,174	20,356	7,616	7,536	1.746
	6	6,989	2,497	15,782	6,043	5,632	1.351
	7	5,715	2,042	12,918	5,035	4,702	1.112
	8	4,771	1,697	10,760	4,299	3,825	0.927



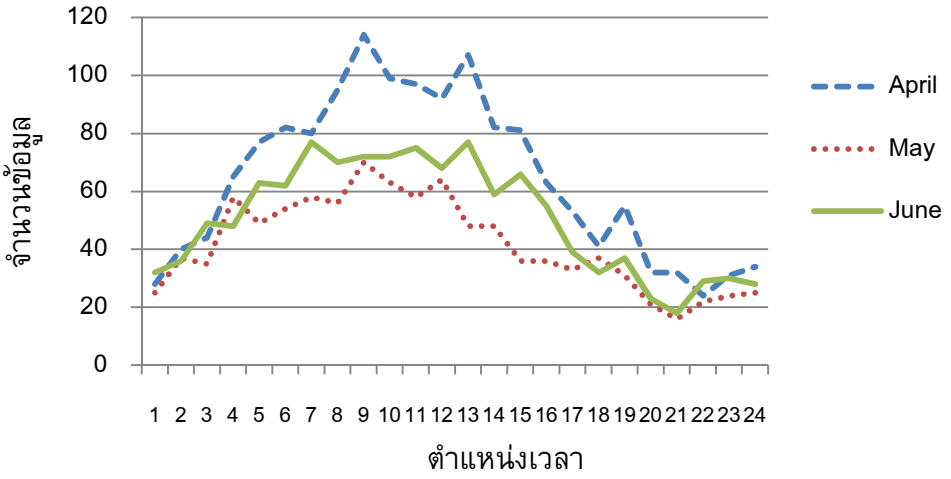
ภาพประกอบ 5.5 ความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ย โดยใช้ข้อมูลจากแหล่งข้อมูลทั้ง 3 รวมกัน

จากภาพประกอบ 5.3 จะเห็นได้ว่ากราฟความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ย ที่จำนวนครั้งในการดึงข้อมูล 2, 3, 7 และ 8 ครั้ง ความล่าในการดึงข้อมูลของรูปแบบ Retrieval scheduling จะมากกว่าปกติ อันเนื่องมาจากบางช่วงเวลาของแหล่งข้อมูลนี้แทบจะไม่มีการแสดงข้อมูลเลย ตัวอย่างแสดงดังภาพประกอบ 5.6 ทำให้การหาความสัมพันธ์ระหว่างตำแหน่งเวลาในการดึงข้อมูลกับจำนวนข้อมูลผิดพลาด จึงทำให้ได้ตำแหน่งเวลาในการดึงข้อมูลที่ไม่เหมาะสมและเมื่อนำตำแหน่งเวลาที่ได้ไปดึงข้อมูลก็จะมีความล่าช้าในการดึงข้อมูลสูง



ภาพประกอบ 5.6 ลักษณะการแสดงข้อมูลของข่าวบันเทิงจากแหล่งข้อมูล CNN

และจากภาพประกอบ 5.4 จะเห็นได้ว่ากราฟความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ยของรูปแบบในการดึงข้อมูลทั้ง 2 รูปแบบมีความใกล้เคียงกันมาก เนื่องจากลักษณะการแสดงข้อมูลของแหล่งข้อมูล REUTERS มีการกระจายตัวคล้ายกับเส้นโค้งปกติ ดังแสดงในภาพประกอบ 5.7 จึงทำให้ตำแหน่งเวลาในการดึงข้อมูลที่ได้จากรูปแบบ Retrieval scheduling เป็นตำแหน่งที่ใกล้เคียงกันหรือเป็นตำแหน่งเดียวกันกับรูปแบบ DORPM ความล่าช้าในการดึงข้อมูลที่ได้จึงใกล้เคียงกันด้วย



ภาพประกอบ 5.7 การกระจายตัวของลักษณะการแสดงข้อมูลของข่าวรอบโลก จากแหล่งข้อมูล REUTERS

เมื่อพิจารณารวมกันทั้ง 3 แหล่งข้อมูลจะพบว่ารูปแบบ DORPM จะช่วยลดความล่าช้าในการดึงข้อมูลได้ดีกว่าแบบ Retrieval scheduling เนื่องจากตำแหน่งเวลาในการดึงข้อมูลที่ได้นั้นคำนวณมาจากลักษณะการแสดงข้อมูลจากแหล่งข้อมูลนั้นโดยตรง จึงทำให้ตำแหน่งเวลาในการดึงข้อมูลที่นำไปใช้มีความสอดคล้องกับลักษณะการแสดงข้อมูลของแหล่งข้อมูลนั้น ผลการทดลองแสดงให้เห็นว่าระยะเวลาในการเรียนรู้ข้อมูล 4 สัปดาห์หรือประมาณ 1 เดือน การดึงข้อมูลโดยรูปแบบ DORPM สามารถลดความล่าช้าเฉลี่ยในการดึงข้อมูลได้มากกว่าแบบ Retrieval scheduling ช่วยให้ผู้ใช้บริการได้รับข่าวสารที่มีความทันสมัยมากยิ่งขึ้น ทั้งยังช่วยให้ตัวรวบรวมข่าวสารจัดสรรทรัพยากรในการดึงข้อมูลได้อย่างมีประสิทธิภาพอีกด้วย

5.3 จำนวนครั้งที่เหมาะสมในการดึงข้อมูล

จากการกำหนดตำแหน่งเวลาในการดึงข้อมูลจะต้องกำหนดจำนวนครั้งในการดึงข้อมูลก่อนจึงจะสามารถกำหนดตำแหน่งเวลาในการดึงข้อมูลได้ เนื่องจากจำนวนครั้งในการดึงข้อมูลมีผลต่อการกำหนดตำแหน่งเวลาในการดึงข้อมูล จำนวนครั้งในการดึงข้อมูลที่ต่างกันจะทำให้ตำแหน่งเวลาในการดึงข้อมูลนั้นต่างกันไปด้วย อย่างไรก็ตามจะเห็นว่าจำนวนครั้งในการดึงข้อมูลจะส่งผลต่อความล่าช้าในการดึงข้อมูล เนื่องจากจำนวนครั้งในการดึงข้อมูลมีจำนวนมากขึ้นความล่าช้าในการดึงข้อมูลจะลดน้อยลง

5.3.1 การออกแบบการทดลอง

เนื่องจากจำนวนครั้งในการดึงข้อมูลมีผลต่อความล่าช้าในการดึงข้อมูล ดังนั้นในการทดลองเพื่อหาความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าในการดึงข้อมูล จึงทดลองเช่นเดียวกับการหาประสิทธิภาพของกลไกการกำหนดตำแหน่งเวลาในการดึงข้อมูล โดยกำหนดตัวแปรต่างๆ ที่ใช้ในการทดลองมีดังต่อไปนี้

- ตัวแปรต้น คือ จำนวนครั้งในการดึงข้อมูล
- ตัวแปรตาม คือ ความล่าช้าในการดึงข้อมูล
- ตัวแปรที่ต้องควบคุม คือ แหล่งข้อมูล และจำนวนข้อมูลที่นำมาใช้

ซึ่งวิธีการในการทดลองดำเนินการตามขั้นตอนดังต่อไปนี้

1) แบ่งข้อมูลที่เก็บรวบรวมมาทั้ง 3 เดือนออกเป็น 2 ส่วน โดยส่วนแรกใช้ข้อมูลในเดือนแรกในการเรียนรู้เพื่อกำหนดตำแหน่งเวลาในการดึงข้อมูล ส่วนที่ 2 ใช้ข้อมูลในอีก 2 เดือนที่เหลือเพื่อคำนวณหาความล่าช้าที่เกิดขึ้นจากตำแหน่งเวลาในการดึงข้อมูลที่ได้จากส่วนแรก

2) นำข้อมูลในเดือนแรกมาเรียนรู้เพื่อกำหนดตำแหน่งเวลาในการดึงข้อมูล โดยแต่ละจำนวนครั้งในการดึงข้อมูลจะมีตำแหน่งเวลาในการดึงข้อมูลที่ต่างกัน

3) นำตำแหน่งเวลาที่ได้จากขั้นตอนที่ 2 มาคำนวณความล่าช้าของข้อมูลอีก 2 เดือนที่เหลืออยู่

4) หาความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าในการดึงข้อมูล

5.3.2 ผลการศึกษา

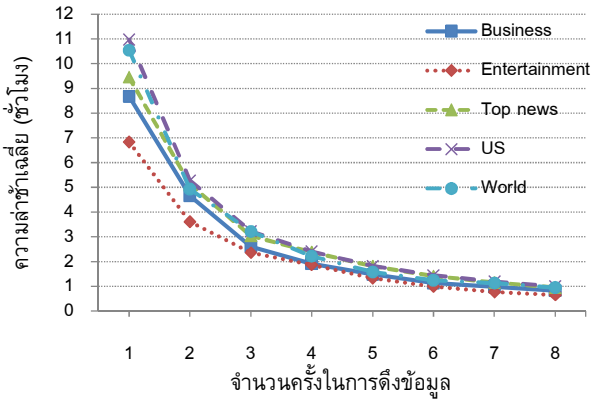
เช่นเดียวกันกับการหาประสิทธิภาพของกลไกการกำหนดตำแหน่งเวลาในการดึงข้อมูล จะมีแหล่งข้อมูลทั้งหมด 3 แหล่งข้อมูล แต่ละแหล่งข้อมูลมี 5 ประเภทข่าว จึงมีชุดข้อมูลทั้งหมด 15 ชุด เพื่อให้ง่ายต่อการหาความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าในการดึงข้อมูล จะพิจารณาโดยแยกพิจารณาทีละแหล่งข้อมูลดังต่อไปนี้

1) ข้อมูลจากแหล่งข้อมูล BBC

ความล่าช้าในการดึงข้อมูลของจำนวนครั้งในการดึงข้อมูลที่ต่างกันจากแหล่งข้อมูล BBC แสดงดังตาราง 5.6 และกราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ยแสดงดังภาพประกอบ 5.7

ตารางที่ 5.6 ความล่าช้าในการดึงข้อมูลของจำนวนครั้งในการดึงข้อมูลที่ต่างกันจากแหล่งข้อมูล BBC

จำนวนครั้งในการดึงข้อมูล	ความล่าช้าในการดึงข้อมูล (ชั่วโมง)					ความล่าช้าเฉลี่ย (ชั่วโมง / ข้อมูล)				
	Business	Entertainment	Top news	US	World	Business	Entertainment	Top news	US	World
#1	17,021	5,817	64,573	18,207	7,057	8.680	6.835	9.460	10.968	10.549
#2	9,145	3,068	35,009	8,765	3,295	4.663	3.605	5.129	5.280	4.925
#3	5,090	2,012	20,783	5,351	2,154	2.596	2.364	3.045	3.223	3.220
#4	3,740	1,592	16,242	3,987	1,485	1.907	1.871	2.379	2.402	2.220
#5	2,906	1,128	12,373	3,031	1,060	1.482	1.325	1.813	1.826	1.584
#6	2,218	842	9,695	2,414	827	1.131	0.989	1.420	1.454	1.236
#7	1,920	654	7,868	1,961	757	0.979	0.769	1.153	1.181	1.132
#8	1,617	553	6,505	1,655	630	0.825	0.650	0.953	0.997	0.942



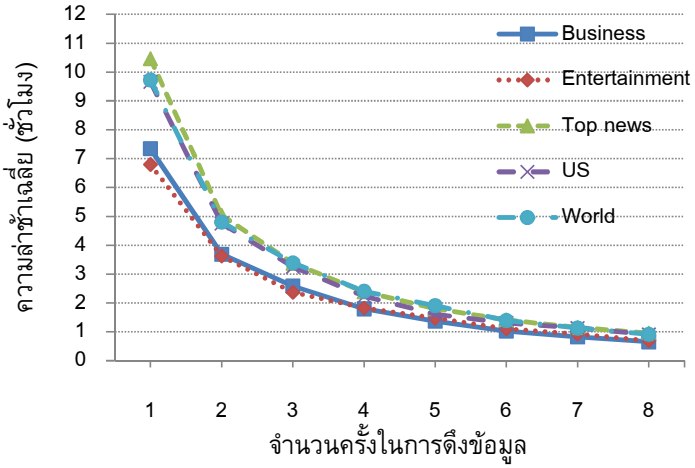
ภาพประกอบ 5.8 ความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ย โดยใช้ข้อมูลจากแหล่งข้อมูล BBC

2) ข้อมูลจากแหล่งข้อมูล CNN

ความล่าช้าในการดึงข้อมูลของจำนวนครั้งในการดึงข้อมูลที่ต่างกันจากแหล่งข้อมูล CNN แสดงดังตาราง 5.7 และกราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ยแสดงดังภาพประกอบ 5.8

ตารางที่ 5.7 ความล่าช้าในการดึงข้อมูลของจำนวนครั้งในการดึงข้อมูลที่ต่างกันจากแหล่งข้อมูล CNN

จำนวนครั้งในการดึงข้อมูล	ความล่าช้าในการดึงข้อมูล (ชั่วโมง)					ความล่าช้าเฉลี่ย (ชั่วโมง / ข้อมูล)				
	Business	Entertainment	Top news	US	World	Business	Entertainment	Top news	US	World
#1	13,089	3,703	19,938	12,563	11,575	7.349	6.794	10.466	9.656	9.719
#2	6,573	1,976	9,586	6,183	5,715	3.691	3.626	5.032	4.752	4.798
#3	4,601	1,293	6,439	4,223	4,036	2.583	2.372	3.380	3.246	3.389
#4	3,204	996	4,542	2,907	2,861	1.799	1.828	2.384	2.234	2.402
#5	2,433	804	3,449	2,100	2,271	1.366	1.475	1.810	1.614	1.907
#6	1,834	603	2,696	1,701	1,660	1.030	1.106	1.415	1.307	1.394
#7	1,467	504	2,193	1,471	1,354	0.824	0.925	1.151	1.131	1.137
#8	1,163	380	1,814	1,181	1,098	0.653	0.697	0.952	0.908	0.922



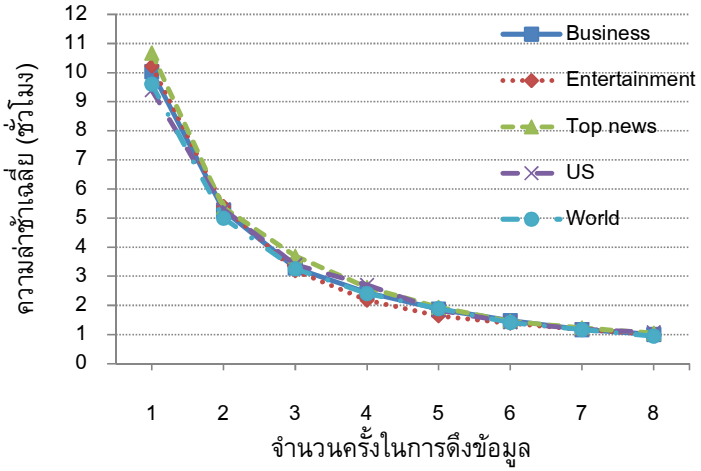
ภาพประกอบ 5.9 ความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ย โดยใช้ข้อมูลจากแหล่งข้อมูล CNN

3) ข้อมูลจากแหล่งข้อมูล REUTERS

ความล่าช้าในการดึงข้อมูลของจำนวนครั้งในการดึงข้อมูลที่ต่างกันจากแหล่งข้อมูล REUTERS แสดงดังตาราง 5.8 และกราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ยแสดงดังภาพประกอบ 5.9

ตารางที่ 5.8 ความล่าช้าในการดึงข้อมูลของจำนวนครั้งในการดึงข้อมูลที่ต่างกันจากแหล่งข้อมูล REUTERS

จำนวนครั้งในการดึงข้อมูล	ความล่าช้าในการดึงข้อมูล (ชั่วโมง)					ความล่าช้าเฉลี่ย (ชั่วโมง / ข้อมูล)				
	Business	Entertainment	Top news	US	World	Business	Entertainment	Top news	US	World
#1	19,902	7,737	24,920	12,720	21,319	10.036	10.221	10.668	9.394	9.599
#2	10,444	4,085	12,528	7,166	11,105	5.267	5.396	5.363	5.292	5.000
#3	6,516	2,417	8,634	4,606	7,243	3.286	3.193	3.696	3.402	3.261
#4	4,837	1,650	6,131	3,663	5,345	2.439	2.180	2.625	2.705	2.407
#5	3,712	1,242	4,534	2,485	4,205	1.872	1.641	1.941	1.835	1.893
#6	2,937	1,052	3,391	1,928	3,145	1.481	1.390	1.452	1.424	1.416
#7	2,328	884	2,857	1,603	2,591	1.174	1.168	1.223	1.184	1.167
#8	1,991	764	2,441	1,463	2,097	1.004	1.009	1.045	1.081	0.944



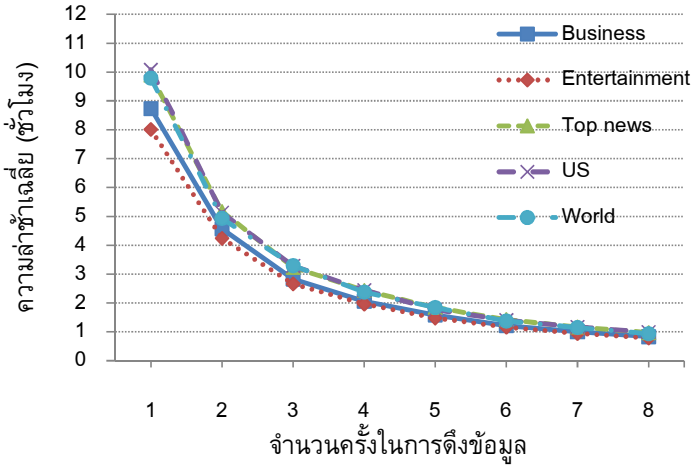
ภาพประกอบ 5.10 ความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ย โดยใช้ข้อมูลจากแหล่งข้อมูล REUTERS

4) ข้อมูลจากทั้ง 3 แหล่งข้อมูลรวมกัน

ความล่าช้าในการดึงข้อมูลของจำนวนครั้งในการดึงข้อมูลที่ต่างกันจากทั้ง 3 แหล่งข้อมูลรวมกัน แสดงดังตาราง 5.9 และกราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ยแสดงดังภาพประกอบ 5.10

ตารางที่ 5.9 ความล่าช้าในการดึงข้อมูลของจำนวนครั้งในการดึงข้อมูลที่ต่างกันจากทั้ง 3 แหล่งข้อมูลรวมกัน

จำนวนครั้งในการดึงข้อมูล	ความล่าช้าในการดึงข้อมูล (ชั่วโมง)					ความล่าช้าเฉลี่ย (ชั่วโมง / ข้อมูล)				
	Business	Entertainment	Top news	US	World	Business	Entertainment	Top news	US	World
#1	50,012	17,257	109,431	43,490	39,951	8.736	8.015	9.888	10.079	9.790
#2	26,162	9,129	57,123	22,114	20,115	4.570	4.240	5.162	5.125	4.929
#3	16,207	5,722	35,856	14,180	13,433	2.831	2.658	3.240	3.286	3.292
#4	11,781	4,238	26,915	10,557	9,691	2.058	1.968	2.432	2.447	2.375
#5	9,051	3,174	20,356	7,616	7,536	1.581	1.474	1.839	1.765	1.847
#6	6,989	2,497	15,782	6,043	5,632	1.221	1.160	1.426	1.400	1.380
#7	5,715	2,042	12,918	5,035	4,702	0.998	0.948	1.167	1.167	1.152
#8	4,771	1,697	10,760	4,299	3,825	0.833	0.788	0.972	0.996	0.937



ภาพประกอบ 5.11 ความสัมพันธ์ระหว่างจำนวนครั้งในการดึงข้อมูลกับความล่าช้าเฉลี่ย โดยใช้ข้อมูลทั้ง 3 แหล่งข้อมูลรวมกัน

จากภาพประกอบ 5.7 – 5.9 จะเห็นได้ชัดว่าไม่ว่าแหล่งข้อมูลใดหรือเป็นข่าวประเภทใดเมื่อจำนวนครั้งในการดึงข้อมูลเพิ่มขึ้นความล่าช้าในการดึงข้อมูลจะลดลง และเมื่อหาความล่าช้าเฉลี่ยของแต่ละประเภทจะพบว่าลักษณะของกราฟคล้ายคลึงกันมากจนบางกราฟแทบจะทับกันสนิท ซึ่งหมายความว่าความล่าช้าในการดึงข้อมูลเฉลี่ยมีค่าเท่าๆ กัน ณ จำนวนครั้งในการดึงข้อมูลที่เท่ากัน ถึงแม้ว่าข่าวบางประเภทอาจมีลักษณะของกราฟแตกต่างไปบ้าง ซึ่งอาจจะมีผลมาจากลักษณะการแสดงข้อมูลของข่าวประเภทนั้น แต่โดยรวมแล้วถือว่ามีลักษณะของกราฟความล่าช้าในการดึงข้อมูลเฉลี่ยที่ใกล้เคียงกันมาก โดยเฉพาะอย่างยิ่งแหล่งข้อมูล REUTERS ที่ไม่ว่าจะเป็นข่าวประเภทใดก็ตามความล่าช้าในการดึงข้อมูลเฉลี่ยที่เท่าๆ กันทุกประเภท

และจากภาพประกอบ 5.10 เมื่อข้อมูลข่าวแต่ละประเภทถูกรวบรวมมาจากแหล่งข้อมูลทั้ง 3 แหล่งเข้าด้วยกัน กราฟแสดงความล่าช้าในการดึงข้อมูลเฉลี่ยกับจำนวนครั้งในการดึงข้อมูล จะยิ่งมีลักษณะที่คล้ายคลึงกันมากยิ่งขึ้น และเมื่อจำนวนครั้งในการดึงข้อมูลมีค่ามากๆ ลักษณะของกราฟเส้นของข่าวแต่ละประเภทจะยิ่งเข้าใกล้กันมากยิ่งขึ้น เมื่อลักษณะของกราฟเป็นเช่นนี้แล้วจึงสามารถสรุปความสัมพันธ์ของจำนวนครั้งในการดึงข้อมูลกับความล่าช้าในการดึงข้อมูลเฉลี่ยได้ดังตารางที่ 5.10

ตารางที่ 5.10 ความสัมพันธ์ของจำนวนครั้งในการดึงข้อมูลกับความล่าช้าในการดึงข้อมูลเฉลี่ย

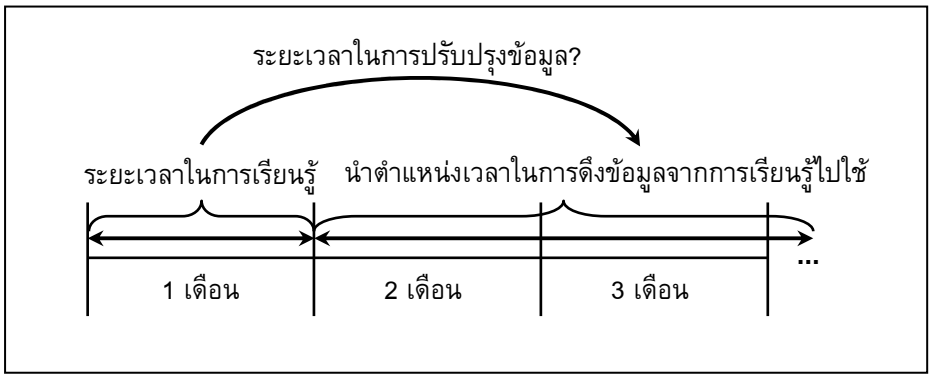
จำนวนครั้งในการดึงข้อมูล (ครั้ง / วัน)	ความล่าช้าในการดึงข้อมูลเฉลี่ย (ชั่วโมง / ข้อมูล)
#1	8 – 10
#2	4 – 5
#3	2.6 – 3.3
#4	2 – 2.5
#5	1.5 – 1.9
#6	1.1 – 1.4
#7	1 – 1.2
#8	< 1

จากตารางที่ 5.10 จะเห็นได้ว่าจำนวนครั้งในการดึงข้อมูลจะมีความล่าช้าในการดึงข้อมูลเฉลี่ยเป็นค่าคงที่ในช่วงหนึ่ง ซึ่งหากสามารถระบุได้ว่าความล่าช้าในการดึงข้อมูลที่ต้องการเป็นเท่าไรก็จะสามารถกำหนดจำนวนครั้งในการดึงข้อมูลที่เหมาะสมได้ เช่น ต้องการให้ดึงข้อมูลโดยที่แต่ละข้อมูลที่ได้รับนั้นมีความล่าช้าเฉลี่ยในการดึงข้อมูลไม่เกิน 3 ชั่วโมง ก็จะ

สามารถกำหนดได้ว่าจำนวนครั้งในการดึงข้อมูลเพียงแค่ 4 ครั้งต่อวันนั้นเพียงพอต่อความล่าช้าที่ต้องการ ซึ่งช่วยให้ตัวรวบรวมข่าวสารใช้ทรัพยากรได้อย่างมีประสิทธิภาพ และช่วยให้แหล่งข้อมูลไม่ต้องแบกรับภาระจากการที่ตัวรวบรวมข่าวสารร้องขอข้อมูลมากเกินความจำเป็น

5.4 ระยะเวลาในการปรับปรุงข้อมูลที่เหมาะสม

เนื่องจากกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลจะมีการเรียนรู้ข้อมูลในระยะหนึ่งก่อนจึงจะสามารถกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลได้ และจากการทดลองพบว่าที่ระยะเวลาในการเรียนรู้ 4 สัปดาห์หรือประมาณ 1 เดือนนั้นเป็นระยะเวลาในการเรียนรู้ที่มีความเหมาะสม อย่างไรก็ตามเมื่อนำกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลไปใช้จริง จะพบว่าเมื่อเวลาผ่านไประยะหนึ่งข้อมูลที่ได้นำมาเรียนรู้เพื่อกำหนดตำแหน่งเวลาในการดึงข้อมูลนั้นอาจมีความล้าสมัย ซึ่งอาจทำให้ตำแหน่งเวลาในการดึงข้อมูลที่ได้นั้นอาจไม่สอดคล้องกับการแสดงข้อมูลในปัจจุบัน ดังนั้นเมื่อเวลาผ่านไประยะหนึ่งจึงควรมีการปรับปรุงข้อมูลที่ใช้ในการเรียนรู้ตำแหน่งเวลาในการดึงข้อมูลใหม่ เพื่อให้ได้ตำแหน่งเวลาในการดึงข้อมูลที่มีความสอดคล้องกับการแสดงข้อมูลในปัจจุบันอยู่เสมอ ระยะเวลาในการปรับปรุงข้อมูลแสดงดังภาพประกอบ 5.11



ภาพประกอบ 5.12 ระยะเวลาในการปรับปรุงข้อมูล

5.4.1 การออกแบบการทดลอง

ในการปรับปรุงข้อมูลที่ใช้ในการเรียนรู้ข้อมูลเพื่อนำไปกำหนดตำแหน่งเวลาในการดึงข้อมูล จะใช้ระยะเวลาในการเรียนรู้เหมือนเดิมทุกครั้งคือที่ระยะเวลา 4 สัปดาห์หรือประมาณ 1 เดือน เนื่องจากผลการทดลองแสดงให้เห็นแล้วว่าที่ระยะเวลา 4 สัปดาห์มีความเหมาะสม แต่สิ่งจะพิจารณาคือระยะเวลาในการปรับปรุงข้อมูล ซึ่งจะส่งผลต่อการกำหนดตำแหน่งเวลาในการดึงข้อมูลจากข้อมูลที่ได้เรียนรู้ โดยกำหนดตัวแปรต่างๆ ที่ใช้ในการทดลองมีดังต่อไปนี้

- ตัวแปรต้น คือ ระยะเวลาในการปรับปรุงข้อมูล
- ตัวแปรตาม คือ ตำแหน่งเวลา และความล่าช้าในการดึงข้อมูล
- ตัวแปรที่ต้องควบคุม คือ ข้อมูลที่นำมาใช้ และระยะเวลาในการเรียนรู้

ซึ่งวิธีการในการทดลองดำเนินการตามขั้นตอนดังต่อไปนี้

1) เตรียมข้อมูลที่จะนำมาใช้ทดลอง โดยสมมติลักษณะการเปลี่ยนแปลงของการแสดงข้อมูลที่เป็นไปได้ ซึ่งแบ่งออกได้เป็น 3 ลักษณะ คือ ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงน้อย เปลี่ยนแปลงมาก และเปลี่ยนแปลงบ่อยๆ

2) แบ่งข้อมูลเป็นทีละสัปดาห์ และปรับปรุงข้อมูลที่ระยะเวลาต่างๆ แต่ยังคงใช้ระยะเวลาในการเรียนรู้เท่าเดิมคือ 1 เดือน เท่ากันทุกระยะเวลาที่ปรับปรุงข้อมูล

3) แต่ละระยะเวลาที่ปรับปรุงข้อมูลในขั้นตอนที่ 2 นำข้อมูลที่ได้ไปกำหนดตำแหน่งเวลาในการดึงข้อมูล

4) เปรียบเทียบตำแหน่งเวลาในการดึงข้อมูลแต่ละตำแหน่งที่ได้กับระยะเวลาในการปรับปรุงข้อมูล

ข้อมูลที่ได้กำหนดขึ้นมาเพื่อใช้ในการทดลองทั้ง 3 ลักษณะ มีรายละเอียดดังต่อไปนี้

แบบที่ 1 ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงน้อย การแสดงข้อมูลในเดือนที่ 2 และ 3 มีลักษณะคล้ายกับในเดือนที่ 1 แต่ในช่วงเวลา 7.00 น. ถึง 8.00 น. มีการแสดงข้อมูลมากกว่าเดิม

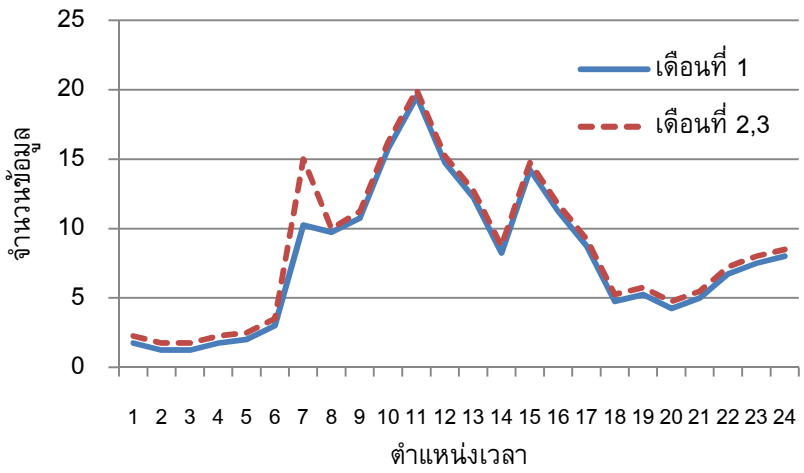
แบบที่ 2 ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงมาก การแสดงข้อมูลในเดือนที่ 2 และ 3 มีลักษณะแตกต่างจากข้อมูลในเดือนที่ 1 ลักษณะการแสดงข้อมูลในเดือนที่ 2 และ 3 จะเปลี่ยนไปเป็นอีกแบบหนึ่ง

แบบที่ 3 ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงบ่อยๆ การแสดงข้อมูลของเดือนที่ 2 และ 3 ในแต่ละสัปดาห์มีการเปลี่ยนแปลงอยู่เสมอแต่มีลักษณะที่คล้ายกับการแสดงข้อมูลแบบเดิมอยู่

รายละเอียดของข้อมูลทั้ง 3 รูปแบบแสดงดังตารางที่ 5.11 ถึง 5.13

ตารางที่ 5.11 ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงน้อย

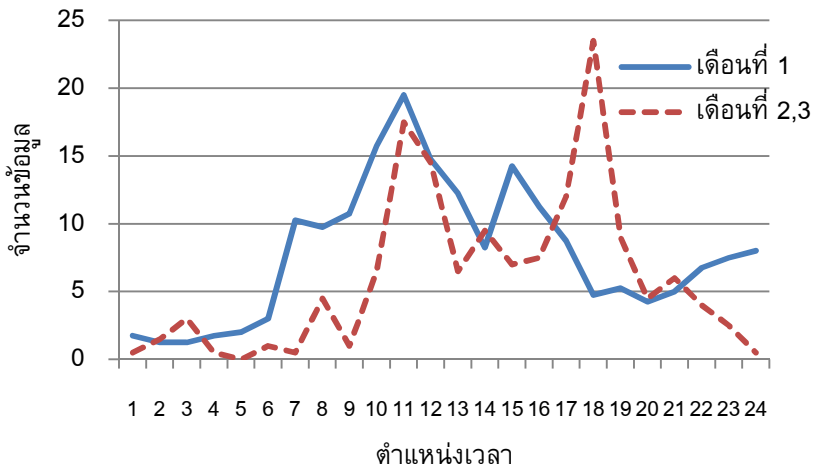
เวลา	จำนวนการแสดงข้อมูล											
	เดือนที่ 1				เดือนที่ 2				เดือนที่ 3			
1	2	2	2	2	2	2	2	2	2	2	2	2
2	1	1	1	1	2	2	2	2	2	2	2	2
3	1	1	1	1	2	2	2	2	2	2	2	2
4	2	2	2	2	2	2	2	2	2	2	2	2
5	2	2	2	2	3	3	3	3	3	3	3	3
6	3	3	3	3	4	4	4	4	4	4	4	4
7	10	10	10	10	15	15	15	15	15	15	15	15
8	10	10	10	10	10	10	10	10	10	10	10	10
9	11	11	11	11	11	11	11	11	11	11	11	11
10	16	16	16	16	16	16	16	16	16	16	16	16
11	20	20	20	20	20	20	20	20	20	20	20	20
12	15	15	15	15	15	15	15	15	15	15	15	15
13	12	12	12	12	13	13	13	13	13	13	13	13
14	8	8	8	8	9	9	9	9	9	9	9	9
15	14	14	14	14	15	15	15	15	15	15	15	15
16	11	11	11	11	12	12	12	12	12	12	12	12
17	9	9	9	9	9	9	9	9	9	9	9	9
18	5	5	5	5	5	5	5	5	5	5	5	5
19	5	5	5	5	6	6	6	6	6	6	6	6
20	4	4	4	4	5	5	5	5	5	5	5	5
21	5	5	5	5	6	6	6	6	6	6	6	6
22	7	7	7	7	7	7	7	7	7	7	7	7
23	8	8	8	8	8	8	8	8	8	8	8	8
24	8	8	8	8	9	9	9	9	9	9	9	9



ภาพประกอบ 5.13 ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงน้อย

ตารางที่ 5.12 ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงมาก

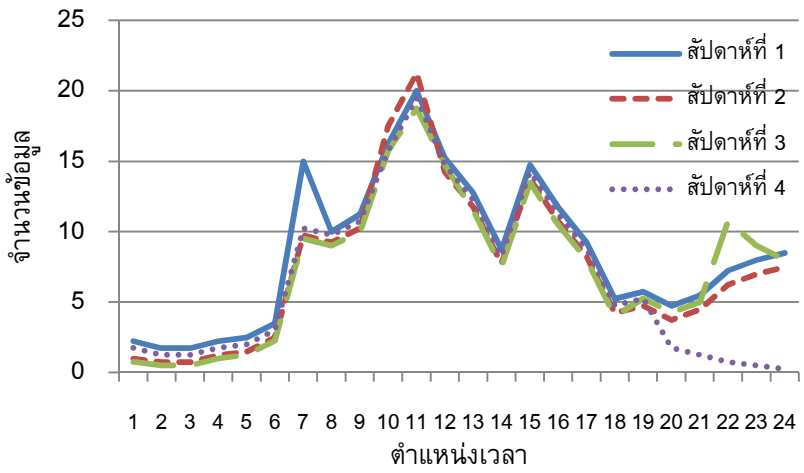
เวลา	จำนวนการแสดงข้อมูล											
	เดือนที่ 1				เดือนที่ 2				เดือนที่ 3			
1	2	2	2	2	0	0	0	0	0	0	0	0
2	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	2	2	2	2	2	2	2	2
4	2	2	2	2	0	0	0	0	0	0	0	0
5	2	2	2	2	0	0	0	0	0	0	0	0
6	3	3	3	3	1	1	1	1	1	1	1	1
7	10	10	10	10	0	0	0	0	0	0	0	0
8	10	10	10	10	2	2	2	2	2	2	2	2
9	11	11	11	11	1	1	1	1	1	1	1	1
10	16	16	16	16	3	3	3	3	3	3	3	3
11	20	20	20	20	9	9	9	9	9	9	9	9
12	15	15	15	15	7	7	7	7	7	7	7	7
13	12	12	12	12	3	3	3	3	3	3	3	3
14	8	8	8	8	5	5	5	5	5	5	5	5
15	14	14	14	14	4	4	4	4	4	4	4	4
16	11	11	11	11	4	4	4	4	4	4	4	4
17	9	9	9	9	6	6	6	6	6	6	6	6
18	5	5	5	5	12	12	12	12	12	12	12	12
19	5	5	5	5	5	5	5	5	5	5	5	5
20	4	4	4	4	2	2	2	2	2	2	2	2
21	5	5	5	5	3	3	3	3	3	3	3	3
22	7	7	7	7	2	2	2	2	2	2	2	2
23	8	8	8	8	1	1	1	1	1	1	1	1
24	8	8	8	8	0	0	0	0	0	0	0	0



ภาพประกอบ 5.14 ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงมาก

ตารางที่ 5.13 ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงบ่อย ๆ

เวลา	จำนวนการแสดงข้อมูล											
	เดือนที่ 1				เดือนที่ 2				เดือนที่ 3			
1	2	2	2	2	2	1	1	2	2	1	1	2
2	1	1	1	1	2	1	1	1	2	1	1	1
3	1	1	1	1	2	1	1	1	2	1	1	1
4	2	2	2	2	2	1	1	2	2	1	1	2
5	2	2	2	2	3	2	1	2	3	2	1	2
6	3	3	3	3	4	3	2	3	4	3	2	3
7	10	10	10	10	15	10	10	10	15	10	10	10
8	10	10	10	10	10	9	9	10	10	9	9	10
9	11	11	11	11	11	10	10	11	11	10	10	11
10	16	16	16	16	16	18	16	16	16	18	16	16
11	20	20	20	20	20	21	19	20	20	21	19	20
12	15	15	15	15	15	14	15	15	15	14	15	15
13	12	12	12	12	13	12	12	12	13	12	12	12
14	8	8	8	8	9	8	8	8	9	8	8	8
15	14	14	14	14	15	14	14	14	15	14	14	14
16	11	11	11	11	12	11	11	11	12	11	11	11
17	9	9	9	9	9	8	8	9	9	8	8	9
18	5	5	5	5	5	4	4	5	5	4	4	5
19	5	5	5	5	6	5	5	5	6	5	5	5
20	4	4	4	4	5	4	4	2	5	4	4	2
21	5	5	5	5	6	5	5	1	6	5	5	1
22	7	7	7	7	7	6	11	1	7	6	11	1
23	8	8	8	8	8	7	9	1	8	7	9	1
24	8	8	8	8	9	8	8	0	9	8	8	0



ภาพประกอบ 5.15 การแสดงข้อมูลในแต่ละสัปดาห์ของเดือนที่ 2 และ 3

5.4.2 ผลการศึกษา

ในการทดลองจะทดสอบกับระยะเวลาในการปรับปรุงข้อมูลที่เปลี่ยนไปทีละ 1 สัปดาห์ แต่เนื่องจากระยะเวลาในการเรียนรู้ข้อมูลถูกกำหนดให้มีระยะเวลา 4 สัปดาห์ ทำให้การปรับปรุงข้อมูลในสัปดาห์ที่ 1 ถึง 3 มีการใช้ข้อมูลเดิมรวมกับข้อมูลใหม่ โดยในการทดลองจะทดสอบตำแหน่งเวลาในการดึงข้อมูลที่มีจำนวนครั้งในการดึงข้อมูลที่แตกต่างกัน และสังเกตการเปลี่ยนแปลงของตำแหน่งเวลาในการดึงข้อมูลของระยะเวลาในการปรับปรุงข้อมูลที่ต่างกัน ซึ่งได้ผลการทดลองดังต่อไปนี้

1) ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงน้อย

ตำแหน่งเวลาในการดึงข้อมูลของระยะเวลาในการปรับปรุงข้อมูลที่ต่างกันของข้อมูลที่มีการเปลี่ยนแปลงน้อย แสดงดังตาราง 5.14

ตารางที่ 5.14 ตำแหน่งเวลาในการดึงข้อมูลของระยะเวลาในการปรับปรุงข้อมูลที่ต่างกันของข้อมูลที่มีการเปลี่ยนแปลงน้อย

ระยะเวลาในการปรับปรุงข้อมูล (สัปดาห์)	ตำแหน่งเวลาในการดึงข้อมูล							
	#1	#2	#3	#4	#5	#6	#7	#8
-	17	13,24	11,17,24	10,13,17,24	8,11,13,17,24	8,11,13,16,19,24	8,11,13,15,17,21,24	7,9,11,13,15,17,21,24
1	17	13,24	11,17,24	10,13,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
2	17	13,24	11,17,24	8,12,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
3	17	13,24	11,16,24	8,12,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
4	17	13,24	11,16,24	8,12,17,24	7,11,15,19,24	7,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
5	17	13,24	11,16,24	8,12,17,24	7,11,15,19,24	7,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
6	17	13,24	11,16,24	8,12,17,24	7,11,15,19,24	7,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
7	17	13,24	11,16,24	8,12,17,24	7,11,15,19,24	7,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
8	17	13,24	11,16,24	8,12,17,24	7,11,15,19,24	7,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24

2) ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงมาก

ตำแหน่งเวลาในการดึงข้อมูลของระยะเวลาในการปรับปรุงข้อมูลที่ต่างกันของข้อมูลที่มีการเปลี่ยนแปลงมาก แสดงดังตาราง 5.15

ตารางที่ 5.15 ตำแหน่งเวลาในการดึงข้อมูลของระยะเวลาในการปรับปรุงข้อมูลที่ต่างกันของข้อมูลที่มีการเปลี่ยนแปลงมาก

ระยะเวลาในการปรับปรุงข้อมูล (สัปดาห์)	ตำแหน่งเวลาในการดึงข้อมูล							
	#1	#2	#3	#4	#5	#6	#7	#8
-	17	13,24	11,17,24	10,13,17,24	8,11,13,17,24	8,11,13,16,19,24	8,11,13,15,17,21,24	7,9,11,13,15,17,21,24
1	18	13,24	12,18,24	11,15,19,24	8,12,15,19,24	8,11,13,16,19,24	8,11,13,15,18,21,24	7,9,11,13,15,18,21,24
2	18	12,21	12,18,24	11,14,18,24	8,12,15,19,24	8,12,15,18,21,24	8,11,13,15,18,21,24	4,8,11,13,15,18,21,24
3	19	12,21	12,18,24	11,14,18,24	3,11,14,18,22	8,12,15,18,21,24	8,11,13,15,18,21,24	3,8,11,13,15,18,20,23
4	19	12,21	12,18,23	11,14,18,23	3,11,14,18,21	3,11,12,15,18,21	3,11,12,15,18,20,23	3,11,12,14,16,18,20,23
5	19	12,21	12,18,23	11,14,18,23	3,11,14,18,21	3,11,12,15,18,21	3,11,12,15,18,20,23	3,11,12,14,16,18,20,23
6	19	12,21	12,18,23	11,14,18,23	3,11,14,18,21	3,11,12,15,18,21	3,11,12,15,18,20,23	3,11,12,14,16,18,20,23
7	19	12,21	12,18,23	11,14,18,23	3,11,14,18,21	3,11,12,15,18,21	3,11,12,15,18,20,23	3,11,12,14,16,18,20,23
8	19	12,21	12,18,23	11,14,18,23	3,11,14,18,21	3,11,12,15,18,21	3,11,12,15,18,20,23	3,11,12,14,16,18,20,23

3) ลักษณะการแสดงข้อมูลที่มีการเปลี่ยนแปลงบ่อยๆ

ตำแหน่งเวลาในการดึงข้อมูลของระยะเวลาในการปรับปรุงข้อมูลที่ต่างกันของข้อมูลที่มีการเปลี่ยนแปลงบ่อยๆ แสดงดังตาราง 5.16

ตารางที่ 5.16 ตำแหน่งเวลาในการดึงข้อมูลของระยะเวลาในการปรับปรุงข้อมูลที่ต่างกันของข้อมูลที่มีการเปลี่ยนแปลงบ่อยๆ

ระยะเวลาในการปรับปรุงข้อมูล (สัปดาห์)	ตำแหน่งเวลาในการดึงข้อมูล							
	#1	#2	#3	#4	#5	#6	#7	#8
-	17	13,24	11,17,24	10,13,17,24	8,11,13,17,24	8,11,13,16,19,24	8,11,13,15,17,21,24	7,9,11,13,15,17,21,24
1	17	13,24	11,17,24	10,13,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
2	17	13,24	11,17,24	10,13,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
3	17	13,24	11,16,24	10,13,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,21,24	7,9,11,13,15,17,21,24
4	17	13,24	11,16,24	10,13,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,20,24	7,9,11,13,15,17,20,24
5	17	13,24	11,16,24	10,13,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,20,24	7,9,11,13,15,17,20,24
6	17	13,24	11,16,24	10,13,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,20,24	7,9,11,13,15,17,20,24
7	17	13,24	11,16,24	10,13,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,20,24	7,9,11,13,15,17,20,24
8	17	13,24	11,16,24	10,13,17,24	8,11,15,19,24	8,11,13,16,19,24	7,10,12,15,17,20,24	7,9,11,13,15,17,20,24

จากทั้ง 3 ลักษณะการแสดงข้อมูลจะพบว่าตำแหน่งเวลาในการดึงข้อมูลของทั้ง 3 แบบจะไม่เปลี่ยนแปลง หลังจากระยะเวลาในการปรับปรุงข้อมูลผ่านไปประมาณ 4 สัปดาห์ ไม่ว่าจำนวนครั้งในการดึงข้อมูลจะเป็นกี่ครั้งก็ตาม เนื่องจากข้อมูลทั้ง 3 ลักษณะที่ใช้ในเดือนที่ 2 และ 3 เป็นข้อมูลที่เหมือนกันและมีลักษณะการแสดงข้อมูลเปลี่ยนไปจากในเดือนที่ 1 จึงทำให้หลังจากสัปดาห์ที่ 4 หรือข้อมูลในเดือนที่ 2 เหมือนกับในเดือนที่ 3 ตำแหน่งเวลาในการดึงข้อมูลจึงไม่เปลี่ยนแปลง

อย่างไรก็ตามจะเห็นว่าในสัปดาห์ที่ 1 ถึง 3 เป็นระยะเวลาที่มีทั้งข้อมูลเก่าคือข้อมูลในเดือนที่ 1 และข้อมูลใหม่คือข้อมูลในสัปดาห์ที่ 1 – 3 ในเดือนที่ 2 รวมกันจึงทำให้ตำแหน่งเวลาในการดึงข้อมูลในระยะนี้ต้องปรับตำแหน่งเพื่อให้สอดคล้องกับข้อมูลที่นำมารวมกัน ตำแหน่งเวลาในการดึงข้อมูลที่ได้จึงแตกต่างจากตำแหน่งเวลาที่ได้จากข้อมูลในเดือนแรก

จากตารางที่ 5.14 ซึ่งเป็นข้อมูลที่มีการเปลี่ยนแปลงน้อย โดยตำแหน่งเวลาที่มีการแสดงข้อมูลเพิ่มมากขึ้นเป็นช่วงเวลา 7.00 น. ถึง 8.00 น. ตำแหน่งเวลาในการดึงข้อมูลบางตำแหน่งเวลาจึงถูกปรับให้ดึงข้อมูลเร็วขึ้นในช่วงเวลานี้ เช่น ที่จำนวนครั้งในการดึงข้อมูล 4 ครั้ง จากตำแหน่งเวลา 10 จะเปลี่ยนไปเป็น 8 หรือที่จำนวนครั้งในการดึงข้อมูล 5 – 7 ครั้ง จากตำแหน่งเวลา 8 จะเปลี่ยนไปเป็น 7 แต่โดยภาพรวมแล้วตำแหน่งเวลาในการดึงข้อมูลจะไม่เปลี่ยนแปลงไปมาก เนื่องจากที่ช่วงเวลาอื่นๆ ยังคงมีลักษณะการแสดงข้อมูลเช่นเดิม

จากตารางที่ 5.15 ซึ่งเป็นข้อมูลที่มีการเปลี่ยนแปลงมาก จะเห็นว่าตำแหน่งเวลาในการดึงข้อมูลจะพยายามปรับเปลี่ยนเพื่อให้สอดคล้องกับลักษณะข้อมูลที่เปลี่ยนไป โดยตำแหน่งเวลาในการดึงข้อมูลจะเปลี่ยนไปมากเนื่องจากลักษณะการแสดงข้อมูลไม่เหมือนแบบเดิม และตำแหน่งเวลาจะเริ่มคงที่เมื่อระยะเวลาในการปรับปรุงข้อมูลเข้าสู่สัปดาห์ที่ 4

จากตารางที่ 5.16 ซึ่งเป็นข้อมูลที่มีการเปลี่ยนแปลงบ่อยๆ จะเห็นว่าตำแหน่งเวลาในการดึงข้อมูลแทบจะไม่เปลี่ยนไปจากเดิม เนื่องจากลักษณะข้อมูลยังคงเป็นแบบเดิม เพียงแต่ในบางสัปดาห์ข้อมูลในบางช่วงอาจเพิ่มขึ้นและข้อมูลในบางช่วงอาจลดลง แต่เพราะใช้ระยะเวลาในการเรียนรู้ประมาณ 1 เดือนจึงทำให้ข้อมูลนั้นเฉลี่ยกันไป ไม่มีผลกระทบหรือมีผลกระทบน้อยต่อการกำหนดตำแหน่งเวลาในการดึงข้อมูล

จากทั้ง 3 ลักษณะการแสดงข้อมูลจึงสามารถสรุปได้ว่าระยะเวลาในการปรับปรุงข้อมูล คือระยะเวลาที่ตำแหน่งเวลาในการดึงข้อมูลสอดคล้องกับลักษณะการแสดงข้อมูลที่เปลี่ยนไป ซึ่งจะเห็นได้ว่าที่ระยะเวลาประมาณ 4 สัปดาห์นั้นเหมาะสมที่จะใช้เป็นระยะเวลาในการปรับปรุงข้อมูล เนื่องจากระยะเวลาในการปรับปรุงข้อมูลควรเป็นระยะเวลาที่เร็วที่สุดที่สามารถกำหนดตำแหน่งเวลาในการดึงข้อมูลที่สอดคล้องกับลักษณะการแสดงข้อมูลที่เปลี่ยนไปเมื่อลักษณะการแสดงข้อมูลมีการเปลี่ยนแปลง ซึ่งระยะเวลาในการปรับปรุงข้อมูลควรเป็นระยะเวลาเดียวกันกับระยะเวลาในการเรียนรู้ข้อมูลนั่นเอง

บทที่ 6

บทสรุปและข้อเสนอแนะ

วิทยานิพนธ์นี้ได้นำเสนอกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS (Determining Optimal Retrieval Points Mechanism for RSS Documents: DORPM) โดยการทำงานของกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS ประกอบด้วย 2 ส่วนสำคัญ คือ ส่วนรวบรวมข้อมูล และส่วนวิเคราะห์ข้อมูล โดยส่วนรวบรวมข้อมูลจะเก็บรวบรวมข้อมูลจากแหล่งข้อมูลมาและแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมกับการนำไปใช้งาน และส่วนวิเคราะห์ข้อมูลจะนำข้อมูลที่ได้จากส่วนแรกมาวิเคราะห์หาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

ในส่วนของการรวบรวมข้อมูลจะทำหน้าที่เตรียมข้อมูลและแปลงข้อมูลให้อยู่ในรูปแบบที่สามารถนำไปวิเคราะห์หาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลในขั้นตอนต่อไป การทำงานในส่วนนี้จะเริ่มจากการเก็บรวบรวมเอกสาร RSS จากแหล่งข้อมูล และสกัดข้อมูลเก็บเอาเฉพาะข้อมูลภายในแท็กที่ต้องใช้เก็บไว้ในฐานข้อมูลของตัวรวบรวมข่าวสาร จากนั้นจะนำข้อมูลที่ได้จากการสกัดข้อมูลภายในแท็ก <pubDate> ซึ่งเป็นเวลาในการแสดงข้อมูลของแต่ละข้อมูล นำไปแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม และแบ่งกลุ่มข้อมูลตามลักษณะการแสดงข้อมูล

ในส่วนของการวิเคราะห์ข้อมูลจะนำข้อมูลที่ได้จากส่วนรวบรวมข้อมูลซึ่งทำการแปลงข้อมูลเรียบร้อยแล้วมาวิเคราะห์หาตำแหน่งในการดึงข้อมูลที่เหมาะสม จะเริ่มจากคำนวณความล่าช้าในการดึงข้อมูลของแต่ละตำแหน่งเวลาในการดึงข้อมูล โดยใช้ลักษณะการแสดงข้อมูลที่ได้จากในส่วนแรกมาคำนวณหาผลต่างของระยะเวลาที่ข้อมูลแสดงกับตำแหน่งเวลาในการดึงข้อมูล เมื่อคำนวณความล่าช้าของแต่ละตำแหน่งเวลาในการดึงข้อมูลที่เป็นไปได้ทั้งหมดแล้ว จะกำหนดตำแหน่งเวลาในการดึงข้อมูลโดยเลือกตำแหน่งเวลาที่มีความล่าช้าในการดึงข้อมูลน้อยที่สุด และนำตำแหน่งเวลาที่ได้ไปใช้ในการดึงข้อมูลในครั้งต่อๆ ไป

6.1 บทสรุปผลการวิจัย

กลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS เป็นกลไกที่สามารถกำหนดตำแหน่งเวลาในการดึงข้อมูลที่มีประสิทธิภาพมากกว่าการกำหนดเวลาในการดึงข้อมูล (Retrieval Scheduling) เนื่องจากเมื่อนำตำแหน่งเวลาที่กำหนดไป

ใช้ในการดึงข้อมูลพบว่ามีความล่าช้าในการดึงข้อมูลที่ลดลง โดยสรุปเป็นประเด็นต่าง ๆ ได้ดังต่อไปนี้

6.1.1 ระยะเวลาในการเรียนรู้ข้อมูล

ในการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลจำเป็นจะต้องเรียนรู้ข้อมูลก่อน โดยระยะเวลาในการเรียนรู้ข้อมูลจะมีผลต่อการกำหนดตำแหน่งเวลาในการดึงข้อมูล ซึ่งระยะเวลาในการเรียนรู้ข้อมูลของแต่ละรูปแบบจะมีระยะเวลาที่แตกต่างกันดังนี้

1) รูปแบบ Retrieval scheduling จะใช้ระยะเวลาเพียงแค่ 2 สัปดาห์ในการเรียนรู้ข้อมูล ผลการทดลองแสดงให้เห็นว่าเมื่อใช้ระยะเวลาที่มากกว่า 2 สัปดาห์ก็ยังคงได้ตำแหน่งเวลาในการดึงข้อมูลเช่นเดิม

2) รูปแบบ DORPM จะใช้ระยะเวลาในการเรียนรู้ 4 สัปดาห์หรือประมาณ 1 เดือน โดยผลการทดลองแสดงให้เห็นว่าเมื่อใช้ระยะเวลาที่มากกว่า 4 สัปดาห์จะได้ตำแหน่งเวลาในการดึงข้อมูลเช่นเดิม

จากระยะเวลาที่ใช้แสดงให้เห็นว่ารูปแบบ DORPM ใช้ระยะเวลาในการเรียนรู้ข้อมูลนานกว่าแบบ Retrieval scheduling ซึ่งทำให้ตำแหน่งเวลาในการดึงข้อมูลที่ได้จากการเรียนรู้มีจำนวนข้อมูลในการเรียนรู้ที่มากกว่า จึงมีโอกาสได้ตำแหน่งเวลาที่มีความผิดพลาดน้อยลง เพราะลักษณะการแสดงข้อมูลในแต่ละวันจะมีลักษณะที่ไม่แน่นอน หากใช้ระยะเวลาในการเรียนรู้น้อยจะมีโอกาสได้ตำแหน่งเวลาที่มีความผิดพลาดได้สูง ดังนั้นระยะเวลาในการเรียนรู้ที่มากกว่าจึงช่วยให้ได้ตำแหน่งเวลาในการดึงข้อมูลที่มีความเหมาะสมมากกว่า

6.1.2 ประสิทธิภาพในการดึงข้อมูล

จากผลการทดลองโดยใช้แหล่งข้อมูล BBC, CNN และ REUTERS เก็บรวบรวมข้อมูลทั้งหมด 3 เดือน ใช้ข้อมูลเดือนแรกในการเรียนรู้ข้อมูล และสองเดือนหลังในการทดสอบ ใช้จำนวนครั้งในการดึงข้อมูล 1 – 8 ครั้ง โดยแต่ละรูปแบบมีความล่าช้าในการดึงข้อมูลดังแสดงในตาราง 6.1

ตารางที่ 6.1 ตารางเปรียบเทียบความล่าช้าเฉลี่ยระหว่างการดึงข้อมูลแบบRetrieval scheduling กับการดึงข้อมูลแบบ DORPM

จำนวนครั้งในการดึงข้อมูล	ความล่าช้าเฉลี่ยในการดึงข้อมูล (ชั่วโมง/ข้อมูล)		คิดเป็น %
	Retrieval scheduling	DORPM	
#1	10.772	9.515	12%
#2	6.202	4.925	21%
#3	4.413	3.123	29%
#4	2.739	2.311	16%
#5	2.053	1.746	15%
#6	1.698	1.351	20%
#7	1.592	1.112	30%
#8	1.432	0.927	35%
		เฉลี่ย	22%

เมื่อพิจารณาความล่าช้าในการดึงข้อมูลจะพบว่ารูปแบบ DORPM จะช่วยลดความล่าช้าในการดึงข้อมูลได้ดีกว่าแบบ Retrieval scheduling โดยทุกๆ จำนวนครั้งในการดึงข้อมูลรูปแบบ DORPM จะมีความล่าช้าเฉลี่ยในการดึงข้อมูลน้อยกว่าแบบ Retrieval scheduling ซึ่งหากคิดจากจำนวนครั้งในการดึงข้อมูลทั้ง 8 แบบ จะได้ว่าการดึงข้อมูลโดยรูปแบบ DORPM สามารถลดความล่าช้าเฉลี่ยในการดึงข้อมูลได้ถึง 22% เมื่อเทียบกับแบบ Retrieval scheduling

6.1.3 การปรับปรุงข้อมูลที่ใช้ในการเรียนรู้

เมื่อผ่านไประยะเวลาหนึ่งข้อมูลที่นำมาใช้ในการเรียนรู้อาจไม่สอดคล้องกับลักษณะการแสดงข้อมูลในปัจจุบัน ดังนั้นหากมีตัวรวบรวมข่าวสารมีการปรับปรุงข้อมูลที่ใช้ในการเรียนรู้อยู่เสมอจะช่วยให้ตำแหน่งเวลาที่ใช้ในการดึงข้อมูลมีความสอดคล้องกับลักษณะการแสดงข้อมูลที่เป็นปัจจุบัน โดยรูปแบบ Retrieval scheduling จะไม่มีการปรับปรุงข้อมูลที่ใช้ในการเรียนรู้ จึงมีโอกาสที่ตำแหน่งเวลาที่ใช้ในการดึงข้อมูลอาจผิดพลาดได้เมื่อลักษณะของการแสดงข้อมูลเปลี่ยนไป แต่รูปแบบ DORPM จะมีการปรับปรุงข้อมูลอยู่เสมอ โดยปรับปรุงข้อมูลทุกๆ เดือน เป็นระยะเวลาเดียวกันกับระยะเวลาในการเรียนรู้ จึงทำให้ตำแหน่งเวลาในการดึงข้อมูลสอดคล้องกับลักษณะการแสดงข้อมูลอยู่ตลอดเวลา เมื่อลักษณะการแสดงข้อมูลเปลี่ยนไปตัวรวบรวมข่าวสารจะสามารถปรับเปลี่ยนตำแหน่งเวลาในการดึงข้อมูลให้สอดคล้องกับการแสดงข้อมูลได้อยู่เสมอ

6.1.4 การปรับเปลี่ยนตำแหน่งเวลาในการดึงข้อมูล

สำหรับกลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS เป็นกลไกที่กำหนดตำแหน่งเวลาในการดึงข้อมูลในแต่ละวันให้เป็นตำแหน่งเดียวกันทุกวัน ซึ่งตำแหน่งเวลาในการดึงข้อมูลที่ได้มาจากการนำลักษณะการแสดงข้อมูลมารวมกันดังนั้นตำแหน่งเวลาที่ได้จึงเป็นตำแหน่งเวลาที่เหมาะสมจะใช้ดึงข้อมูลทุกวัน เพราะเป็นตำแหน่งเวลาที่กำหนดมาจากการแสดงข้อมูลรวมกันไม่ได้มาจากการแสดงข้อมูลวันใดวันหนึ่ง

หากต้องการเปลี่ยนรูปแบบการดึงข้อมูลจากที่กำหนดให้เป็นตำแหน่งเดียวกันทุกวัน เปลี่ยนไปเป็นแบบที่สามารถปรับเปลี่ยนตำแหน่งเวลาในการดึงข้อมูลในแต่ละวันได้ จำเป็นจะต้องทราบลักษณะการแสดงข้อมูลในแต่ละวันก่อนจึงจะสามารถปรับเปลี่ยนตำแหน่งเวลาในการดึงข้อมูลให้สอดคล้องกันได้ แต่ในทางปฏิบัติเราไม่สามารถทราบถึงลักษณะการแสดงข้อมูลในแต่ละวันล่วงหน้าได้ ดังนั้นการปรับเปลี่ยนตำแหน่งเวลาในการดึงข้อมูลในแต่ละวันจึงต้องใช้การทำนายลักษณะการแสดงข้อมูล โดยใช้แบบจำลองการแสดงข้อมูลที่สามารถทำนายลักษณะการแสดงข้อมูลล่วงหน้าได้อย่างแม่นยำซึ่งทำได้ค่อนข้างยากเนื่องลักษณะการแสดงข้อมูลนั้นขึ้นอยู่กับแหล่งข้อมูล จึงเป็นเรื่องยากที่จะสามารถทำนายลักษณะการแสดงข้อมูลได้อย่างแม่นยำ

6.2 ปัญหาและอุปสรรค

เนื่องจากข้อมูลที่นำมาใช้ในการทดลองไม่มีข้อมูลที่เป็นมาตรฐาน เช่น Benchmark จึงจำเป็นจะต้องเก็บรวบรวมข้อมูลเพื่อนำมาใช้เอง โดยในการเก็บข้อมูลได้เลือกแหล่งข้อมูลที่เป็นที่นิยมและมีมาตรฐานในระดับนานาชาติมากที่สุด จึงเลือกแหล่งข้อมูลที่นำมาใช้ 3 แหล่งข้อมูล คือ BBC, CNN และ REUTERS ซึ่งเป็นแหล่งข้อมูลที่มีความน่าเชื่อถือสูงเมื่อเทียบกับแหล่งข้อมูลอื่นๆ ในการเก็บข้อมูลจะตั้งเวลาในการดึงข้อมูลทุกๆ 2 ชั่วโมง ซึ่งเป็นเวลาที่เหมาะสมเนื่องจากเป็นช่วงเวลาที่ทำให้ได้ข้อมูลครบถ้วน หากดึงข้อมูลช้าเกินไปอาจทำให้ข้อมูลบางข้อมูลหายไปได้

อย่างไรก็ตามในการเก็บข้อมูลรวมระยะเวลาทั้งหมด 3 เดือน อาจมีบางวันที่ระบบเครือข่ายอินเตอร์เน็ตมีปัญหาทำให้การดึงข้อมูลบางช่วงล่าช้า จึงทำให้อาจได้ข้อมูลที่ไม่ครบถ้วนและไม่สามารถเก็บข้อมูลย้อนหลังได้ ซึ่งแก้ปัญหาดังกล่าวโดยตัดข้อมูลในวันดังกล่าวทิ้งไปเพื่อไม่ให้มีผลกระทบต่อการกำหนดตำแหน่งเวลาในการดึงข้อมูล

6.3 ข้อเสนอแนะและงานวิจัยในอนาคต

กลไกการกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลสำหรับเอกสาร RSS เป็นกลไกที่ช่วยกำหนดตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล อย่างไรก็ตามตำแหน่งเวลาที่ได้นั้นจะเป็นตำแหน่งเดียวกันทุกวันและจำนวนครั้งในการดึงข้อมูลในแต่ละวันจะเท่ากันทุกวัน ไม่สามารถปรับเปลี่ยนได้ หากต้องการให้กลไกมีความสามารถในการปรับเปลี่ยนตำแหน่งเวลาได้เหมาะสมกับลักษณะการแสดงข้อมูลในแต่ละวัน จำเป็นต้องทำนายลักษณะการแสดงข้อมูลได้อย่างแม่นยำซึ่งกลไกยังไม่สามารถทำได้ในขณะนี้

งานวิจัยในอนาคตสำหรับตัวรวบรวมข่าวสารจึงมีแนวโน้มที่จะต้องใช้การสร้างแบบจำลองเพื่อมาทำนายลักษณะการแสดงข้อมูล และนำลักษณะการแสดงข้อมูลที่ได้มากำหนดตำแหน่งเวลาในการดึงข้อมูลซึ่งสามารถปรับเปลี่ยนได้ตามลักษณะการแสดงข้อมูลที่ทำนายในแต่ละวัน แบบจำลองจึงมีผลอย่างมากต่อการกำหนดตำแหน่งเวลาในการดึงข้อมูล ดังนั้นหากสามารถสร้างแบบจำลองที่ทำนายลักษณะการแสดงข้อมูลได้อย่างแม่นยำจะช่วยให้ตัวรวบรวมข่าวสารมีประสิทธิภาพในการดึงข้อมูลเป็นอย่างมาก

อีกประเด็นหนึ่งที่น่าสนใจคือการนำลักษณะเนื้อหาของข้อมูลมาหาความสัมพันธ์ของเหตุการณ์ที่เชื่อมโยงกันระหว่างแต่ละข้อมูล ซึ่งจะช่วยทำให้เราทราบได้ว่าข้อมูลอะไรเป็นที่สนใจของสังคมอยู่ในขณะนี้ และอาจจัดหมวดหมู่ข้อมูลที่มีเนื้อหาสอดคล้องกันไว้ด้วยกัน เมื่อมีข้อมูลใหม่ๆ เข้ามาจะได้ติดตามความคืบหน้าของเหตุการณ์นั้นๆ ได้ดียิ่งขึ้น

บรรณานุกรม

ชารวีย์ แสงขำ. 2552. เทคนิคการคัดกรองวิดีโออัตโนมัติสำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโปรโตคอล TCP/IP. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ สงขลา.

เชาวนันท์ ขุนดำ และ ลัดดา ปรีชาวีรกุล. 2553. กลไกการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล. การประชุมวิชาการวิทยาการคอมพิวเตอร์และวิศวกรรมคอมพิวเตอร์แห่งชาติ ครั้งที่ 14 (NCSEC 2010). เชียงใหม่, ประเทศไทย, 17 – 19 พฤศจิกายน 2553. หน้า 270 – 275.

วิชุตา แก้วนพรัตน์. 2552. กลไกแจ้งสารสนเทศที่ปลอดภัยด้วยเทคโนโลยี RSS สำหรับอุปกรณ์สื่อสารเคลื่อนที่ที่รองรับโพรโทคอล TCP/IP. วิทยานิพนธ์วิทยาศาสตรมหาบัณฑิต คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ สงขลา.

BBC. 2011. News feeds from the BBC. http://www.bbc.co.uk/news/10628494 (accessed 22/4/2011).

Blogspot. 2010. Push Technology. http://mikerakmae.blogspot.com/2009/01/gsc010750122765work38-push-technology.html (accessed 16/9/2010).

Cho, J., and Molina, H. G., 2000. Synchronizing a Database to Improve Freshness. Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '00).

Cho, J., and Molina, H.G., 2003, Estimating Frequency of Change. *ACM TOIT*, 3(3), August 2003.

CNN. 2011. RSS (Really Simple Syndication) – CNN.com. http://edition.cnn.com/services/rss/ (accessed 22/4/2011).

Doveloper in thai. 2010. Client pull/ Server push. http://develop.do.in.th/?p=290 (accessed 16/9/2010).

Edwards, G., McCurley, K., and Tomlin, J., 2001. Adaptive Model from Optimizing Performance of an Incremental Web Crawler. The 10th World Wide Web Conference (2001), pp.106 – 113.

Finkelstein, E. 2005. Syndicating Web Sites with RSS Feeds For Dummies. Wiley Publishing: NJ.

Gallaugher, J., 1996. The Critical Choice of Client Server Architecture: A Comparison of Two and Three Tier Systems, A future issue of Information Systems Management, Auerbach Publications, New York.

Gardner, W., Mulvey, E.P., and Shaw, E.C., 1995. Regression Analyses of Counts and Rates: Poisson, Overdispersed Poisson, and Negative Binomial Models. Psychological Bulletin., Vol. 118, No. 3, pp. 392 – 404.

Han, Y.G., Lee, S.H., Kim, J.H., and Kim, Y., 2008. A New Aggregation Policy for RSS Services, Proceeding of the 2008 international workshop on Context enabled source and service selection, integration and adaptation, Vol. 292.

Khundam, Ch., and Preechaveerakul, L. 2011. Determining Optimal Retrieval Points Mechanism for RSS Documents. 3rd International Conference on Advanced Computer Control (ICACC 2011). Harbin, China, January 18 – 20, 2011. pp.644 – 649.

Kim, S.J., and Lee, S.H., 2007. Estimating the Change of Web Pages, in Proceedings of the International Conference on Computational Science 2007 (Beijing, China), pp.798 – 805.

Lipschuts, S. and Schiller, J.J., 1995. Theory and Problems of Finite Mathematics. McGraw-Hill, USA. pp.256 – 381.

Manager Online. 2010. Manager Online. http://www.manager.co.th/rss/ (accessed 22/4/2011).

Miryam, W., 1997. The Pull of Push, http://www.cio.com/WebMaster/070197_push_ content.html. (accessed 22/4/2011).

Netlab. 2010. Poisson process. http://www.netlab.tkk.fi/opetus/s38143/luennot/ E_poisson.pdf (accessed 16/9/2010).

REUTERS. 2011. Reuters News RSS Feeds. http://www.reuters.com/tools/rss (accessed 16/9/2010).

RSS in Thai. 2010. การอ่าน RSS feed. http://www.rss.in.th/2007/07/28/ reading-rss-feed/ (accessed 16/9/2010).

Sia, K., and Cho, J., 2005. Efficient Monitoring Algorithm for Fast News Alert, technical report, University of California, Los Angeles.

Sia, K., Cho, J., and Cho, Y., 2007. Efficient Monitoring Algorithm for Fast News Alert, IEEE Trans. Knowledge and Data Eng., Vol.19, pp. 950 – 961.

Sia, K.C., Cho, J., Hino, K., Chi, Y., Zhu, S., and Tseng, B.L., 2007. Monitoring RSS Feeds Based on User Browsing Pattern, in Proceedings of the International Conference on Weblogs and Social Media (Boulder Colorado, March 2007), pp.161 – 168.

Stevens W. R. 1994. TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley.

Umbach, K. W. 1997. What is "Push Technology". http://www.library.ca.gov/crb/97 /notes/V4n6.pdf (accessed 22/4/2011).

W3C. 2009. Extensible Markup Language (XML). http://www.w3c.org/XML/ (accessed 22/4/2011).

W3School. 2010. RSS Tutorial. http://www.w3schools.com/rss/ (accessed 16/9/2010).

Wikipedia. 2010. Pull Technology. http://en.wikipedia.org/wiki/ Pull_technology (accessed 16/9/2010).

Wikipedia. 2010. Poisson process. http://en.wikipedia.org/wiki/Poisson_process #Definition (accessed 16/9/2010).

ภาคผนวก

ภาคผนวก ก.

ผลงานตีพิมพ์

เรื่อง	กลไกการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล
งานประชุมวิชาการ	วิทยาการคอมพิวเตอร์และวิศวกรรมคอมพิวเตอร์แห่งชาติ ครั้งที่ 14 (NCSEC 2010)
สถานที่	จังหวัดเชียงใหม่ ประเทศไทย
วันที่	17 – 19 พฤศจิกายน 2553

กลไกการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

Discovering Optimal Retrieval Points Mechanism for RSS feeds

เชาวนันท์ ขุนดำ และลัดดา ปรีชาวีรกุล
ห้องปฏิบัติการวิจัยเทคโนโลยีระบบสารสนเทศและการวิจัยประยุกต์ ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยสงขลานครินทร์ หาดใหญ่ สงขลา 90110 ประเทศไทย
Email: {s5210220147, ladda.p}@psu.ac.th

บทคัดย่อ

เทคโนโลยี RSS (Really Simple Syndication) ทำให้ผู้ใช้สามารถรับข่าวสารใหม่ ๆ ได้ทันต่อเหตุการณ์ โดยการทำงานของ RSS เครื่องแม่ข่ายที่ให้บริการ RSS จะมีตัวรวบรวมข่าวสาร (Aggregator) ทำหน้าที่ดึงข้อมูลจากเว็บไซต์ต่างๆ และจะต้องเป็นผู้กำหนดการตั้งเวลาในการดึงข้อมูลเป็นช่วงเวลาที่เท่ากันเอง ซึ่งช่วงเวลาที่ตั้งไว้ในบางครั้งอาจไม่มีการเปลี่ยนแปลงข้อมูลเลย หรือมีโอกาสได้ข่าวในเวลาที่ช้ากว่าความเป็นจริง งานวิจัยนี้จึงนำเสนอกลไกการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล โดยเรียนรู้จากข้อมูลโดยตรง ซึ่งผลการทดลองแสดงให้เห็นว่าตำแหน่งเวลาในการดึงข้อมูลที่ได้ สามารถลดความล่าช้าได้ถึง 21% เมื่อเทียบกับงานวิจัยก่อนหน้านี้

คำสำคัญ: RSS, ตำแหน่งเวลาในการดึงข้อมูล

1. บทนำ

ปัจจุบันภาษา XML (Extensible Markup Language) [1] เป็นภาษาที่มีการนำมาใช้อย่างแพร่หลายเพื่อส่งผ่านข้อมูลกันบนอินเทอร์เน็ต เทคโนโลยี RSS (Really Simple Syndication) [2] เป็นตัวอย่างหนึ่งที่มีการส่งข้อมูลโดยใช้มาตรฐาน XML เพื่อรวบรวมและเผยแพร่ข่าวสารหรือเนื้อหาต่างๆ ให้ผู้ใช้สามารถรับข้อมูลที่ทันต่อเหตุการณ์ ดังจะเห็นจากบล็อก (Blog) ส่วนบุคคล เว็บไซต์ต่าง ๆ ในปัจจุบันที่มีการนำ เทคโนโลยี RSS มาใช้กันอย่างแพร่หลาย จากความสามารถดังกล่าวนี้ทำให้ง่ายในการเข้าถึงข้อมูลจากหลาย ๆ แหล่ง อย่างไรก็ตามการทำงานของตัวรวบรวมข่าวสารโดยทั่วไปจะทำงานโดยตั้งเวลาในการดึงข้อมูลเป็นช่วงเวลาที่เท่ากัน เช่น ตั้งเวลาในการดึงข้อมูลทุก ๆ 2 ชั่วโมง เป็นต้น จากการตั้งเวลาดังกล่าว ทำให้การดึงข้อมูลในบางครั้งอาจไม่มีการเปลี่ยนแปลงข้อมูลเลย หรือมีโอกาสได้ข่าวในเวลาที่ช้ากว่าความเป็นจริง ดังนั้นการมีกลไกในการหาตำแหน่งเวลาการดึงข้อมูลที่ดีจะช่วยเพิ่มประสิทธิภาพการทำงานของตัวรวบรวมข่าวสาร ทำให้สามารถดึงข้อมูลได้ในเวลาที่เหมาะสม เพื่อให้ได้ข่าวสารล่าสุดและเกิดความล่าช้าในการดึงข้อมูลน้อยที่สุด บทความนี้จึงนำเสนอกลไกการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล โดยจะไม่สร้างแบบจำลองการแสดงข่าวเพื่อนำแบบจำลองมาทำนายเวลาในการดึงข้อมูล แต่จะเรียนรู้จากข้อมูลโดยตรง เนื่องจากผู้วิจัยเห็นว่าการสร้างแบบจำลองเพื่อนำมาทำนายหาตำแหน่งเวลาในการดึงข้อมูลนั้น จะมีโอกาสผิดพลาดมากกว่าการหาตำแหน่งเวลาจากข้อมูลโดยตรง เพราะการแสดงข่าวนั้นมีความไม่แน่นอนสูง

เนื้อหาของบทความในส่วนที่ 2 จะกล่าวถึง เทคโนโลยี RSS ส่วนที่ 3 จะกล่าวถึงงานวิจัยที่เกี่ยวข้อง ส่วนที่ 4 จะกล่าวถึงสถาปัตยกรรมการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล ส่วนที่ 5 การพัฒนากลไกหาตำแหน่งเวลาในการดึงข้อมูลและผลการศึกษา และส่วนที่ 6 เป็นบทสรุป

2. เทคโนโลยี RSS

RSS ย่อมาจาก Really Simple Syndication เป็นข้อมูลที่อยู่ในรูปของภาษา XML พัฒนาขึ้นเพื่อรวบรวมและเผยแพร่เนื้อหาหรือข่าวสารของเว็บไซต์ที่มีการเปลี่ยนแปลงบ่อย โดยผู้ใช้สามารถรับข่าวสารจากเว็บไซต์ที่ให้บริการ RSS ได้ด้วยการใช้โปรแกรมรวบรวมข่าวสารที่เรียกว่า Reader หรือ Aggregator ซึ่งมีหลักการทำงานคล้ายกับโปรแกรมรับอีเมล์ เพียงแต่ผู้รับบริการต้องลงทะเบียนรับข่าวสารที่สนใจจากตัวรวบรวมข่าวสาร โดยตัวรวบรวมข่าวสารจะรวบรวมเอกสาร RSS จากเว็บไซต์ต่างๆ และตรวจสอบการเปลี่ยนแปลงข้อมูล แล้วแสดงผลข้อมูลล่าสุดให้อัตโนมัติ ซึ่งข้อมูลที่ได้จะเป็นเพียงหัวข้อข่าว หรือรายละเอียดโดยย่อเท่านั้น ส่วนเนื้อหา หรือข้อความหลักของข่าวนั้นจะมีลิงค์เชื่อมโยงให้อีกทีหนึ่ง โครงสร้างการทำงานของ RSS แสดงดังรูปที่ 1 ประกอบด้วย 3 ส่วน คือ

1) ผู้เผยแพร่ข่าวสาร (Publisher) คือ เว็บไซต์ที่ให้บริการข้อมูลข่าวสารในรูปแบบเอกสาร RSS
2) ตัวรวบรวมข่าวสาร (Aggregator) ซึ่งทำหน้าที่เป็นตัวแทนรวบรวมข่าวสารจากเว็บไซต์ต่างๆ
3) ผู้รับบริการข้อมูลข่าวสาร (Subscriber) คือ ผู้รับบริการข้อมูลข่าวสารจากเว็บไซต์ต่างๆ

รูปที่ 1 โครงสร้างการทำงานของ RSS

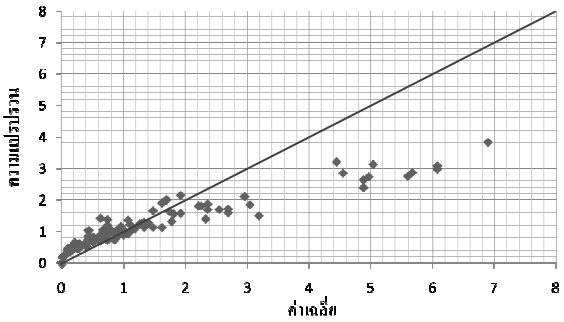
3. งานวิจัยที่เกี่ยวข้อง

จากการนำเทคโนโลยี RSS มาใช้อย่างแพร่หลาย เพื่อรวบรวมและเผยแพร่ข่าวสารให้มีความทันสมัยอยู่เสมอ แต่ตัวรวบรวมข่าวสารโดยทั่วไปจะทำงานโดยตั้งเวลาในการดึงข้อมูลเป็นช่วงเวลาที่เท่า ๆ กัน เช่น ตั้งเวลาในการดึงข้อมูลทุก ๆ 2 ชั่วโมง เป็นต้น ทำให้การดึงข้อมูลในบางครั้งอาจไม่มีการเปลี่ยนแปลงข้อมูลเลย หรือมีโอกาสได้ข่าวในเวลาที่ช้ากว่าความเป็นจริง ซึ่งงานวิจัยหลายงานที่เกี่ยวข้องส่วนใหญ่จะมุ่งเน้นไปที่รูปแบบในการดึงข้อมูลเพื่อจะหาเวลาที่เหมาะสมในการดึงข้อมูล ดังเช่นในปี ค.ศ. 2000 Cho และ Molina [5] ได้ศึกษาความถี่ในการเปลี่ยนแปลงของเว็บ เพื่อเพิ่มประสิทธิภาพให้กับ Web crawler ในการดึงข้อมูลของเว็บนั้น โดยได้นิยาม “fresh” และ “age” ขึ้นมาเพื่อใช้วัดความใหม่และอายุของข้อมูลที่มี ในงานวิจัยได้ใช้การแจกแจงแบบปัวส์ซองมาทำนายการเปลี่ยนแปลงที่เกิด ซึ่งทำให้ Web crawler สามารถดึงข้อมูลให้มีความใหม่เพิ่มขึ้น 35%

ปี ค.ศ. 2005 Sia และ Cho [6, 7] ได้ศึกษาวิธีการที่จะทำให้ตัวรวบรวมข่าวสารมีความล่าช้าในการดึงข้อมูลน้อยที่สุด โดยมีรูปแบบในการดึงข้อมูลสองแบบคือ แบบ Resource allocation และแบบ Retrieval scheduling ซึ่งใช้การแจกแจงแบบปัวส์ซองมาสร้างแบบจำลองในการแสดงข้อมูลแล้วหาตำแหน่งในการดึงข้อมูลที่ดีที่สุด ซึ่งเมื่อเทียบกับแบบตั้งเวลาเป็นช่วงเวลาที่เท่า ๆ กัน แบบ Resource allocation และ Retrieval scheduling ลดความล่าช้าได้ 33% และ 12% ตามลำดับต่อมาในปี ค.ศ. 2007 ได้ปรับปรุงรูปแบบในการดึงข้อมูลจากเดิมที่ดูเฉพาะการแสดงข้อมูลจากแหล่งข้อมูลอย่างเดียว เปลี่ยนมาดูรูปแบบการใช้งานของผู้ใช้ด้วย

ปี ค.ศ. 2008 Han และคณะ [8] ได้นำเสนอรูปแบบใหม่ในการดึงข้อมูลโดยดูจากจำนวนข้อมูลที่จะขาดหายไปเป็นหลัก ซึ่งปรับปรุงมาจากแบบ Resource allocation จากงานวิจัย [6, 7]

จากงานวิจัย[6, 7] พบว่าในการหาตำแหน่งในการดึงข้อมูลจะสร้างแบบจำลองขึ้นจากลักษณะของข้อมูลซึ่งใช้การแจกแจงแบบปัวส์ซอง จากนั้นจะหาตำแหน่งในการดึงข้อมูลที่ทำให้เกิดความล่าช้าน้อยที่สุด โดยใช้วิธีการที่เสนอขึ้นมาใหม่ ซึ่งจะเห็นว่าการหาตำแหน่งเวลาตามวิธีการนั้นจะขึ้นอยู่กับแบบจำลองการแสดงข้อมูล หากมีแบบจำลองที่มีความแม่นยำในการทำนายการแสดงข้อมูล ก็จะทำให้การหาตำแหน่งในการดึงข้อมูลมีความแม่นยำมากขึ้น และทำให้ความล่าช้าในการดึงข้อมูลลดลงตามไปด้วย อย่างไรก็ตามเมื่อพิจารณาแบบจำลองที่ใช้ พบว่าแบบจำลองดังกล่าวได้ถูกนำมาประยุกต์จาก[5] ซึ่งเป็นการทำนายข้อมูลแบบไม่ขึ้นกับเวลา (Time-independent Random Process) แต่เนื่องจากการสร้างแบบจำลองการแสดงข่าว ช่วงเวลามีผลต่อการแสดงข้อมูล จึงใช้แบบจำลองการแจกแจงที่ขึ้นอยู่กับเวลา (Time-dependent Random Process) นอกจากนี้การเลือกใช้ลักษณะการแสดงข้อมูลด้วยการแจกแจงแบบปัวส์ซอง จะได้ผลการทำนายที่ดีก็ต่อเมื่อ ค่าเฉลี่ยกับความแปรปรวนต้องมีค่าใกล้เคียงกัน ($\mu_X \approx \sigma_X^2$) [3] และจากการนำข้อมูลการแสดงข่าวของ BBC ใน 1 เดือน มาสร้างกราฟกระจายระหว่างค่าเฉลี่ยกับความแปรปรวน ดังรูปที่ 2 พบว่าค่า μ_X กับ σ_X^2 มีแนวโน้มที่จะเท่ากัน แต่เมื่อพิจารณาถึงค่าของ $\frac{\left|\mu_X - \sigma_X^2\right|}{\mu_X}$ และ $\frac{\left|\mu_X - \sigma_X^2\right|}{\sigma_X^2}$ พบว่า เมื่อเทียบเป็นเปอร์เซ็นต์แล้วจะมีความแตกต่างกันมาก เช่น ค่าเฉลี่ย $(\mu_X) = 0.3$ ความแปรปรวน $(\sigma_X^2) = 0.5$ ค่าของ $\frac{\left|\mu_X - \sigma_X^2\right|}{\mu_X} = \frac{0.2}{0.3} = 0.66$ ซึ่งเท่ากับ 66% และ $\frac{\left|\mu_X - \sigma_X^2\right|}{\sigma_X^2} = \frac{0.2}{0.5} = 0.4$ ซึ่งเท่ากับ 40% นอกจากนี้ยังมีความไม่แน่นอนในการแสดงข่าวสูงเมื่อมีการกำหนดค่าในแต่ละช่วงเวลาว่ามีจำนวนการแสดงข่าวเป็นเท่าไร ดังนั้นการใช้การแจกแจงแบบปัวส์ซองสร้างแบบจำลองจึงมีโอกาสเกิดความคลาดเคลื่อนจากความเป็นจริงได้มาก ซึ่งแบบจำลองดังกล่าวข้างต้นถือเป็นตัวแบบการแสดงข้อมูลที่เหมือนกันทุกวัน ดังนั้นหากได้แบบจำลองที่ไม่ดี อาจส่งผลต่อการกำหนดตำแหน่งเวลาในการดึงข้อมูลเป็นอย่างมาก ผู้วิจัยจึงนำเสนอกลไกในการค้นหาตำแหน่งเวลาในการดึงข้อมูล (Discovering Optimal Retrieval Points Mechanism: DORPM) เพื่อให้เกิดความล่าช้าในการดึงข้อมูลน้อยที่สุด

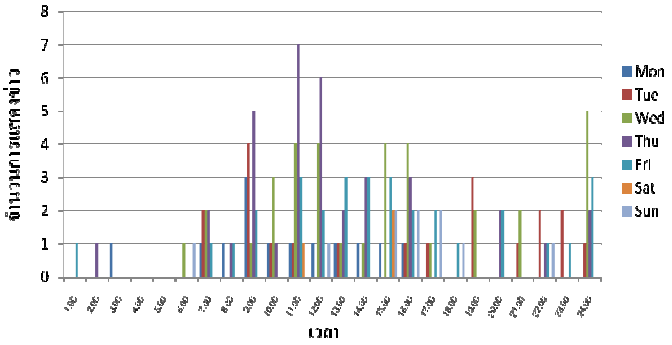


รูปที่ 2 กราฟกระจายระหว่างค่าเฉลี่ยกับความแปรปรวนของข่าวจาก BBC ในเดือน เม.ย. 53

4. สถาปัตยกรรมการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

4.1 การออกแบบสถาปัตยกรรมการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

ในการออกแบบสถาปัตยกรรมการหาตำแหน่งเวลาในการดึงข้อมูล ผู้วิจัยได้ศึกษาลักษณะการแสดงข่าวเศรษฐกิจจากแหล่งข่าว BBC เป็นเวลา 1 สัปดาห์ โดยแบ่งช่วงเวลาในการดึงข้อมูลออกเป็นช่วงๆ ละ 1 ชม. เท่ากัน ดังรูปที่ 3 พบว่าลักษณะการแสดงข่าวในแต่ละวันจะมีลักษณะการแสดงข่าวที่คล้ายกัน คือ ช่วงเวลา 0.00 น. – 6.00 น. จะไม่มีการแสดงข่าวเลยหรือมีค่อนข้างน้อย เมื่อเทียบกับช่วงเวลา 6.00 น. – 12.00 น. อีกทั้งในการแบ่งช่วงเวลาในการดึงข้อมูลออกเป็นช่วงเวลาที่เท่ากัน ในกรณีที่ ณ เวลาที่ต้องดึงข้อมูลแต่ไม่มีข้อมูลที่จะแสดง ก็ทำให้เสียทรัพยากรในการดึงข้อมูลโดยเปล่าประโยชน์ นอกจากนี้ เมื่อศึกษาการดึงข้อมูลโดยแบ่งออกเป็นช่วงเวลาที่ห่างกันมากขึ้น โดยได้แบ่งออกเป็น 4 ช่วงคือ 06.00 น. 12.00 น. 18.00 น. และ 24.00 น. พบว่า หากแบ่งช่วงกว้างเกินไปก็อาจส่งผลเกิดความล่าช้าในการดึงข้อมูล ทำให้การแสดงข่าวไม่เป็นปัจจุบัน จากปัญหาดังกล่าวข้างต้น การหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลจะช่วยให้มีการดึงข้อมูลโดยใช้ทรัพยากรอย่างมีประสิทธิภาพและลดความล่าช้าของการดึงข้อมูล

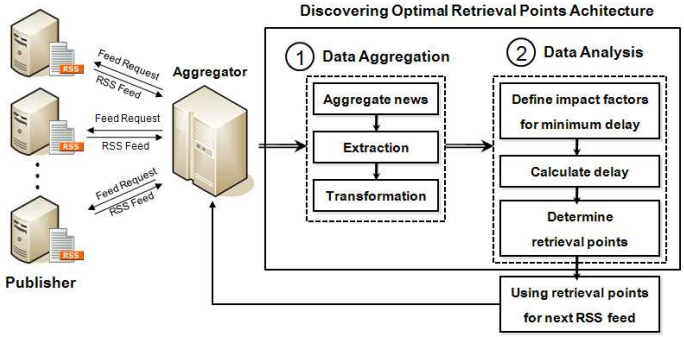


รูปที่ 3 ลักษณะการแสดงข่าวเศรษฐกิจของ BBC ระหว่างวันที่ 5-12 เม.ย. 53

4.2 การทำงานของสถาปัตยกรรมเวลาที่เหมาะสมในการดึงข้อมูล

การทำงานของสถาปัตยกรรมการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล (Discovering Optimal Retrieval Point Architecture: DORPA) ประกอบด้วย 2 ส่วนสำคัญ คือ ส่วนรวบรวมข้อมูล และส่วนวิเคราะห์ข้อมูล

1. **ส่วนรวบรวมข้อมูล (Data Aggregation)** ทำหน้าที่รวบรวมข่าวสารจากแหล่งข่าวต่างต่างๆ โดยมีขั้นตอนการทำงานดังนี้
   1) รวบรวมข่าวสารจากแหล่งข่าวต่างๆ
   2) สกัดข้อมูลโดยนำแท็กเวลาในการแสดงข่าว มาสกัดเอาข้อมูลวันและเวลาเก็บเอาไว้ในฐานข้อมูล
   3) แปลงข้อมูลวันและเวลาจัดให้อยู่ในรูปแบบที่เหมาะสม โดยรวบรวมจำนวนข่าวเป็นช่วง ๆ ละ 1 ชม. เท่า ๆ กัน เพื่อให้ง่ายต่อการนำข้อมูลไปวิเคราะห์ในขั้นตอนต่อไป
2. **ส่วนวิเคราะห์ข้อมูล (Data Analysis)** ทำหน้าที่วิเคราะห์ข้อมูลเพื่อหาตำแหน่งเวลาที่เหมาะสมที่สุดในการดึงข้อมูล โดยนำข้อมูลที่ได้จากส่วนที่ 1 มาคำนวณหาตำแหน่งในการดึงข้อมูลที่ทำให้เกิดความล่าช้าน้อยที่สุด ประกอบด้วย 3 ขั้นตอน ดังนี้
   1) กำหนดปัจจัยที่ส่งผลต่อความล่าช้าในการดึงข้อมูล ซึ่งจะประกอบด้วย ตำแหน่งเวลาในการแสดงข่าว, จำนวนข่าวที่แสดงในช่วงเวลานั้น และตำแหน่งในการดึงข้อมูล
   2) คำนวณความล่าช้าที่เกิดขึ้น โดยนำปัจจัยดังกล่าวมาคำนวณความล่าช้าที่ตำแหน่งเวลาในการดึงข้อมูลต่างๆ
   3) กำหนดตำแหน่งเวลาในการดึงข้อมูลที่ทำให้เกิดความล่าช้าน้อยที่สุด



รูปที่ 4 สถาปัตยกรรมการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล (DORPA)

4.3 คำนิยามที่ใช้ในสถาปัตยกรรมการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล

เพื่อให้ง่ายต่อการเข้าใจ จึงขอนิยามปัจจัยต่างๆที่ใช้ในการสร้างสถาปัตยกรรมการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูลดังต่อไปนี้

บทนิยามที่ 1 กำหนดให้ T คือ เซตของตำแหน่งเวลาในแต่ละวัน ซึ่ง $T = \{1, 2, \ldots, 24\}$ โดยที่ 1 คือเวลา 1.00 น. , 2 คือเวลา 2.00 น. ,..., 24 คือเวลา 24.00 น. เราจะกล่าวว่า τ คือ ตำแหน่งในการดึงข้อมูล ซึ่ง $\tau \in T$ โดยที่ $\tau = 1$ คือการดึงข้อมูล ณ เวลา 1.00 น. , $\tau = 2$ คือการดึงข้อมูล ณ เวลา 2.00 น. ,..., $\tau = 24$ คือ การดึงข้อมูล ณ เวลา 24.00 น. □

บทนิยามที่ 2 กำหนดให้ $i \in T$ เป็นตำแหน่งเวลา โดย τ_j เป็นตำแหน่งในการดึงข้อมูล จะได้ว่าเวลาที่ดึงข้อมูลช้าไป ณ เวลา i (d_i) คือ ผลต่างระหว่างเวลาที่ i กับเวลาในการดึงข้อมูล ดังนี้

$$d_i = \tau_j - i \text{ โดย } \tau_j \text{ เป็นเวลาที่ใกล้ที่สุดที่ } i \le \tau_j$$ □

บทนิยามที่ 3 กำหนดให้ η_i เป็นจำนวนข่าวในช่วงเวลา i -1 ถึง i ในแต่ละวัน โดยที่ η_1 คือ จำนวนข่าวตั้งแต่เวลา 0.00-0.59 น., η_2 คือ จำนวนข่าวตั้งแต่เวลา 1.00-1.59 น. , ..., η_{24} คือ จำนวนข่าวตั้งแต่เวลา 23.00-23.59 น. □

บทนิยามที่ 4 กำหนดให้ $i \in T$ เป็นตำแหน่งเวลา โดย η_i เป็นจำนวนข่าวที่แสดง ณ เวลา i และ τ_j เป็นตำแหน่งในการดึงข้อมูล จะได้ว่าความล่าช้าในการดึงข้อมูล ณ เวลา i (D_i) คือ ผลคูณระหว่างจำนวนข่าวที่แสดง ณ เวลานั้นคูณกับเวลาที่ดึงข้อมูลช้าไป ดังนี้

$$D_i = \eta_i d_i = \eta_i(\tau_j - i) \text{ โดย } \tau_j \text{ เป็นเวลาที่ใกล้ที่สุดที่ } i \le \tau_j$$ □

* กำหนดให้ $D_i(k)$ หมายถึงความล่าช้าในการดึงข้อมูล ณ เวลา i ของวันที่ k และ $\eta_i(k)$ หมายถึงจำนวนข่าวที่แสดง ณ เวลา i ของวันที่ k

บทตั้งที่ 1 กำหนดให้ $i \in T$ เป็นตำแหน่งเวลา จะได้ว่าผลรวมความล่าช้าใน 1 ช่วงของการดึงข้อมูล หรือผลรวมความล่าช้าในช่วง τ_j ถึง τ_{j+1} คือ

$$\sum_{i=\tau_j}^{\tau_{j+1}} D_i = \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(\tau_{j+1} - i) \text{ โดย } i \text{ เป็นเวลาระหว่าง } \tau_j \text{ ถึง } \tau_{j+1}$$ □

พิสูจน์ จากช่วงของการดึงข้อมูลที่พิจารณา คือ τ_j ถึง τ_{j+1}
จะได้ว่า ถ้า $i \le \tau_j$ ความล่าช้าที่เกิดขึ้นจะเป็นของช่วงก่อนหน้านี้
และถ้า $i > \tau_{j+1}$ ความล่าช้าที่เกิดขึ้นจะเป็นของช่วงถัดไป
ดังนั้น i ที่พิจารณาจะอยู่ในช่วง $\tau_j + 1 \le i \le \tau_{j+1}$
จาก τ_{j+1} เป็นเวลาที่ใกล้ที่สุดที่ $i \le \tau_{j+1}$
จะได้ว่าเวลาที่ช้าไป คือ $\tau_{j+1} - i$
ดังนั้นผลรวมความล่าช้าในช่วง τ_j ถึง τ_{j+1}
คือ $\sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(\tau_{j+1} - i)$ ❖

บทตั้งที่ 2 กำหนดให้มีการแสดงข้อมูล ณ เวลา $i \in T$ โดย η_i เป็นจำนวนข่าวที่แสดง ณ เวลา i และมีการดึงข้อมูล m ครั้ง ณ ตำแหน่งเวลา $\tau_1, \tau_2, ..., \tau_m$ จะได้ว่าผลรวมความล่าช้าสะสมในหนึ่งวัน คือ

$$\sum_{j=1}^{m} \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(\tau_{j+1} - i) \quad \text{โดย } \tau_{m+1} = \tau_1 + 24 \qquad \square$$

พิสูจน์ จากการดึงข้อมูล m ครั้ง ณ ตำแหน่งเวลา $\tau_1, \tau_2, ..., \tau_m$ จะแบ่งเวลาออกเป็น $m+1$ ช่วง คือ ช่วงเวลา 1 ถึง τ_1, τ_1+1 ถึง τ_2, τ_2+1 ถึง τ_3, ... , $\tau_{m-1}+1$ ถึง τ_m, τ_m+1 ถึง 24 จากบทตั้งที่ 1 จะได้ว่าผลรวมความล่าช้าในแต่ละช่วงคือ $\sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(\tau_{j+1} - i)$

แต่ช่วงเวลา τ_m+1 ถึง 24 ตำแหน่งในการดึงข้อมูลจะเป็น τ_1 ในวันถัดไป ดังนั้นเวลาที่ดึงข้อมูลช้าไปของช่วงนี้คือ $\tau_1 + 24 - i$

นั่นคือ $\sum_{i=\tau_m+1}^{24} D_i = \sum_{i=\tau_m+1}^{24} \eta_i(\tau_1 + 24 - i)$

และจากช่วงเวลา 1 ถึง τ_1 เป็นช่วงเวลาเดียวกันกับ $1 + 24$ ถึง $\tau_1 + 24$ ในวันถัดไป ซึ่งใช้ค่า η_i จากวันเดิม ดังนั้นเวลาที่ดึงข้อมูลช้าไปของช่วงนี้คือ $\tau_1 + 24 - i$ และผลรวมความล่าช้าของช่วงเวลา 1 ถึง τ_1 คือ

$$\sum_{i=1}^{\tau_1} \eta_i(\tau_1 - i) = \sum_{i=24+1}^{\tau_1+24} \eta_i(\tau_1 + 24 - i) \qquad (1)$$

จะได้ผลรวมความล่าช้าในแต่ละช่วงรวมกัน คือ

$$\sum_{i=1}^{\tau_1} \eta_i(\tau_1 - i) + \sum_{i=\tau_1+1}^{\tau_2} \eta_i(\tau_2 - i) + ... + \sum_{i=\tau_{m-1}+1}^{\tau_m} \eta_i(\tau_m - i) + \sum_{i=\tau_m+1}^{24} \eta_i(\tau_1 + 24 - i) \qquad (2)$$

จากสมการ (1) แทนในสมการ (2) จะได้

$$\sum_{i=24+1}^{\tau_1+24} \eta_i(\tau_1 + 24 - i) + \sum_{i=\tau_1+1}^{\tau_2} \eta_i(\tau_2 - i) + ... + \sum_{i=\tau_{m-1}+1}^{\tau_m} \eta_i(\tau_m - i) + \sum_{i=\tau_m+1}^{24} \eta_i(\tau_1 + 24 - i)$$

$$= \sum_{i=\tau_1+1}^{\tau_2} \eta_i(\tau_2 - i) + \sum_{i=\tau_2+1}^{\tau_3} \eta_i(\tau_3 - i) + ... + \sum_{i=\tau_{m-1}+1}^{\tau_m} \eta_i(\tau_m - i) + \sum_{i=\tau_m+1}^{24} \eta_i(\tau_1 + 24 - i)$$

$$+ \sum_{i=24+1}^{\tau_1+24} \eta_i(\tau_1 + 24 - i)$$

$$= \sum_{i=\tau_1+1}^{\tau_2} \eta_i(\tau_2 - i) + \sum_{i=\tau_2+1}^{\tau_3} \eta_i(\tau_3 - i) + ... + \sum_{i=\tau_{m-1}+1}^{\tau_m} \eta_i(\tau_m - i) + \sum_{i=\tau_m+1}^{\tau_1+24} \eta_i(\tau_1 + 24 - i)$$

$$= \sum_{j=1}^{m} \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(\tau_{j+1} - i) \quad \text{โดย } \tau_{m+1} = \tau_1 + 24 \qquad \text{❖}$$

บทตั้งที่ 3 กำหนดให้มีการดึงข้อมูล n วัน ในแต่ละวันมีการแสดงข้อมูล ณ เวลา $i \in T$ โดย $\eta_i(k)$ เป็นจำนวนข่าวที่แสดง ณ เวลา i ของวันที่ k และมีการดึงข้อมูล m ครั้งต่อวัน ณ ตำแหน่งเวลา $\tau_1, \tau_2, ..., \tau_m$ ตำแหน่งเดียวกันทุกวัน จะได้ว่าผลรวมความล่าช้าสะสม n วัน ที่ $\tau = \{\tau_1, \tau_2, ..., \tau_m\}$ คือ

$$D(n, \tau) = \sum_{k=1}^{n} \sum_{j=1}^{m} \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(k)(\tau_{j+1} - i)$$

$$\text{โดย } \tau_{m+1} = \tau_1 + 24 \text{ และ } k = 1, 2, ..., n \qquad \square$$

พิสูจน์ เนื่องจากจำนวนข่าวที่แสดงในแต่ละวันมีความแตกต่างกัน

ดังนั้นกำหนดให้ $\eta_i(k)$ เป็นจำนวนข่าวที่แสดง ณ เวลา i ของวันที่ k จากบทตั้งที่ 2 ผลรวมความล่าช้าสะสมในแต่ละวัน คือ

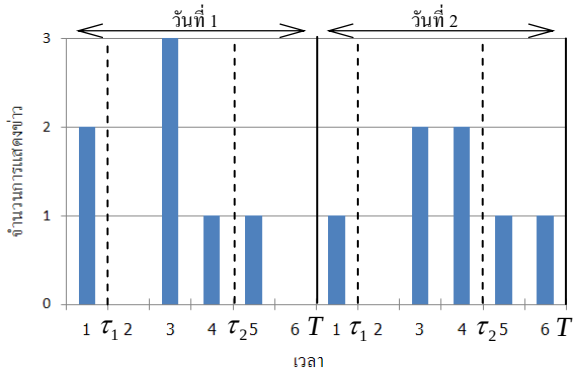
$\sum_{j=1}^{m} \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(k)(\tau_{j+1} - i)$ โดย $\tau_{m+1} = \tau_1 + 24$ และ $k = 1, 2, ..., n$

จะได้ผลรวมความล่าช้าสะสม n วัน คือ

$$\sum_{j=1}^{m} \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(1)(\tau_{j+1} - i) + \sum_{j=1}^{m} \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(2)(\tau_{j+1} - i) + ... + \sum_{j=1}^{m} \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(n)(\tau_{j+1} - i)$$

จากการดึงข้อมูล $\tau_1, \tau_2, ..., \tau_m$ เป็นตำแหน่งเดียวกันทุกวัน ดังนั้นผลรวมความล่าช้าสะสม n วัน ที่ $\tau = \{\tau_1, \tau_2, ..., \tau_m\}$ คือ

$$\sum_{k=1}^{n} \sum_{j=1}^{m} \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(k)(\tau_{j+1} - i) \quad \text{โดย } \tau_{m+1} = \tau_1 + 24 \text{ และ } k = 1, 2, ..., n \qquad \text{❖}$$



รูปที่ 5 การแสดงข้อมูล 2 วัน แต่ละวันแบ่งออกเป็น 6 ช่วงเวลา

ตัวอย่างที่ 1 อธิบายการหาผลรวมความล่าช้าสะสม 2 วัน สมมติให้แหล่งข่าวมีการแสดงข่าวดังรูปที่ 5 โดยกำหนดตำแหน่งเวลาในการดึงข้อมูล 2 ครั้ง คือ $\tau_1 = 1$ และ $\tau_2 = 4$

จะได้ $i \in T = \{1, 2, 3, 4, 5, 6\}$

$\eta_1(1), \eta_2(1), ..., \eta_6(1) = 2, 0, 3, 1, 1, 0$ ตามลำดับ

$\eta_1(2), \eta_2(2), ..., \eta_6(2) = 1, 0, 2, 2, 1, 1$ ตามลำดับ

จะได้ความล่าช้าในแต่ละช่วง คือ

$D_1(1) = \eta_1(1)(\tau_1 - 1) = 2(1 - 1) = 0$

$D_2(1) = \eta_2(1)(\tau_2 - 2) = 0(4 - 2) = 0$

$D_3(1) = \eta_3(1)(\tau_2 - 3) = 3(4 - 3) = 3$

$D_4(1) = \eta_4(1)(\tau_2 - 4) = 1(4 - 4) = 0$

$D_5(1) = \eta_5(1)(\tau_1 + 6 - 5) = 1(1 + 6 - 5) = 2$

$D_6(1) = \eta_6(1)(\tau_1 + 6 - 6) = 0(1 + 6 - 6) = 0$

$D_1(2) = \eta_1(2)(\tau_1 - 1) = 1(1 - 1) = 0$

$D_2(2) = \eta_2(2)(\tau_2 - 2) = 0(4 - 2) = 0$

$D_3(2) = \eta_3(2)(\tau_2 - 3) = 2(4 - 3) = 2$

$D_4(2) = \eta_4(2)(\tau_2 - 4) = 2(4 - 4) = 0$

$D_5(2) = \eta_5(2)(\tau_1 + 6 - 5) = 1(1 + 6 - 5) = 2$

$D_6(2) = \eta_6(2)(\tau_1 + 6 - 6) = 1(1 + 6 - 6) = 1$

ดังนั้นผลรวมความล่าช้าสะสม 2 วัน คือ

$$\sum_{k=1}^{2} \sum_{j=1}^{2} \sum_{i=\tau_j+1}^{\tau_{j+1}} \eta_i(k)(\tau_{j+1} - i) = 0 + 0 + 3 + 0 + 2 + 0 + 0 + 0 + 2 + 0 + 2 + 1 = 10$$

4.4 กลไกการหาตำแหน่งเวลาในการดึงข้อมูล

ในการเรียนรู้ข้อมูลการแสดงข่าวในแต่ละวันที่เพิ่มเข้ามา จะทำให้ตำแหน่งเวลาในการดึงข้อมูลอาจเปลี่ยนแปลงได้ขึ้นอยู่กับการแสดงข้อมูล ดังนั้นจำเป็นต้องมีวิธีการหาตำแหน่งในการดึงข้อมูล โดยใช้ตำแหน่งเดียวกันทุกวันที่ทำให้มีความล่าช้าสะสมน้อยที่สุด

กำหนดให้แหล่งข้อมูลมีการดึงข้อมูล m ครั้ง/วัน จะได้ว่าความล่าช้าสะสมใน 1 วัน $D(1)$ มีตำแหน่งในการดึงข้อมูลที่ทำให้เกิดความล่าช้าสะสมน้อยที่สุด คือ $\tau(1) = \{\tau_1, \tau_2, ..., \tau_m\}$ และ $D(2)$ มีตำแหน่งการดึงข้อมูล คือ $\tau(2)$, ..., $D(n)$ มีตำแหน่งการดึงข้อมูล คือ $\tau(n)$

ดังนั้น $D(1), D(2), ..., D(n)$ เป็นความล่าช้าสะสมที่น้อยที่สุดของ 1,2,..., n วัน ตามลำดับ โดย $\tau(1), \tau(2), ..., \tau(n)$ เป็นตำแหน่งการดึงข้อมูลที่ทำให้เกิดความล่าช้าสะสมน้อยที่สุด ซึ่งอาจมีตำแหน่งที่แตกต่างกัน แต่เนื่องจากตำแหน่งในการดึงข้อมูลต้องเป็นตำแหน่งเดียวกันทุกวัน ดังนั้นจะต้องเลือกตำแหน่งการดึงข้อมูลเพียงแค่ตำแหน่งเดียว โดยจะเลือก $\tau(j) \in \{\tau(1), \tau(2), ..., \tau(n)\}$ ที่ทำให้ $\sum_{i=1}^{n} |D(i, \tau(j)) - D(i, \tau(i))|$ มีค่าน้อยที่สุด

ตัวอย่างที่ 2 อธิบายการหาตำแหน่งเวลาในการดึงข้อมูล โดยให้แหล่งข่าวมีการแสดงข่าวดังรูปที่ 5 และกำหนดให้ดึงข้อมูล 2 ครั้ง/วัน จะได้ว่าตำแหน่งในการดึงข้อมูลที่ทำให้เกิดความล่าช้าสะสมน้อยที่สุดของวันแรกคือ $\tau(1) =$ {1,3} โดยมีความล่าช้าสะสม $D(1, \tau(1)) =$ 5 และตำแหน่งในการดึงข้อมูลที่ทำให้เกิดความล่าช้าสะสมน้อยที่สุดของ 2 วัน คือ $\tau(2) =$ {1,4} โดยมีความล่าช้าสะสม $D(2, \tau(2)) = 10$

จะได้ว่า $D(1, \tau(1)) = 5$ $D(2, \tau(1)) = 14$

$D(1, \tau(2)) = 5$ $D(2, \tau(2)) = 10$

ดังนั้น $\tau(j) = \tau(2)$ ทำให้ $\sum_{i=1}^{2} |D(i, \tau(j)) - D(i, \tau(i))|$ มีค่าน้อยที่สุด นั่นคือจะเลือกตำแหน่งเวลา 1 กับ 4 เป็นตำแหน่งเวลาในการดึงข้อมูลทุกวัน

5. การพัฒนากลไกหาตำแหน่งเวลาในการดึงข้อมูล

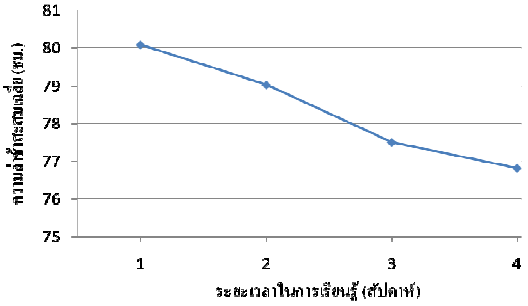
ในการพัฒนากลไกหาตำแหน่งเวลาในการดึงข้อมูล ประกอบด้วย 3 ขั้นตอนซึ่งมีรายละเอียดการพัฒนา ดังนี้

1. การรวบรวมข่าวสาร

ใช้ชุดข้อมูลที่ได้เก็บข้อมูลเองจากเว็บไซต์ข่าวสารที่ให้บริการ RSS คือ BBC [9] CNN [10] และ Reuters[11] โดยแบ่งประเภทของข่าวออกเป็น 5 ประเภท คือ ข่าวเศรษฐกิจ ข่าวบันเทิง ข่าวเด่น ข่าวสหรัฐอเมริกาและข่าวรอบโลก และทำการเก็บข้อมูล 3 เดือนตั้งแต่เมษายน 2553 ถึงมิถุนายน 2553 มีจำนวนข่าวทั้งหมด 42,375 ข่าว แบ่งออกเป็นข่าวของ BBC ทั้งหมด 17,599 ข่าว CNN ทั้งหมด 10,638 ข่าว และ Reuters ทั้งหมด 14,138 ข่าว

2. การเรียนรู้ตำแหน่งเวลาในการดึงข้อมูล

จากข้อมูลที่ได้เก็บรวบรวม โดยแบ่งช่วงเวลาในการแสดงข่าวออกเป็นช่วง ๆ ละ 1 ชม. จะได้ข้อมูลที่มีลักษณะในการแสดงข้อมูลในแต่ละวันที่คล้ายกัน ซึ่งจะใช้คุณลักษณะนี้มาทำการหาตำแหน่งเวลาในการดึงข้อมูล โดยทดลองด้วยระยะเวลาในการเรียนรู้ที่ต่างกัน รูปที่ 6 แสดงระยะเวลาในการเรียนรู้กับความล่าช้าสะสมเฉลี่ยที่เกิดขึ้น



รูปที่ 6 ผลของระยะเวลาในการเรียนรู้ที่เพิ่มขึ้น

จะเห็นได้ว่าเมื่อระยะเวลาในการเรียนรู้มากขึ้นความล่าช้าสะสมเฉลี่ยที่เกิดขึ้นมีแนวโน้มที่ลดลง

3. กำหนดตำแหน่งเวลาในการดึงข้อมูล

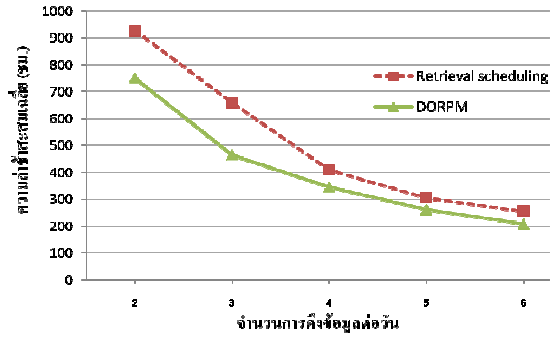
นำข้อมูลมาทำการเรียนรู้เป็นเวลา 1 เดือน แล้วกำหนดตำแหน่งในการดึงข้อมูลด้วยวิธีการดังกล่าวข้างต้น โดยเลือกตำแหน่งที่ทำให้ $\sum_{i=1}^{n} |D(i, \tau(j)) - D(i, \tau(i))|$ มีค่าน้อยที่สุด

5.1 ผลการศึกษาและบทวิจารณ์

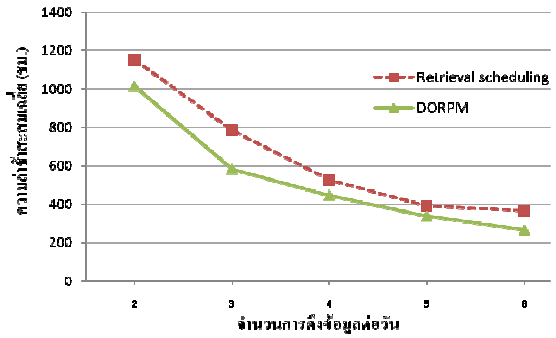
จากผลการเรียนรู้ที่ได้ พบว่าระยะเวลาในการเรียนรู้ที่เพิ่มมากขึ้นความล่าช้าที่ได้จะลดลง จึงเลือกใช้ข้อมูล 1 เดือนเพื่อทำการเรียนรู้และข้อมูลอีก 2 เดือนที่เหลือเพื่อคำนวณหาค่าความล่าช้าที่เกิดขึ้นจากตำแหน่งเวลาในการดึงข้อมูลที่ได้เรียนรู้ในเดือนแรก ในการทดลองเพื่อศึกษาผลกระทบของรูปแบบในการหาตำแหน่งเวลาในการดึงข้อมูลกับความล่าช้าสะสมที่เกิดขึ้น จะทำการเปรียบเทียบโดยแบ่งออกเป็น 2 รูปแบบ คือ

1) Retrieval scheduling ตำแหน่งเวลาในการดึงข้อมูลจะเป็นไปตามวิธีการใน [6, 7]
2) DORPM ตำแหน่งเวลาในการดึงข้อมูลจะเป็นไปตามตำแหน่งที่มีความล่าช้าสะสมน้อยที่สุด

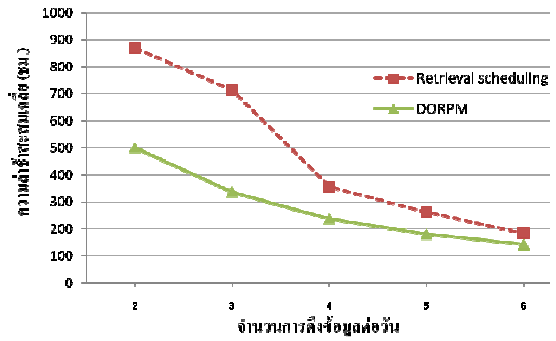
ผลการทดลองแสดงให้เห็นว่า ตำแหน่งในการดึงข้อมูลแบบ DORPM ทำให้มีความล่าช้าสะสมเฉลี่ยน้อยกว่าแบบ Retrieval scheduling ดังรูปที่ 7-10



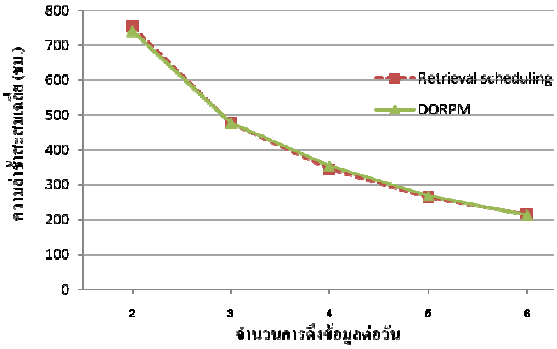
รูปที่ 7 ประสิทธิภาพของการดึงข้อมูลแต่ละรูปแบบ โดยใช้ข้อมูลจากทั้ง 3 แหล่งข่าว



รูปที่ 8 ประสิทธิภาพของการดึงข้อมูลแต่ละรูปแบบ โดยใช้ข้อมูลจาก BBC



รูปที่ 9 ประสิทธิภาพของการดึงข้อมูลแต่ละรูปแบบ โดยใช้ข้อมูลจาก CNN



รูปที่ 10 ประสิทธิภาพของการดึงข้อมูลแต่ละรูปแบบ โดยใช้ข้อมูลจาก REUTERS

จากผลการทดลองจะเห็นได้ว่ารูปแบบการหาตำแหน่งเวลาในการดึงข้อมูลแบบ DORPM ทำให้เกิดความล่าช้าในการดึงข้อมูลน้อยเมื่อเทียบกับ Retrieval scheduling จากรูปที่ 9 จะเห็นว่ารูปแบบ Retrieval scheduling มีความล่าช้าสะสมที่มากผิดปกติ เนื่องจากการสร้างแบบจำลองขึ้นอยู่กับลักษณะของข้อมูลที่ใช้การแจกแจงแบบปัวส์ซอง ดังนั้นถ้าตำแหน่งในการดึงข้อมูลที่ได้นั้นไม่มีการแสดงข้อมูล ก็จะทำให้เกิดความผิดพลาดในการหาตำแหน่งของการดึงข้อมูล และจากรูปที่ 10 จะเห็นว่าความล่าช้าสะสมที่ได้มีค่าใกล้เคียงกันเนื่องจากการแสดงข้อมูลของแหล่งข่าวนี้มีการแสดงข่าวสม่ำเสมอ จึงทำให้การดึงข้อมูลทั้ง 2 รูปแบบมีความล่าช้าสะสมใกล้เคียงกัน

6. บทสรุป

งานวิจัยนี้ได้นำเสนอรูปแบบในการหาตำแหน่งที่เหมาะสมในการดึงข้อมูลจากเดิมที่ต้องสร้างแบบจำลองในการแสดงข้อมูลก่อนแล้วจึงนำแบบจำลองนั้นมาหาตำแหน่งในการดึงข้อมูล [6, 7] ซึ่งในการสร้างแบบจำลองแบบเดิมจะใช้ข้อมูลเพียงแค่ 2 สัปดาห์เท่านั้น ทำให้มีโอกาสผิดพลาดได้สูง โดยกลไกที่นำเสนอขึ้น (DORPM) จะทำให้เกิดความล่าช้าในการดึงข้อมูลน้อยลงหากใช้ระยะเวลาในการเรียนรู้เพิ่มขึ้น ผลการทดลองแสดงให้เห็นว่าข้อมูลในการเรียนรู้ 1 เดือน สามารถลดความล่าช้าได้ถึง 21% เมื่อเทียบกับงานวิจัย [6, 7] อย่างไรก็ตามหากใช้เวลาเรียนรู้ที่มากขึ้นก็จะสามารถลดความล่าช้าลงได้อีก ช่วยให้ผู้ใช้ได้รับข่าวสารที่มีความทันสมัยมากขึ้น ทั้งยังช่วยให้ตัวรวบรวมข่าวสารจัดสรรทรัพยากรในการดึงข้อมูลได้อย่างมีประสิทธิภาพอีกด้วย

7. เอกสารอ้างอิง

[1] W3C, “Extensible Markup Language (XML)”, [Online]. Available: http://www.w3.org/XML/ (July 13, 2010).

[2] W3Schools, “RSS Tutorial”, [Online]. Available: http://www.w3schools.com/rss/ (July 13, 2010).

[3] W. Gardner, E.P. Mulvey, and E.C. Shaw, “Regression Analyses of Counts and Rates: Poisson, Overdispersed Poisson, and Negative Binomial Models”. *Psychological Bulletin*., Vol. 118, No. 3, pp. 392-404, 1995.

[4] S. Lipschuts and J.J. Schiller, “Theory and Problems of Finite Mathematics”, McGraw-Hill, USA, 1995.

[5] J. Cho and H. Garcia-Molina, “Synchronizing a Database to Improve Freshness”. *Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’00),* 2000.

[6] K. Sia, J. Cho, “Efficient Monitoring Algorithm for Fast News Alert”, technical report, University of California, Los Angeles, 2005.

[7] K. Sia, J. Cho and Y. Cho “Efficient Monitoring Algorithm for Fast News Alert”, *IEEE Trans. Knowledge and Data Eng.*, VOL. 19, pp. 950-961, 2007.

[8] Y.G. Han, S.H. Lee, J.H. Kim and Y. Kim, “A News Aggregation Policy for RSS Services”, *Proceeding of the 2008 international workshop on Context enabled source and service selection, integration and adaptation*, Vol. 292, 2008.

[9] BBC.co.uk [Online]. Available: http://news.bbc.co.uk/2/hi/help/3223484.stm (July 13, 2010).

[10] CNN.com [Online]. Available: http://www.cnn.com/services/rss/ (July 13, 2010).

[11] REUTERS.com [online]. Available: http://www.reuters.com/tools/rss (July 13, 2010).

ภาคผนวก ข.

ผลงานตีพิมพ์

เรื่อง	Determining Optimal Retrieval Points Mechanism for RSS Documents
งานประชุมวิชาการ	2011 3rd International Conference on Advanced Computer Control (ICACC 2011)
สถานที่	Harbin, China
วันที่	18 – 20 มกราคม 2554

2011 3rd International Conference on Advanced Computer Control (ICACC 2011)

Determining Optimal Retrieval Points Mechanism for RSS Documents

Chaowanan Khundam and Ladda Preechaveerakul
Department of Computer Science
Faculty of Science, Prince of Songkla University
HatYai, Songkhla, Thailand
e-mail: {s5210220147, ladda.p}@psu.ac.th

***Abstract*— Really Simple Syndication or RSS Technology helps users to receive updated contents using an aggregator from web sites. The aggregator is scheduled at time intervals to feed automatically. However, setting time intervals may cause a delay between the publication of new contents at a publisher site and the appearance at the aggregator, or no updated contents may occur. Then, we propose a mechanism to determine an optimal retrieval point for RSS feeds. The mechanism reduces the delay of retrieving contents. The study results show that the retrieval point can reduce 21% of the delay in retrieving contents in comparison with a previous research.**

Keywords- Aggregator; Really Simple Syndication; Retrieval Point

I. INTRODUCTION

Currently, Extensible Markup Language (XML) [1] is a meta-markup language widely used in the Internet, and enables a general availability and interchange structured information. Each XML-based markup language is called an application. One of the XML applications is Really Simple Syndication (RSS). RSS [2] is mostly used in weblog and news web sites to share and view updated contents from different sites using an aggregator. The aggregator polls specified URLs, using HTTP, to find an information feed under a setup time interval, for example, every 2 hours. However, setting time intervals may cause a delay between the publication of new contents at a publisher site and the appearance at the aggregator, or no updated contents may occur. Most current researches focus on a retrieval model to find an optimal retrieval time. In the year 2000, Cho and Garcia-Molina [5] studied the web change frequency to enhance the retrieving of web information using a web crawler. They have defined "freshness" and "age" to measure what the latest information is. This research used Poisson distribution to predict the occurrences. The research increased web crawler retrieving new information up 35%. In 2005, Sia and Cho [6, 7] studied how the blog aggregator gets a minimal delay to retrieve new postings. They defined two main retrieval policies: Resource allocation and Retrieval scheduling, used Poisson distribution to generate posting model for finding the best retrieving point. In comparison with a periodic retrieving posting aggregator, the resource allocation and the retrieval scheduling reduces the delay by 33% and 12%, respectively. In 2007, they modified the retrieval approach from a single retrieval to the clients' usage.

In 2008, Han et al. [8] proposed a news aggregation policy to minimize the number of missing posting within an aggregation. This work was modified from [6, 7].

Therefore, we propose a new mechanism to find an optimal retrieval point for RSS feeds to enhance the efficiency of the aggregator, to receive optimal retrieval points for the latest contents, and to get minimal delay in retrieving contents.

The remainder of this paper is organized as follows: Section 2 explains RSS Technology. We propose Determining Optimal Retrieval Points Mechanism (DORPM) and how it works in Section 3. The implementation and study results are shown in Section 4. Finally, we conclude our work in Section 5.

II. RSS TECHNOLOGY

RSS stands for Really Simple Syndication, Rich Site Summary, or RDF (Resource Description Framework) Site Summary [2]. RSS is an XML application used to keep track of updated contents such as weblog, news headlines, etc. It syndicates contents from several URLs to one location. Users can receive the contents from RSS service providers using an RSS reader or an aggregator. The aggregator works similar to an e-mail program except a subscriber must register to an aggregated system. The system aggregates RSS documents from several URLs, checks latest contents, and displays them automatically. Each feed is shown only a headline or a short detail, a whole content is linked to a publishing web site. An RSS structure in Fig. 1 consists of 3 main components as follows:

1. A publisher is a web site which provides RSS documents.
2. An aggregator is a module to aggregate any RSS documents from multiple web sites.
3. A subscriber is a user who subscribes to receive RSS documents.

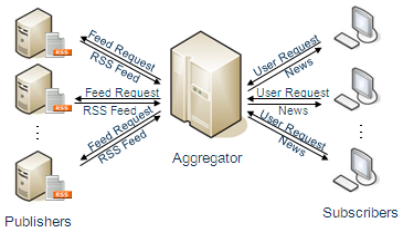


Figure 1. RSS Structure.

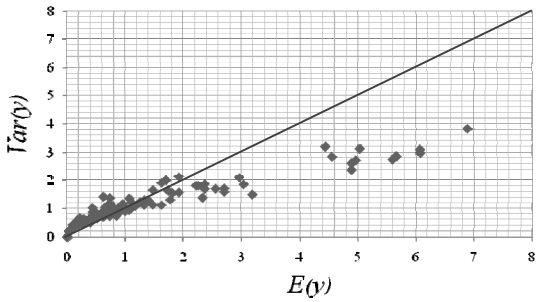


Figure 2. Distribution between expected value and variance of BBC news in April, 2010.

III. Determining Optimal Retrieval Points Mechanism (DORPM)

A. Preliminary Study

From [6, 7], the retrieval point is generated from the retrieval model which is based on Poisson distribution. They proposed a new scheduling algorithm to find a minimum delay of retrieving contents. This means that the accuracy of the retrieval point depends on the proposed retrieval model. However, this retrieval model has been applied from [5] which is a time-independent random process. In contrast, the retrieval model for postings is time-dependency. Additionally, using Poisson process to find a good retrieval point, the expected value approximately equals to its variance ($E(y_i) \approx Var(y_i)$) [3]. Therefore, we focused on two points of view. First, we observed posting contents from BBC for a month, divided into 1 hour per time interval. Then, a distribution between the expected value and the variance was plotted as shown in Fig. 2. We found that $E(y)$ and $Var(y)$ seems to be equal. However, we also noticed that a percentage of $\frac{|E(y_i)-Var(y_i)|}{E(y_i)}$ and a percentage of $\frac{|E(y_i)-Var(y_i)|}{Var(y_i)}$ was very different. For example, $E(y_i)=0.3$, $Var(y_i)=0.5$, $\frac{|E(y_i)-Var(y_i)|}{E(y_i)} = \frac{0.2}{0.3} \approx 0.66$ or 66%, and $\frac{|E(y_i)-Var(y_i)|}{Var(y_i)} = \frac{0.2}{0.5} = 0.4$ or 40%. From this point of view, using Poisson distribution to generate a retrieval model may cause an error and considerably affect the determined retrieval points. Second, we investigated a posting characteristic from BBC for one week, setting up time to show new contents every 2 hours.

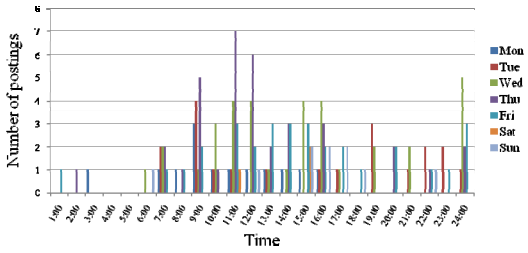


Figure 3. Posting Characteristic from BBC on April 5-12, 2010.

Fig. 3 shows that setting time interval may cause the aggregator to experience a delay in retrieving contents or no update content may occur. Therefore, a new approach to find the optimal retrieval points for RSS feeds called Determining Optimal Retrieval Points Mechanism (DORPM) is proposed. To clarify terms and conditions used in this paper, some definitions are defined in Section 3.*C*.

B. Design Concept

The DORPM is designed to optimize retrieval points for feeding RSS documents efficiently and minimizing the delay between the publications of new content at a source and its appearance at the aggregator. DORPM consists of 2 parts: Data Syndication and Data Analysis.

1) Data Syndication aggregates RSS documents with the following operations:
   a) Gathering RSS documents from multiple web sites.
   b) Extracting date and time attributes in each RSS document to store in the data repository.
   c) Transforming date and time into appropriate format by dividing time intervals per hour for simplicity.

2) Data Analysis finds the optimal retrieval points with the following operations:
   a) Defining factors that cause a delay in retrieving contents such as posting time, a number of postings, and retrieval points.
   b) Calculating a delay which occurs between posting time and retrieval points.
   c) Determining retrieval points to get minimal delay.

C. DORPM Terminology

Definition 1 Let T be a set of daily times defined by $T=\{1,2,...,24\}$, where 1 is 01:00, 2 is 02:00,…, and 24 is 00:00.

Definition 2 An interval i is the time between $i-1$ to i, where i = 1 means 00:00 – 00:59, i = 2 means 01:00 – 01:59, …, i = 24 means 23:00 – 23:59**Definition 3** Let τ is a retrieval point, $\tau \in T$ where $\tau = 1$ is retrieving at 01:00, $\tau = 2$ is retrieving at 02:00,…, $\tau = 24$ is retrieving at 00:00.

Definition 4 Given an interval $i \in T$ and Let η_i be a number of contents of the interval i, where η_1 is a number of contents from 00:00 to 00:59, η_2 is a number of contents

from 01:00 to 01:59,…, η_{24} is a number of contents from 23:00 to 23:59.

Definition 5 Let $\alpha_{i,j}$ be a posting time, where i is an interval and j^{th} is the order of posting.

Definition 6 Given an interval $i \in T$, a k^{th} retrieval point $\tau_k \in T$ and posting time $\alpha_{i,j}$. A total delay of retrieving contents in the daily interval i defined by

$$\sum_{j=1}^{\eta_i}\left(\tau_k - \alpha_{i,j}\right) \quad \text{where } \tau_k \text{ is the closest time that } i \le \tau_k$$

Lemma 1 Given an interval $i \in T$. A total delay from τ_k to τ_{k+1} is

$$\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_i}\left(\tau_{k+1} - \alpha_{i,j}\right)$$

Lemma 2 Given $\alpha_{i,j}$ and η_i be a posting time and a number of contents, respectively. Let $\tau_1, \tau_2, ..., \tau_m$ be retrieval points, a daily total cumulative delay is

$$\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1} - \alpha_{i',j}\right)$$

where $\tau_{m+1} = \tau_1 + 24$ and $i' \equiv (i-1)(\mathrm{mod}\,24)+1$

Lemma 3 Given n-day retrieving contents, where $\alpha(l)_{i,j}$ and $\eta(l)_i$ be a posting time and a number of contents at l^{th} day, respectively. We set up m retrievals at time $\tau_1, \tau_2, ..., \tau_m$ every day, a total cumulative delay for n-days is

$$\sum_{l=1}^{n}\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta(l)_{i'}}\left(\tau_{k+1} - \alpha(l)_{i',j}\right)$$

where $\tau_{m+1} = \tau_1 + 24$ and $i' \equiv (i-1)(\mathrm{mod}\,24)+1$

To easily calculate time point, a unit of the posting time is rounded into hour, for example, 09:56 is rounded into 10:00, 13:10 is rounded into 14:00, etc.

Definition 7 Given an interval $i \in T$, a k^{th} retrieval point $\tau_k \in T$ and posting time $\alpha_{i,j}$. A total delay of rounded-time retrieving contents in the interval i define by

$$\eta_i\left(\tau_k - i\right) \quad \text{where } \tau_k \text{ is the closest time that } i \le \tau_k$$

Lemma 4 Given the time point $i \in T$. A total delay of rounded-time retrieval contents from τ_k to τ_{k+1} is

$$\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_i\left(\tau_{k+1} - i\right)$$

Lemma 5 Given η_i be a number of contents. We schedule m retrievals at time $\tau_1, \tau_2, ..., \tau_m$, a daily total cumulative delay of rounded-time contents is

$$\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}\left(\tau_{k+1} - i'\right)$$

where $\tau_{m+1} = \tau_1 + 24$ and $i' \equiv (i-1)(\mathrm{mod}\,24)+1$

Lemma 6 Given n-day retrieving contents, where $\eta(l)_i$ be a number of contents at l^{th} day. We set up m retrievals at time $\tau_1, \tau_2, ..., \tau_m$ every day, an n-day total cumulative delay of rounded-time contents is

$$\sum_{l=1}^{n}\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta(l)_{i'}\left(\tau_{k+1} - i'\right)$$

where $\tau_{m+1} = \tau_1 + 24$ and $i' \equiv (i-1)(\mathrm{mod}\,24)+1$

The following theorem shows that the retrieval points of actual-time contents is not different from the retrieval points of rounded-time contents.

Theorem 1 Given a set of retrieval points τ where $\tau = \{\tau_1, \tau_2, ..., \tau_m\}$. If the rounded-time contents retrieving at τ have a daily minimum total cumulative delay, then τ makes the actual-time retrieving contents have a daily minimum total cumulative delay.

Proof Let $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}\left(\tau_{k+1} - i'\right)$ be a daily total cumulative delay of rounded-time contents, and $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1} - \alpha_{i',j}\right)$ be a daily total cumulative delay of actual-time contents.

Let $\tau = \{\tau_1, \tau_2, ..., \tau_m\}$ be the retrieval points making $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}\left(\tau_{k+1} - i'\right)$ minimum.

We will prove that $\tau = \{\tau_1, \tau_2, ..., \tau_m\}$ is the retrieval points making $\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1} - \alpha_{i',j}\right)$ minimum.

$$\begin{aligned}
\sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}\left(\tau_{k+1} - \alpha_{i',j}\right) &= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}(\tau_{k+1} - i' + i' - \alpha_{i',j}) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\left(\sum_{j=1}^{\eta_{i'}}(\tau_{k+1} - i') + \sum_{j=1}^{\eta_{i'}}(i' - \alpha_{i',j})\right) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\left(\eta_{i'}(\tau_{k+1} - i') + \sum_{j=1}^{\eta_{i'}}(i' - \alpha_{i',j})\right) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}(\tau_{k+1} - i') + \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\sum_{j=1}^{\eta_{i'}}(i' - \alpha_{i',j}) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}(\tau_{k+1} - i') + \sum_{i=1}^{24}\sum_{j=1}^{\eta_i}(i - \alpha_{i,j}) \\
&= \sum_{k=1}^{m}\sum_{i=\tau_k+1}^{\tau_{k+1}}\eta_{i'}(\tau_{k+1} - i') + C,
\end{aligned}$$

where $C = \sum_{i=1}^{24}\sum_{j=1}^{\eta_i}\left(i - \alpha_{i,j}\right)$ is the total delay occurred in every hour each day, C is constant.

Since $\tau = \{\tau_1, \tau_2, ..., \tau_m\}$ making $\sum_{k=1}^{m} \sum_{i=\tau_k+1}^{\tau_{k+1}} \eta_{i'}(\tau_{k+1} - i')$ minimum, $\tau = \{\tau_1, \tau_2, ..., \tau_m\}$ making $\sum_{k=1}^{m} \sum_{i=\tau_k+1}^{\tau_{k+1}} \eta_{i'}(\tau_{k+1} - i') + C$ minimum.

Therefore, $\tau = \{\tau_1, \tau_2, ..., \tau_m\}$ is the retrieval point making $\sum_{k=1}^{m} \sum_{i=\tau_k+1}^{\tau_{k+1}} \sum_{j=1}^{\eta_{i'}} (\tau_{k+1} - \alpha_{i',j})$ minimum.

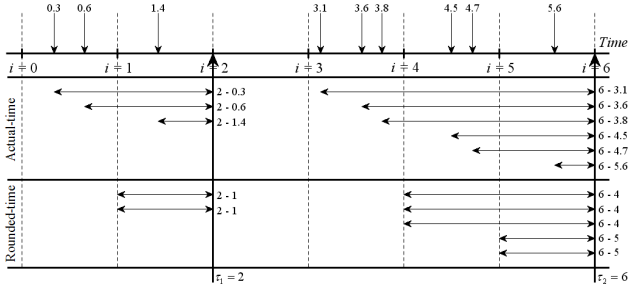


Figure 4. Delay occurred at τ_1 and τ_2 of actual-time and rounded-time.

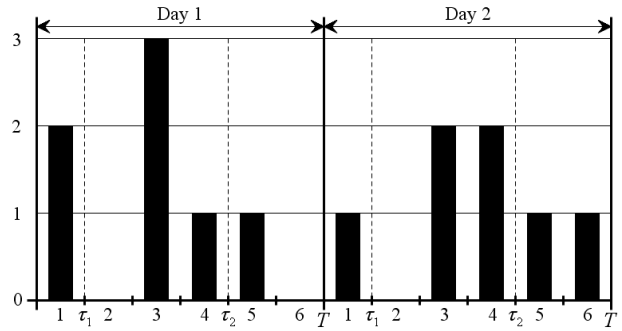


Figure 5. A number of contents posted within 2 days.

Corollary 1 Given a set of retrieval points τ where $\tau = \{\tau_1, \tau_2, ..., \tau_m\}$. If the rounded-time contents retrieving at τ have an *n*-day minimum total cumulative delay, then τ makes the actual-time retrieving contents be an *n*-day minimum total cumulative delay.

Example 1 Find an actual-time and a rounded-time delay at $\tau_1 = 2$ and $\tau_2 = 6$ from Fig. 4.

Actual-time Delay = (2-0.3)+(2+0.6)+(2-1.4)+(6-3.1)+(6-3.6)+(6-3.8)+(6-4.5)+(6-4.7)+(6-5.6) = 13.4

Rounded-time Delay = (2-1)+(2-1)+(6-4)+(6-4)+(6-4)+(6-5)+(6-5) = 10

From Example 1, a rounded-time delay equals to a number of posting times a difference between a retrieval point and an interval as defined in definition 7.

Example 2 Find a cumulative delay occurred in two days, suppose that a publisher posted contents as shown in Fig. 5.

Let retrieval points are $\tau_1 = 1$ and $\tau_2 = 4$,

$i \in T = \{1, 2, 3, 4, 5, 6\}$,

$\eta_1(1), \eta_2(1), ..., \eta_6(1) = 2, 0, 3, 1, 1, 0$ respectively, and

$\eta_1(2), \eta_2(2), ..., \eta_6(2) = 1, 0, 2, 2, 1, 1$ respectively

The delays in each interval are

$D_1(1) = \eta_1(1)(\tau_1 - 1) = 2(1 - 1) = 0$

$D_2(1) = \eta_2(1)(\tau_2 - 2) = 0(4 - 2) = 0$

$D_3(1) = \eta_3(1)(\tau_2 - 3) = 3(4 - 3) = 3$

$D_4(1) = \eta_4(1)(\tau_2 - 4) = 1(4 - 4) = 0$

$D_5(1) = \eta_5(1)(\tau_1 + 6 - 5) = 1(1 + 6 - 5) = 2$

$D_6(1) = \eta_6(1)(\tau_1 + 6 - 6) = 0(1 + 6 - 6) = 0$

$D_1(2) = \eta_1(2)(\tau_1 - 1) = 1(1 - 1) = 0$

$D_2(2) = \eta_2(2)(\tau_2 - 2) = 0(4 - 2) = 0$

$D_3(2) = \eta_3(2)(\tau_2 - 3) = 2(4 - 3) = 2$

$D_4(2) = \eta_4(2)(\tau_2 - 4) = 2(4 - 4) = 0$

$D_5(2) = \eta_5(2)(\tau_1 + 6 - 5) = 1(1 + 6 - 5) = 2$

$D_6(2) = \eta_6(2)(\tau_1 + 6 - 6) = 1(1 + 6 - 6) = 1$

Therefore, a 2-day cumulative delay is

$$\sum_{k=1}^{2} \sum_{j=1}^{2} \sum_{i=\tau_{j+1}}^{\tau_{j+1}} \eta_i(k)(\tau_{j+1} - i) = 0+0+3+0+2+0+0+0+2+0+2+1 = 10$$

Example 3 Find retrieval points where a publisher posted contents as shown in Fig. 5. Suppose that the time is set to retrieve contents twice a day. A minimum total cumulative delay of the first day: $\tau(1) = \{1,3\}$ where a cumulative delay: $D(1, \tau(1)) = 5$. A minimum total cumulative delay of the second day: $\tau(2) = \{1,4\}$ where a cumulative delay: $D(2, \tau(2)) = 10$.

Then, $D(1, \tau(1)) = 5$ $D(2, \tau(1)) = 14$

$D(1, \tau(2)) = 5$ $D(2, \tau(2)) = 10$

Therefore, $\tau(j) = \tau(2)$ minimize $\sum_{i=1}^{2} D(i, \tau(j))$ i.e. retrieval points 1 and 4 are chosen to retrieve contents every day.

IV. Implementation Of Determining Retrieval Points Mechanism

The implementation is based on investigating the posting characteristics. Data was syndicated for three months during April to June, 2010 with 42,375 news articles which came from BBC [9] 17,599 news, CNN [10] 10,638 news, and Reuters [11] 14,138 news, and classified into five groups: business, entertainment, top, US, and World news. Then, a process to transform tag <PubDate> and publication time was invoked. We set up each interval per hour for finding a number of posting contents per interval each day. Then, the cumulative delay was calculated, depending on a number of days and retrieval points as explained in Example 2. After

that, choosing only one retrieval point, $\tau(j) \in \{\tau(1), \tau(2), ..., \tau(n)\}$, given a minimum delay from $\tau(1), \tau(2), ..., \tau(n)$.

A. Study Results and Discussion

Our study divides a posting time in each interval (1 hour per interval) which each day contains similar posting characteristics. Then, finding retrieval points use different durations. Fig. 6 shows the durations with the average of cumulative delays. The graph reveals that the more durations increased, the less average of cumulative delays were reduced. Therefore, we collected data from the first month to learn, and used the data from the second and third month to calculate delays using retrieval points that occurred from data in the first month.

The experiment studied the effects of retrieval points and cumulative delays occurred with 2 policies: retrieval scheduling from [6, 7] and our approach, DORPM which retrieval points depend on the retrieval point at the minimum delay. Fig. 7-10 show the average cumulative delays in DORPM is less than the retrieval scheduling from [6, 7].

The experiment reveals that DORPM minimizes the delay of retrieving contents compared to Retrieval scheduling. Furthermore, Fig. 9 shows the unusual result of the cumulative delays because of using Poisson process to generate the model. This means that the delays depend on the retrieval points. If the retrieval point from the model has no contents to retrieve, this retrieval model will cause an error. However, Fig. 10 shows the similarity between retrieval scheduling and DORPM because of the regular posting of the number of contents.

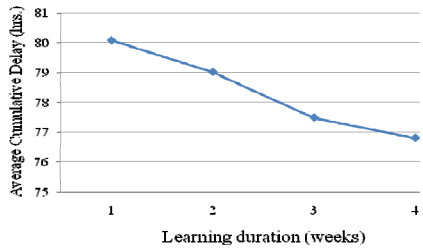


Figure 6. The average of cumulative delays depend on durations.

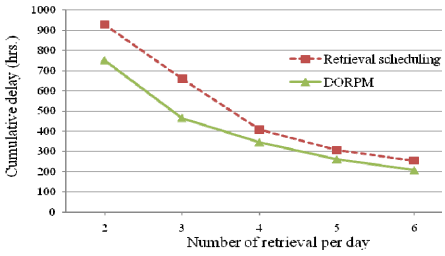


Figure 7. The effective of retrieving contents using data from 3 sources: BBC,CNN, and Reuters.

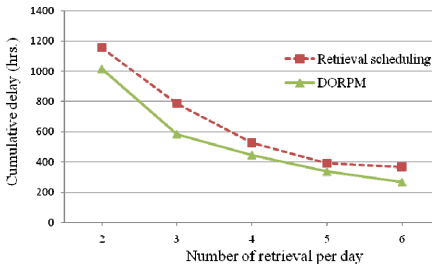


Figure 8. The effective of retrieving contents using data from BBC.

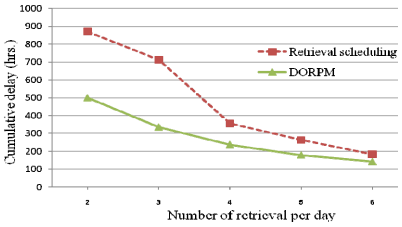


Figure 9. The effective of retrieving contents using data from CNN.

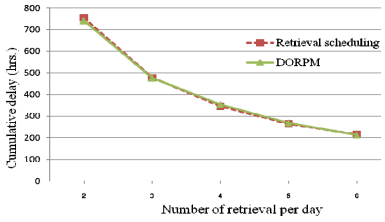


Figure 10. The effective of retrieving contents using data from Reuters.

V. CONCLUSION

This research proposes a new approach to find an optimal retrieval point for RSS feeds called Discovering Optimal Retrieval Point Mechanism (DORPM) instead of the previous research in [6,7] which generates a retrieval model before finding the optimal retrieval point. The new mechanism reduces the delay of retrieving contents. This experiment has been conducted for only one month, and the retrieval point has been reduced 21% of the delay in retrieving contents in comparison with a previous research [6, 7]. If more time were allotted for this experiment, it is very possible that the delay time could be lessened. Therefore, if increasing the time to retrieve data, the delay will be minimized, enabling the users to receive updated information, and make the aggregator work more efficiently.

REFERENCES

[1] W3C, "Extensible Markup Language (XML)", [Online]. Available: http://www.w3.org/XML/.

[2] W3Schools, "RSS Tutorial", [Online]. Available: http://www.w3schools.com/rss/ .

[3] W. Gardner, E.P. Mulvey, and E.C. Shaw, "Regression Analyses of Counts and Rates: Poisson, Overdispersed Poisson, and Negative Binomial Models". *Psychological Bulletin*., Vol. 118, 1995.

[4] S. Lipschuts and J.J. Schiller, "Theory and Problems of Finite Mathematics", McGraw-Hill, USA, 1995.

[5] J. Cho and H. Garcia-Molina, "Synchronizing a Database to Improve Freshness". *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '00),* 2000.

[6] K. Sia, J. Cho, "Efficient Monitoring Algorithm for Fast News Alert", technical report, University of California, Los Angeles, 2005.

[7] K. Sia, J. Cho and Y. Cho "Efficient Monitoring Algorithm for Fast News Alert", *IEEE Trans. Knowledge and Data Eng.*, Vol. 19, pp. 950-961, 2007.

[8] Y.G. Han, S.H. Lee, J.H. Kim and Y. Kim, "A New Aggregation Policy for RSS Services", Proceeding of the 2008 international workshop on Context enabled source and service selection, integration and adaptation, Vol. 292, 2008.

[9] BBC.co.uk [Online]. Available: http://news.bbc.co.uk/2/hi/help/3223484.stm

[10] CNN.com [Online]. Available: http://www.cnn.com/services/rss/.

[11] REUTERS.com [online]. Available: http://www.reuters.com/tools/rss.

ประวัติผู้เขียน

ชื่อ สกุล นายเชาวนันท์ ขุนดำ

รหัสประจำตัวนักศึกษา 5210220147

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วท.บ. (คณิตศาสตร์)	มหาวิทยาลัยสงขลานครินทร์	2552

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนผู้ช่วยวิจัยคณะวิทยาศาสตร์ (RA) ประจำปีการศึกษา 2552

การตีพิมพ์เผยแพร่ผลงาน

เชาวนันท์ ขุนดำ และ ลัดดา ปรีชาวีรกุล. 2553. กลไกการหาตำแหน่งเวลาที่เหมาะสมในการดึงข้อมูล. การประชุมวิชาการวิทยาการคอมพิวเตอร์และวิศวกรรมคอมพิวเตอร์แห่งชาติ ครั้งที่ 14 (NCSEC 2010). เชียงใหม่, ประเทศไทย, 17 – 19 พฤศจิกายน 2553. หน้า 270 – 275.

Khundam, Ch., and Preechaveerakul, L. 2011. Determining Optimal Retrieval Points Mechanism for RSS Documents. 3rd International Conference on Advanced Computer Control (ICACC 2011). Harbin, China, January 18 – 20, 2011. pp.644 – 649.