การวิเคราะห์รูปแบบในการกระจายการทำดัชนี สำหรับเครื่องจักรสืบค้นบนอินเตอร์เน็ต

ANALYSIS OF DISTRIBUTED INDEXING TOPOLOGIES FOR INTERNET SEARCH ENGINE

โกศล เตือนวีระเดช
KOSOL TUANVEERADEJ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาเทคโนโลยีการสนเทศ
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2547

ISBN 974-9709-85-3

การวิเคราะห์รูปแบบในการกระจายการทำดัชนี สำหรับเครื่องจักรสืบค้นบนอินเตอร์เน็ต

ANALYSIS OF DISTRIBUTED INDEXING TOPOLOGIES FOR INTERNET SEARCH ENGINE

โกศล เตือนวีระเดช

KOSOL TUANVEERADEJ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2547

ISBN 974-9709-85-3

ANALYSIS OF DISTRIBUTED INDEXING TOPOLOGIES FOR INTERNET SEARCH ENGINE

KOSOL TUANVEERADEJ

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2004

ISBN 974-9709-85-3

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การวิเคราะห์รูปแบบในการกระจายการทำดัชนีสำหรับเครื่องจักรสืบค้นบนอินเทอร์เน็ต

ANALYSIS OF DISTRIBUTED INDEXING TOPOLOGIES FOR INTERNET SEARCH ENGINE

ชื่อนักศึกษา นายโกศล เตือนวีระเดช

รหัสประจำตัว 42067013

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ผู้ควบคุมวิทยานิพนธ์ ผศ.อัครินทร์ คุณกิตติ

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
ผศ.อัครินทร์	คุณกิตติ	
รศ.ดร.รัตติกร	วรากูลศิริพันธุ์	
ผศ.ดร.จันทร์บูรณ์	สถิตวิริยวงศ์	
ผศ.ดร.อาริต	ธรรมโน	
ผศ.ดร.โชติพัชร์	ภรณวลัย	

วัน/เดือน/ปี ที่สอบ 19 พฤษภาคม 2547 เวลา 15.30 น. เป็นต้นไป

สถานที่สอบ ณ ห้อง M 23 (ชั้นลอย) อาคารเรียนรวมและปฏิบัติการคณะเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัยรับรองแล้ว

(ผศ.ดร.จารุวัตร เจริญสุข)

คณบดีบัณฑิตวิทยาลัย

วันที่ 28 **เดือน** กรกฎาคม **พ.ศ.** 2547

หัวข้อวิทยานิพนธ์	การวิเคราะห์รูปแบบในการกระจายการทำดัชนี สำหรับเครื่องจักรสืบค้นบนอินเตอร์เน็ต
นักศึกษา	นายโกศล เตือนวีระเดช
รหัสประจำตัว	42067013
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
พ.ศ.	2547
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.อัครินทร์ คุณกิตติ

บทคัดย่อ

การทำดัชนีเว็บเพจของเครื่องจักรสืบค้นบนอินเตอร์เน็ตที่มีการใช้ Web Robot ในการทำดัชนีและเก็บข้อมูลเว็บเพจมีการทำงานแบบรวมศูนย์ ซึ่งใช้เวลาในการทำดัชนีและทำให้เกิดปริมาณการสื่อสารจำนวนมาก จึงเกิดแนวคิดในการกระจายกระบวนการทำดัชนีเว็บเพจจากส่วนกลางไปออกไป เพื่อลดปริมาณการสื่อสารและลดเวลาในการทำดัชนี ในการศึกษาครั้งนี้เป็นการศึกษาการทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ตแบบกระจายการทำดัชนี โดยทำการวิเคราะห์หาความสัมพันธ์ระหว่างตัวแปรได้แก่ ปริมาณการสื่อสารข้อมูล เวลาในการทำดัชนีข้อมูล และวิเคราะห์รูปแบบการกระจายการทำดัชนี โดยมีวัตถุประสงค์เพื่อลดปริมาณการสื่อสารข้อมูลระหว่างการทำงาน และหารูปแบบการกระจายกระบวนการทำดัชนีที่ใช้เวลา และปริมาณการสื่อสารในระหว่างการทำดัชนีน้อยที่สุด จากการทดลองการพบว่าเวลาที่ใช้ในการทำดัชนีแบบกระจายการทำดัชนีลดลง โดยรูปแบบการกระจายที่ดีที่สุดได้แก่รูปแบบการกระจายที่มีการกระจายกระบวนการ Preindexing ทั้งหมดออกไปอยู่บนโฮสต์เดียวกัน และอยู่ใกล้เว็บเซิร์ฟเวอร์ ซึ่งนอกจากใช้เวลาในการทำดัชนีลดลงแล้ว ปริมาณการสื่อสารข้อมูลในระหว่างกระบวนการทำดัชนีก็ลดลงได้อย่างมาก

Thesis Title	Analysis of Distributed Indexing Topologies for Internet Search Engine
Student	Mr. Kosol Turnveeradej
Student ID.	42067013
Degree	Master of Science
Program	Information Technology
Year	2004
Thesis Advisor	Assist.Prof. Akharin Khunkitti

ABSTRACT

Indexing Web pages by Web Robot in Internet Search Engine have centralize characteristic that produce a lot of communication traffic load and use a lot of indexing process time. To reduce communication traffic and indexing time, process in central part distributed. This thesis study how Distributed Internet Search Engine work and analyze parameter involve such as Communication Traffic load, Process time, and analyze distribution topologies of Distributed Indexing Internet Search Engine. This aim to reduce communication traffic loads and discovers the optimum distribution topology. From the experimental indexing time for Distributed Indexing Internet Search Engine has reduced. The most optimum topology is the topology that distributes Preindexing process to web server or nearest to web server, in addition to that communication traffic in this topology was reduced.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ผศ. อัครินทร์ คุณกิตติ ที่ได้ให้ความช่วยเหลือ ให้คำชี้แนะ ให้คำปรึกษาและแนวทางในการแก้ปัญหา จัดหาอุปกรณ์เพื่อนำมาใช้ในการทดลอง ตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบพระคุณ คณะกรรมการสอบวิทยานิพนธ์และหัวข้อวิทยานิพนธ์ทุกท่านที่ให้คำแนะนำ และชี้แนะแนวทางจนทำให้วิทยานิพนธ์เล่มนี้เสร็จสมบูรณ์ลงได้

ขอขอบคุณเพื่อนๆ ที่เป็นกำลังใจและให้ความช่วยเหลือในหลายด้าน ตลอดจนเจ้าหน้าที่ของคณะเทคโนโลยีสารสนเทศ ที่ช่วยอำนวยความสะดวกเป็นอย่างดี และผู้ที่มิได้กล่าวถึงอีกหลายท่านที่มีส่วนช่วยให้วิทยานิพนธ์เล่มนี้เสร็จสมบูรณ์

สำหรับคุณความดีอันใดที่จะเกิดขึ้นจากวิทยานิพนธ์เล่มนี้ ข้าพเจ้าขอมอบให้ แด่บิดามารดาผู้เป็นที่รักและเคารพยิ่ง ตลอดจนครูบาอาจารย์ผู้ประสิทธิประสาทวิชาความรู้และประสบการณ์ที่ดีให้แก่ข้าพเจ้า

โกศล เตือนวีระเดช

สารบัญ

หน้า

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญตาราง (ต่อ)

สารบัญตาราง (ต่อ)

ตารางที่ หน้า

สารบัญรูป

รูปที่ หน้า

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การเติบโตอย่างรวดเร็วของอินเตอร์เน็ต ทำให้จำนวนข้อมูลที่อยู่บนอินเตอร์เน็ตมีจำนวนเพิ่มขึ้นอย่างรวดเร็ว และแหล่งของข้อมูลสารสนเทศจำนวนมหาศาล แต่ข้อมูลสารสนเทศที่มีอยู่เหล่านี้มีการกระจายและไม่มีรูปแบบที่แน่นอน ทำให้การสืบค้น จัดหมวดหมู่ และการนำข้อมูลที่มีอยู่มาใช้ประโยชน์เป็นไปได้ยาก จากปัญหาที่เกิดขึ้นจึงเกิดการพัฒนาระบบที่ช่วยในการสืบค้นข้อมูล ทำดัชนีและจัดหมวดหมู่ของข้อมูลที่มีอยู่บนอินเตอร์เน็ตเรียกว่าเครื่องจักรสืบค้นบนอินเตอร์เน็ต (Internet Search Engine)

การทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ต สามารถแบ่งการทำงานออกเป็นสามส่วนหลักได้แก่ ส่วนติดต่อผู้ใช้ ส่วนฐานข้อมูล ส่วนงานเบื้องหลัง หรือส่วนของ Web Robot ในการทำงานส่วนติดต่อผู้ใช้จะรับข้อมูลในการค้น จากผู้ใช้นำไปทำการค้นหาในฐานข้อมูลและส่งผลลัพธ์กลับไปให้ผู้ใช้ ส่วนฐานข้อมูลเป็นส่วนที่เก็บข้อมูลสำหรับค้นคืนข้อมูลหรือเรียกว่า ฐานข้อมูลดัชนี (Index Database) ซึ่งเป็นส่วนที่ได้จากการทำงานของส่วนงานเบื้องหลัง สำหรับส่วนงานเบื้องหลัง หรือ Web Robot มีหน้าที่ในการเก็บรวบรวมข้อมูลจากเว็บเพจโดยการร้องขอเว็บเพจจากเว็บเซิร์ฟเวอร์ นำมาทำดัชนีและเก็บข้อดัชนีที่ได้ลงฐานข้อมูลดัชนี

เครื่องจักรสืบค้นที่มีลักษณะการทำงานซึ่งรวมการทำดัชนีไว้ที่ส่วนกลาง จะต้องโหลดเว็บเพจ และทำดัชนีข้อมูลโดยเป็นหน้าที่ในส่วนกลาง ทำให้เกิดปริมาณข้อมูลที่ส่งผ่านเครือข่ายเป็นจำนวนมหาศาลโดยการคำนวณเบื้องต้นปริมาณข้อมูลที่ส่งผ่านระหว่างกันประมาณได้เท่ากับขนาดของเว็บเพจที่มี และเนื่องจากการเพิ่มขึ้นของขนาดเครือข่ายและปริมาณข้อมูลของ World Wide Web (WWW) โดยมีปริมาณเว็บเพจที่เพิ่มขึ้นเป็นปริมาณ 1,500,000 หน้า/วัน [11] ดังนั้นหากต้องทำดัชนีเอกสารเหล่านี้ จะต้องใช้เวลาในการทำงานเท่าไหร่ และเกิดปริมาณข้อมูลที่ส่งผ่านเครือข่ายมากเท่าใด

แนวทางที่มีการใช้มากในการพัฒนาทำดัชนีเว็บเพจของเครื่องจักรสืบค้น ได้แก่การกระจายกระบวนการในการทำดัชนีเว็บเพจ โดยกระบวนการที่กระจายออกไปยังโฮสต์หลายๆ โฮสต์ มีการทำงานขนานกัน และมีการทำงานอิสระไม่ขึ้นต่อกัน แล้วส่งผลลัพธ์จากการทำงานกลับไปส่วนกลางเพื่อปรับปรุงฐานข้อมูลดัชนี ทำให้ได้การทำดัชนีโดยรวมใช้เวลาในการประมวลผลลดลง เนื่องจากการประมวลผลขนานกันของ กระบวนการทำดัชนีที่กระจายออกไป นอกจากนี้

ปริมาณการสื่อสารข้อมูลในระหว่างการทำดัชนีก็จะลดลงด้วย ซึ่งจากการวิจัยพบว่าอัตราส่วนของขนาดข้อมูลดัชนี มีขนาดลดลงโดยมีขนาดประมาณ 15% ของ Input [10][12] จะเห็นได้ว่าหรือลดลงประมาณ 85% ทำให้สามารถใช้ทรัพยากรเครือข่ายได้ประสิทธิภาพสูงขึ้น

ในงานวิจัยนี้จะแสดงและวิเคราะห์ ความสัมพันธ์ของ ภาระงาน ปริมาณการสื่อสารข้อมูลที่เกิดขึ้น และตัวแปรต่างในระหว่างการทำงานเพื่อหารูปแบบที่เหมาะสมในการกระจายการทำดัชนี เพื่อพัฒนาเครื่องจักรสืบค้นบนอินเตอร์เน็ต ให้มีประสิทธิภาพมากขึ้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. ศึกษากระบวนการทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ต กระบวนการในการทำดัชนี และการค้นคืนสารสนเทศบนอินเตอร์เน็ต
2. ศึกษาความสัมพันธ์ระหว่าง ปริมาณการสื่อสารกับปริมาณภาระงานในการทำงานของ เครื่องจักรสืบค้นบนอินเตอร์เน็ตแบบกระจายการทำดัชนี
3. เพื่อพัฒนาการทำงานของเครื่องจักรสืบค้นแบบกระจายการทำดัชนีให้มีประสิทธิภาพ

1.3 ทฤษฎีหรือแนวความคิดที่ใช้ในการศึกษา

เนื่องจากกระบวนการในการทำดัชนีเว็บเพจของ เครื่องจักรสืบค้นบนอินเตอร์เน็ตเป็นกระบวนการที่ใช้เวลาประสิทธิภาพของเครื่องในการประมวลผล รวมทั้งทรัพยากรระบบเครือข่ายในการรับส่งข้อมูลเป็นจำนวนมาก เพื่อลดเวลาและจำนวนข้อมูลที่ต้องส่งผ่านเครือข่าย และใช้ทรัพยากรเครือข่ายให้เกิดประโยชน์มากขึ้น

แนวคิดของงานวิจัยนี้คือ ทำการกระจายการทำดัชนีออกไป โดยกระบวนการที่กระจายออกไปมีการทำงานขนานกันไปและไม่ขึ้นต่อกัน ทำให้ได้ประสิทธิภาพการทำงานโดยรวมที่ดีกว่าโดยใช้เวลาในการทำดัชนีน้อยลง และมีปริมาณสื่อสารข้อมูลที่น้อยกว่าวิธีการทำดัชนีข้อมูลแบบเดิม

1.4 ขอบเขตของงานวิจัย

ในการศึกษาหลักการทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ต จะทำการศึกษาโดยเน้นความสำคัญที่ขั้นตอนในการทำดัชนีเว็บเพจซึ่งเป็นขั้นตอนที่สำคัญในการสร้างฐานข้อมูลดัชนี ซึ่งเป็นหัวใจในการทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ต โดยทำการศึกษามุ่งเน้นเพื่อหารูปแบบการกระจายกระบวนการทำดัชนีที่เหมาะสม ระหว่างเวลาในการทำดัชนี ปริมาณข้อมูลที่ส่ง

ผ่านระบบเครือข่ายในระหว่างการทำดัชนี และการใช้พื้นที่ในการเก็บข้อมูลบนดิสก์ โดยพารามิเตอร์ที่นำมาใช้ในการศึกษาครั้งนี้ได้แก่ เวลาที่ใช้ในการประมวลผล ปริมาณข้อมูลที่ส่งผ่านเครือข่าย หรือพื้นที่ที่ใช้ในการจัดเก็บแคชบนดิสก์

ข้อจำกัดของการศึกษาได้แก่ จำกัดความเร็วหน่วยประมวลผลกลาง และจำนวนหน่วยความจำที่ใช้ในแต่ละเครื่องเท่ากันทุกเครื่อง โดยใช้คอมพิวเตอร์จำนวน 4 เครื่อง ความเร็วหน่วยประมวลผล 350 MHz หน่วยความจำ 128 MB ทำการจำลองเซิร์ฟเวอร์ และการทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ตบนเครือข่ายภายใน และโปรแกรมที่ใช้สำหรับทำการทดลองได้แก่ BDDbot [16]

1.5 ขั้นตอนการศึกษา

ขั้นตอนในการดำเนินการประกอบด้วยขั้นตอนหลักดังนี้

1. กำหนดหัวข้อ เป้าหมาย จุดประสงค์ และขอบเขตของวิทยานิพนธ์
2. ศึกษาทฤษฎีและหลักการพื้นฐานของเครื่องจักรสืบค้นบนอินเตอร์เน็ต
3. จัดหาอุปกรณ์เครื่องมือในการศึกษา
4. ออกแบบการทดลอง
5. ทำการทดลอง และวัดผล
6. วิเคราะห์และประเมินผลการทดลอง
7. จัดทำเอกสารประกอบวิทยานิพนธ์

1.6 นิยามศัพท์และข้อตกลงเบื้องต้น

ในการศึกษาและวิจัยครั้งนี้เพื่อความเข้าใจตรงกันในการทำความเข้าใจในเนื้อหาของวิทยานิพนธ์ฉบับนี้ และลดความสับสนจึงขอนิยามศัพท์ที่ใช้ในการอ้างอิงดังนี้

การอ้างอิงถึงเครื่องจักรสืบค้นบนอินเตอร์เน็ต หรือเครื่องจักรสืบค้น ให้เป็นความหมายเดียวกันคือเป็นเครื่องจักรสืบค้นที่ทำการเก็บข้อมูลเว็บเพจเพื่อทำดัชนี และให้บริการค้นหาข้อมูลที่อยู่บนอินเตอร์เน็ต โดยจะใช้ตัวย่อ OSE (Original Search Engine)

เครื่องจักรสืบค้น แบบกระจายการทำดัชนี คือเครื่องจักรสืบค้นบนอินเตอร์เน็ตที่มีการกระจายหน้าที่การทำดัชนีเว็บเพจออกไป โดยจะใช้ตัวย่อ DISE (Distributed Index Search Engine)

กระบวนการในการทำดัชนีโดยรวมเรียกว่า Indexing process (Pi) และกระบวนการในการทำดัชนีกำหนดเป็นสองขั้นตอนใหญ่ได้แก่ Preindexing (Pp) และ Indexmap ซึ่งในกระบวนการ Preindexing จะแตกการทำงานออกเป็นส่วนย่อย ได้แก่

1. กระบวนการในการกำจัด HTML Tag เรียกว่า RMHTML (P1)
2. กระบวนการในการ กำจัดคำที่เป็น Stop word และ function word เรียกว่า RMSTOP (P2)
3. กระบวนการในการทำให้คำที่มีรูปแบบต่างกันกลับไปอยู่ในรูปแบบรากศัพท์ เรียกว่า STEMWORD (P3)
4. กระบวนการ สร้างไฟล์ดัชนีข้อมูลเรียกว่า INDEXMAP (P4)

1.7 ประโยชน์ที่คาดว่าจะได้รับ

1. ทราบถึงกลไกการทำงาน และข้อด้อยในการทำงานของเครื่องจักรสืบค้น
2. สามารถวิเคราะห์หารูปแบบที่เหมาะสมในการกระจายการทำดัชนี
3. สามารถนำผลการวิจัยไปใช้ในการพัฒนา เครื่องจักรสืบค้นแบบกระจายการทำดัชนี

1.8 รายละเอียดเนื้อหาในแต่ละบท

ในบทที่ 2 จะกล่าวถึงหลักการทำงานเบื้องต้นกี่ยวกับการทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ต พารามิเตอร์ที่สำคัญที่ใช้ในการวิเคราะห์การทำงาน

ในบทที่ 3 จะกล่าวถึงทฤษฎีที่ใช้เป็นแนวคิดในการวิจัย การทดลองเบื้องต้นและผลการทดลองเพื่อ วิเคราะห์หาความสัมพันธ์ของกระบวนการในขั้นตอนของกระบวนการการทำดัชนี และเสนอแนวทางที่น่าจะเหมาะสมสำหรับกระบวนการในการกระจายการทำดัชนี

ในบทที่ 4 นำเสนอผลการทดลองเพื่อหารูปแบบการกระจายการทำดัชนี ที่เหมาะสม และทดลองตามแนวคิดที่คาดว่าน่าจะดีที่สุดในขั้นตอนการทำดัชนี

ในบทที่ 5 จะเป็นบทสรุปผลงานวิจัยในการกระจายการทำดัชนีทั้งหมด และข้อเสนอแนะในงานวัจัยที่น่าจะทำต่อไปในอนาคต

บทที่ 2
หลักการทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ต

ในบทนี้กล่าวถึงทฤษฎีพื้นฐานเบื้องหลังการทำงานของ เครื่องจักรสืบค้น และการทำงานของเว็บเซิร์ฟเวอร์ จุดประสงค์เพื่อช่วยให้เข้าใจหลักการทำงานเพื่อเป็นพื้นฐานในการเข้าใจ การทำงานของเครื่องจักรสืบค้นและงานวิจัยในบทต่อไป

2.1 ประวัติความเป็นมาของเครื่องจักรสืบค้นบนอินเตอร์เน็ต

ในช่วงแรกของอินเตอร์เน็ต และยังไม่มี World Wide Web วิธีการในการเก็บและค้นคืนไฟล์จะทำผ่าน File Transfer Protocol (FTP) เซิร์ฟเวอร์ ซึ่งเป็นวิธีการที่ง่ายและสะดวกในการแบ่งปันโปรแกรม และไฟล์ผ่านอินเตอร์เน็ต หลักการทำงานของ FTP คือเมื่อผู้ดูแลระบบต้องการแบ่งปันไฟล์ในเครื่องกับผู้ใช้ อื่นๆ พวกเขาจะติดตั้งโปรแกรมที่เรียกว่า FTP เซิร์ฟเวอร์ เมื่อมีใครที่ต้องการค้นหาหรือใช้ไฟล์นั้นจะต้องติดต่อเข้าสู่เครื่องนั้นโดยผ่านโปรแกรมอีกอันหนึ่งเรียกว่า FTP ไคลเอนต์

ในระยะแรกการแบ่งปันไฟล์กับผู้ใช้คนอื่นๆ บนอินเตอร์เน็ตเป็นอย่างนั้นคือเมื่อต้องการแบ่งปันไฟล์กับผู้ใช้คนอื่นๆ จะต้องติดตั้ง FTP เซิร์ฟเวอร์ จนกระทั่งมี Anonymous FTP เพื่อเป็นที่รวบรวมและแบ่งปันการใช้ไฟล์โดยอนุญาตให้ผู้ใช้สามารถส่ง และรับไฟล์ได้ง่ายขึ้น

ถึงแม้จะมี Anonymous FTP เซิร์ฟเวอร์ขึ้นมาแล้วแต่ไฟล์ที่สำคัญๆ ยังคงกระจายอยู่ใน FTP เล็กๆ และสามารถค้นหาและรู้ได้โดยการบอกต่อๆ กันไปหรือการบอกผ่านทาง e-mail

Archie จึงเป็นเครื่องจักรสืบค้นตัวแรกที่ทำการเก็บข้อมูลโดยรวบรวมรายชื่อและโปรแกรมที่มีอยู่บน FTP เซิร์ฟเวอร์ และสามารถให้ผู้ใช้ได้ค้นหาไฟล์ที่ต้องการ หรือกล่าวได้ว่า Archie เป็นเครื่องจักรสืบค้นที่ทำการรวบรวมและทำดัชนีรายชื่อไฟล์ ที่มีอยู่บน FTP เซิร์ฟเวอร์

การทำงานของ Gopher คล้ายๆ กับการทำงานของ FTP เซิร์ฟเวอร์แต่แทนที่จะเป็น การแบ่งปันการใช้ไฟล์ Gopher จะเป็นการแบ่งปันการใช้เอกสารร่วมกันในรูปของข้อความไม่มี รูปภาพ และทำให้เกิด Veronica ซึ่งเป็นเครื่องจักรสืบค้นสำหรับ Gopher เซิร์ฟเวอร์โดยการทำดัชนีชื่อไฟล์ที่มีอยู่ใน Gopher เซิร์ฟเวอร์

ภายหลังเมื่อ World Wide Web เริ่มรู้จักกันมากขึ้นจึงเริ่มมีเครื่องจักรสืบค้นบนเว็บขึ้นเรียกว่า World Wide Web Wanderer ซึ่งเป็น web robot ตัวแรกที่ถูกออกแบบมาเพื่อติดตามการ

เติบโตของเว็บ ในช่วงเริ่มต้น WWW Wanderer รวบรวมเฉพาะรายชื่อของเว็บเซิร์ฟเวอร์ ต่อมาจึงมีการรวบรวมรายชื่อ URL และฐานข้อมูลที่ได้จากการเก็บรวมรวม URL เรียกว่า Wandex

การทำงานของ Wanderer ในตอนนั้นทำให้เกิดปัญหาเกี่ยวกับประสิทธิภาพของเครือข่ายที่ต่ำลง เนื่องจาก Wanderer อาจจะทำการดึงข้อมูลจากเพจเดียวกัน เป็นร้อยครั้งในหนึ่งวัน

ต่อมาจึงมีการพัฒนา ALIWEB ซึ่งมีการทำงานเหมือนกับ Archie แต่ไม่มีการใช้ web robot ในการเก็บรวบรวมข้อมูล แต่จะให้ผู้ดูแลเว็บทำการส่งข้อมูลเกี่ยวกับเว็บที่ต้องการให้แสดงรายการข้อดีของวิธีการนี้คือไม่มีการใช้ robot ในการเก็บข้อมูลและไม่เกิดการใช้ทรัพยากร bandwidth โดยสิ้นเปลือง

ข้อเสียของ ALIWEB ก็คือไฟล์ดัชนีจะต้องสร้างโดยผู้แลเว็บและซึ่งส่วนใหญ่ไม่รู้ว่าจะสร้างได้อย่างไรทำให้ไม่มีใครส่งไฟล์ดัชนีไปยัง ALIWEB และทำให้ฐานข้อมูลของ ALIWEB มีขนาดเล็กกว่า ฐานข้อมูลที่สร้างโดย robot

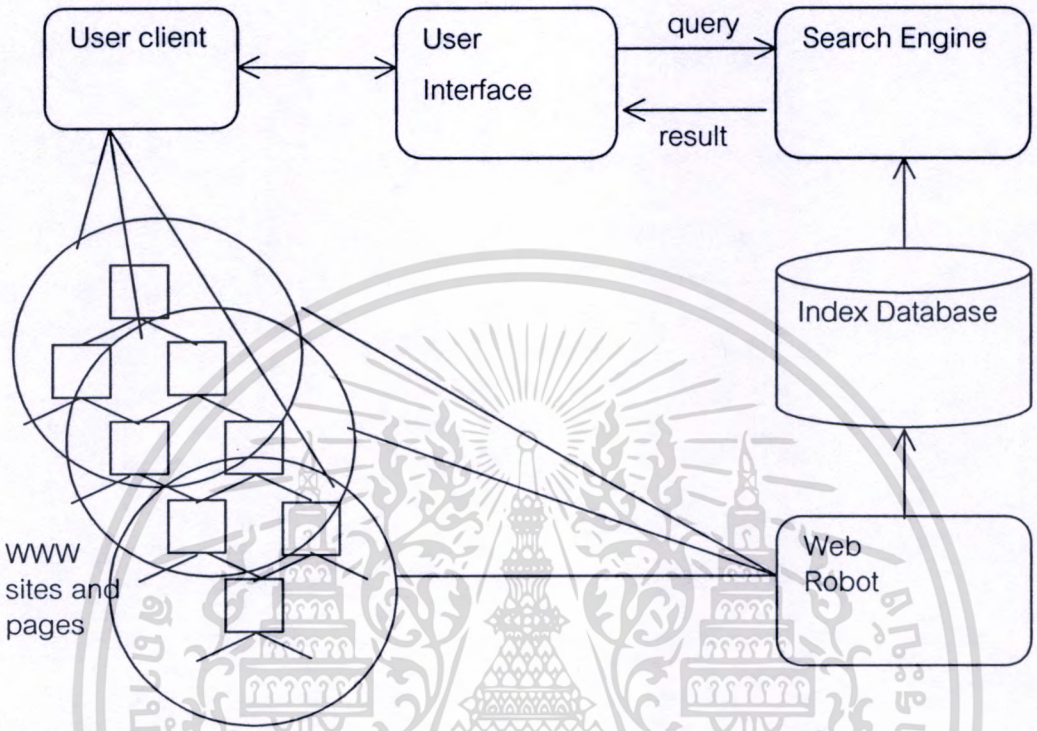
ต่อมาจึงเกิดเครื่องจักรสืบค้นที่ใช้ robot ในการทำงานมากขึ้นและสามตัวแรกได้แก่ Jumpstation, WWW Worm, Repository-Base Software Engineering (RBSE) ซึ่งในช่วงแรกๆ เป็นการเก็บเฉพาะข้อมูลที่เป็น Title และ URLs ของแต่ละเว็บเพจเท่านั้นต่อมาจึงเกิดเครื่องจักรสืบค้นที่ทำดัชนีทั้งเพจขึ้นในชื่อ WebCrawler

นอกจากเครื่องจักรสืบค้นทั้งหลายที่กล่าวมาแล้วเนื่องจากในช่วงแรกการใช้ robot ยังไม่มีการใส่ความฉลาดเข้าไปในการทำดัชนีจึงเกิดแนวคิดที่เรียกว่า Search Directory ซึ่งมีการจำแนกประเภทของเว็บเพจและมีการจัดหมวดหมู่โดยใช้คนช่วย ตัวอย่างเช่น Galaxy และ Yahoo เป็นต้น

2.2 การทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ต

การทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ต สามารถแบ่งการทำงานออกเป็นสามส่วนใหญ่ได้แก่ ส่วนติดต่อผู้ใช้ (Front End) เป็นส่วนที่ติดต่อกับผู้ใช้ในการรับคำค้น (Query) จากผู้ใช้นำไปค้นหาในฐานข้อมูลดัชนีและแสดงผลลัพธ์จากการค้นหาให้กับผู้ใช้ ส่วนต่อมาคือส่วนของฐานข้อมูล (Database) เป็นส่วนที่เก็บข้อมูลสำหรับค้นคืนข้อมูลหรือเรียกว่า ฐานข้อมูลดัชนี (Index Database) ซึ่งข้อมูลที่ได้เกิดจากการทำงานของ Web Robot ซึ่งจะทำการเก็บรวบรวมข้อมูลและทำดัชนีข้อมูล ส่วนสุดท้ายได้แก่ งานเบื้องหลัง (Back End) ซึ่งประกอบด้วย Web Robot ทำหน้าที่ในการเก็บรวบรวมข้อมูลจากเว็บเพจโดยการร้องขอจากเว็บเซิร์ฟเวอร์ (Web Server) นำมาทำดัชนี และปรับปรุงข้อมูลลงในฐานข้อมูลดัชนี การทำงานของ Web Robot เราสามารถแบ่งการทำงานออกเป็นสองส่วน ได้แก่ส่วนในการค้นหารวบรวมข้อมูล (Gatherer) ซึ่งทำ

หน้าที่ในการค้นหาเว็บเพจเพื่อนำมาทำดัชนี และส่วนที่มีหน้าที่ทำดัชนีข้อมูล (Indexer) ซึ่งทำหน้าที่ในการประมวลผลข้อมูลที่ได้ให้กลายเป็นดัชนีข้อมูลและนำดัชนีที่ได้ปรับปรุงลงสู่ฐานข้อมูลดัชนี เพื่อใช้ในการค้นคืนของเครื่องจักรสืบค้นต่อไป



รูปที่ 2.1 สถาปัตยกรรมของเครื่องจักรสืบค้นบนอินเตอร์เน็ต [2]

2.2.1 Web Robot

Web robot เป็นส่วนสำคัญที่ทำหน้าที่ในการทำดัชนีข้อมูลเว็บเพจ ซึ่งในกระบวนการทำดัชนีสามารถแบ่งออกเป็นส่วนย่อยๆ ได้ดังนี้

กระบวนการในการ Crawling

กระบวนการในการกำจัด HTML Tag (RMHTML)

กระบวนการในการกำจัด Stop word (RMSTOP)

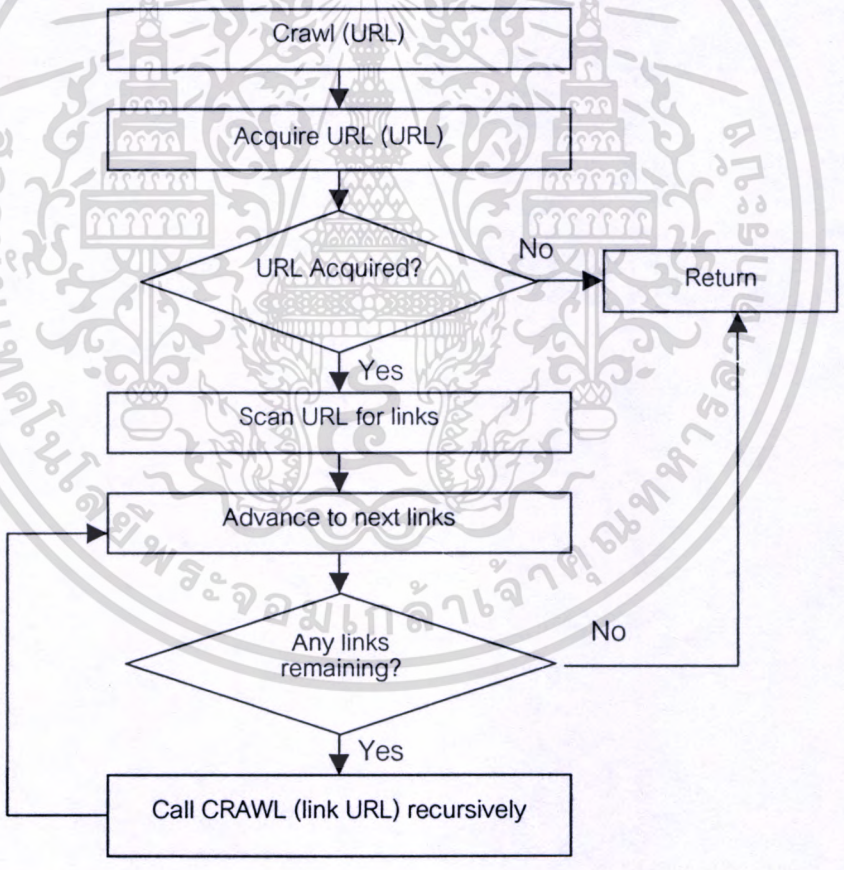
กระบวนการในการแปลงคำศัพท์ให้อยู่ในรูปรากศัพท์ (STEMWORD)

และกระบวนการในการทำดัชนี (INDEXMAP)

2.2.2 Crawling

กระบวนการในการ Crawling คือกระบวนการในการค้นหาและรวบรวมเว็บเพจก่อนที่จะนำเว็บเพจนั้นเข้ากระบวนการในการทำดัชนี โดยกระบวนการในการ crawling จะมีลำดับการทำงานดังรูปที่ 2.2

เริ่มต้นที่เว็บเพจหนึ่งเป็นจุดเริ่มต้นโดยให้ผู้ใช้เป็นผู้กำหนด กระบวนการในการ crawling เริ่มต้นโดยการตรวจสอบว่าเว็บเพจหรือ url ที่ได้มานั้นมีอยู่จริงหรือและสามารถเข้าถึงได้หรือไม่ เมื่อพบว่าเว็บเพจนั้นมีอยู่จริงและสามารถเข้าถึงได้ จะทำการโหลดเว็บเพจนั้นมา เพื่อทำการประมวลผลทำการตรวจสอบว่ามี URL ลิงค์ใดภายในเว็บเพจนั้นหรือไม่ และทำการรวบรวม URL ลิงค์ซึ่งมีอยู่ในเว็บเพจนั้นเข้าสู่คิวเพื่อตรวจสอบและทำการโหลดต่อไป ทำอย่างนี้ไปจนกระทั่งในคิวไม่มี URL เหลือให้ตรวจสอบ



รูปที่ 2.2 แผนภาพลำดับการทำงาน Crawling ส่วนค้นหาเว็บเพจของ Web Robot

2.2.3 กระบวนการในการกำจัด HTML Tag (RMHTML)

เมื่อเว็บเพจได้ถูกโหลด และรวบรวมไว้จากกระบวนการ Crawling ขั้นตอนต่อไป คือการนำเว็บเพจที่ได้เข้าสู่กระบวนการ RMHTML ซึ่งเป็นกระบวนการกำจัดส่วนที่เป็น HTML tag ออก เพื่อให้เหลือเพียงส่วนที่เป็นข้อมูล และเนื้อหาของเว็บเพจ

2.2.4 กระบวนการในการกำจัด Stop word (RMSTOP)

ในกระบวนการ RMSTOP เป็นกระบวนการในการกำจัด Stop word หรือ Function word ซึ่ง Stop word หรือ Function Word คือคำที่ปรากฏบ่อย แต่ไม่ได้แสดงความหมายของเนื้อหา หรือไม่ได้เป็นคำสำคัญในเนื้อหา ซึ่งบางครั้งเราเรียกคำเหล่านี้ว่า Negative word โดยเราสามารถรวบรวมและสร้างเป็น Negative dictionary หรือ Stop List [7] ในกระบวนการที่จะกำจัด RMSTOP เนื้อหาของเว็บเพจจะถูกโหลดเข้ามาเพื่อทำการตรวจสอบและค้นหาคำที่เป็น Stop word โดยการตรวจสอบกับ stop List ซึ่งในตารางที่ 2.1 เป็นตัวอย่างบางส่วนของ stop word ซึ่งจะต้องถูกกำจัดไปในกระบวนการนี้

การใช้ Stop list ช่วยให้ประสิทธิภาพในการทำดัชนีเพิ่มขึ้น ทำให้มีความถูกต้องมากขึ้นในการค้นคืน และช่วยให้กระบวนการในการทำดัชนีทำได้รวดเร็วขึ้นโดยพิจารณาจากความจริงที่ stop list ประกอบด้วยคำที่ปรากฏบ่อยครั้งและไม่มีความหมายเฉพาะหรือมีความสำคัญในการบ่งชี้เอกสาร โดยปรกติเอกสารทั่วไปประกอบด้วยคำทั่วไปและคำที่เป็น stop word โดยมีคำปรกติประมาณร้อยละ 50 หากทำการกำจัดคำที่เป็น stop word ถูกกำจัดออกไปจากเอกสารจะทำให้จำนวนอักษรที่ผ่านกระบวนการทำดัชนีลดลงอย่างมาก ดังนั้นเวลาที่ใช้ในทำดัชนีเอกสารในขั้นตอนต่อไปจึงลดลงด้วย และเหตุผลอีกข้อในการกำจัด stop words ได้แก่ความถูกต้องของดัชนี เนื่องจากคำที่จะเป็นดัชนีของเอกสารควรเป็นคำที่มีความหมายและบ่งบอกว่าเอกสารนั้นเกี่ยวกับเรื่องใด หากในการทำดัชนีไม่ได้กำจัดคำที่เป็น stop word แล้วจะทำให้ดัชนีข้อมูลที่ได้ไม่สามารถระบุถึงเอกสารที่ตรงและถูกต้องได้ นอกจากนี้การกำจัด stop word ทำให้ปริมาณข้อมูลมีขนาดเล็กลงทำให้การทำดัชนีมีประสิทธิภาพมากขึ้น

ตารางที่ 2.1 บางส่วนของคำที่เป็น Stop Word

A	AMONGST	BECAME
ABOUT	AN	BECAUSE
AFTER	AND	BECOME
AFTERWARDS	ANOTHER	BECOMES
AGAIN	ANY	BECOMING

AGAINTST	ANYHOW	BEEN
ALL	ANYONE	BEFORE
ALMOST	ANYTHING	BEFOREHAND
ALONE	ANYWHERE	BEHIND
ALONG	ARE	BEING
ALREADY	AROUND	BELOW
ALSO	AS	BESIDE

2.2.5 **กระบวนการในการแปลงคำให้อยู่ในรูปรากศัพท์** (STEMWORD)

หลังจากการกำจัดคำที่เป็น Stop word และ function word ไปแล้วคำศัพท์ส่วนที่เหลือจะเป็นคำศัพท์ที่จะเป็นตัวแทนของข้อมูลในเว็บเพจนั้นได้ แต่บางครั้งรูปแบบของคำศัพท์ที่มีรากศัพท์เดียวกันอาจมีหลายรูปแบบ เพื่อลดความหลากหลายของรูปแบบคำศัพท์ และเพิ่มประสิทธิภาพในการค้นคืนของข้อมูลดัชนี จึงจำเป็นต้องมีการแปลงคำศัพท์ที่มีรากศัพท์เดียวกัน แต่มีหลายรูปแบบแตกต่างกันให้อยู่ในรูปแบบเดียวกันซึ่งเรียกว่ากระบวนการ STEMWORD

ตารางที่ 2.2 บางส่วนของคำที่เป็น Suffix List

ABILITIES	ACEOUSNESS	AGES
ABILITY	ACEOUSNESSES	AGING
ABLE	ACIDOUS	AGINGFUL
ABLED	ACIDOUSLY	AGINGLY
ABLEDLY	ACIES	AIC
ABLENESS	ACIOUSNESS	AICAL
ABLER	ACIOUSNESSES	AICALLY
ABLES	ACITIES	AICALS
ABLING	ACITY	AICISM
ABLINGFUL	ACY	AICISMS
ABLINGLY	AE	AICS
ABLY	AGE	AL
ACEOUS	AGED	ALISATION

โดยกำจัดคำอุปสรรค (Prefix) และคำต่อท้าย (Suffix) เพื่อให้คำที่ได้อยู่ในรูปของรากศัพท์ และลดความหลากหลายในรูปแบบของคำเช่น analysis, analyzing, analyzer และ analyzed ให้อยู่ในรากศัพท์ "analy" ซึ่งในตารางที่ 2.2 แสดงบางส่วนของ Suffix list โดยในการวิจัยครั้งนี้เราได้ algorithm ในการแปลงคำให้อยู่ในรูปแบบรากศัพท์ของ Martin Porter [14] ในกระบวนการแปลงคำศัพท์ให้อยู่ในรูปรากศัพท์

2.2.6 กระบวนการทำดัชนี (INDEXMAP)

เป็นกระบวนการในการรวบรวมคำศัพท์ที่เหมือนกัน เพื่อหาความถี่ของคำที่ปรากฏและตำแหน่งที่ปรากฏในเอกสาร และทำการถ่วงน้ำหนักคำ เพื่อเป็นเครื่องมือในการกำหนดความสำคัญของคำศัพท์นั้นๆ ในเอกสาร โดยส่วนใหญ่ใช้ฟังก์ชันของความถี่คำที่ปรากฏในเอกสาร เพื่อแทนค่าความสำคัญของคำในเอกสาร

คำที่มีค่าความสำคัญสูงพอ จะถูกกำหนดอยู่ในชุดของเอกสารโดยอาจจะมีหรือไม่มีการถ่วงน้ำหนัก โดยบางครั้งกรณีการทำดัชนีเป็นระบบฐานสอง (Binary Mode) จะกำหนดค่าของคำศัพท์เป็น 1 เมื่อคำศัพท์มีค่าในเอกสาร โดยไม่สนใจความถี่ของคำที่ปรากฏในเอกสาร ส่วนกรณีการทำดัชนีที่มีการให้น้ำหนักของคำ (Weighted Indexing System) น้ำหนักของคำเป็นส่วนที่แสดงถึง ความสำคัญของคำในเอกสาร ส่วนคำที่มีค่าของความสำคัญไม่สูงพอก็จะมีน้ำหนักของคำที่น้อยกว่า

หลังจากนั้นข้อมูลที่ได้จากขั้นตอนการคำนวณน้ำหนักดัชนีที่กล่าวมาข้างต้นจะถูกสร้างเป็นฐานข้อมูลดัชนีของเครื่องจักรสืบค้น เรียกว่า Inverted File ในรูปที่ 2.3 ซึ่งแสดงคำศัพท์และรายการของเอกสารที่มีความสัมพันธ์ต่อคำศัพท์นั้น หรือเป็นข้อมูลที่บอกว่าคำนั้นๆ เป็นดัชนีอยู่ในเอกสารชุดใดบ้าง และ Direct File ในรูปที่ 2.4 ซึ่งจะแสดงเอกสารและรายการของคำศัพท์ที่มีความสัมพันธ์ต่อเอกสารนั้น และใช้ในการบ่งชี้ว่าในเอกสารประกอบด้วยคำใดที่เป็นดัชนี

		Information Item		
		Doc 1	Doc 2	Doc 3
Topic	Term 1	1	0	1
	Term2	1	1	0
	Term 3	0	1	1

รูปที่ 2.3 ตัวอย่างของ Invert File

Information Item		Topic: Term 1	Term 2	Term 3
	Doc 1	1	1	0
	Doc 2	0	1	1
	Doc 3	1	0	1

รูปที่ 2.4 ตัวอย่างของ Direct File

จากกระบวนการในการทำดัชนี จะเห็นได้ว่าการทำงานของ Web robot สามารถแบ่งออกเป็นส่วนๆ และในการทำดัชนีใช้เวลาในการประมวลผล และทำให้เกิดปริมาณปริมาณการสื่อสารข้อมูลจำนวนมาก จากการร้องขอและการส่งเว็บเพจเพื่อนำมาทำดัชนีในส่วนกลาง

2.3 การทำงานของ BDDBot

ในการศึกษาครั้งนี้ผู้วิจัยได้ใช้เครื่องจักรสืบค้นบนอินเตอร์เน็ตที่มีชื่อว่า BDDBot เป็นต้นแบบในการศึกษา เนื่องจาก BDDBot เป็นเครื่องจักรสืบค้นที่มีการทำงานพื้นฐานของเครื่องจักรสืบค้นเกือบทั้งหมดและพัฒนาโดยภาษาจาวาทำให้สามารถทำงานได้บนแพลตฟอร์มต่างกันได้

BDDBot พัฒนาขึ้นเพื่อให้ผู้ที่สนใจศึกษาการทำงานของเครื่องจักรสืบค้น สามารถนำมาใช้และทำความเข้าใจการทำงานของเครื่องจักรสืบค้นบน ผู้ใช้สามารถนำ BDDBot มาใช้งาน และปรับปรุงเปลี่ยนแปลงการทำงานได้ภายใต้เงื่อนไขลิขสิทธิ์แบบ GNU

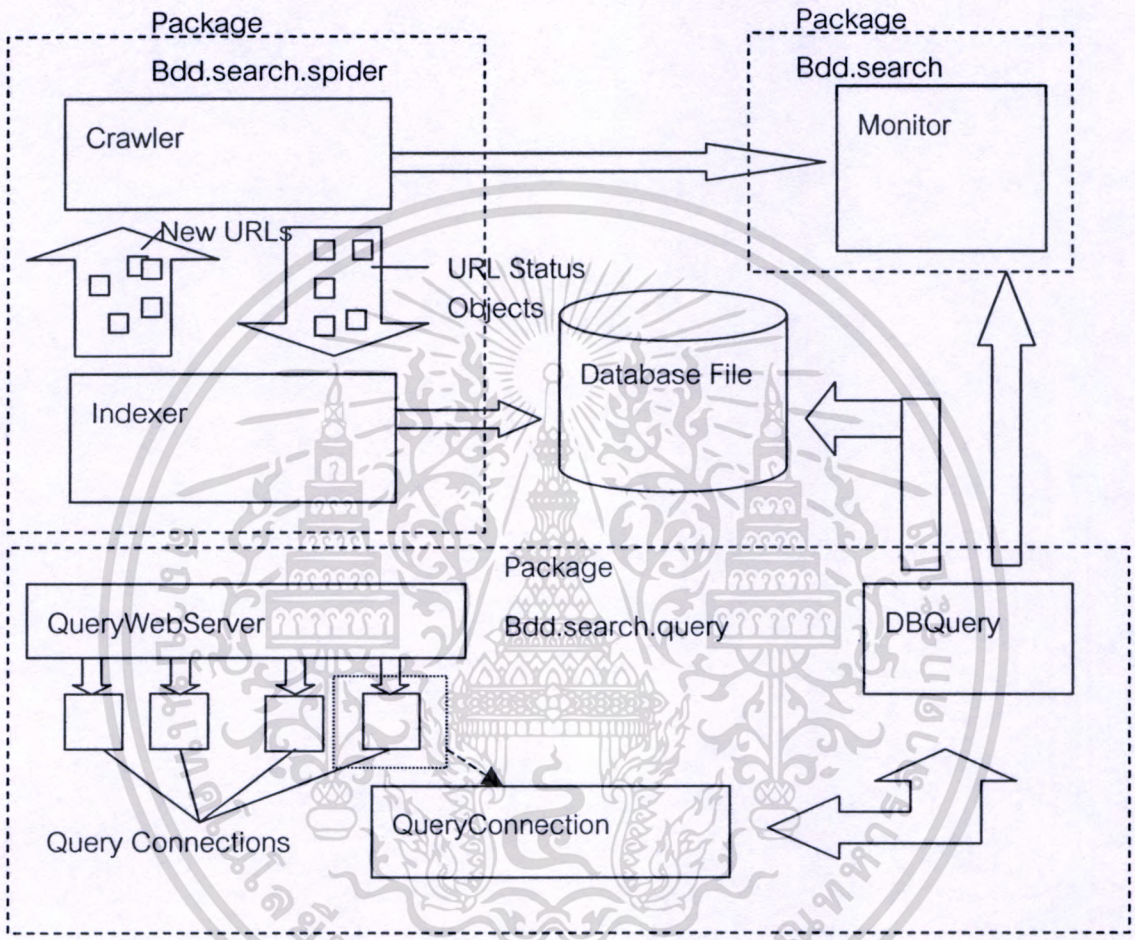
BDDBot พัฒนาจากภาษาจาวา ซึ่งเป็นภาษาที่มีประสิทธิภาพและรวมฟังก์ชันต่างๆ เกี่ยวกับการทำงานบนเครือข่ายไว้ในตัวมันเอง และสามารถทำงานโดยไม่มีปัญหาในการทำงานข้ามแพลตฟอร์ม แต่ข้อจำกัดก็มีเช่นกันได้แก่ ในการทำงานของ BDDBot เครื่องคอมพิวเตอร์ที่เป็นแพลตฟอร์ม Macintosh จะทำให้เกิดปัญหาได้เนื่องจากว่า BDDBot มีการทำงานแบบ Thred แต่เครื่องคอมพิวเตอร์ Macintosh ไม่สนับสนุนการทำงานแบบ Thread แต่อย่างไรก็ตามเป็นที่ทราบกันดีอยู่ดีว่า แพลตฟอร์มของ Mac ไม่ใช่ทางเลือกที่ดีเพื่อเป็นเซิร์ฟเวอร์

2.3.1 การทำงาน และสถาปัตยกรรมของ BDDbot

เพื่อความเข้าใจการทำงานเบื้องต้นของ BDDBot จึงขออธิบายเกี่ยวกับสถาปัตยกรรมและการทำงานของ BDDBot เพื่อความเข้าใจในภาพรวมการทำงาน

การทำงานของ BDDBot มีการทำงานพื้นฐานเหมือนกับเครื่องจักรสืบค้นอื่นๆ มีได้แก่ การ Crawling เว็บเพจ การทำดัชนีเว็บเพจ และมีโปรแกรมสำหรับค้นคืนข้อมูลจากดัชนี โดยในรูปที่ 2.4 เป็นรูปแสดงสถาปัตยกรรมการทำงาน ของ BDDBot โดยรูปสี่เหลี่ยมเป็นตัวแทนของ ส่วน

ประกอบของเครื่องจักรสืบค้น หรือ class ในภาษาจาวาซึ่งจะแสดงเฉพาะ class ที่สำคัญเพื่อไม่ให้ยุ่งเหยิงเกินไป เส้นประรอบๆ สี่เหลี่ยมแสดงถึง package ซึ่งระบุส่วนที่อยู่ของ class ซึ่งประกอบกันขึ้นเป็น BDDBot ประกอบด้วย package bdd.search, bdd.search.spider, bdd.search.query และ package ที่ไม่ได้แสดงไว้ได้แก่ bdd.util



รูปที่ 2.4 สถาปัตยกรรมของ BDDBot [13]

โดยใน package bdd.search ประกอบด้วย ส่วนที่เป็นส่วนหลักของ BDDbot ซึ่งได้แก่ Mornitor ซึ่งเป็นส่วนสำหรับใช้ตรวจสอบดูการทำงานของ BDDbot ซึ่งใน package bdd.search ยังประกอบด้วย package ย่อยสอง package ได้แก่ bdd.search.spider และ bdd.search.query ซึ่งเทียบได้กับส่วนของ Web robot และส่วนของ User Interface ในเครื่องจักรสืบค้นทั่วไปตามลำดับ

2.3.1.1 Package bdd.search.spider

Package bdd.search.spider เทียบได้กับส่วนของ Web robot ในเครื่องจักรสืบค้นทั่วไป ประกอบด้วย class ที่ทำหน้าที่ในการ Crawling ชื่อว่า Crawler ซึ่งทำหน้าที่ในการค้นหาและรวบรวมเว็บเพจเพื่อนำมาทำดัชนี และ class ที่ทำหน้าที่ในการทำดัชนีเรียกว่า Indexer

กระบวนการในการทำดัชนีเริ่มต้นโดยการอ่านไฟล์ urls.txt และ rules.txt ไฟล์เพื่อรับ URL เริ่มต้นในการทำงานจาก urls.txt และข้อกำหนดจาก rules.txt หลังจากนั้น จึงเริ่มต้นกระบวนการ Crawling จาก URL ที่กำหนดไว้และสร้าง URLStatus ออปเจ็คเพื่อเก็บสถานะของ URL นั้นๆ และจึงส่ง URLStatus ไปยัง Indexer ซึ่งจะทำการอ่าน URLStatus และเริ่มกระบวนการทำดัชนี และสร้างฐานข้อมูลดัชนี

2.3.1.2 Package bdd.search.query

Package bdd.search.query ประกอบด้วย classes ที่ทำหน้าที่ในการรับคำค้นจากผู้ใช้ เรียกว่า QueryWebServer และ class ที่นำคำค้นนั้นไปค้นหาจากฐานข้อมูลเรียกว่า DBQuery และ class ซึ่งเป็นตัวกลางเชื่อมระหว่าง QueryWebServer และ DBQuery เรียกว่า QueryConnection

เมื่อมีการรับคำค้นเข้ามา QueryWebServer จะทำการสร้าง QueryConnection ไปยัง DBQuery จากนั้น DBQuery จะทำหน้าที่ในการค้นหา และส่งผลลัพธ์จากการค้นหาให้ผู้ใช้

2.3.1.3 Package bdd.util

Package bdd.util ประกอบด้วย class ซึ่งเป็นเครื่องมือช่วยในการทำงานของ BDDBot เช่น class FIFOQueue ซึ่งทำหน้าที่ในการจัดการคิวแบบ FIFO, class Stemmer ทำหน้าที่ในกระบวนการ STEMWORD, class ResultConclussion ทำหน้าที่ในการสรุปผลการทำงาน และ MyDataInputStream เป็น class ที่ช่วยในการจัดการเกี่ยวกับ InputStream

2.4 สรุปบท

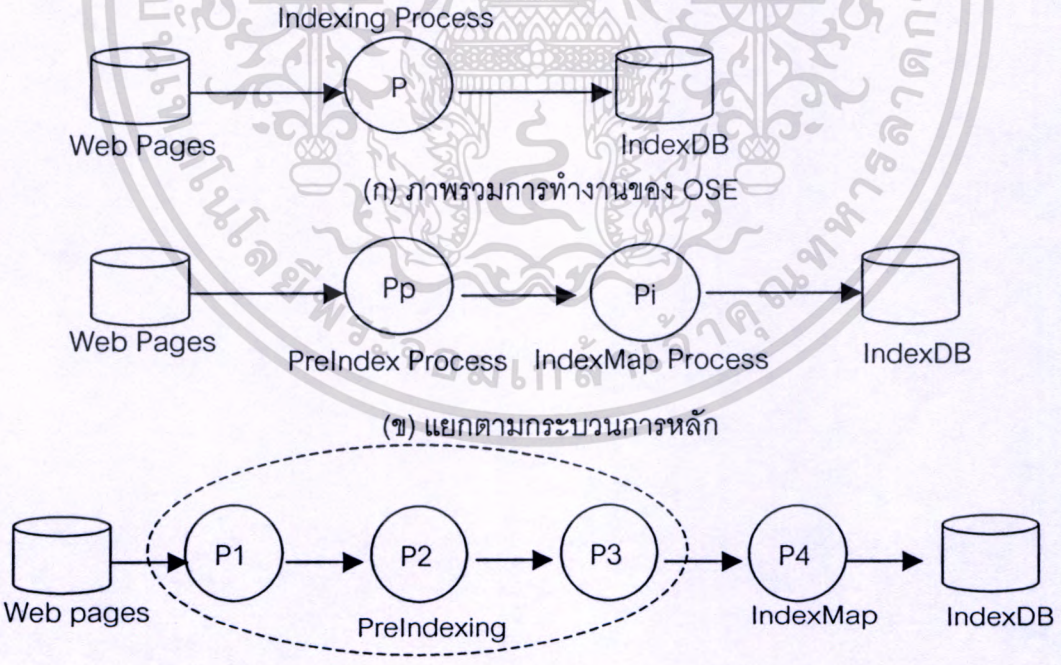
การทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ตประกอบด้วยส่วนสำคัญสามส่วนได้แก่ ส่วน ของงานเบื้องหลังได้แก่ Web robot หรือ Spider ซึ่งทำหน้าที่ในการค้นหาและรวบรวมเว็บเพจเพื่อนำมาทำดัชนี และสร้างส่วนที่เป็นฐานข้อมูลดัชนี ซึ่งเป็นส่วนสำคัญที่สองซึ่งเก็บข้อมูลดัชนี ของเว็บเพจที่ได้จากการทำงานของ Web robot และส่วนที่สามได้แก่ ส่วนติดต่อผู้ใช้ทำหน้าที่ในการรับคำค้นจากผู้ใช้ และนำคำค้นนั้นไปค้นหาในฐานข้อมูลดัชนี เมื่อได้ผลลัพธ์จึงส่งผลลัพธ์กลับไปยังผู้ใช้

บทที่ 3
การกระจายการทำงานของเครื่องจักรสืบค้น

ในบทนี้จะกล่าวถึงแนวคิดและทฤษฎีที่ผู้วิจัยใช้สำหรับเป็นแนวทางในการวิจัย โดยอธิบายแนวคิด และกระบวนการในการทดลอง โดยแบ่งออกเป็นสองส่วนเป็นสองส่วนได้แก่ การหลักการและแนวคิดที่ใช้ในการวิจัย และส่วนอธิบายทดลองเพื่อทดสอบแนวคิด โดยแนวคิดอธิบายเพื่อทราบถึงกระบวนการและความสัมพันธ์ของกระบวนการทำดัชนีแบบเก่า (OSE) พารามิเตอร์ที่เกี่ยวข้อง เวลาในการประมวลผล ขนาดของผลลัพธ์จากกระบวนการขั้นต่างๆ ในการทำดัชนี และการทำงานของ เครื่องจักรสืบค้นแบบกระจาย (DISE) และทำการอธิบายและวิธีการในการทดลอง

3.1 ลักษณะกระบวนการทำดัชนีของ OSE

เมื่อพิจารณากระบวนการในการทำดัชนีของ OSE กระบวนการในการทำดัชนีเป็นกระบวนการต่อเนื่องกันไปดังรูปที่ 3.1 โดยกระบวนการในการทำดัชนีจะต้องทำให้เสร็จสิ้นในครั้งเดียวต่อเอกสาร HTML หนึ่งหน้า แล้วจึงนำดัชนีที่ได้มารวมกันเป็นฐานข้อมูลดัชนี



รูปที่ 3.1 ลำดับการทำงานในกระบวนการทำดัชนี

จากการวิเคราะห์พบว่า กระบวนการในการทำดัชนีสามารถแยกเป็นสองกระบวนการหลักได้แก่ หนึ่งกระบวนการ PreIndexing ซึ่งเป็นกระบวนการเตรียมข้อมูลก่อนการทำดัชนี ประกอบด้วย กระบวนการในการกำจัด HTML tag หรือ RMHTML (P1), กระบวนการ RMSTOP (P2), กระบวนการในการทำคำศัพท์ให้อยู่ในรูปรากศัพท์ STEMWORD (P3) กระบวนการหลักส่วนที่สองได้แก่ กระบวนการถ่วงน้ำหนักคำและทำดัชนี หรือเรียกว่า INDEXMAP (P4) โดยกระบวนการแต่ละขั้นมีการทำงานต่อเนื่องกันไป จนกระทั่งได้ผลลัพธ์สุดท้ายได้แก่ฐานข้อมูลดัชนี

3.1.1 พิจารณาเกี่ยวกับภาระงาน

เมื่อพิจารณาเกี่ยวกับภาระงานในการทำดัชนีซึ่งแปรผันตรงกับประมาณข้อมูลเข้าที่ใช้ในการประมวลผลหรือสามารถแสดงได้เป็นรูปแบบของสมการได้ดังสมการที่ 3.1

$$W_i = P_i f(S_i) \tag{3.1}$$

W_i คือภาระงานที่ i

P_i คือค่าคงที่ของกระบวนการที่ i

S_i คือขนาดของข้อมูลเข้าของกระบวนการที่ i

ซึ่งจะเห็นว่าเมื่อขนาดของข้อมูลเข้าในกระบวนการมีขนาดเพิ่มขึ้นภาระงานในการประมวลผลจะเพิ่มขึ้นด้วย ซึ่งภาระงานในที่นี้ก็คือเวลาในการประมวลผลที่เพิ่มขึ้นเมื่อขนาดข้อมูลเข้ามีขนาดใหญ่ขึ้นนั้นเอง หรือกล่าวได้ว่า W คือเวลา T ที่ใช้ในการประมวลผล

3.1.2 พิจารณาเกี่ยวกับปริมาณการสื่อสารข้อมูล

ในการทำงานของเครื่องจักรสืบค้นแบบ OSE ซึ่งมีการทำงานแบบ รวมศูนย์การทำดัชนีที่ส่วนกลาง ปริมาณข้อมูลที่ส่งผ่านเครือข่ายระหว่างกันจะมีขนาด เท่ากับขนาดของเว็บเพจ บวกด้วยขนาดของ Overhead ของการสื่อสาร โดยสามารถแสดงได้ดังในสมการ 3.2

$$D_T = \sum_{i=1}^{n} F_i S_i + \text{Communication overhead} \tag{3.2}$$

D_T คือขนาดข้อมูลทั้งหมดที่ส่งผ่านเครือข่ายระหว่างการทำงาน

F_i คือจำนวนของ HTML ไฟล์ในโฮสต์ที่ i

S_i คือขนาดของ HTML ไฟล์โดยเฉลี่ยที่ i

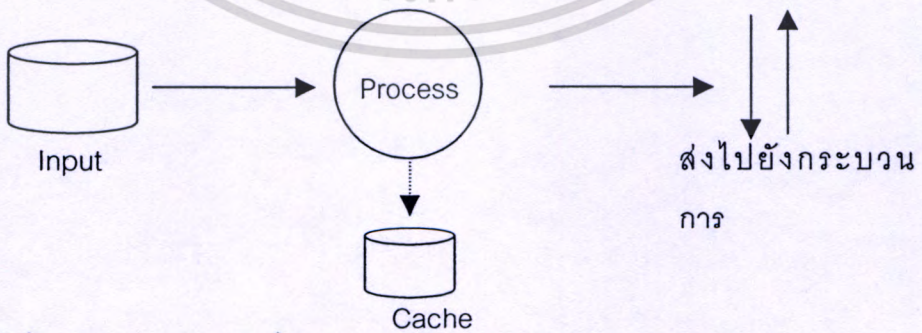
โดยในแต่ละขั้นตอนของกระบวนการทำดัชนีจะมีข้อมูลบางส่วนที่จะถูกกำจัดออกไปโดยจากการทดลองในตารางที่ 3.1 ซึ่งแสดงขนาดของข้อมูลเข้าและขนาดของข้อมูลดัชนี ทำให้เห็นว่าขนาดของผลลัพธ์ลดลงโดยขนาดของผลลัพธ์ขึ้นกับข้อมูลที่ต้องการเก็บและกระบวนการในการทำดัชนีที่ใช้ โดยที่ขนาดข้อมูลเข้า 100MB ผลลัพธ์ของการทำงานของ FreeWAIS-sf มีขนาดเท่ากับ 56 MB หรือมีขนาดเท่ากับ 56% ของข้อมูลเข้า และผลลัพธ์ของ Harvest มีขนาด 2904 KB หรือประมาณ 3% ของข้อมูลเข้าส่วนเครื่องจักรสืบค้นอื่นเกิดการทำงานผิดพลาดเนื่องจากการใช้หน่วยความจำหมดไป โดยประมาณผลลัพธ์ที่ 10 MB ของ FFW มีขนาด 664 KB และผลลัพธ์ของ Isite มีขนาด 2904 KB และผลลัพธ์ของ SWISH มีขนาด 2816 KB หรือมีขนาดเท่ากับ 6%, 29% และ 28% ของขนาดข้อมูลเข้าตามลำดับ

ตารางที่ 3.1 ขนาดผลลัพธ์ข้อมูลดัชนี [14]

Engine	10K	100K	1MB	10MB	100MB
FreeWAIS-sf	24K	176K	768K	7080K	56000K
FFW	8K	40K	176K	664K	Core dump
Harvest	5K	29K	168K	736K	2904K
Isite	4K	40K	344K	2904K	Core dump
SWISH	8K	56K	304K	2816K	Core dump

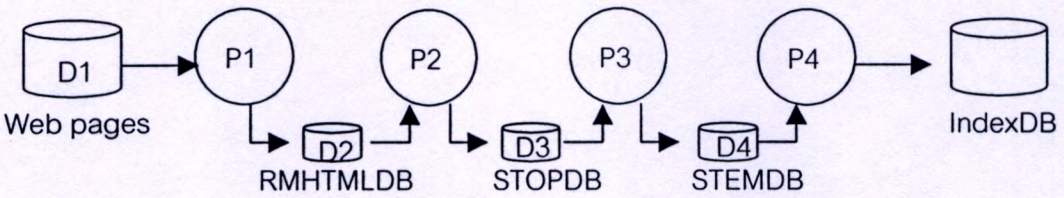
3.2 ลักษณะการทำงานของ DISE

การทำงานของ DISE มีการทำงานเหมือนกับ OSE และมีการเขียนผลลัพธ์ในแต่ละขั้นลงดิสก์ เก็บไว้สำหรับส่งต่อไปยังกระบวนการถัดไปที่อยู่ต่างเครื่องหรืออยู่บนเครื่องเดียวกันโดยสามารถแสดงได้ดังรูป 3.2



รูปที่ 3.2 เมื่อผ่านกระบวนการหนึ่งผลลัพธ์จะถูกส่งไปยังกระบวนการถัดไป

ดังนั้นปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำงานของ DISE จึงมีขนาดเท่ากับขนาดของ ผลลัพธ์ในแต่ละกระบวนการ ซึ่งกระบวนการโดยรวมของ DISE สามารถแสดงได้ดังรูป 3.3



รูปที่ 3.3 ภาพรวมของกระบวนการในการทำดัชนีของ DISE

3.2.1 กระบวนการ RMHTML (P1)

เมื่อเว็บเพจผ่านเข้าสู่กระบวนการ P1 ซึ่งเป็นกระบวนการในการกำจัด HTML Tag ที่อยู่ในเว็บเพจเพื่อให้เหลือข้อมูลที่เป็นเนื้อหาของเว็บเพจเพียงอย่างเดียว ผลลัพธ์ในกระบวนการนี้ถูกจัดเก็บเป็นไฟล์ฐานข้อมูลชื่อ rmhtml.db ซึ่งเก็บข้อมูลได้แก่จำนวนเว็บเพจ (NUMBER_OF_URL), URL ที่อ้างถึง (URL) และเนื้อหาภายใน URL (PAGE_CONTENT) โดยมีรูปแบบไฟล์ ดังรูป 3.4 [16]

```
NUMBER_OF_URL(INTEGER) URL(STRING)\n
CONTENT_LENGTH(LONG) PAGE_CONTENT(STRING)\n
URL(STRING)\n
CONTENT_LENGTH(LONG) PAGE_CONTENT(STRING)\n
URL(STRING)\n
CONTENT_LENGTH(LONG) PAGE_CONTENT(STRING)\n
URL(STRING)\n
CONTENT_LENGTH(LONG) PAGE_CONTENT(STRING)\n
.
.
.
```

รูปที่ 3.4 รูปแบบของไฟล์ rmhtml.db

3.2.2 กระบวนการ RMSTOP (P2)

กระบวนการ RMSTOP หรือ P2 เป็นกระบวนการในการกำจัด Stop word หรือคำที่ไม่มีความหมายต่อเนื้อหาในการใช้เป็นดัชนีของเพจนั้นๆ โดยผลลัพธ์ของกระบวนการนี้เก็บอยู่ในไฟล์ฐานข้อมูลชื่อ rmstop.db2 โดยเก็บข้อมูล จำนวน URL (NUMBER_OF_URL), จำนวนคำใน URL (NUMBER_OF_WORD), คำที่อยู่ใน URL (WORD), ความถี่ของคำนั้นที่พบใน URL (WORD_FRQ), ตำแหน่งแรกของคำนั้น (WORD_POS) โดยมีการเก็บในรูปแบบดังรูปที่ 3.5 [16]

```
NUMBER_OF_URL(INTEGER)  NUMBER_OF_WORD(INTEGER)  URL(STRING)  \n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
.
.
.
END_OF_URL(INTEGER  -1)  NUMBER_OF_WORD(INTEGER)  URL(STRING)  \n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
.
.
```

รูปที่ 3.5 รูปแบบไฟล์ rmstop.db

3.2.3 กระบวนการ STEMWORD (P3)

กระบวนการ STEMWORD (P3) เป็นกระบวนการในการแปลงคำศัพท์ที่มีรูปแบบหลากหลายแต่มีรากศัพท์ในรูปเดียวกัน ให้อยู่ในรูปรากศัพท์ โดยในการวัจัยครั้งนี้ผู้ใช้ในกระบวนการในแปลงรูปคำศัพท์ของ Martin Porter [15] โดยผลลัพธ์จากกระบวนการนี้เก็บในไฟล์ฐานข้อมูลชื่อ stem.db3 โดยมีรูปแบบดังรูป 3.6 [16]

```
NUMBER_OF_URL(INTEGER)  NUMBER_OF_WORD(INTEGER)  URL(STRING)  \n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
.
.
.
END_OF_URL(INTEGER  -1)  NUMBER_OF_WORD(INTEGER)  URL(STRING)  \n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
WORD_FRQ(INTEGER) WORD_POS(INTEGER) WORD(STRING)\n
.
.
.
```

รูปที่ 3.6 รูปแบบไฟล์ stem.db3

3.2.4 กระบวนการ INDEXMAP (P4)

กระบวนการ INDEXMAP ประกอบด้วยกระบวนการในการให้น้ำหนักแก่คำที่มีอยู่ในเอกสารและทำการจับคู่คำกับเอกสารเพื่อใช้ในการค้นคืน โดยในไฟล์ดัชนีเก็บข้อมูล พอยต์เตอร์ชี้ไปยังคำค้น (POINTER_TO_WORD), จำนวน URL ที่มีอยู่ใน ฐานข้อมูลดัชนี (NUMBER_OF_URL), URL, คำค้น (WORD), น้ำหนักของคำนั้น(SCORE) และหมายเลข URL ที่คำนั้นปรากฏอยู่ (URL_NUMBER) โดยเก็บอยู่ในไฟล์ฐานข้อมูลชื่อ main.db และมีรูปแบบดังรูป 3.7 [16]

```
POINTER_TO_WORD(LONG) NUMBER_OF_URL(INTEGER)\n
URL(STRING) \n
URL(STRING) \n
URL(STRING) \n
.
.
.
WORD(STRING) \n
URL_NUMBER(INTEGER) SCORE(BYTE) URL_NUMBER(INTEGER) SCORE(BYTE)
... 0 \n
WORD(STRING) \n
URL_NUMBER(INTEGER) SCORE(BYTE) URL_NUMBER(INTEGER) SCORE(BYTE)
... 0 \n
WORD(STRING) \n
URL_NUMBER(INTEGER) SCORE(BYTE) URL_NUMBER(INTEGER) SCORE(BYTE)
... 0 \n
.
.
.
```

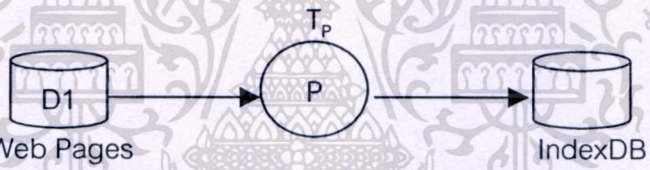
รูปที่ 3.7 รูปแบบของไฟล์ main.db

3.3 การวิเคราะห์

กระบวนการในการทำดัชนีของเครื่องจักรสืบค้นบนอินเตอร์เน็ตสามารถแบ่งและแตกออกเป็นกระบวนการย่อย และสามารถกระจายกระบวนการย่อยเหล่านี้ออกไปเพื่อกระจายภาระการทำดัชนีข้อมูลเว็บเพจ เพื่อเพิ่มประสิทธิภาพในการทำดัชนี ทำให้การทำดัชนีข้อมูลเร็วขึ้น ซึ่งในกระบวนการย่อยที่ได้ทำการแตกกระบวนการออกมา ได้แก่ RMHTML, RMSTOP, STEMWORD, INDEXMAP โดยแบ่งส่วนการทำงานในการทำดัชนีออกเป็นสองส่วนได้แก่ ในส่วนกลาง และส่วนกระบวนการที่กระจายออกไป โดยกระบวนการทำดัชนีที่กระจายออกไปได้แก่ กระบวนการ RMHTML, RMSTOP, STEMWORD ในส่วนกลางได้แก่กระบวนการ INDEXMAP ซึ่งมีหน้าที่ในการรวบรวมผลลัพธ์จากกระบวนการทำดัชนีที่กระจายออกไป และสร้างเป็นฐานข้อมูลดัชนี

3.3.1 วิเคราะห์การทำงานของ OSE

เวลาในการประมวลผลของ OSE ซึ่งในกระบวนการทำดัชนีมีการทำงานต่อเนื่องกันไปโดยไม่แยกออกจากกัน เวลาในการประมวลผล และความสัมพันธ์ของข้อมูลเข้าและข้อมูลที่ออกจากกระบวนการสามารถแสดงได้ดังสมการ 3.3 และ 3.4



รูปที่ 3.7 เวลาในการประมวลผลและขนาดข้อมูลเข้าของ OSE

$$T_{total\ OSE} = T_p(D_1) \tag{3.3}$$

$$D_{index} = g(D_1) \tag{3.4}$$

$T_{total\ OSE}$ คือเวลาในการทำดัชนีของ OSE

D_{index} คือขนาดข้อมูลดัชนี

D_1 คือขนาดของข้อมูลเว็บเพจ

$T_p(D_1)$ คือฟังก์ชันของเวลาในการทำดัชนีในรูป D1

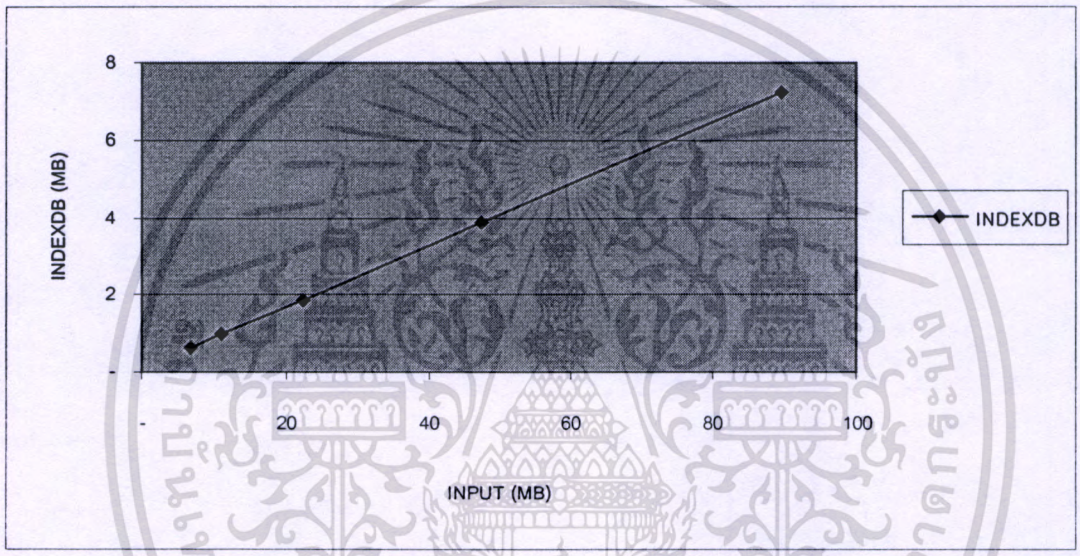
$g(D_1)$ คือฟังก์ชันขนาดข้อมูลในรูปของ D1

ซึ่งความสัมพันธ์ ของขนาดข้อมูลในแต่ละกระบวนการ จากการทดลองเบื้องต้นพบว่าความสัมพันธ์ของขนาดข้อมูลเข้ากับข้อมูลดัชนี และเวลาในการทำดัชนีของ OSE สามารถแสดง

ได้ในรูปที่ 3.8 และ 3.9 ตามลำดับ โดยในการทดลองได้ทำการสร้างเว็บไซต์ที่มีขนาด 20MB, 50MB 100MB, 200MB และ 400MB ซึ่งเป็นขนาดของข้อมูลเว็บไซต์ ซึ่งรวมถึงไฟล์ชนิดอื่นๆ ที่ไม่ใช่ HTML ด้วย

ตารางที่ 3.2 ผลการทดลองขนาดดัชนี OSE

INPUT	6.88	10.88	22.23	47.41	89.45
INDEXDB	0.62	0.98	1.87	3.86	7.22



รูปที่ 3.8 กราฟความสัมพันธ์ของขนาดข้อมูล OSE

เมื่อทำการวิเคราะห์โดยใช้ Regression analysis (ดูในภาคผนวก ค) จากสมการแนวโน้ม ซึ่งแสดงความสัมพันธ์ของขนาดข้อมูล Input และขนาดข้อมูลดัชนีในกระบวนการทำดัชนีของ OSE ในตารางที่ 3.3 พบว่า

ตารางที่ 3.3 สมการความสัมพันธ์ของขนาดข้อมูล OSE

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$D_{index} = 0.079700381\ D_1 + 0.091912319$	0.999964067
2	Logarithm	$D_{index} = 2.461081198 Ln(D_1) - 4.851229834$	0.898280683
3	Polynomial	$D_{index} = 0.000002831\ D_1^2 + 0.079427256\ D_1 +$	0.999964665

		0.095397065	
4	Polynomial	$D_{index} = 0.000001232\ D_1^3 - 0.000170082\ D_1^2 + 0.085573970\ D_1 + 0.049528325$	0.999986098
5	Power	$D_{index} = 0.099413084\ D_1^{0.951106761}$	0.999817504
6	Exponential	$D_{index} = 0.748398296e^{0.027738100\ D1}$	0.902649279

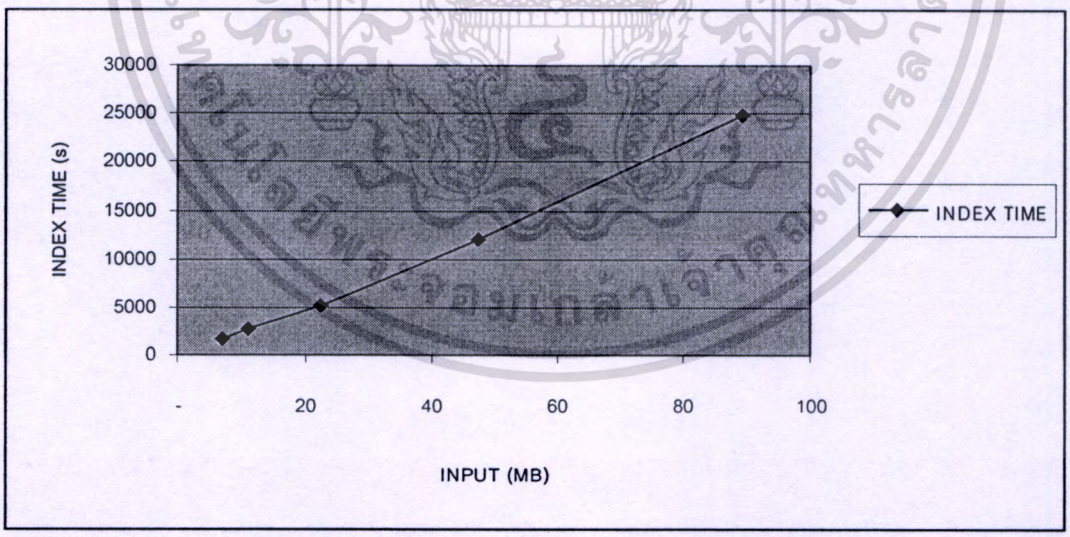
สรุปความสัมพันธ์

พบว่าลักษณะความสัมพันธ์เป็นแบบ Linear

อธิบาย เนื่องจากให้ค่า R^2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ในรูปแบบ Polynomial กำลัง 2 เนื่องจากค่าสัมประสิทธิ์ของตัวแปรกำลัง 2 มีค่าใกล้เคียง 0 และเมื่อเปรียบเทียบกับสัมประสิทธิ์ของตัวแปรกำลัง 1 มีค่าน้อยมากประมาณ 0.004%

ตารางที่ 3.4 ผลการทดลองเวลาในการประมวลผล OSE

INPUT	6.88	10.88	22.23	47.41	89.45
INDEX TIME	1,573	2,593	5,232	11,916	24,778



รูปที่ 3.9 เวลาในการทำดัชนีของ OSE

เมื่อทำการวิเคราะห์โดยใช้ Regression analysis (ดูในภาคผนวก ค) จากสมการแนวโน้ม ซึ่งแสดงความสัมพันธ์ของเวลาในการประมวลผลกับขนาดข้อมูล Input ในกระบวนการทำดัชนี ของ OSE ในตารางที่ 3.5

ตารางที่ 3.5 ตารางสมการความสัมพันธ์ของเวลาในการประมวลผลและขนาดข้อมูลเข้า OSE

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$T_{total\ OSE} = 280.7482010\ D_1 - 711.5858579$	0.9975957
2	Logarithm	$T_{total\ OSE} = 8532.8475187 Ln(D_1) - 17693.7599473$	0.8681708
3	Polynomial	$T_{total\ OSE} = 0.6308101\ D_1^2 + 219.8840339\ D_1 + 64.9683725$	0.9999832
4	Polynomial	$T_{total\ OSE} = -0.0004064\ D_1^3 + 0.6878539\ D_1^2 + 217.8562306\ D_1 + 80.1004875$	0.9999834
6	Power	$T_{total\ OSE} = 199.1227826\ D_1^{1.0664291}$	0.9992761
7	Exponential	$T_{total\ OSE} = 1904.1601612e^{0.0312580\ D1}$	0.9112701

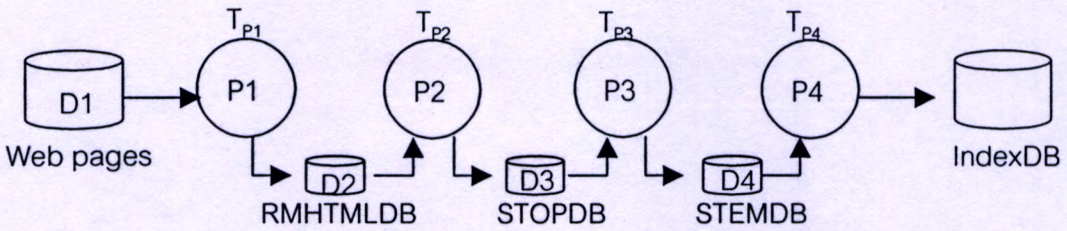
สรุปความสัมพันธ์

พบว่าลักษณะความสัมพันธ์เป็นแบบ Polynomial กำลัง 2

อธิบาย เนื่องจากให้ค่า R^2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ในรูปแบบ Polynomial กำลัง 3 เนื่องจากค่าสัมประสิทธิ์ของตัวแปรกำลัง 3 มีค่าน้อยมากเมื่อเทียบกับสัมประสิทธิ์ของตัวแปรกำลังสอง และเมื่อขนาดข้อมูล D_1 มากขึ้นเวลาในการทำดัชนีเป็นลบซึ่งเป็นไปไม่ได้จึงไม่ใช้รูปแบบความสัมพันธ์ polynomial กำลัง 3 และเมื่อ X

3.3.2 วิเคราะห์การทำงานของ DISE

ในการทำงานของ DISE ที่กระบวนการในการทำดัชนีสามารถทำงานโดยไม่ต้องต่อเนื่องกันไปได้ โดยการทำงานสามารถแยกออกจากกันและสามารถทำงานขนานกันไป ซึ่งการทำงานของ DISE สามารถแสดงได้ดังรูป 3.10



รูปที่ 3.10 เวลาในการประมวลผลและขนาดข้อมูลเข้าในแต่ละกระบวนการ DISE

เวลาของการทำดัชนีของ DISE ซึ่งมีการทำงานในกระบวนการย่อยซึ่งไม่ต้องทำงานต่อเนื่องกันไปในคราวเดียว จึงสามารถเขียนในรูปของผลรวมของเวลาในการประมวลผลของกระบวนการย่อย P1 ถึง P4 ซึ่งสามารถแสดงได้ในรูปสมการ 3.5

$$T_{total\ DISE} = T_{P1}(D_1) + T_{P2}(D_2) + T_{P3}(D_3) + T_{P4}(D_4) \quad (3.5)$$

D_1 คือขนาดของข้อเว็บเพจที่เข้าสู่กระบวนการ P_1

D_2 คือขนาดของ RMHTMLDB ที่เข้าสู่กระบวนการ P_2

D_3 คือขนาดของ RMSTOP ที่เข้าสู่กระบวนการ P_3

D_4 คือขนาดของ STEMDB ที่เข้าสู่กระบวนการ P_4

$T_{total\ DISE}$ คือเวลารวมที่ใช้ในการทำดัชนีแบบ DISE

$T_{pi}(D_i)$ คือฟังก์ชันของเวลาของกระบวนการที่ i ในรูปของ Input ที่ i

โดยความสัมพันธ์ของขนาดข้อมูลผลลัพธ์และข้อมูลเข้า ที่เข้าสู่กระบวนการต่างๆ สามารถเขียนได้ในรูปของ ฟังก์ชันความสัมพันธ์กับข้อมูลของกระบวนการที่ i -1 ของข้อมูลที่เข้าในกระบวนการดังแสดงในสมการ 3.6 ถึง 3.9

$$D_2 = g_1(D_1) \quad (3.6)$$

$$D_3 = g_2(D_2) \quad (3.7)$$

$$D_4 = g_3(D_3) \quad (3.8)$$

$$D_5 = g_4(D_4) \quad (3.9)$$

เมื่อ $g_i\ (D_i)$ คือฟังก์ชันของข้อมูลในกระบวนการ p ที่ i ในรูปของ Input ของกระบวนการ p ที่ i

โดยเราสามารถหาความสัมพันธ์ของข้อมูลในแต่ละกระบวนการ จากการทดลองการทำงานงานของ DISE ซึ่งทดลองโดยการใช้ข้อมูลเว็บไซต์ที่มีขนาด 20MB, 50MB 100MB, 200MB และ 400MB ซึ่งเป็นขนาดของข้อมูลเว็บไซต์ ซึ่งรวมถึงไฟล์ชนิดอื่นๆ ที่ไม่ใช่ HTML ด้วย ส่วนขนาดของ INPUT คือขนาดของ HTML ซึ่งที่ขนาดเว็บไซต์ 20MB, 50MB, 100MB, 200MB, 400MB มีขนาด 6.88MB, 10.88MB, 22.33MB, 47.41MB และ 89.45MB ตามลำดับ โดยให้ DISE ทำดัชนีข้อมูลเว็บเพจที่อยู่บนเครื่องเดียวกัน ซึ่งได้ผลการทดลองดังนี้

ตารางที่ 3.5 ผลการทดลองขนาดของข้อมูลในกระบวนการต่าง P1 ถึง P4

WEB size (MB)	20	50	100	200	400
INPUT	6.88	10.88	22.23	47.41	89.45
RMHTMLDB	4.97	7.63	15.78	33.47	63.04
RMSTOPDB	1.80	2.92	6.00	12.45	24.32
STEMDB	1.32	2.16	4.43	9.18	18.11
INDEXDB	0.62	0.98	1.87	3.86	7.22

ตารางที่ 3.6 เวลาในการประมวลผล DISE

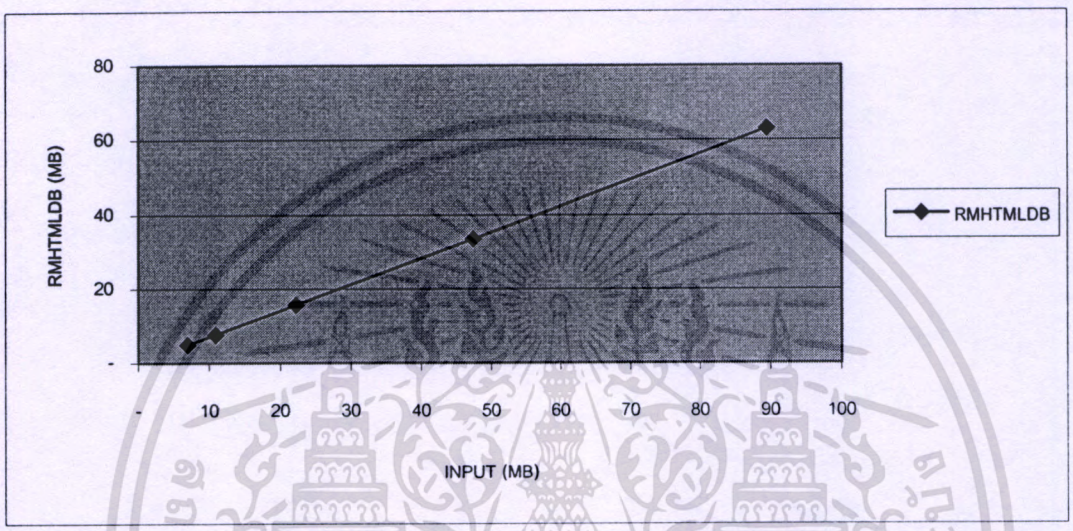
WEB size (MB)	20	50	100	200	400
RMHTML (s)	929	1,622	3,202	9,490	24,759
RMSTOP (s)	290	466	934	1,984	3,823
STEMWORD (s)	149	258	501	1,094	2,166
INDEXMAP (s)	437	750	1,526	3,423	6,730
Total time (s)	1,805	3,096	6,164	15,991	37,478

จากผลการทดลองที่ได้จะนำไปหาความสัมพันธ์ของข้อมูลในกระบวนการต่างๆ โดยใช้เส้นแนวโน้ม และสมการความสัมพันธ์ โดยแยกเป็นความสัมพันธ์ของขนาดข้อมูลในกระบวนการต่างๆ และความสัมพันธ์ของเวลาในการประมวลผลในกระบวนการ P1 ถึง P4 ที่มีความสัมพันธ์กับข้อมูลเข้าในแต่ละกระบวนการ

3.3.2.1 ความสัมพันธ์ของข้อมูลเข้าและข้อมูลออกในกระบวนการ P1

ตารางที่ 3.7 แสดงผลการทดลองขนาดข้อมูล INPUT และ RMHTMLDB

INPUT	6.88	10.88	22.23	47.41	89.45
RMHTMLDB	4.97	7.63	15.78	33.47	63.04



รูปที่ 3.11 กราฟความสัมพันธ์ของข้อมูลในกระบวนการ P1

เมื่อทำการวิเคราะห์โดยใช้ Regression analysis จากสมการแนวโน้มซึ่งแสดงความสัมพันธ์ของขนาดข้อมูลเข้าในกระบวนการ P1 INPUT และผลลัพธ์จากกระบวนการ P1 RMHTMLDB ในตารางที่ 3.8

ตารางที่ 3.8 สมการความสัมพันธ์ของข้อมูลในกระบวนการ P1

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$D_2 = 0.704034577\ D_1 + 0.077475026$	0.999993279
2	Logarithm	$D_2 = 21.733427035 Ln(D_1) - 43.567119548$	0.897763831
3	Polynomial	$D_2 = -0.000021675\ D_1^2 + 0.706125924\ D_1 + 0.050791942$	0.999993728
4	Power	$D_2 = 0.722230352\ D_1^{0.994085614}$	0.999919681
5	Exponential	$D_2 = 5.968169405 e^{0.028935542\ D1}$	0.899257811

สรุปความสัมพันธ์

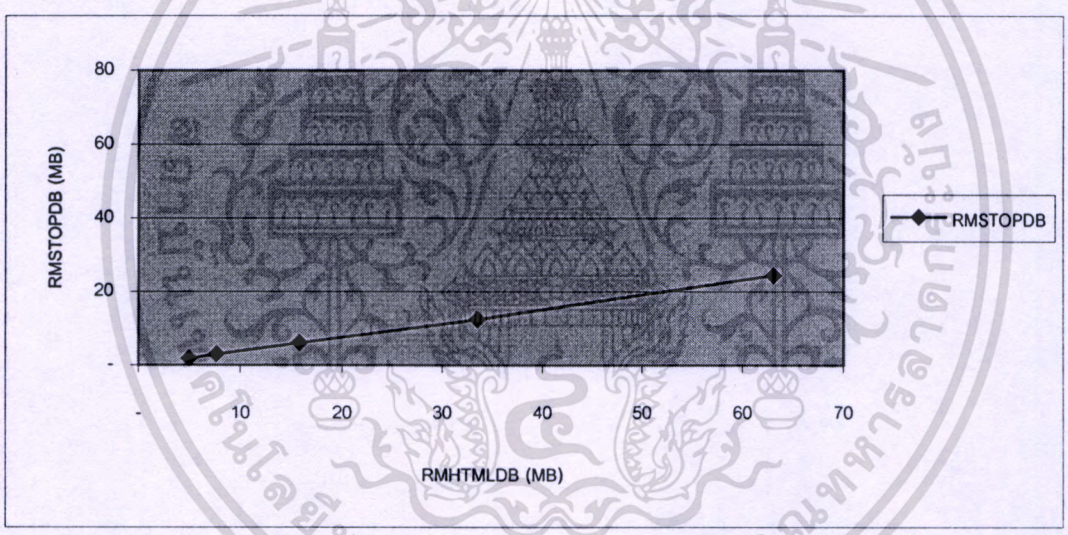
พบว่าลักษณะความสัมพันธ์เป็นแบบ Linear

อธิบาย เนื่องจากให้ค่า R^2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ในรูปสมการ polynomial เนื่องจากมีค่าสัมประสิทธิ์ที่ตัวแปรกำลังสองใกล้เคียง 0 และเมื่อเทียบกับสัมประสิทธิ์ของตัวแปรกำลังหนึ่งพบว่ามีค่าน้อยมากประมาณ 0.003% หรือประมาณได้ว่าอยู่ในรูป Linear

3.3.2.2 ความสัมพันธ์ของข้อมูลเข้าและข้อมูลออกในกระบวนการ P2

ตารางที่ 3.9 แสดงผลการทดลองขนาดข้อมูล RMHTMLDB และ RMSTOPDB

RMHTMLDB	4.97	7.63	15.78	33.47	63.04
RMSTOPDB	1.80	2.92	6.00	12.45	24.32



รูปที่ 3.12 กราฟความสัมพันธ์ระหว่างข้อมูลในกระบวนการ P2

เมื่อทำการวิเคราะห์โดยใช้ Regression analysis จากสมการแนวโน้มซึ่งแสดงความสัมพันธ์ของขนาดข้อมูล RMSTOPDB และขนาดข้อมูล RMHTMLDB ในกระบวนการ P2 ในตารางที่ 3.10

ตารางที่ 3.10 สมการความสัมพันธ์ระหว่างข้อมูลในกระบวนการ P2

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$D_3 = 0.385695981\ D_2 - 0.135568699$	0.999586058
2	Logarithm	$D_3 = 8.405948022 Ln(D_2) - 14.121009118$	0.891849353
3	Polynomial	$D_3 = 0.000431778\ D_2^2 + 0.356312224\ D_2 + 0.129831846$	0.999880543
4	Power	$D_3 = 0.363130354\ D_2^{1.012908239}$	0.999615198
5	Exponential	$D_3 = 2.210667111e^{0.041629608\ D2}$	0.898895365

สรุปความสัมพันธ์

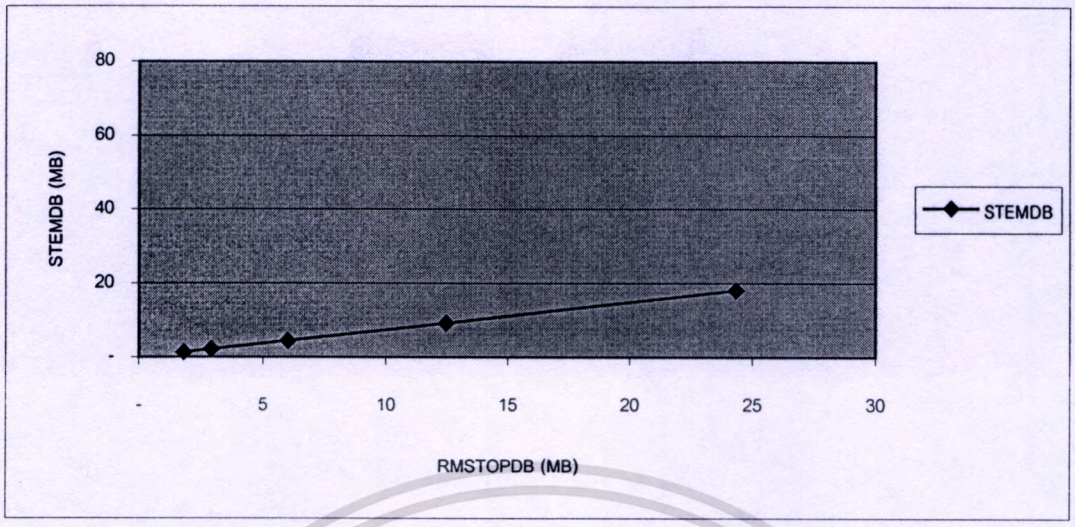
พบว่าลักษณะความสัมพันธ์เป็นแบบ Linear

อธิบาย เนื่องจากให้ค่า R2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ในรูป สมการ polynomial เนื่องจากมีค่าสัมประสิทธิ์ที่ค่ายกกำลังระดับสองมีค่าใกล้เคียง 0 และเมื่อเทียบกับสัมประสิทธิ์ของตัวแปรกำลังหนึ่งมีค่าน้อยมากประมาณ 0.1% หรือประมาณได้ว่าเป็น Linear

3.3.2.3 ความสัมพันธ์ของข้อมูลเข้าและข้อมูลออกในกระบวนการ P3

ตารางที่ 3.11 แสดงผลการทดลองขนาดข้อมูล RMSTOPDB และ STEMDB

RMSTOPDB	1.80	2.92	6.00	12.45	24.32
STEMDB	1.32	2.16	4.43	9.18	18.11



รูปที่ 3.13 กราฟความสัมพันธ์ระหว่างข้อมูลในกระบวนการ P3

เมื่อทำการวิเคราะห์โดยใช้ Regression analysis จากสมการแนวโน้มซึ่งแสดงความสัมพันธ์ของขนาดข้อมูล STEMDB และขนาดข้อมูล RMSTOPDB ในกระบวนการ P3 ตารางที่ 3.12

ตารางที่ 3.12 สมการความสัมพันธ์ระหว่างข้อมูลในการบวนการ P3

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$D_4 = 0.745034590\ D_3 - 0.033689774$	0.999976006
2	Logarithm	$D_4 = 6.173421546 Ln(D_3) - 4.273684805$	0.889434624
3	Polynomial	$D_4 = 0.000577477\ D_3^2 + 0.729856774\ D_3 + 0.018733945$	0.999997236
4	Power	$D_4 = 0.734160980\ D_3^{1.003684225}$	0.999987403
5	Exponential	$D_4 = 1.658282291 e^{0.107920180\ D3}$	0.892442398

สรุปความสัมพันธ์

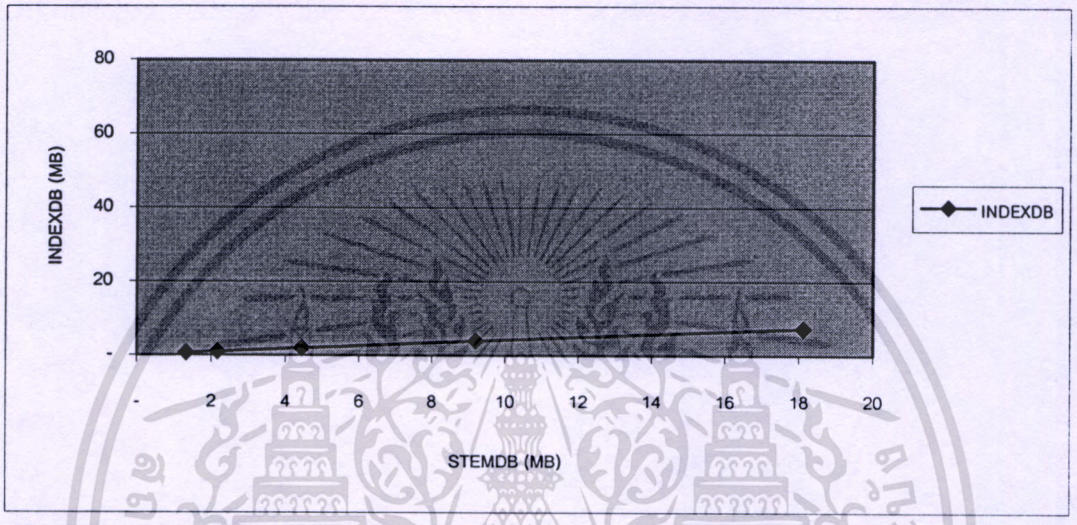
พบว่าลักษณะความสัมพันธ์เป็นรูปแบบ Linear

อธิบาย เนื่องจากให้ค่า R2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ ในรูป สมการ polynomial เนื่องจากมีค่าสัมประสิทธิ์ที่ค่ายกกำลังระดับสองมีค่าใกล้เคียง 0 และเมื่อเทียบกับสัมประสิทธิ์ของตัวแปรกำลังหนึ่งมีค่าน้อยมากประมาณ 0.08% หรือใกล้เคียง Linear

3.3.2.4 ความสัมพันธ์ของข้อมูลเข้าและข้อมูลออกในกระบวนการ P4

ตารางที่ 3.13 แสดงผลการทดลองขนาดข้อมูล STEMDB และ INDEXDB

STEMDB	1.32	2.16	4.43	9.18	18.11
INDEXDB	0.62	0.98	1.87	3.86	7.22



รูปที่ 3.14 กราฟความสัมพันธ์ระหว่างข้อมูลในกระบวนการ P4

เมื่อทำการวิเคราะห์โดยใช้ Regression analysis จากสมการแนวโน้มซึ่งแสดงความสัมพันธ์ของขนาดข้อมูล STEMDB และขนาดข้อมูล INDEXDB ในกระบวนการ P4 ตารางที่ 3.14

ตารางที่ 3.14 สมการความสัมพันธ์ของข้อมูลในกระบวนการ P4

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$D_5 = 0.393777445\ D_4 + 0.137443868$	0.999515428
2	Logarithm	$D_5 = 2.437224832 Ln(D_4) - 0.820224105$	0.900203816
3	Polynomial	$D_5 = -0.001808908\ D_4^2 + 0.429187932\ D_4 + 0.046705005$	0.999928469
4	Power	$D_5 = 0.472788807\ D_4^{0.940876743}$	0.99980925
5	Exponential	$D_5 = 0.764052026e^{0.136357032\ D4}$	0.893192213

สรุปความสัมพันธ์

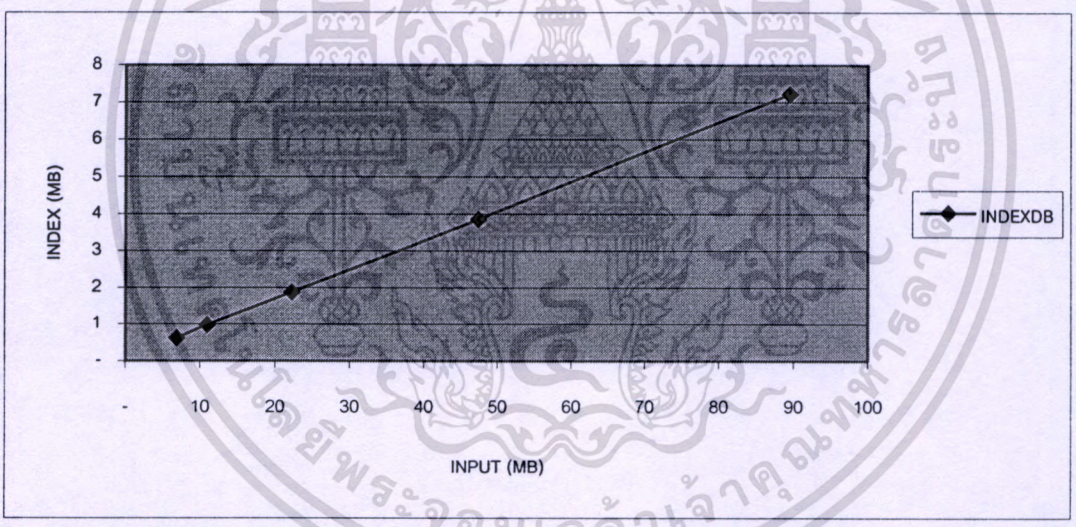
พบว่าลักษณะความสัมพันธ์เป็นแบบ Linear

อธิบายเนื่องจากให้ค่า R^2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ในรูปสมการ polynomial เนื่องจากเมื่อขนาดข้อมูลสูงขึ้นให้ค่าผลลัพธ์ติดลบ และค่าสัมประสิทธิ์ของตัวแปรกำลังสองเทียบกับตัวแปรกำลังหนึ่งมีค่าน้อยมากประมาณ 0.4% หรือประมาณได้ว่าอยู่ในรูปแบบ Linear

3.3.2.5 ความสัมพันธ์ของข้อมูลเข้าและข้อมูลดัชนีกระบวนรวม

ตารางที่ 3.15 แสดงผลการทดลองขนาดข้อมูล INPUT และ INDEXDB

INPUT	6.88	10.88	22.23	47.41	89.45
INDEXDB	0.62	0.98	1.87	3.86	7.22



รูปที่ 3.15 กราฟความสัมพันธ์ของข้อมูลในกระบวนการทำดัชนีรวม DISE

เมื่อทำการวิเคราะห์โดยใช้ Regression analysis จากสมการแนวโน้มซึ่งแสดงความสัมพันธ์ของขนาดข้อมูล INPUT และขนาดข้อมูล INDEXDB ในกระบวนการ P3 ตารางที่ 3.16

ตารางที่ 3.16 สมการความสัมพันธ์ของข้อมูลในกระบวนการทำดัชนีรวม DISE

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$D_5 = 0.079700381\ D_1 + 0.091912319$	0.999964067
2	Logarithm	$D_5 = 2.461081198 Ln(D_1) - 4.851229834$	0.898280683
3	Polynomial	$D_5 = 0.000002831\ D_1^2 + 0.079427256\ D_1 + 0.095397065$	0.999964665
4	Power	$D_5 = 0.099413084\ D_1^{0.951106761}$	0.999817504
5	Exponential	$D_5 = 0.748398296e^{0.027738100\ D1}$	0.902649279

สรุปความสัมพันธ์

พบว่าลักษณะความสัมพันธ์เป็นแบบ Linear

อธิบาย เนื่องจากให้ค่า R^2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ในรูปสมการ polynomial เนื่องจากมีค่าสัมประสิทธิ์ที่ค่ายกกำลังระดับสองใกล้เคียง 0 และเมื่อเทียบสัมประสิทธิ์ของตัวแปรกำลังสองกับสัมประสิทธิ์ของตัวแปรกำลังหนึ่งพบว่ามีค่าน้อยมาก 0.004% หรือใกล้เคียง Linear

3.3.2.6 สรุปความสัมพันธ์ของข้อมูลเข้าและข้อมูลออกในแต่ละกระบวนการ

ความสัมพันธ์ของขนาดข้อมูลเข้า D_1 และผลลัพธ์ D_2 ในกระบวนการ P1 ความสัมพันธ์อยู่ในรูป Linear

ตารางที่ 3.17 สรุปความสัมพันธ์ของข้อมูลเข้าและข้อมูลออกในแต่ละกระบวนการ

ฟังก์ชัน	รูปแบบ	สัมประสิทธิ์	สมการความสัมพันธ์
$D_2 = g_1(D_1)$	Linear	0.7040	$D_2 = 0.704034577D_1 + 0.077475026$
$D_3 = g_2(D_2)$	Linear	0.3857	$D_3 = 0.385695981D_2 - 0.135568699$
$D_4 = g_3(D_3)$	Linear	0.7450	$D_4 = 0.745034590D_3 - 0.033689774$
$D_5 = g_4(D_4)$	Linear	0.3938	$D_5 = 0.393777445D_4 + 0.137443868$
$D_5 = g(D_1)$	Linear	0.0797	$D_5 = 0.079700381D_1 + 0.091912319$

สรุป ดังนั้นจากสมการความสัมพันธ์ข้างต้น อาจกล่าวได้ว่าความสัมพันธ์ของข้อมูลในกระบวนการต่าง มีลักษณะเป็น Linear โดย

กระบวนการ P1 ข้อมูลผลลัพธ์ D_2 มีขนาดประมาณ 70% ของ D_1

กระบวนการ P2 ขนาดข้อมูลผลลัพธ์ D_3 มีขนาดประมาณ 38% ของ D_2

กระบวนการ P3 ขนาดผลลัพธ์ D_4 มีขนาดประมาณ 74% ของ D_3

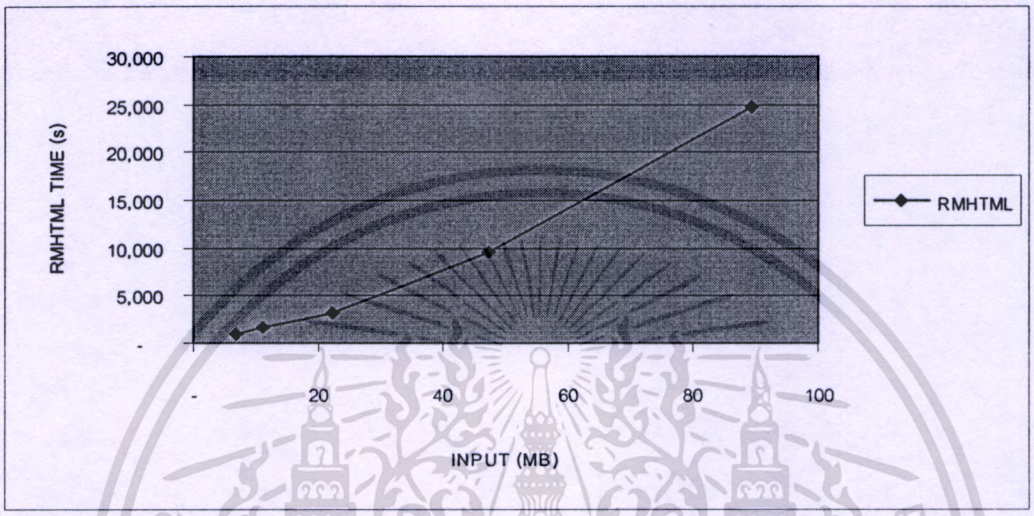
กระบวนการ P4 ขนาดผลลัพธ์ D_5 มีขนาดประมาณ 39% ของ D_4

ขนาดข้อมูลดัชนี มีขนาดโดยประมาณ 8% ของขนาดเว็บเพจ

3.3.2.7 ความสัมพันธ์ระหว่างเวลาและขนาดข้อมูลในกระบวนการ P1

ตารางที่ 3.18 แสดงผลการทดลองเวลา และขนาดข้อมูล INPUT ในการประมวลผล P1

INPUT (MB)	6.88	10.88	22.23	47.41	89.45
RMHTML (S)	929	1,622	3,202	9,490	24,759



รูปที่ 3.16 กราฟความสัมพันธ์ของเวลากับขนาดข้อมูล INPUT ในกระบวนการ P1

ตารางที่ 3.19 สมการความสัมพันธ์ของเวลากับขนาดข้อมูล INPUT ในกระบวนการ P1

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$T_{p1} = 289.02853077D_1 - 2222.58327167$	0.97963802
2	Logarithm	$T_{p1} = 8510.84994670Ln(D_1) - 18842.50793885$	0.80025177
3	Polynomial	$T_{p1} = 1.90586199\ D_1^2 + 105.14005934\ D_1 + 123.61445255$	0.9998300
4	Polynomial	$T_{p1} = -0.00900923\ D_1^3 + 3.17029353\ D_1^2 + 60.19183036\ D_1 + 459.03246445$	0.9999153
5	Polynomial	$T_{p1} = -0.00178255\ D_1^4 + 0.28588164\ D_1^3 - 11.66109116\ D_1^2 + 317.08786370\ D_1 - 790.39782854$	1.00000000000
6	Power	$T_{p1} = 75.59965882\ D_1^{1.26472307}$	0.99260781
7	Exponential	$T_{p1} = 1074.35764358e^{0.03773972\ D1}$	0.93818324

สรุปความสัมพันธ์

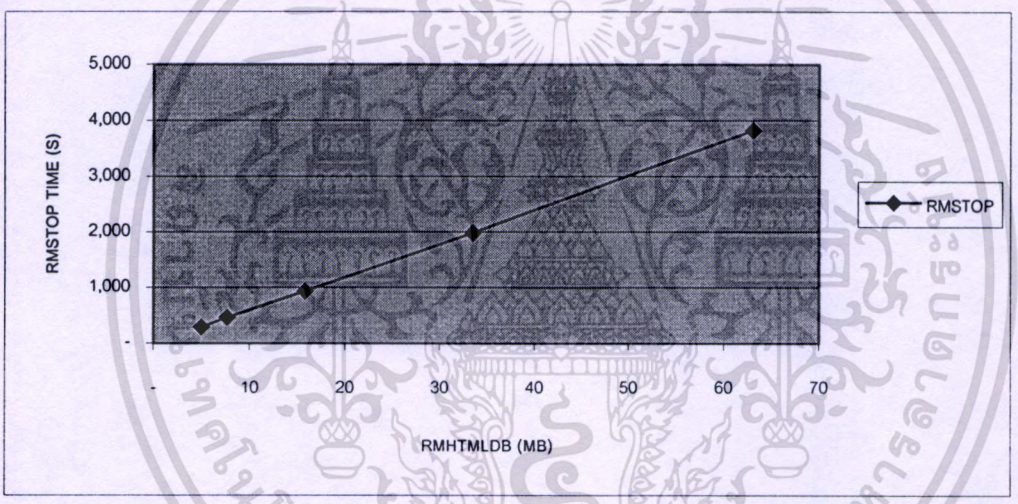
พบว่าลักษณะความสัมพันธ์เป็นแบบ Polynomial กำลัง 2

อธิบาย เนื่องจากให้ค่า R^2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ ในรูป สมการ polynomial กำลัง 3 เนื่องจากเมื่อขนาดข้อมูลเพิ่มขึ้นทำให้เวลามีค่าติดลบซึ่งเป็นไปไม่ได้

3.3.2.8 ความสัมพันธ์ของเวลา และขนาดข้อมูลในกระบวนการ P2

ตารางที่ 3.20 แสดงผลการทดลองเวลา และขนาดข้อมูล RMHTMLDB ในกระบวนการ P2

RMHTMLDB (MB)	4.97	7.63	15.78	33.47	63.04
RMSTOP (s)	290.21	465.65	934.30	1,984.29	3822.75



รูปที่ 3.17 กราฟความสัมพันธ์ของเวลา และข้อมูลในกระบวนการ P2

ตารางที่ 3.21 สมการความสัมพันธ์ของเวลา และข้อมูลในกระบวนการ P2

ลำดับที่	ชนิด	สมการความสัมพันธ์	R2
1	Linear	$T_{p2} = 60.642461533\ D_2 - 15.335625443$	0.999818929
2	Logarithm	$T_{p2} = 1322.600800887 Ln(D_2) - 216.902630896$	0.893333075
3	Polynomial	$T_{p2} = 0.048808030\ D_2^2 + 57.320929673\ D_2 + 14.665179132$	0.999971182
4	Polynomial	$T_{p2} = 0.001319999\ D_2^3 - 0.081942487\ D_2^2 +$	0.99998152

		60.606759794 D_2 - 2.736473167	
5	Polynomial	T_{p2} = -0.000511328 D_2^4 + 0.061122435 D_2^3 – 2.214002609 D_2^2 + 86.902960117 D_2 - 94.230859722	1.0000000001
6	Power	T_{p2} = 58.673558744 $D_2^{1.006139370}$	0.999737195
7	Exponential	T_{p2} = 352.453033140$e^{0.041403010\ D2}$	0.9012500

สรุปความสัมพันธ์

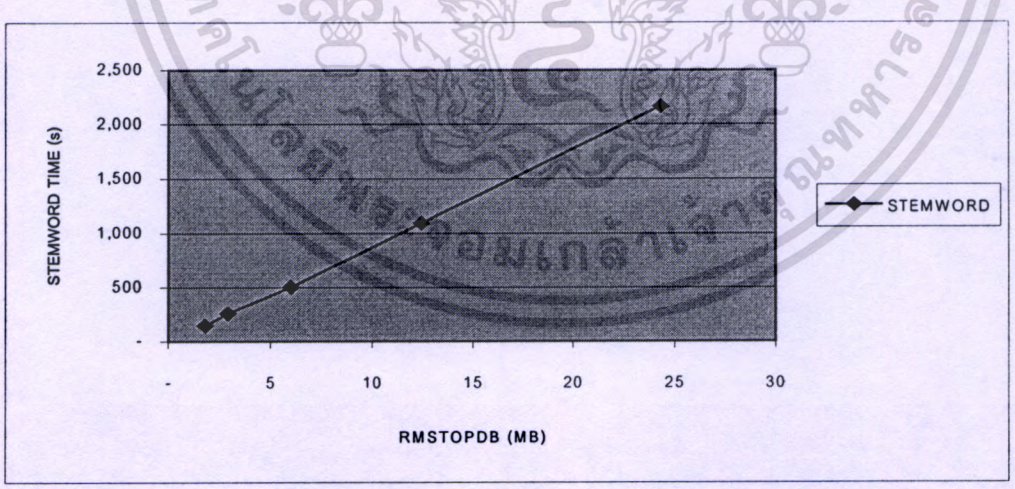
พบว่าลักษณะความสัมพันธ์เป็นแบบ Polynomial กำลัง 2

อธิบาย เนื่องจากให้ค่า R^2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ในรูป สมการ polynomial กำลัง 3 เนื่องจากให้ค่า R^2 ใกล้เคียงสมการ Polynomial กำลังสองแต่เมื่อ D_2 เท่ากับ 0 ให้ค่าเวลาติดลบซึ่งเป็นไปไม่ได้

3.3.2.9 ความสัมพันธ์ของเวลาและขนาดข้อมูลในกระบวนการ P3

ตารางที่ 3.22 แสดงผลการทดลองของเวลา และขนาดข้อมูล RMSTOPDB ในกระบวนการ P3

RMSTOPDB	1.80	2.92	6.00	12.45	24.32
STEMWORD	149.30	258.00	501.33	1,094.33	2165.92



รูปที่ 3.18 กราฟความสัมพันธ์ของเวลา และข้อมูลในกระบวนการ P3

ตารางที่ 3.23 สมการความสัมพันธ์ของเวลา และข้อมูลในกระบวนการ P3

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$T_{p3} = 89.549015425\ D_3 - 16.820378343$	0.99978475
2	Logarithm	$T_{p3} = 740.411393326 Ln(D_3) - 523.511985407$	0.88543467
3	Polynomial	$T_{p3} = 0.131117768\ D_3^2 + 86.102850155\ D_3 - 4.917426936$	0.99986050
4	Polynomial	$T_{p3} = -0.032530527\ D_3^3 + 1.358727157\ D_3^2 + 74.505928988\ D_3 + 17.989410723$	0.99991967
5	Polynomial	$T_{p3} = -0.026930433\ D_3^4 + 1.162976685\ D_3^3 - 14.663584176\ D_3^2 + 148.379457634\ D_3 - 77.186951731$	1.00000000
6	Power	$T_{p3} = 83.135346902\ D_3^{1.020353415}$	0.999392747
7	Exponential	$T_{p3} = 190.178449341 e^{0.109801872\ D3}$	0.893365096

สรุปความสัมพันธ์

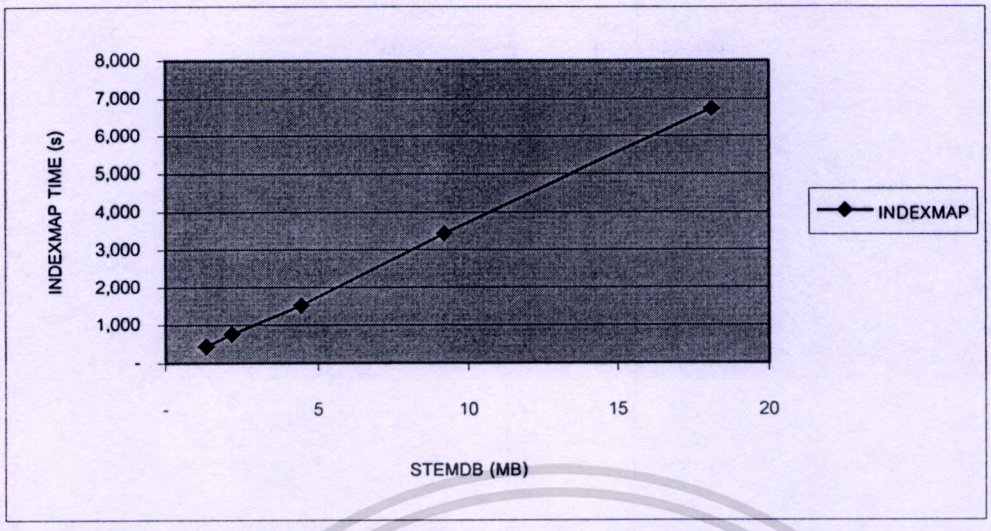
พบว่าความสัมพันธ์เป็นแบบ Polynomial กำลัง 2

อธิบาย เนื่องจากให้ค่า R^2 ใกล้เคียง 1 มากที่สุดและไม่เลือกความสัมพันธ์ในรูปสมการ polynomial กำลัง 3 เนื่องจากเมื่อ D_3 มีขนาดใหญ่ขึ้นเวลาในการประมวลผลติดลบ

3.3.2.10 ความสัมพันธ์ของเวลาและขนาดข้อมูลในกระบวนการ P4

ตารางที่ 3.24 แสดงผลการทดลองของเวลาและขนาดข้อมูล STEMDB ในกระบวนการ P4

STEMDB	1.32	2.16	4.43	9.18	18.11
INDEXMAP	437.10	749.96	1,526.28	3,423.09	6730.35



รูปที่ 3.19 กราฟความสัมพันธ์ของเวลา และข้อมูลในกระบวนการ P4

ตารางที่ 3.25 สมการความสัมพันธ์ของเวลาและข้อมูลในกระบวนการ P4

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$T_{p4} = 376.312019805\ D_4 - 77.057558566$	0.999757078
2	Logarithm	$T_{p4} = 2313.258671627 Ln(D_4) - 967.959741050$	0.888195436
3	Polynomial	$T_{p4} = -0.148873658\ D_4^2 + 379.226313576\ D_4 - 84.525394438$	0.999760142
4	Polynomial	$T_{p4} = -0.467600571\ D_4^3 + 12.953376902\ D_4^2 + 287.629452942\ D_4 + 49.056107306$	0.999968171
5	Polynomial	$T_{p4} = -0.174447355\ D_4^4 + 5.273857073\ D_4^3 - 43.930827977\ D_4^2 + 481.136093365\ D_4 - 134.521402395$	1.0000000002
6	Power	$T_{p4} = 328.881402374\ D_4^{1.045800423}$	0.999685321
7	Exponential	$T_{p4} = 562.784252198e^{0.151039892\ D4}$	0.886925597

สรุปความสัมพันธ์

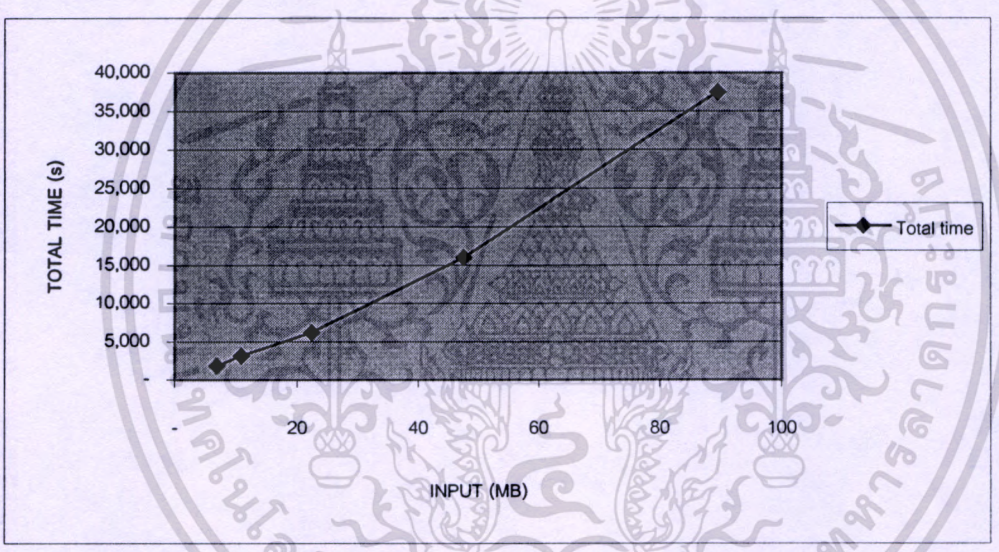
พบว่ารูปแบบความสัมพันธ์เป็นแบบ Linear

อธิบาย เนื่องจากให้ค่า R^2 ใกล้เคียง 1 มากที่สุดและไม่เลือกความสัมพันธ์ในรูปสมการ polynomial กำลัง 2 เนื่องเมื่อ D_4 มีค่ามากให้ค่าเวลาติดลบ

3.3.2.11 ความสัมพันธ์ของเวลาและขนาดข้อมูลในกระบวนการรวม DISE

ตารางที่ 3.26 แสดงผลการทดลองเวลาและขนาดข้อมูล STEMDB ในกระบวนการรวม DISE

STEMDB	1.32	2.16	4.43	9.18	18.11
INDEXMAP	437.10	749.96	1,526.28	3,423.09	6730.35



รูปที่ 3.20 กราฟความสัมพันธ์ของเวลาและข้อมูลในกระบวนการรวม DISE

ตารางที่ 3.27 สมการความสัมพันธ์ของข้อมูลในกระบวนการรวม DISE

ลำดับที่	ชนิด	สมการความสัมพันธ์	R^2
1	Linear	$T_{total\ DISE}$ = 432.181247970 D_1 - 2379.243776655	0.989608293
2	Logarithm	$T_{total\ DISE}$ = 12905.142980378Ln(D_1) - 27795.269234093	0.831292197
3	Polynomial	$T_{total\ DISE}$ = 2.024484894 D_1^2 + 236.847360192	0.9999020

		D_1 + 112.983818696	
4	Polynomial	$T_{total\ DISE}$ = -0.008886078 D_1^3 + 3.271631597 D_1^2 + 192.513575311 D_1 + 443.816652858	0.9999395
5	Polynomial	$T_{total\ DISE}$ = -0.00224138288 D_1^4 + 0.36190950455 D_1^3 - 15.37734107482 D_1^2 + 515.53446697436 D_1 - 1127.21611881840	0.99999999998
6	Power	$T_{total\ DISE}$ = 182.906058704 $D_1^{1.168640312}$	0.99652711
7	Exponential	$T_{total\ DISE}$ = 2146.206246276$e^{0.034587368\ D1}$	0.926543586

สรุปความสัมพันธ์

พบว่ารูปแบบความสัมพันธ์เป็นแบบ Polynomial กำลัง 2

อธิบาย เนื่องจากให้ค่า R2 ใกล้เคียง 1 มากที่สุด และไม่เลือกความสัมพันธ์ ในรูป สมการ polynomial กำลัง 3 เนื่องจากให้ค่าเวลาในการประมวลผลติดลบเมื่อขนาดข้อมูลเพิ่มขึ้น

3.3.2.12 สรุปความสัมพันธ์ของเวลาในการประมวลผลและข้อมูลแต่ละกระบวนการ

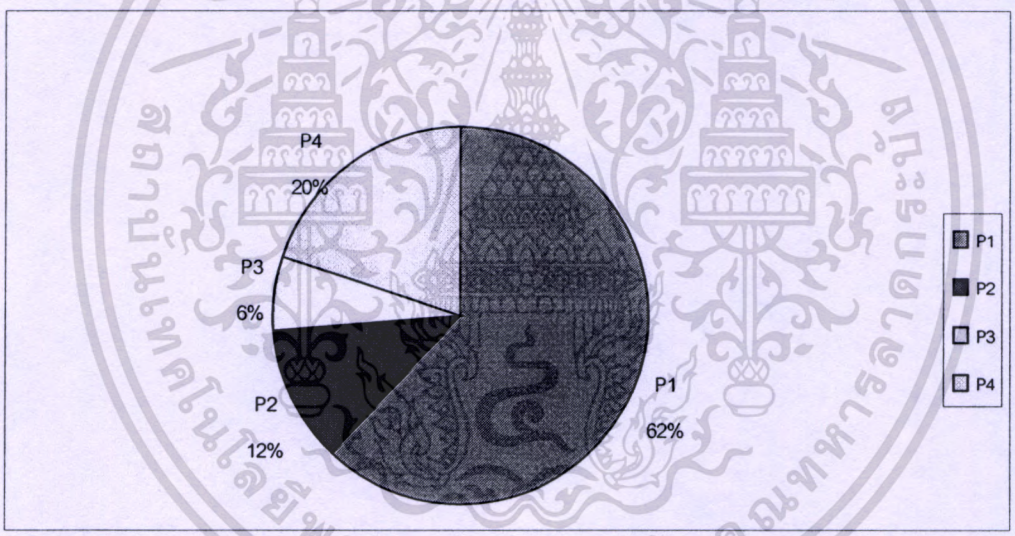
ตารางที่ 3.28 สรุปความสัมพันธ์ของเวลาและข้อมูลในแต่ละกระบวนการ

ฟังก์ชัน	รูปแบบ	สมการความสัมพันธ์
$T_{P1}(D_1)$	Polynomial	T_{p1} = 1.90586199 D_1^2 + 105.14005934 D_1 + 123.61445255
$T_{P2}(D_2)$	Polynomial	T_{p2} = 0.048808030 D_2^2 + 57.320929673 D_2 + 14.665179132
$T_{P3}(D_3)$	Polynomial	T_{p3} = 0.131117768 D_3^2 + 86.102850155 D_3 - 4.917426936
$T_{P4}(D_4)$	Linear	y = 376.312019805x - 77.057558566
$T_{total\ DISE}$ (D_1)	Polynomial	y = 2.024484894x^2 + 236.847360192x + 112.983818696

สรุป จากสมการความสัมพันธ์ข้างต้น อาจกล่าวได้ว่าความสัมพันธ์ของเวลาในการประมวลผลต่อขนาดข้อมูลในกระบวนการต่าง เพิ่มขึ้นเมื่อขนาดข้อมูลเพิ่มขึ้นอย่างมีความสัมพันธ์กัน โดยมีลักษณะเป็น Linear หรือ polynomial กำลัง 2

3.4 การกระจายการทำดัชนี

จากการทดลองพบว่าเวลาในการทำดัชนีเพิ่มขึ้นเมื่อ ขนาดข้อมูลเข้ามีขนาดเพิ่มขึ้นโดยเพิ่มขึ้นเป็นสัดส่วนโดยตรงกับขนาดข้อมูลในช่วงข้อมูลที่ทดสอบมีขนาดเล็ก และคาดว่าจะเพิ่มขึ้นอย่างมากเมื่อขนาดข้อมูลที่จะทำดัชนีมีขนาดใหญ่ขึ้น จากตารางที่ 3.6 เมื่อพิจารณาการกระจายตัวของเวลาในกระบวนการต่างๆ พบว่าเวลามีการกระจายดังรูป 3..21 โดยในกระบวนการ P1 มีการใช้เวลาเป็นสัดส่วน 62% ของเวลา กระบวนการ P2 ใช้เวลาเป็นสัดส่วน 12% P3 ใช้เวลาเป็น 6% และ P4 ใช้เวลาเป็น 20% ของเวลาทั้งหมดในการทำดัชนี



รูปที่ 3.21 การกระจายตัวของเวลาในแต่ละกระบวนการ

ดังนั้นในการกระจายกระบวนการทำดัชนีที่ i ออกไปเป็นจำนวน n เวลาในการประมวลผลในกระบวนการที่ i จะลดลง เนื่องจากเวลาในการประมวลผลแต่ละกระบวนการมีการทำงานขนานกันไป และขนาดข้อมูลไม่เท่ากันฉะนั้นเวลาในการประมวลผลที่กระบวนการที่ i จะเป็นเวลามากที่สุดที่กระบวนการที่กระจายออกไปใช้ ซึ่งสามารถแสดงได้ดังสมการ 3.10

$$T_{Rpi} = MAX\,(Tp_{i,1}, Tp_{i,2}, Tp_{i,3}, \ldots\ Tp_{i,n}) \qquad 3.10$$

T_{Rpi} คือเวลาในกระบวนการที่ i เมื่อกระจายออกไปจำนวน n

$Tp_{i,n}$ คือเวลาที่ใช้ในกระบวนการที่ i ตัวที่ n

n คือจำนวนกระบวนการที่กระจายออกไป

เพราะฉะนั้น เมื่อทำการะกระจาย Tp_i ออกไปเป็นจำนวน n โฮสต์ เวลารวมของกระบวนการในการทำดัชนี เมื่อ $MAX\,(Tp_{i,n})$ คือเวลามากที่สุดในการประมวลผลกระบวนการ i ในจำนวน n กระบวนการที่กระจายออกไป ฉะนั้นเวลารวมของกระบวนการในการทำดัชนีอาจหาได้จากสมการ

$$Tp = MAX\,(Tp_{1,n}) + MAX\,(Tp_{2,m}) + MAX\,(Tp_{3,o}) + Tp_4 \qquad 3.11$$

เมื่อ n คือจำนวนกระบวนการที่กระจายออกไปของกระบวนการที่ 1

m คือจำนวนกระบวนการที่กระจายออกไปของกระบวนการที่ 2

o คือจำนวนกระบวนการที่กระจายออกไปของกระบวนการที่ 3

เมื่อพิจารณาขนาดของข้อมูลผลลัพธ์ในแต่ละกระบวนการ โดยในกระบวนการ P1 ผลลัพธ์เป็นอัตราส่วน 0.70 ของข้อมูล INPUT และผลลัพธ์ในกระบวนการ P2 มีขนาดเป็นอัตราส่วน 0.27 ของ INPUT ผลลัพธ์ในกระบวนการ P3 มีขนาด 0.20 ของ INPUT และผลลัพธ์ในกระบวนการ P4 มีขนาด 0.08 ของ INPUT ดังแสดงในตาราง 3.29

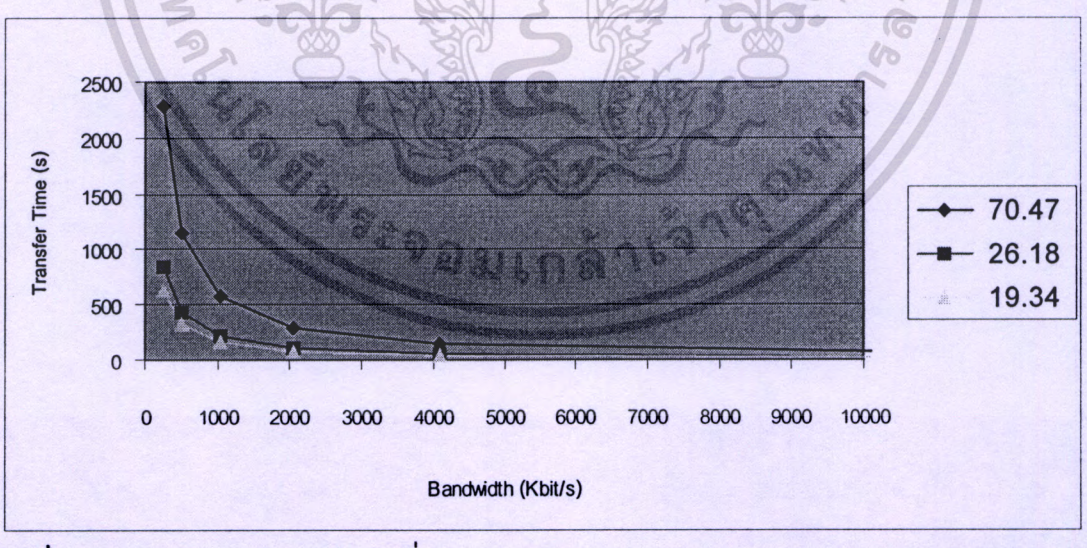
ตารางที่ 3.29 ขนาดของผลลัพธ์ในกระบวนการต่างเทียบกับขนาดของ INPUT

ผลลัพธ์ในการบวนการ	ขนาดของผลลัพธ์เทียบกับ INPUT	ขนาดของผลลัพธ์ที่ลดลงเทียบกับ INPUT
RMHTMLDB	0.70	0.30
RMSTOPDB	0.27	0.73
STEMDB	0.20	0.80
INDEXDB	0.08	0.92

ดังนั้นในการกระจายกระบวนการ RMHTML ออกไป ข้อมูลที่ส่งให้กระบวนการถัดไปจะเท่ากับ 0.7 ของ Input ของข้อมูลเดิม หรือข้อมูลที่ส่งจาก RMHTML ไปยังกระบวนการ RMSTOP มีขนาดลดลงเป็นอัตรา 0.3 ของเว็บเพจ และในกระบวนการ RMSTOP ขนาดข้อมูลที่ส่งจาก RMSTOP จะมีขนาดเท่ากับ 0.27 ของขนาดข้อมูลเว็บเพจ ถ้าหากกระบวนการ RMHTML และ RMSTOP ถูกระจายไปอยู่บนเครื่องเดียวกัน ข้อมูลที่ลดลงจะมีขนาดเท่ากับข้อมูลผลลัพธ์ของกระบวนการ RMSTOP และถ้าหากระบวนการ STEMDB มีการกระจายและอยู่บนเครื่องเดียวกับ RMHTML และ RMSTOP ปริมาณข้อมูลที่ส่งผ่านเครือข่ายไปยัง กระบวนการ INDEXMAP จะมีขนาดเท่ากับ 0.2 ของข้อมูลเว็บเพจหรือลดลง 80% แต่ในกรณีที่ กระบวนการต่างๆ มีการกระจายและมีการทำงานอยู่บนโฮสต์ คนละโฮสต์ ปริมาณข้อมูลที่ส่งผ่านเครือข่ายทั้งหมด จะเท่ากับผลรวมของขนาดข้อมูล RMHTMLDB, RMSTOPDB, STEMDB รวมกันซึ่งอาจจะมีขนาดมากกว่าขนาดข้อมูลเว็บเพจเดิมดังนั้นรูปแบบการกระจายที่เหมาะสมจึงเป็นเรื่องสำคัญ

3.5 Bandwidth และเวลาในการทำดัชนี

เมื่อพิจารณาเกี่ยวกับเวลาในการทำดัชนี และขนาดของ Bandwidth ที่เชื่อมต่อผู้วิจัยได้จัดการทดลองเพื่อดูแนวโน้มของเวลาที่ใช้ในการรับ-ส่งข้อมูล โดยจัดการทดลองโดยการส่งข้อมูลขนาด 70, 26, 19 MB ผ่านเครือข่ายที่มี bandwidth ขนาด 256, 512, 1024, 2048, 4096, 10240 Mbit/s ตามลำดับซึ่งจากการทดลองได้ผลดังนี้



รูปที่ 3.22 เวลาในการรับ-ส่งข้อมูลที่ Bandwidth 256, 512, 1024, 2048, 4096, 10240 Mbit/s

ซึ่งจากการทดลองจะเห็นได้ว่าเมื่อขนาดของ Bandwidth เพิ่มขึ้นเวลาที่ใช้ในการรับ-ส่ง ข้อมูลลดลงอย่างมากดังนั้นในการกระจายการทำดัชนีเมื่อมีการกระจายกระบวนการในการทำดัชนีออกไป เวลาที่ใช้ในการรับ-ส่งข้อมูลลดลงซึ่งทำให้เวลาที่ใช้ในการทำดัชนีโดยรวมลดลงด้วย

3.6 รูปแบบในการกระจายการทำดัชนี

เมื่อทำการกระจายกระบวนการในการทำดัชนีออกไป และมีส่วนที่ทำหน้าที่ในส่วนกลางในการรวมผลลัพธ์ดังนั้นรูปแบบการกระจายที่เป็นไปได้ในการกระจายการทำดัชนีเป็นอย่างไร ในการศึกษาครั้งนี้เนื่องจากมีการแบ่งกระบวนการในการทำดัชนีออกเป็นสี่กระบวนการย่อย ได้แก่ RMHTML RMSTOP STEMWORD และ INDEXMAP โดยกระบวนการ INDEXMAP ซึ่งเป็นกระบวนการซึ่งทำหน้าที่รวบรวมผลลัพธ์จะไม่ถูกกระจายออกไป

รูปแบบในการกระจายกระบวนการทำดัชนี สามารถแบ่งออกเป็นรูปแบบหลักได้สองรูปแบบ รูปแบบแรกได้แก่ กระจายกระบวนการทำดัชนีไปยังโฮสต์ เพื่อทำการประมวลผล แล้วจึงส่งผลลัพธ์กลับสู่ส่วนกลางเพื่อรวบรวมผลลัพธ์ที่ส่วนกลางเท่านั้น รูปแบบที่สองได้แก่ กระจายกระบวนการโดยแบ่งกระบวนการย่อย โดยมีกระบวนการรวมผลลัพธ์ของกระบวนการย่อยจากกระบวนการก่อนหน้า จากนั้นจึงส่งผลลัพธ์ที่มีการรวมแล้วส่งไปยังกระบวนการถัดไป โดยมีรูปแบบการกระจายเป็นแบบลำดับชั้น

3.7 ผลการทดลองและวิเคราะห์

เมื่อมีการกระจายการกระบวนการย่อยในการทำดัชนีออกไป ภาระงานในการทำดัชนีข้อมูลของเครื่องจักรสืบค้นบนอินเตอร์เน็ตจะถูกกระจายออกไป เวลาในการประมวลผลจึงลดลงเนื่องจากการทำงานในกระบวนการย่อยที่ มีการทำงานคู่ขนานกันไป โดยเวลาที่ลดลงมีปัจจัยขึ้นอยู่กับขนาดข้อมูลที่แต่ละกระบวนการรับผิดชอบ ซึ่งถ้าหากขนาดข้อมูลที่รับผิดชอบมีการกระจายอย่างเท่ากัน เวลาในการทำงานจะลดลงโดยเป็นสัดส่วนแปรผกผันกับจำนวนกระบวนการที่กระจายออกไป แต่ถ้าหากปริมาณข้อมูลที่รับผิดชอบโดยกระบวนการมีการกระจายไม่เท่ากัน เวลาในการประมวลผลในกระบวนการจะมีค่า เท่ากับค่ามากที่สุดของเวลาในการประมวลผลของกระบวนการที่กระจายออกไป

ในส่วนของปริมาณข้อมูลที่ส่งผ่านเครือข่าย ปริมาณข้อมูลที่ลดลงในแต่ละกระบวนการ มีผลต่อปริมาณข้อมูลที่ส่งผ่านเครือข่าย โดยถ้าหากกระบวนการที่กระจายออกไป อยู่บนเครื่องเดียวกันมากเท่าไร ปริมาณข้อมูลที่ส่งผ่านเครือข่ายจะลดลงได้มากขึ้นเท่านั้น และถ้าหากกระบวน

การย่อย ในแต่ละขั้นตอนมีการแยกส่วนอยู่บนต่างโฮสต์มากขึ้นปริมาณข้อมูลที่ส่งผ่านเครือข่ายจะมากขึ้น และอาจจะมากกว่าปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำดัชนีของ OSE

ดังนั้นผู้วิจัยจึงคาดว่าการกระจายกระบวนการในการทำดัชนีที่ให้ประสิทธิภาพดีที่สุดน่าจะเป็นกระบวนการที่มีการกระจายกระบวนการย่อย RMHTML, RMSTOP, STEMWORD อยู่บนโฮสต์เดียวกัน โดยกระจายออกไปเป็นจำนวน n และมีกระบวนการในการรวมและสร้างดัชนีที่ส่วนกลางเพื่อคอยรวบรวมผลลัพธ์จากกระบวนการที่กระจายออกไป

บทที่ 4
การทดลองเปรียบเทียบ

ในบทนี้จะกล่าวถึงการทดลองที่จัดขึ้นเพื่อทดลองเปรียบเทียบเพื่อหารูปแบบการกระจายการทำดัชนีที่ดีที่สุดโดยดูจากเวลาที่ใช้ในการทำดัชนีและปริมาณข้อมูลที่ส่งผ่านเครือข่ายซึ่งน้อยที่สุด โดยผู้วิจัยได้จัดการทดลองขึ้นจำนวน 7 การทดลองโดยมีรูปแบบการกระจายการทำดัชนีที่แตกต่างกัน โดยการทดลองที่ 1 เป็นการทดลองการทำงานของ OSE เพื่อใช้เป็นอ้างอิงเปรียบเทียบกับเวลา และปริมาณข้อมูลที่ส่งผ่านเครือข่ายของ OSE และการทดลองที่ 2 ถึงการทดลองที่ 7 เป็นการทดลองเพื่อดูค่าเวลาและปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำดัชนีของ DISE รูปแบบต่างๆ

4.1 สภาพแวดล้อมในการทดลอง และรูปแบบการทดลอง

ในการทดลองทางผู้วิจัยได้ทำการจัดสภาพแวดล้อมพื้นฐานในการทดลองได้แก่เครื่องคอมพิวเตอร์จำนวน 4 เครื่องโดยมีคุณสมบัติดังนี้

หน่วยประมวลผลกลาง ได้แก่ Cyrix II ความเร็ว 350 MHz

หน่วยความจำ 128 MB

เครือข่ายภายในความเร็วสูงสุด 10 Mbit/s

ติดตั้งระบบปฏิบัติการ Free BSD 4.8

เว็บเซิร์ฟเวอร์ Apache 1.3

และ Linux sun jdk 1.3.1 ซึ่งจำเป็นสำหรับติดตั้งและใช้งานเครื่องจักรสืบค้น BDDBot

การเตรียมข้อมูลชุดข้อมูลที่ใช้ในการวิจัยได้แก่เว็บเพจซึ่งเป็นหนังสืออิเล็คทรอนิคส์ ในรูปแบบของ HTML ไฟล์ และทำการจัดชุดข้อมูลเป็นขนาดโดยประมาณ 20, 50, 100, 200, 400, 800 เมกะไบต์ ซึ่งขนาดของเว็บไซต์เป็นขนาดของไฟล์ซึ่งเป็น HTML รูปภาพและไฟล์ชนิดอื่นๆ ประกอบกัน

ในการออกแบบการทดลองเราได้ทำการวิเคราะห์รูปแบบการกระจายการทำดัชนีออกเป็นสองแบบใหญ่ได้แก่ หนึ่งกระจายการทำดัชนีโดยกระบวนการทำดัชนีที่กระจายออกไปจะส่งผลลัพธ์สู่ส่วนกลางโดยตรงซึ่งได้แก่รูปแบบในการทดลองที่ 2, 3, และ 4 รูปแบบที่สองได้แก่ กระจายกระบวนการทำดัชนีออก เป็นลำดับขั้นโดยกระบวนการก่อนหน้าจะส่งผลลัพธ์สู่กระบวนการถัดไปเพื่อรวบรวมผลลัพธ์ เป็นลำดับขั้น การทดลองที่ 5, 6, 7 เพื่อหาเวลาที่ใช้ในการทำดัชนีของรูปแบบ

การกระจายแบบต่างให้ครอบคลุมโดยมีการทดลองที่ 1 ซึ่งเป็นการทดลองการทำงานแบบ OSE เพื่อใช้เป็นส่วนสำหรับอ้างอิงเปรียบเทียบ

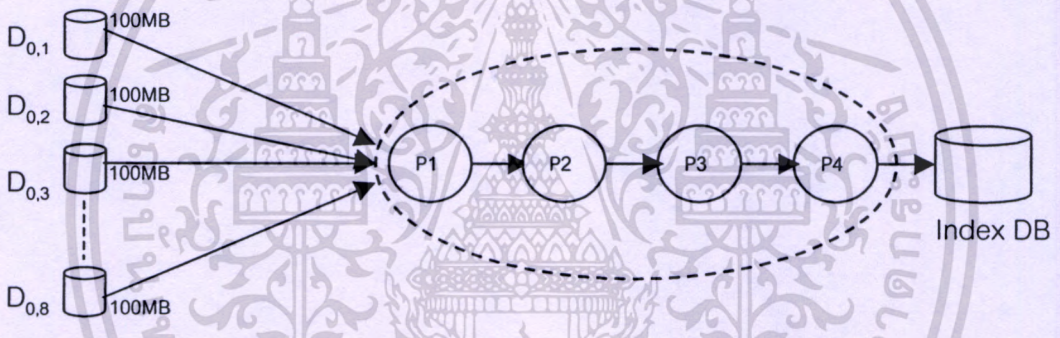
4.2 การทดลอง

4.2.1 การทดลองที่ 1 การทำงานของ OSE

วัตถุประสงค์ เพื่อหาเวลาที่ใช้ในการทำดัชนีและปริมาณข้อมูลที่ส่งผ่านเครือข่าย ในการทำดัชนีเว็บเพจขนาด 800 MB เพื่อใช้เป็นเวลาอ้างอิงของกระบวนการทำดัชนีแบบ OSE

4.2.1.1 วิธีการทดลอง

ตั้งเว็บไซต์จำนวน 8 ไซต์โดย แต่ละเว็บไซต์มีขนาดโดยประมาณ 100 MB และทำการทำดัชนีข้อมูลของเว็บเพจทั้ง 8 ไซต์โดยใช้การทำงานแบบ OSE รูปแบบในการทดลองสามารถแสดงได้ดังรูปที่ 4.1



รูปที่ 4.1 แผนภาพการทดลองที่ 1

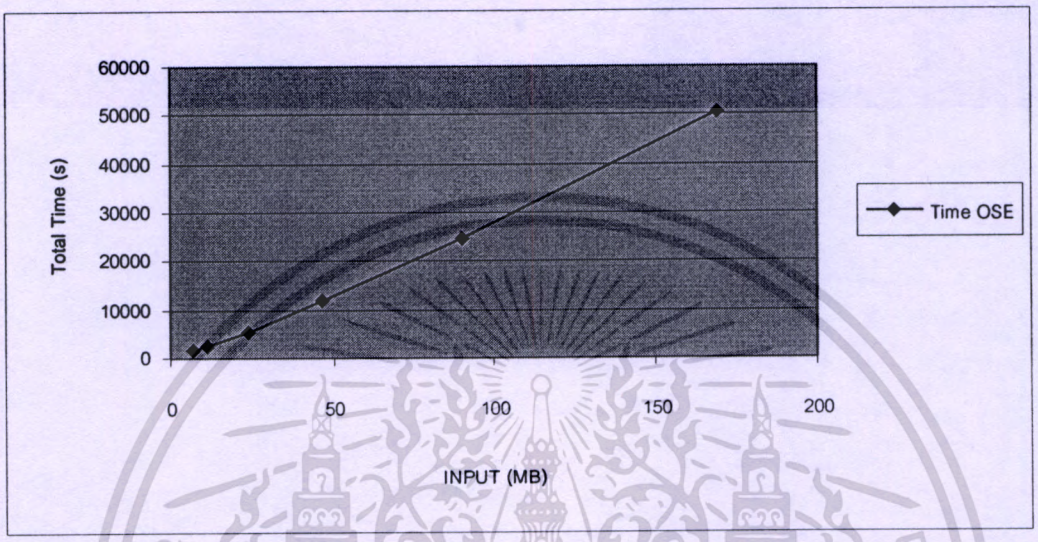
4.2.1.2 ผลการทดลองที่ 1

ตารางที่ 4.1 ขนาดข้อมูลและเวลาที่ใช้ในกระบวนการทำดัชนี

ขนาดข้อมูลเว็บเพจ (MB)	ผลการทดลอง	ผลจากการคำนวณ	เปอร์เซ็นต์ความต่าง
INPUT (MB)	169.53		
INDEXDB (MB)	13.86	13.60	-1.91%
เวลาในการทำดัชนี (s)	50,571	55,448	-8.80%

4.2.1.3 สรุปผลการทดลอง

ปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 1 มีขนาดเท่ากับ 169.53 MB และเมื่อแสดงเป็นกราฟความสัมพันธ์ระหว่างเวลาการทำงานและขนาดของข้อมูล INPUT โดยต่อเนื่องจากการทดลองเบื้องต้น จะแสดงได้ดังในรูปที่ 4.2



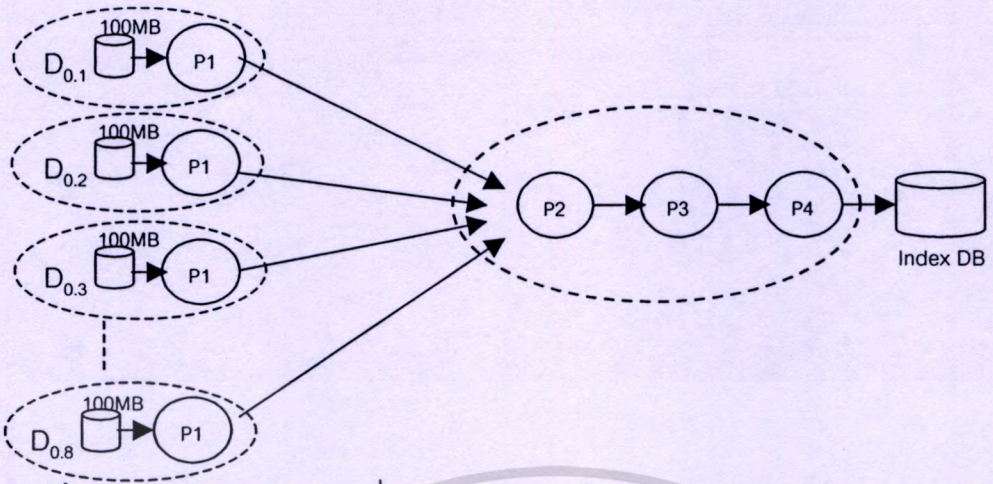
รูปที่ 4.2 เวลารวมที่ใช้ในการทำดัชนี ของ OSE

4.2.2 การทดลองที่ 2 DISE แบบกระจาย P1

วัตถุประสงค์การทดลอง เพื่อหาเวลาการทำงานของ DISE ที่มีกระบวนการย่อย RMHTML กระจายออกไป และหาปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำงานรูปแบบดังกล่าว

4.2.2.1 วิธีการทดลอง

ตั้งเว็บไซต์จำนวน 8 ไซต์แต่ละเว็บไซต์มีขนาดประมาณ 100 MB และทำการทำดัชนีข้อมูลของเว็บเพจทั้ง 8 ไซต์โดยใช้การทำงานแบบ DISE โดยมีการกระจายกระบวนการ RMHTML ออกไปอยู่บนโฮสต์ในเครื่องเดียวกับเว็บไซต์ โดยรูปแบบการทดลองสามารถแสดงได้ดังรูปที่ 4.3



รูปที่ 4.3 แผนภาพการทดลองที่ 2

4.2.2.2 ผลการทดลอง

ตารางที่ 4.2 ขนาดของ RMHTMLDB เมื่อผ่านกระบวนการ RMHTML

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
INPUT (MB)	30.98	18.73	21.56	25.67	16.54	22.24	15.28	18.54
RMHTMLDB (MB)	22.02	13.29	15.26	17.47	12.47	15.91	10.94	13.28

ตารางที่ 4.3 ผลการทดลองเพื่อหาเวลาที่ใช้ในกกระบวนการ P1 ของชุดข้อมูลต่างๆ

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
Total time P1 (s)	5,247	3,985	4,668	4,583	3,223	4,590	2,960	4,076

ตารางที่ 4.4 เวลาที่ใช้ในการประมวลผล P2-P4

กระบวนการ	เวลา (s)
Merge RMHTMLDB	163
RMSTOP (P2)	7,628
Merge RMSTOPDB	0
STEMWORD (P3)	4,084
Merge STEMDB	0
INDEXMAP (P4)	8,497
เวลาที่ใช้ในกระบวนการ P2-P4	21,215

4.2.2.3 สรุปผลการทดลองที่ 2

ในการทดลองที่ 2 ปริมาณข้อมูลที่ส่งผ่านเครือข่ายในระหว่างการทดลองมีขนาดเท่ากับ 120.64 MB เมื่อเทียบกับการทำงานของ OSE พบว่าลดลงเท่ากับ 29 % ส่วนเวลาในการทำดัชนีเว็บเพจใช้เวลารวมทั้งหมดเท่ากับ 26,462 วินาที เมื่อเปรียบเทียบกับการทำงานของ OSE พบว่าเท่ากับ O.52 ของเวลาการทำงานของ OSE หรือลดลง 48% ซึ่งกราฟแสดงเปรียบเทียบเวลาการทำงานของ OSE และเวลาที่ใช้ในการทดลองที่ 2

ตารางที่ 4.5 สรุปปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 2

ปริมาณข้อมูลที่ส่งผ่านเครือข่าย OSE	169.53
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายการทดลองที่ 2	120.64
ลดลง	29%

ตารางที่ 4.6 ตารางสรุปเวลาที่ใช้ในการทำดัชนี

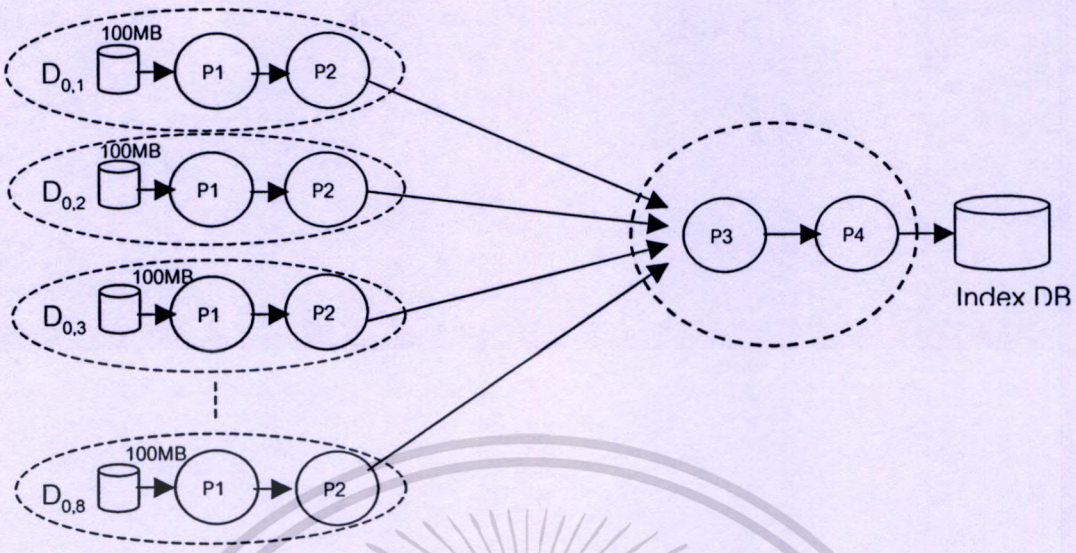
กระบวนการ	เวลาจากการทดลอง (s)	เวลาจากการคำนวณ (s)	เปอร์เซ็นต์ความต่าง
เวลามากที่สุดของกระบวนการ P1 (s)	5,247	5,204	- 0.82%
เวลาที่ใช้ในกระบวนการ P2-P4 (s)	21,215	24,842	+17.10%
เวลารวมในการทำดัชนี (s)	26,462	30,055	13.58%
อัตราส่วนเทียบกับ OSE	0.52	-	

4.2.3 การทดลองที่ 3 DISE แบบกระจาย P1 และ P2

วัตถุประสงค์การทดลอง เพื่อหาเวลาการทำงานของ DISE ที่มีกระบวนการย่อย RMHTML และ RMSTOP กระจายออกไปอยู่บนเครื่องเดียวกัน และหาปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำงานรูปแบบดังกล่าว

4.2.3.1 วิธีการทดลอง

ตั้งเว็บไซต์จำนวน 8 ไซต์โดย แต่ละเว็บไซต์มีขนาดประมาณ 100 MB และทำการทำดัชนีข้อมูลของเว็บเพจทั้ง 8 ไซต์โดยใช้การทำงานแบบ DISE โดยมีการกระจายกระบวนการ RMHTML และ RMSTOP ออกไปโดยทั้งสองกระบวนการอยู่บนเครื่องเดียวกัน และอยู่บนโฮสต์เดียวกับเว็บไซต์ โดยรูปแบบในการทดลองสามารถแสดงได้ดังรูปที่ 4.4



รูปที่ 4.4 แผนภาพการทดลองที่ 3

4.2.3.2 ผลการทดลอง

ตารางที่ 4.7 ขนาดข้อมูลเมื่อผ่านกระบวนการ RMHTML และ RMSTOP

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
INPUT (MB)	30.98	18.73	21.56	25.67	16.54	22.24	15.28	18.54
RMSTOPDB (MB)	7.77	4.68	6.37	6.51	4.55	6.10	4.17	5.08

ตารางที่ 4.8 เวลาในการประมวลผล P1-P2 ของ กระบวนการที่กระจายออกไป

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
Total time P1-P2 (s)	6,562	4,795	5,638	5,768	3,958	5,551	3,592	4,727

ตารางที่ 4.9 เวลาในการประมวลผล P3-P4

กระบวนการ	เวลา (s)
MERGE RMSTOP	49
STEMWORD (P3)	4,084
MERGE STEM	0
INDEXMAP (P4)	8,497
Total Time P3-4	13,297

4.2.3.3 สรุปผลการทดลอง

ในการทดลองที่ 3 ปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 3 เท่ากับ 45.24 เมื่อเปรียบเทียบกับขนาดข้อมูลที่ส่งผ่านเครือข่ายในการทำงานของ OSE พบว่า ลดลงจาก 169.53 เท่ากับ 73%

ตารางที่ 4.10 สรุปผลปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 3

ปริมาณข้อมูลที่ส่งผ่านเครือข่าย OSE (MB)	169.53
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายการทดลองที่ 3 (MB)	45.24
ลดลง	73%

เมื่อเปรียบเทียบเวลาในการทำดัชนีพบว่าเวลาในการทำดัชนีของการทดลองที่ 3 เท่ากับ 19,859 วินาทีลดลงจากเวลาในการทำดัชนีแบบ OSE ซึ่งใช้เวลา 50,571 วินาที เป็นอัตราส่วนเท่ากับ 0.39 ของเวลาประมวลผลแบบ OSE

ตารางที่ 4.11 ตารางสรุปผลเวลาในการทำดัชนี

กระบวนการ	เวลาจากการทดลอง (s)	เวลาจากการคำนวณ (s)	เปอร์เซ็นต์ความต่าง
เวลามากที่สุดใน P1-P2 (s)	6,562	5,916	-9.84%
เวลาใน P3-P4 (s)	13,297	16,830	+26.5%
เวลารวมที่ใช้ในการทำดัชนี (s)	19,859	22,746	+14.50%
อัตราส่วนเทียบกับ OSE	0.39	-	

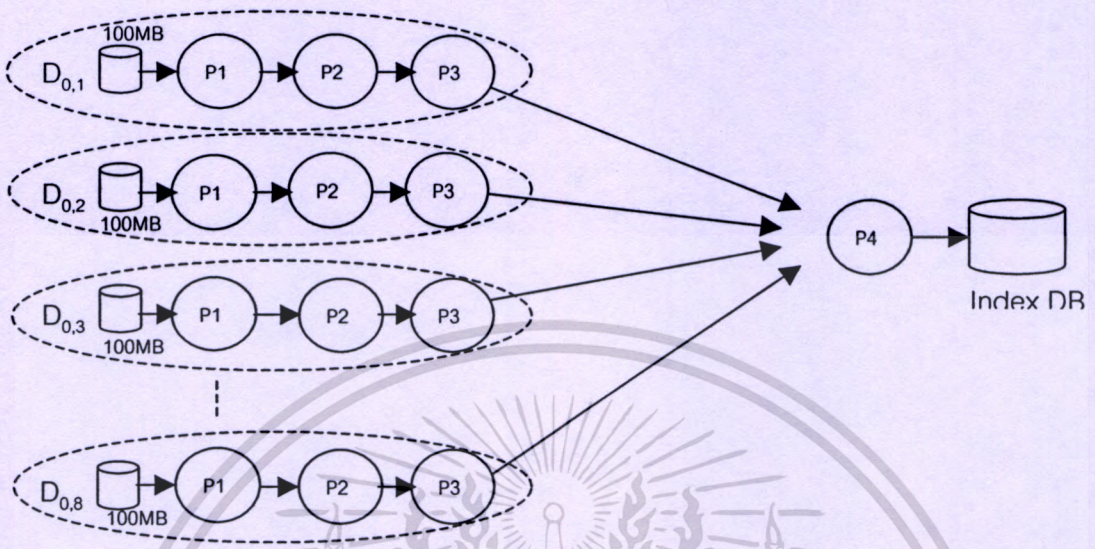
4.2.4 การทดลองที่ 4 DISE แบบกระจาย P1 P2 และ P3

วัตถุประสงค์การทดลอง เพื่อหาเวลาการทำงานของ DISE ที่มีกระบวนการย่อย RMHTML RMSTOP และ STEMWORD กระจายออกไปโดยทั้งสามกระบวนการอยู่บนเครื่องเดียวกัน และหาปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำงานรูปแบบดังกล่าว

4.2.4.1 วิธีการทดลอง

ตั้งเว็บไซต์จำนวน 8 ไซต์โดยแต่ละเว็บไซต์มีขนาดประมาณ 100 MB และทำการทำดัชนีข้อมูลของเว็บเพจทั้ง 8 ไซต์โดยใช้การทำงานแบบ DISE โดยมีการกระจายกระบวนการ

RMHTML และ RMSTOP ออกไปโดยทั้งสองกระบวนการอยู่บนเครื่องเดียวกัน และอยู่บนโฮสต์เดียวกับเว็บไซต์ โดยรูปแบบในการทดลองสามารถแสดงได้ดังรูปที่ 4.5



รูปที่ 4.5 แผนภาพการทดลองที่ 4

4.2.4.2 ผลการทดลอง

ตารางที่ 4.12 ขนาดข้อมูลเมื่อผ่านกระบวนการ RMHTML RMSTOP และ STEMWORD

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
INPUT (MB)	30.98	18.73	21.56	25.67	16.54	22.24	15.28	18.54
STEMDB (MB)	5.67	3.46	4.82	4.81	3.32	4.50	3.08	3.75

ตารางที่ 4.13 เวลาในกระบวนการ P1 - P3

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
Total Time P1-P3 (s)	7,226	5,198	6,187	6,381	4,337	6,132	3,927	5,102

ตารางที่ 4.14 เวลาในกระบวนการ P4

กระบวนการ	เวลา (s)
MERGE STEM	36
INDEXMAP (P4)	8,537
Total Time P4	9,230

4.2.4.3 สรุปผลการทดลอง

ในการทดลองที่ 4 ปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 4 เท่ากับ 33.40 MB เมื่อเทียบกับปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำงานของ OSE จาก 169.53 MB พบว่ามีปริมาณลดลง 80%

ตารางที่ 4.15 สรุปผลการทดลองปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 4

ปริมาณข้อมูลที่ส่งผ่านเครือข่าย OSE (MB)	169.53
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายการทดลองที่ 4 (MB)	33.40
ลดลง	80%

เมื่อพิจารณาเวลาที่ใช้ในการทดลองที่ 4 เวลารวมทั้งกระบวนการในการทำดัชนีเท่ากับ 16,456 วินาที เมื่อเทียบกับเวลาในการทำดัชนีแบบ OSE ใช้เวลาเท่ากับ 50,571 วินาที พบว่าลดลงเป็นอัตราส่วนเท่ากับ 0.33 รูปที่ 4.8 แสดงกราฟเปรียบเทียบเวลาที่ใช้ในกระบวนการทำดัชนีแบบ OSE และเวลาที่ใช้ในการทดลองที่ 4

ตารางที่ 4.16 ตารางสรุปผลเวลาในการทำดัชนี

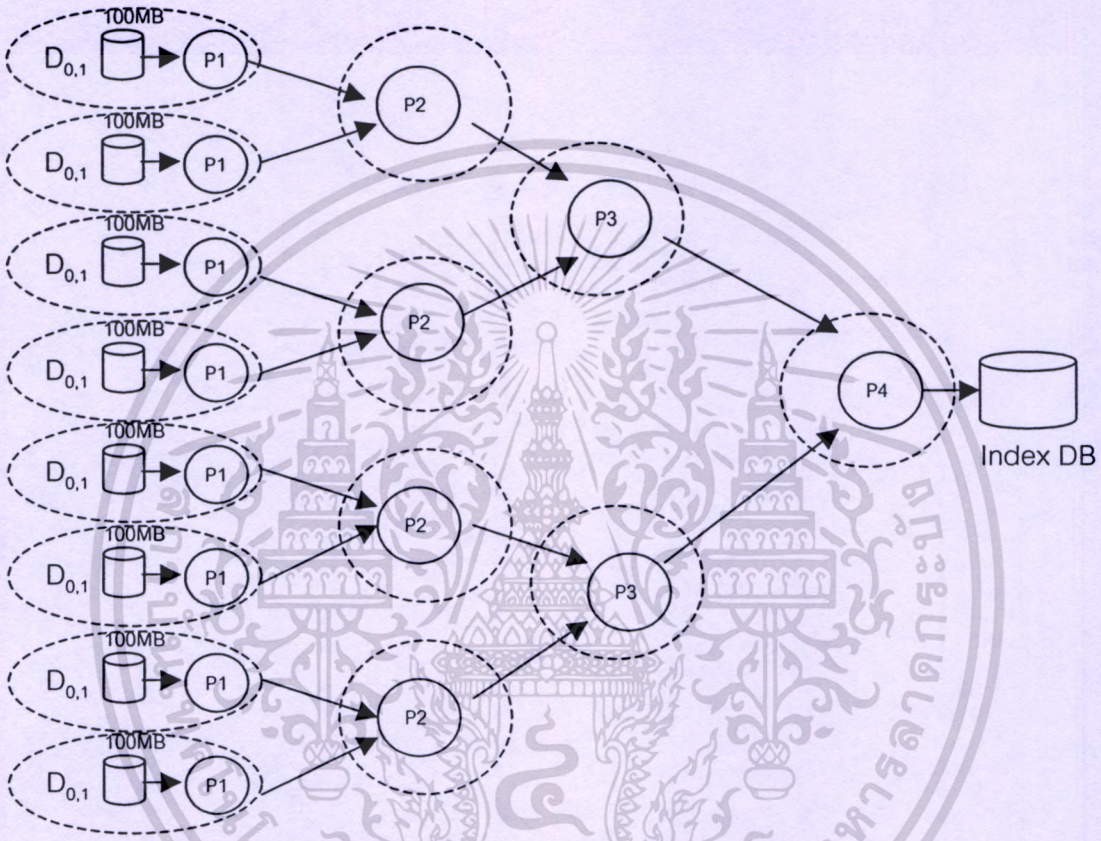
กระบวนการ	เวลาจากการทดลอง (s)	เวลาจากการคำนวณ (s)	เปอร์เซ็นต์ความต่าง
เวลามากที่สุดใน P1-P3 (s)	7,226	6,569	-9.09%
เวลาใน P4 (s)	9,230	12,483	+35.2%
เวลารวมที่ใช้ในการทำดัชนี (s)	16,456	19,709	+19.0%
อัตราส่วนเทียบกับ OSE	0.33		

4.2.5 การทดลองที่ 5 DISE แบบ Hierarchy

วัตถุประสงค์การทดลอง เพื่อหาเวลาการทำงานของ DISE ที่มีการกระจายกระบวนการย่อยเป็นลำดับขั้น โดยทั้งสามกระบวนการย่อยที่กระจายออกไปอยู่บนโฮสต์คนละเครื่อง และศึกษาปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำงานรูปแบบดังกล่าว

4.2.5.1 วิธีการทดลอง

ตั้งเว็บไซต์จำนวน 8 ไซต์โดย แต่ละเว็บไซต์มีประมาณ 100 MB และทำการทำดัชนีข้อมูลของเว็บเพจทั้ง 8 ไซต์โดยใช้การทำงานแบบ DISE โดยมีการกระจายกระบวนการ RMHTML อยู่ที่เดียวกับเว็บไซต์ และมีการกระจายกระบวนการ RMSTOP และ STEMWORD ออกเป็นลำดับขั้นโดยกระบวนการย่อยอยู่บนโฮสต์ที่แยกต่างหากกัน รูปแบบในการทดลองแสดงได้ดังรูปที่ 4.6



รูปที่ 4.6 แผนภาพการทดลองที่ 5

4.2.5.2 ผลการทดลอง

ตารางที่ 4.17 ขนาดข้อมูลในกระบวนการ P1

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
INPUT (MB)	30.98	18.73	21.56	25.67	16.54	22.24	15.28	18.54
RMHTMLDB (MB)	22.02	13.29	15.26	17.47	12.47	15.91	10.94	13.28

ตารางที่ 4.18 ขนาดข้อมูลในกระบวนการ P2

Process time	ข้อมูล (1,4)(1)	ข้อมูล (2,3)(2)	ข้อมูล (5,6)(3)	ข้อมูล (7,8)(4)
RMSTOPDB (MB)	14.28	11.06	10.66	9.25

ตารางที่ 4.19 ขนาดข้อมูลในกระบวนการ P3

Process time	(1)+(2) =(5)	(3)+(4) =(6)
STEMDB (MB)	18.75	14.65

ตารางที่ 4.20 เวลาในการประมวลผลกระบวนการ P1

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
Total Time of P1 (s)	5,247	3,985	4,668	4,583	3,223	4,590	2,960	4,076

ตารางที่ 4.21 เวลาในการประมวลผลกระบวนการ P2

Process time	ข้อมูล (1,4)(1)	ข้อมูล (2,3)(2)	ข้อมูล (5,6)(3)	ข้อมูล (7,8)(4)
Merge RMHTMLDB	8	7	6	6
RMSTOP (P2)	2,432	1,855	1,981	1,625
Total Time of P2	2,441	1,862	1,988	1,631

ตารางที่ 4.22 เวลาในการประมวลผลกระบวนการ P3

Process time	(1)+(2) (5)	(3)+(4) (6)
MERGE RMSTOP	6	5
STEM (P3)	2,362	2,016
Total Time of P3	2,368	2,021

ตารางที่ 4.23 เวลาในการประมวลผล P4

กระบวนการ	เวลา (s)
MERGE STEM	8.22
INDEXMAP (P4)	10,156
Total Time P4	11,007

4.2.5.3 สรุปผลการทดลอง

ในการทดลองที่ 5 ปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 5 เท่ากับ 199.28 MB เมื่อเทียบกับปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำดัชนีข้อมูลของ OSE มีขนาด 169.53 MB พบว่าเพิ่มขึ้น 18%

ตารางที่ 4.24 ปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 5

กระบวนการ	ปริมาณข้อมูล (MB)
ปริมาณข้อมูลที่ส่งผ่านเครือข่าย OSE	169.53
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายกระบวนการ P1 ไป P2	120.64
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายกระบวนการ P2 ไป P3	45.24
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายกระบวนการ P3 ไป P4	33.4
รวมปริมาณข้อมูลส่งผ่านเครือข่าย	199.28
เพิ่มขึ้น	18%

เมื่อพิจารณาเวลาการทำดัชนีในการทดลองที่ 5 ใช้เวลาในการทำดัชนีเท่ากับ 21,063 วินาทีเมื่อเทียบกับเวลาที่ใช้ในการทำดัชนีของ OSE เท่ากับ 50,571 วินาทีพบว่าลดลงเท่ากับ 0.42 ของเวลาในการทำดัชนีของ OSE รูปที่ 4.10 เวลาในการทดลองที่ 5 เทียบกับเวลา OSE

ตารางที่ 4.25 เวลาในการทำดัชนีข้อมูล

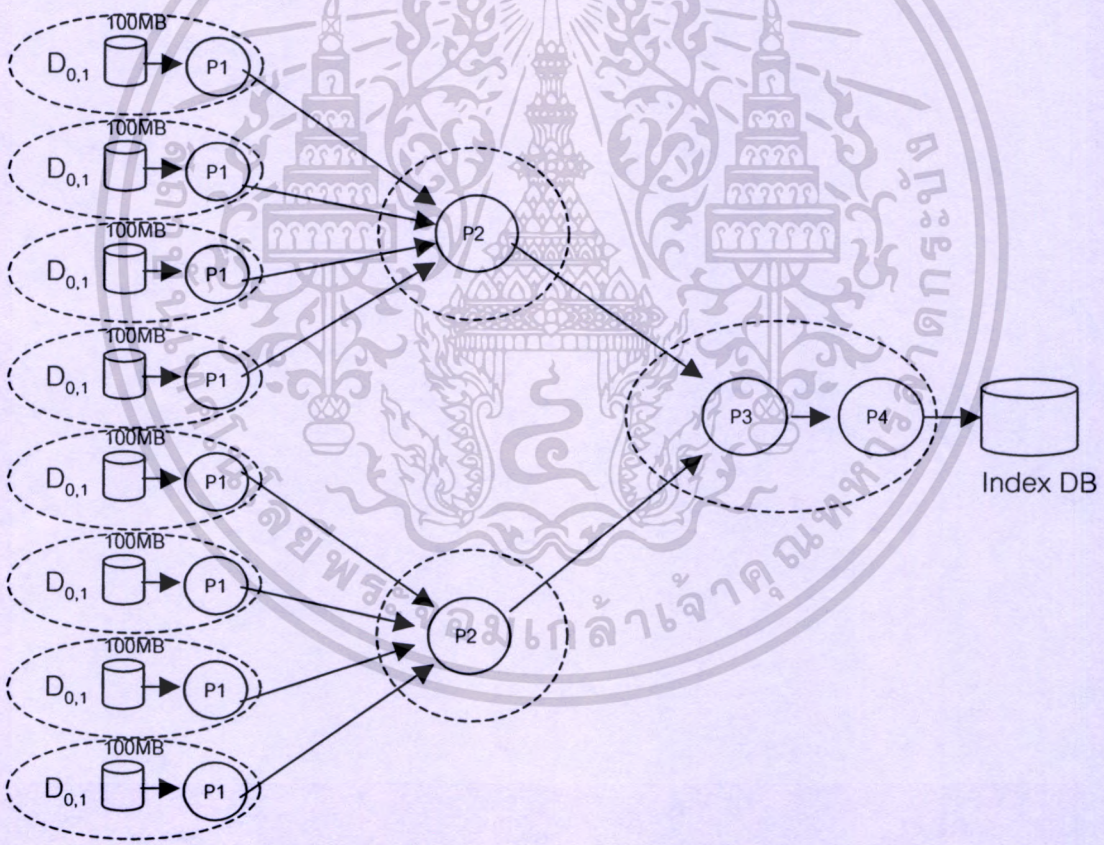
กระบวนการ	เวลาจากการทดลอง (s)	เวลาจากการคำนวณ (s)	เปอร์เซ็นต์ความต่าง
เวลามากที่สุดใน P1 (s)	5,247	5,204	-0.82%
เวลามากที่สุดใน P2 (s)	2,441	2,236	-0.22%
เวลามากที่สุดใน P3 (s)	2,368	2,260	-4.56%
เวลาใน P4 (s)	11,007	12,491	+13.48%
เวลารวมที่ใช้ในการทำดัชนี	21,063	-	+5.35%
อัตราส่วนเทียบกับ OSE	0.42	-	-

4.2.6 การทดลองที่ 6 DISE แบบ Hierarchy P3, P4 อยู่ในส่วนกลาง

วัตถุประสงค์การทดลอง เพื่อหาเวลาการทำงานของ DISE ที่มีการกระจายกระบวนการย่อยเป็นลำดับขั้น โดยทั้งสามกระบวนการย่อยที่กระจายออกไปอยู่บนโฮสต์คนละเครื่อง และศึกษาปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำงานรูปแบบดังกล่าว

4.2.6.1 วิธีการทดลอง

ตั้งเว็บไซต์จำนวน 8 ไซต์โดยแต่ละเว็บไซต์มีขนาดประมาณ 100 MB และทำการทำดัชนีข้อมูลของเว็บเพจทั้ง 8 ไซต์โดยใช้การทำงานแบบ DISE โดยมีการกระจายกระบวนการ RMHTML อยู่ที่เดียวกับเว็บไซต์ และมีการกระจายกระบวนการ RMSTOP และ STEMWORD ออกเป็นลำดับขั้น โดยกระบวนการย่อยอยู่บนโฮสต์ที่แยกต่างหากกันรูปแบบในการทดลองสามารถแสดงได้ดังรูปที่ 4.7



รูปที่ 4.7 แผนภูมิภาพการทดลองที่ 6

4.2.6.2 ผลการทดลอง

ตารางที่ 4.26 ขนาดข้อมูลเมื่อผ่านกระบวนการ RMHMTL

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
INPUT (MB)	30.98	18.73	21.56	25.67	16.54	22.24	15.28	18.54
RMHTMLDB (MB)	22.02	13.29	15.26	17.47	12.47	15.91	10.94	13.28

ตารางที่ 4.27 ขนาดข้อมูลเมื่อผ่านกระบวนการ RMSTOP

Data size	ชุดที่ 1+2+3+4 (1)	ชุดที่ 5+6+7+8 (2)
RMSTOPDB (MB)	25.34	19.91

ตารางที่ 4.28 เวลาในการประมวลผล P1

ข้อมูลชุดที่	ชุดที่ 1	ชุดที่ 2	ชุดที่ 3	ชุดที่ 4	ชุดที่ 5	ชุดที่ 6	ชุดที่ 7	ชุดที่ 8
Total of P1 (s)	5,247	3,985	4,668	4,583	3,223	4,590	2,960	4,076

ตารางที่ 4.29 เวลาในการประมวลผล P2

กระบวนการ	ชุดที่ 1+2+3+4 (1)	ชุดที่ 5+6+7+8 (2)
Merge RMHTMLDB	39,558	29,438
RMSTOP	4,250,931	3,711,802
Total Time of P2	4,290,679	3,741,470

ตารางที่ 4.30 เวลาในการประมวลผล P3 - P4

กระบวนการ	เวลา (s)
MERGE RMSTOP	10,715
STEM	4,555,475
MERGE RMSTOP	5
STEM	10,140,982
Total Time of P3-4	15,504,782

4.2.6.3 สรุปผลการทดลอง

ในการทดลองที่ 6 ปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 6 เท่ากับ 165.88 MB เมื่อเทียบกับปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำดัชนีข้อมูลของ OSE เท่ากับ 169.53 MB พบว่าลดลงเท่ากับ 2.2%

ตารางที่ 4.31 สรุปผลปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 6

กระบวนการ	ขนาดข้อมูล (MB)
ปริมาณข้อมูลที่ส่งผ่านเครือข่าย OSE	169.53
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายกระบวนการ P1 ไป P2	120.64
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายกระบวนการ P2 ไป P3	45.24
รวมปริมาณข้อมูลส่งผ่านเครือข่าย	165.88
ลดลง	2.2%

เมื่อพิจารณาเวลาการทำดัชนีการทดลองที่ 6 ใช้เวลาในการทำดัชนี 25,042 วินาที เมื่อเทียบกับเวลาในการทำดัชนีของ OSE เท่ากับ 50,571 วินาที พบว่าลดลงโดยเทียบกับเวลาในการทำดัชนีของ OSE เท่ากับ 0.5 ของเวลาในการทำงานของ OSE

ตารางที่ 4.32 สรุปผลเวลาในการทำดัชนีในการทดลองที่ 6

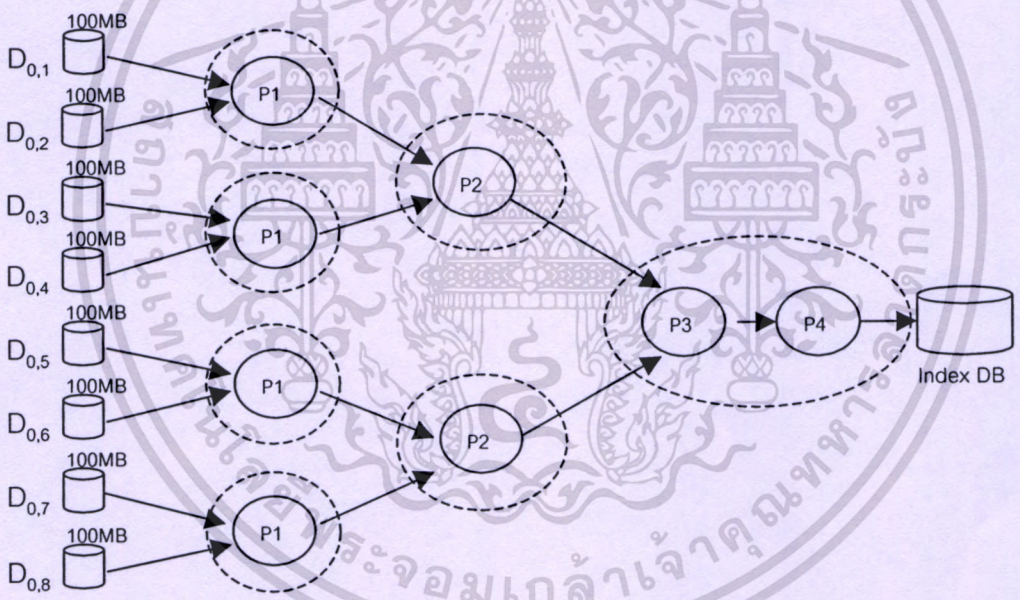
กระบวนการ	เวลาจากการทดลอง (s)	เวลาจากการคำนวณ (s)	เปอร์เซ็นต์ความต่าง
เวลามากที่สุดใน P1 (s)	5,247	5,204	-0.82%
เวลามากที่สุดใน P2 (s)	4,291	4,253	-0.88%
เวลามากที่สุดใน P3-P4 (s)	15,505	16,765	+8.13%
เวลารวมที่ใช้ในการทำดัชนี	25,042	26,222	+0.003%
อัตราส่วนเทียบกับ OSE	0.50	-	-

4.2.7 การทดลองที่ 7 DISE แบบ Hierarchy P1 ไม่อยู่บนเว็บเซิร์ฟเวอร์และ P3, P4 อยู่ส่วนกลาง

วัตถุประสงค์การทดลอง เพื่อหาเวลาการทำงานของ DISE ที่มีการกระจายกระบวนการย่อยเป็นลำดับขั้น โดยทั้งสามกระบวนการย่อยที่กระจายออกไปอยู่บนโฮสต์คนละเครื่อง และศึกษาปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำงานรูปแบบดังกล่าว

4.2.7.1 วิธีการทดลอง

ตั้งเว็บไซต์จำนวน 8 ไซต์โดย แต่ละเว็บไซต์มีขนาดประมาณ 100 MB และทำการทำดัชนีข้อมูลของเว็บเพจทั้ง 8 ไซต์โดยใช้การทำงานแบบ DISE โดยมีการกระจายกระบวนการ RMHTML อยู่ที่เดียวกับเว็บไซต์ และมีการกระจายกระบวนการ RMSTOP และ STEMWORD ออกเป็นลำดับขั้น โดยกระบวนการย่อยอยู่บนโฮสต์ที่ แยกต่างหากกัน รูปแบบในการทดลองสามารถแสดงได้ดังรูปที่ 4.8



รูปที่ 4.8 แผนภูมิภาพการทดลองที่ 7

4.2.7.2 ผลการทดลอง

ตารางที่ 4.33 ตารางแสดงขนาดข้อมูลที่กระบวนการ P1

ข้อมูลชุดที่	ชุดที่ 1+4 (1)	ชุดที่ 2+3 (2)	ชุดที่ 5+6 (3)	ชุดที่ 7+8 (4)
INPUT (MB)	56.65	40.29	38.77	33.82
RMHTMLDB (MB)	39.48	28.55	28.38	24.22

ตารางที่ 4.34 ตารางแสดงขนาดข้อมูลที่กระบวนการ P2

ข้อมูลชุดที่	ชุดที่ (1)+(2)	ชุดที่ (3)+(4)
RMSTOPDB (MB)	25.34	19.91

ตารางที่ 4.35 ตารางแสดงเวลาในการประมวลผลที่ P1

ข้อมูลชุดที่	ชุดที่ 1+4 (1)	ชุดที่ 2+3 (2)	ชุดที่ 5+6 (3)	ชุดที่ 7+8 (4)
Total Time of P1 (s)	9,829	7,208	9,258	7,542

ตารางที่ 4.36 ตารางเวลาในการประมวลผลที่ P2

ข้อมูลชุดที่	ชุดที่ (1)+(2)	ชุดที่ (3)+(4)
Merge RMHTMLDB	20	14
RMSTOP (P2)	4,267	3,792
Total Time P2 (s)	4,287	3,807

ตารางที่ 4.37 ตารางเวลาในการประมวลผล P3-P4

กระบวนการ	เวลา (s)
MERGE RMSTOP	11
STEM (P3)	4,157
MERGE STEM	0
INDEXMAP (P4)	8,583
Total Time P3-4	13,412

4.2.7.3 สรุปผลการทดลอง

ปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 5 ปริมาณข้อมูลทั้งหมดที่ส่งผ่านเครือข่ายเท่ากับ 335.41 MB เมื่อเทียบกับปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทำดัชนีข้อมูลของ OSE เท่ากับ 169.53 MB พบว่าเพิ่มขึ้น 98%

เมื่อพิจารณาเวลาในการประมวลผลในการทดลองที่ 7 เวลาที่ใช้เท่ากับ 27,528 วินาทีเมื่อเทียบกับเวลาในการทำดัชนีของ OSE พบว่าลดลง เมื่อเทียบกับเวลาของ OSE เป็นอัตราส่วน 0.54 ของเวลาที่ใช้ใน OSE

ตารางที่ 4.38 สรุปปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลองที่ 7

กระบวนการ	ขนาดข้อมูล (MB)
ปริมาณข้อมูลที่ส่งผ่านเครือข่าย OSE	169.53
ปริมาณข้อมูลเข้าสู่ P1	169.53
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายกระบวนการ P1 ไป P2	120.64
ปริมาณข้อมูลที่ส่งผ่านเครือข่ายกระบวนการ P2 ไป P3	45.24
รวมปริมาณข้อมูลส่งผ่านเครือข่าย	335.41
เพิ่มขึ้น	0.98

ตารางที่ 4.39 สรุปเวลาในการทำดัชนีในการทดลองที่ 7

กระบวนการ	เวลาจากการทดลอง (s)	เวลาจากการคำนวณ (s)	เปอร์เซ็นต์ความต่าง
เวลามากที่สุดใน P1 (s)	9,829	12,053	22.6%
เวลามากที่สุดใน P2 (s)	4,287	4,135	-3.55%
เวลามากที่สุดใน P3-P4 (s)	13,412	16,525	+23.2%
เวลารวมที่ใช้ในการทำดัชนี	27,528	32,713	18.8%
อัตราส่วนเทียบกับ OSE	0.54	-	-

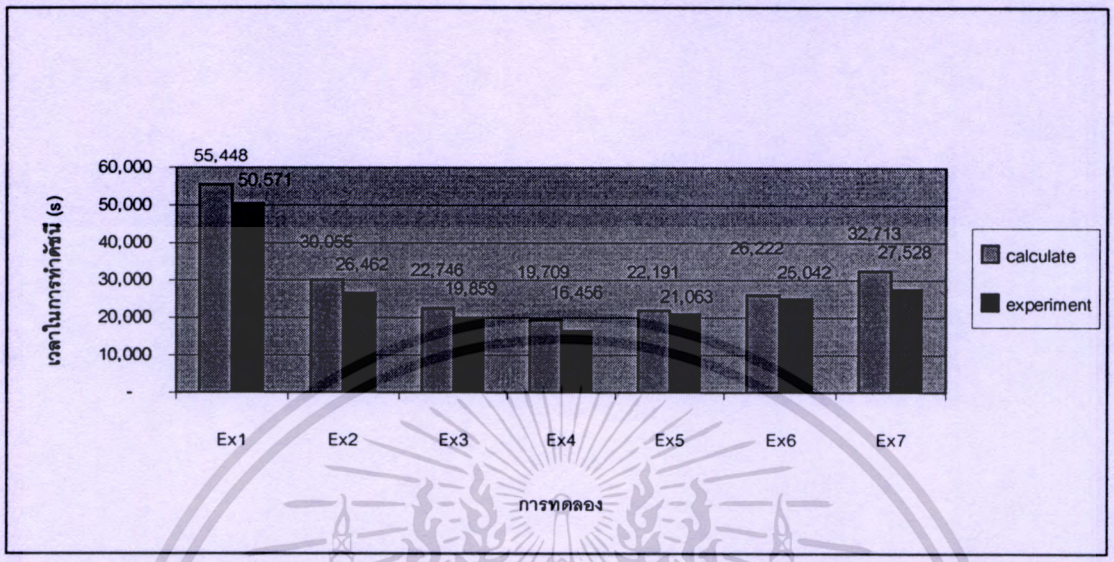
4.3 สรุปผลการทดลอง

ผลการทดลองที่จัดขึ้นเพื่อเป็นการทดลองเพื่อหารูปแบบการกระจายกาทำดัชนีเว็บเพจโดยที่ใช้เวลา และปริมาณข้อมูลที่ส่งผ่านเครือข่ายน้อยที่สุด

ในการทดลองที่ 1 เป็นการทดลองที่ทำเพื่อใช้อ้างอิงเพื่อเปรียบเทียบทั้งในเรื่องเวลาและปริมาณการสื่อสารข้อมูลของ OSE การทดลองที่ 2 ถึงการทดลองที่ 7 เป็นการทดลองโดยใช้การทำงานแบบ DISE เพื่อหารูปแบบการกระจายการทำดัชนีที่เหมาะสม ซึ่งจะใช้เวลาในการทำดัชนีและปริมาณการสื่อสารข้อมูลที่น้อยที่สุด

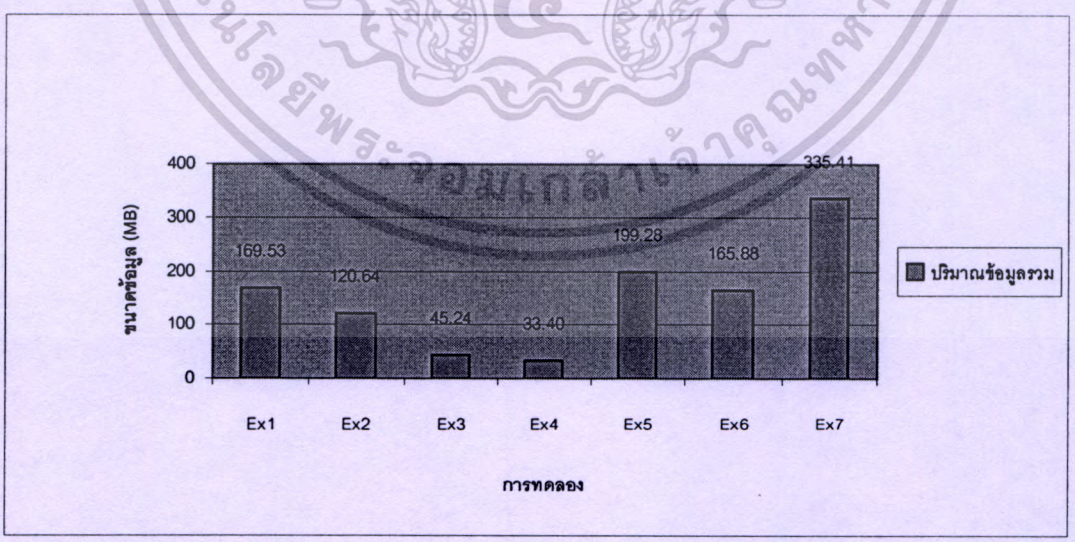
จากการทดลองสรุปได้ดังนี้ รูปแบบการกระจายการทำดัชนีที่ใช้เวลาในการทำดัชนีน้อยที่สุดได้แก่รูปแบบในการทดลองที่ 4 เนื่องจากมีกระบวนการที่ทำงานขนานกันไปเป็นจำนวนมากที่สุด จากการวิเคราะห์เบื้องต้นในบทที่ 3 เวลาในการประมวลผลที่ลดลงขึ้นกับจำนวนกระบวนการ

ที่กระจายออกไปโดยในรูปที่ 4.9 แสดงผลของเวลาที่ใช้ในการทดลองต่างๆ เปรียบเทียบกัน และเทียบกับเวลาในการทำดัชนีที่ได้จากการคำนวณ



รูปที่ 4.9 เปรียบเทียบเวลาในการทำดัชนีในการทดลอง

ส่วนปริมาณข้อมูลที่ส่งผ่านเครือข่าย รูปแบบการกระจายการทำดัชนีที่มีการส่งข้อมูลผ่านเครือข่ายน้อยที่สุดได้แก่รูปแบบที่ 4 เช่นกันแต่ในการทดลองที่ 5 ถึงการทดลองที่ 7 พบว่าปริมาณข้อมูลที่ส่งผ่านเครือข่ายลดลงน้อยมากหรือเพิ่มขึ้นเช่นในการทดลองที่ 7 ปริมาณข้อมูลที่ส่งผ่านเครือข่ายเพิ่มขึ้นถึง 98% หรือเกือบเท่าตัว



รูปที่ 4.10 เปรียบเทียบปริมาณข้อมูลที่ส่งผ่านเครือข่ายในการทดลอง

ดังนั้นอาจสรุปได้ว่ารูปแบบการกระจายกระบวนการในการทำดัชนีในรูปแบบการทดลองที่ 4 น่าจะเป็นรูปแบบที่เหมาะสมที่สุด

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ในงานวิจัยนี้ได้ศึกษาการทำงานของเครื่องจักรสืบค้นบนอินเตอร์เน็ตโดยมุ่งเน้นที่กระบวนการในการทำดัชนีข้อมูล และในการวิจัยครั้งนี้ได้ทำการแบ่งกระบวนการในการทำดัชนีออกเป็น 4 กระบวนการย่อย เพื่อทำการศึกษาความสัมพันธ์ของข้อมูลเข้าและข้อมูลออกในแต่ละกระบวนการ และความสัมพันธ์ของปริมาณข้อมูลเข้ากับเวลาในการประมวลผลแต่ละกระบวนการ โดยความสัมพันธ์ของขนาดข้อมูลเข้าและข้อมูลออกในแต่ละกระบวนการ สามารถสรุปได้ว่าอยู่ในรูปแบบของสมการ Linear ซึ่งจากความสัมพันธ์ของ INPUT กับ RMHTMLDB ในกระบวนการ RMHTML พบว่า RMHTMLDB จะมีขนาด 0.7040 ของ INPUT จากความสัมพันธ์ระหว่าง RMHTMLDB กับ RMSTOPDB ในกระบวนการ RMSTOP พบว่า RMSTOPDB จะมีขนาด 0.3857 ของ RMHTMLDB จากความสัมพันธ์ระหว่าง RMSTOP กับ STEMDB ในกระบวนการ STEMWORD พบว่า STEMDB มีขนาด 0.7450 ของ RMSTOP จากความสัมพันธ์ของ STEMDB กับ INDEXDB ในกระบวนการ INDEXMAP พบว่า INDEXDB จะมีขนาด 0.3938 ของ STEMDB

ส่วนความสัมพันธ์ของเวลาและขนาดข้อมูลเข้าในกระบวนการย่อยมีลักษณะเป็น Polynomial ซึ่งจากกราฟความสัมพันธ์เห็นว่าใกล้เคียงเส้นตรง ซึ่งในกระบวนการ RMHTML มีลักษณะเป็น Polynomial และมีกำลังสูงสุดเท่ากับ 2 ในกระบวนการ RMSTOP มีความสัมพันธ์อยู่ในรูป Polynomial กำลัง 2 กระบวนการ STEMWORD มีความสัมพันธ์ในลักษณะ Linear และในกระบวนการ INDEXMAP มีความสัมพันธ์ในรูป Polynomial กำลัง 2

นอกจากนี้จากผลการทดลองเราจะเห็นว่ากระบวนการทำดัชนีเว็บเพจโดยวิธีแบบ DISE ทำให้เวลาที่ใช้ในการทำดัชนีลดลงเมื่อเทียบกับการทำงานแบบ OSE พบว่าเวลาในการทำดัชนีสามารถลดลงได้โดยประมาณ 50% หรือมากกว่า ขึ้นกับจำนวนของกระบวนการที่กระจายออกไปและรูปแบบในการกระจาย รูปแบบการกระจายการทำดัชนีที่ใช้เวลาในการทำดัชนีน้อยที่สุดได้แก่ การกระจายในรูปแบบที่ 4 ซึ่งมีการกระจายกระบวนการ P1 P2 P3 ออกไปอยู่บนโฮสต์เดียวกัน ซึ่งเมื่อพิจารณาจากกระบวนการพบว่าเปรียบเทียบได้กับการกระจายกระบวนการ Preindexing ออกไปซึ่งเป็นรูปแบบที่ทำให้กระบวนการทำดัชนีรวดเร็ว และมีปริมาณการสื่อสารข้อมูลน้อยที่สุด ส่วนในรูปแบบการกระจายที่ 5 ถึง 7 แม้จะมีการกระจายกระบวนการทำดัชนีออกไปเป็นจำนวนมากกว่า แต่เวลาการทำดัชนีที่ใช้ไม่ได้น้อยกว่าการกระจายในรูปแบบที่กระจาย Preindexing ออกไป

5.2 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับผู้ที่สนใจหรือตั้งใจจะใช้งานวิจัยนี้เพื่อทำการศึกษาในขั้นถัดไป ทางผู้วิจัยมีข้อเสนอแนะดังนี้ ในการวิเคราะห์รูปแบบการทำงานการทำงานเพื่อให้ครอบคลุมการปัจจัยเกี่ยวกับ Bandwidth และมูลค่าของการสื่อสารข้อมูลที่เกิดขึ้น ผู้วิจัยเสนอว่าควรวิเคราะห์โดยใช้ Cost Analysis ช่วยในการวิเคราะห์ที่ละเอียดและครอบคลุมมากขึ้น เนื่องจากในการสื่อสารผ่านเครือข่ายภายในแม้ปริมาณการสื่อสารมากแต่มูลค่าทรัพยากรที่ใช้อาจจะคุ้มค่าต่อปริมาณสื่อสารข้อมูลที่เกิดขึ้น และการกระจายเฉพาะส่วน Preindexing ทำให้ในส่วนของกระบวนการ Indexing สามารถมีกระบวนการให้คะแนนคำในเอกสารแตกต่างกันได้ หากไม่ต้องการให้เกิดภาระงานที่เว็บเซิร์ฟเวอร์จึงอาจจัดตั้ง Preindexing เซิร์ฟเวอร์ขึ้นต่างหากเพื่อทำการ Preindexing เว็บใน Domain แล้วจึงส่งผลลัพธ์สู่ส่วนกลาง

เอกสารอ้างอิง

[1] Danny B. Lange., Mitsuru Oshima., "Programming and Deploying Java Mobile Agents with Aglets," *Addison Wesley*, August 1998.

[2] Yuwono, B., and Lee, D.L., "WISE: A World Wide Web resource database system," *IEEE Transaction on Knowledge and Data Engineering 8 (4)*, 1996, 548-554.

[3] David Kotz., Robert S. Gray., "Mobile Agents and the Future of the Internet, " *Department of Computer Science / Thayer School of Engineering Dartmonth College*, May 1999.

[4] Brian Brewington., Robert Gray., Karsuhiro Moizumi., David Kotz., George Cybenko., Daniela Rus., "Mobile agents in distributed information retrieval," *Department of Computer Science / Thayer School of Engineering Dartmonth College*, 1999.

[5] David Pallman., "Programming Bots, Spiders, and Intelligent Agents in Microsoft Visual C++," *Microsoft Press*, 1999.

[6] Craig A.Knoblock, "Search the World Wide Web," *IEEE Expert Intelligent System & Their Applications*, January-Febuary 1997.

[7] Salton, G., and McGill, M., "Introduction to Modern Information Retrieval," *McGraw-Hill*, New York NY, 1983.

[8] Alfonso Fuggetta, Gian Pietro Picco and Giovanni Vigna, "Understanding Code Mobility", *IEEE Transaction on Software Engineering, Vol.24, No. 5*, May 1998, pp. 342-361.

[9] C. Mic Bowman.; Peter B. Danzig., Darren R.Hardy ., "Harvest: A Scalable, Customizable Discovery and Access System" ,*Technical Report CU-CS-732-94 Department of Computer Science University of Colorado-Boulder*, Aug 1995.

[10] Darren R. Hardy., Michael F.Schwartz, "Customized Information Extraction as a Basic for Resource Discovery", *Technical Report CU-CS-707-94, To appear, ACM Transactions on Computer Systems*, March 1994.

[11] Mark P.Sinka, David W.Corne. "A Large Benchmark Dataset for Web Document Clustering" *Soft Computing Systems : Design, Management and Applications, of Frontiers in Artificial Intelligence and Applications, vol. 87*,2002. Pp. 881-890

[12] YingLian Xia, David O'Hallaron, Michael K. Reiter "A Secure Distributed Search System" *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Computing HPDC-11*, 2002

[13] Wes Sonnenreich, Tim Macinta. " Web Developer.com Guide to Search Engines" *John Wiley & Sons, Inc*, 1998.

[14] Sandra L. Esler and Michael L. Nelson "NASA Indexing Benchmarks: Evaluating Text Search Engines" *Journal of Network and Computer Applications, 1997*, Pp. 339-353

[15] http://www.tartarus.org/~martin/PorterStemmer/

[16] http://www.twmacinta.com/bddbot/

ภาคผนวก ก

รายการคำศัพท์ที่ไม่มีความหมายต่อเนื้อหาที่ใช้ในการวิจัยนี้

(Stopword List)

ภาคผนวก ก.

รายการคำศัพท์ที่ไม่มีความหมายต่อเนื้อหาที่ใช้ในการโปรแกรม BDDbot

A	An	Became	cannot	different
About	And	Because	canst	discussed
Above	Another	Become	certain	do
According	Any	Becomes	cf	does
Across	Anybody	Becoming	cfrd	doesn't
After	Anyhow	Been	choose	doing
Afterwards	Anyone	Before	conducted	dost
Against	Anything	Beforehand	considered	doth
Albeit	Anyway	Behind	contrariwise	double
All	Anywhere	Being	cos	down
Almost	Apart	Below	could	dual
Alone	Are	Beside	crd	due
Along	Around	Besides	cu	during
Already	As	Between	day	each
Also	At	Beyond	described	either
Although	Author	Both	describes	else
Always	Av	But	designed	elsewhere
Among	Available	By	determine	enough
Amongst	Be	Can	determined	Et
Etc	Ff	Has	hindmost	insomuch
Even	First	Hast	his	instead
Ever	For	Hath	hither	into
Every	Formerly	Have	hitherto	investigated
Everybody	Forth	He	how	inward
Everyone	Forward	Hence	however	inwards
Everything	Found	Henceforth	howsoever	is
Everywhere	From	Her	I	it
Except	Front	Here	Ie	its
Excepted	Further	Hereabouts	If	itself

Excepting	Furthermore	Hereafter	In	just
Exception	Furthest	Hereby	Inasmuch	kind
Exclude	General	Herein	Inc	kg
Excluding	Given	Hereto	Include	km
Exclusive	Get	Hereupon	Included	last
Far	Go	Hers	Including	latter
Farther	Had	Herself	Indeed	latterly
Farthest	Halves	Him	Indoors	less
Few	Hardly	Himself	Inside	lest
Let	Ms	Not	otherwise	provided
Like	Much	Nothing	ought	provides
Little	Must	Notwithstanding	our	quite
Ltd	My	Now	ours	rather
Made	Myself	Nowadays	ourselves	really
Many	Namely	Nowhere	out	related
May	Nbsp	Obtained	outside	report
Maybe	Need	Of	over	required
Me	Neither	Off	own	results
Meantime	Never	Often	per	round
Meanwhile	Nevertheless	Ok	performance	s
Might	Next	On	performed	said
More	No	Once	perhaps	sake
Moreover	Nobody	One	plenty	same
Most	None	Only	possible	sang
Mostly	Nonetheless	Onto	present	save
More	Noone	Or	presented	saw
Mr	Nope	Other	presents	see
Mrs	Nor	Others	provide	seeing
Seem	Slew	Sprung	there	throughout
Seemed	Slung	Srd	thereabout	thru
Seeming	Slunk	Stave	thereabouts	thus
Seems	Smote	Staves	thereafter	thy

Seen	So	Still	thereby	thyself
Seldom	Some	Studies	therefore	till
Selected	Somebody	Such	therein	to
Selves	Somehow	Supposing	thereof	together
Sent	Someone	Tested	thereon	too
Several	Something	Than	thereto	toward
Sfrd	Sometime	That	thereupon	types
Shalt	Sometimes	The	these	towards
She	Somewhat	Thee	they	unable
Should	Somewhere	Their	this	under
Shown	Spake	Them	those	underneath
Sideways	Spat	Themselves	thou	unless
Significant	Spoke	Then	though	unlike
Since	Spoken	Thence	thrice	until
Slept	Sprang	Thenceforth	through	up
Upon	Well	Wherefore	whichever	will
Upward	Were	Whereof	whichsoever	wilt
Upwards	What	Whereon	while	with
Us	Whatever	Wheresoever	whilst	within
Use	Whatsoever	Whereto	whither	without
Used	When	Wherefrom	who	worse
Using	Whence	Wherein	whoa	worst
Various	Whenever	Whereinto	whoever	would
Very	Whensoever	Whereunto	whole	wow
Via	Where	Whereupon	whom	ye
Vs	Whereabouts	Wherever	whomever	yet
Want	Whereafter	Wherewith	whomsoever	year
Was	Whereas	Whether	whose	yippee
We	Whereat	Whew	whosoever	you
Week	Whereby	Which	why	your

ภาคผนวก ข

Regression analysis

ภาคผนวก ข.

Regression analysis

Regression analysis คือการศึกษาถึงลักษณะความสัมพันธ์ระหว่างตัวแปร ว่าตัวแปรตัวหนึ่งหรือหลายตัว มีอิทธิผลต่อตัวแปรอีกตัวหนึ่งอย่างไร ในรูปใดโดยแสดงลักษณะความสัมพันธ์นั้นออกมาในรูปของสมการถดถอย ประโยชน์หนึ่งที่ได้จากการทราบความสัมพันธ์ระหว่างตัวแปร คือ จะสามารถทำนายค่าของตัวแปรหนึ่งที่สนใจ จากข้อมูลของตัวแปรอื่น โดยสามารถเขียนสมการความสัมพันธ์สำหรับการคำนวณเส้นแนวโน้มได้หลายรูปแบบดังนี้

เส้นตรง

คำนวณหาค่ากำลังสองน้อยที่สุดในแบบเส้นตรงแทนด้วยสมการดังต่อไปนี้

$y = mx + b$

โดย m คือความลาดเอียงและ b คือจุดตัดแกน

โพลีโนเมียล

คำนวณหาค่ากำลังสองน้อยที่สุดใช้ค่าจุดต่างๆ โดยการใช้สมการดังต่อไปนี้

$y = b + c_1x + c_2x^2 + c_3x^3 + \ldots + c_6x^6$

โดย b และ c1...c6 คือค่าคงที่

แบบลอการิทึม

คำนวณหาค่ากำลังสองน้อยที่สุดใช้ค่าจุดต่างๆ โดยการใช้สมการดังต่อไปนี้

$y = c \ln x + b$

โดย c และ b คือค่าคงที่และ e คือฟังก์ชันลอการิทึมธรรมชาติ

ยกกำลัง

คำนวณหาค่ากำลังสองน้อยที่สุดในใช้ค่าจุดต่างๆ โดยการใช้สมการดังต่อไปนี้

$y = ce^{bx}$

โดย c และ b คือค่าคงที่และ e คือฐานของลอการิทึมธรรมชาติ

ยกกำลัง

คำนวณหาค่ากำลังสองน้อยที่สุดในใช้ค่าจุดต่างๆ โดยการใช้สมการดังต่อไปนี้

$y = cx^b$

โดย c และ b คือค่าคงที่

ค่า R-squared

$R^2 = 1 - (SSE/SST)$

โดย

SSE The Error Sum of Squares โดย SSE = $(Y_i - \hat{Y}_i)^2$

SST Total Sum of Squares โดย SST = $(\sum Y_i^2) - (\sum Y_i)^2 / n$

กำลังสองของสัมประสิทธิ์สหสัมพันธ์ r^2 เรียกว่าสัมประสิทธิ์การตัดสินใจ (coefficient of determination) **ใช้วัดอิทธิพลของตัวแปรตัวหนึ่งว่ามีผลต่อตัวแปรอีกตัวหนึ่ง มากน้อยเพียงใด สัมประสิทธิ์การตัดสินใจมีค่าได้ตั้งแต่ 0 ถึง 1 และเมื่อ R2 มีค่าใกล้ 1 มากเท่าใดระดับของความสัมพันธ์ยิ่งน่าเชื่อถือมากขึ้น**

ภาคผนวก ค

ผลงานวิจัยที่เกี่ยวข้องกับวิทยานิพนธ์และได้รับการตีพิมพ์

K. Tuanveeradej, and A. Khunkitti, "Distributed Indexing Search Engine", ICEP 2004, Phuket, Thailand, 22-23 January, 2004.

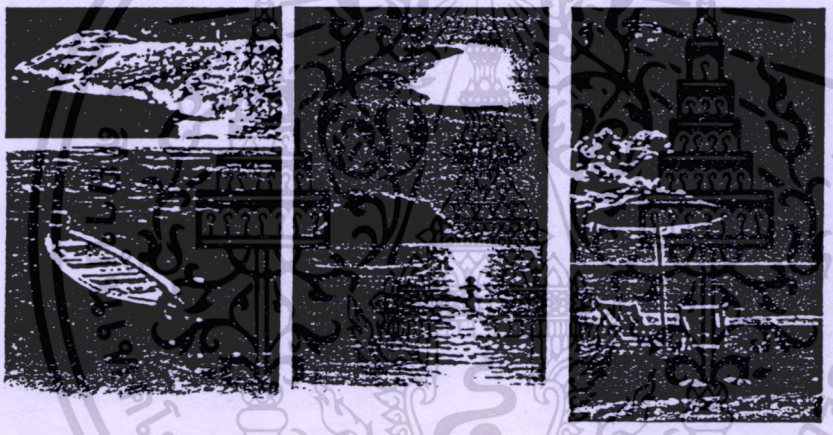
PROCEEDINGS

The 4th Information and Computer Engineering Postgraduate Workshop 2004

ICEP 2004

22nd - 23rd January 2004

Phuket, Thailand



Prof Jun Marai, KEIO University, Japan

Organised and Sponsored by
Department of Computer Engineering,
Faculty of Engineering,
Prince of Songkla University, Thailand
and Prof Jun Marai, KEIO University, Japan

In cooperation with
IEEE ComSoc, Thailand Chapter,
ECTI (Electrical Engineering/Electronics,
Computer, Telecommunications and Information
Technology Association of Thailand)

ISBN 974-644-518-9

Distributed Indexing Search Engine

K. Tuanveeradej and A. Khunkitti

Faculty of Information technology
Kingmongkut's Institute of Technology Ladkrabang
Chalongkrung Road, Ladkrabang Bangkok Thailand 10520
Email: s2067013@kmitl.ac.th

Abstract

Rapid growth in data volume, and data diversity render Internet-accessible information increasingly difficult to locate relevant information. Web search engine has an important role in addressing this problem, but Search Engine have centralize characteristic that make unnecessary network traffic load and densely workload at server that effect system performance. This paper will introduce a way Distributed Internet Search Engine can address this problem, parameter involve in analyzing, and relation between the parameter such as network load, Indexing workload, Space utilization of Distributed Internet Search Engine. This paper tend to reduce network load and find out optimum distributed indexing form for Internet Search Engine, Indexing workload and space utilization that optimizes Distributed Indexing Internet Search Engine.

1. . Introduction

The Rapid growth of Internet, and WWW increase volume of accessible-information, and make it diversely. While they are growing the difficulty to find relevant information is increasing. To use this accessible-information effectively, Internet Search Engine has an important role to gather, specific-topic, and index of this accessible-information to make it flexible search.

Internet search engine can separated in three parts. Front-end with interface to communicate with user to receive query and send result back to user. Database that keep indexed data of web pages. Back-end or web robot that collects web pages by crawling web pages by hyperlink and gets web pages to make indexed database for Search engine. Because of characteristic of accessible-information in Internet and WWW is distributed and diversely, but process of gathering and indexing was centralized. This cause load in network increasingly and also slowly to make index up to date. By distributed process of indexing to host that contain accessible-information we can reduce communication traffic and indexing workload in central part.

In this paper we discuss how to distribute process of indexing, analysis relation between three-parameters communication traffic, Indexing workload, Space utilization propose to reduce cost of operation and optimize distributed form of indexing in Distributed Internet Search Engine. Sections 2 introduce architecture of search engine in old fashion, format of index database, function of web robot to crawling and indexing to build index database. Section 3 introduce distributed indexing search engine, architecture and the ways that indexer can be distributed. Section 4 analyses the effective of distributed indexing search engine compare with old fashion of search engine in three variables workload, Communication traffic, and disk space utilization.

2. Internet Search Engine

Internet Search Engine can separate into three parts by it's function, Front-end, Index Database, and Back-End. User connect to search engine by front-end that it function for receive query from user and send query to search in index database then return relevant results to user.

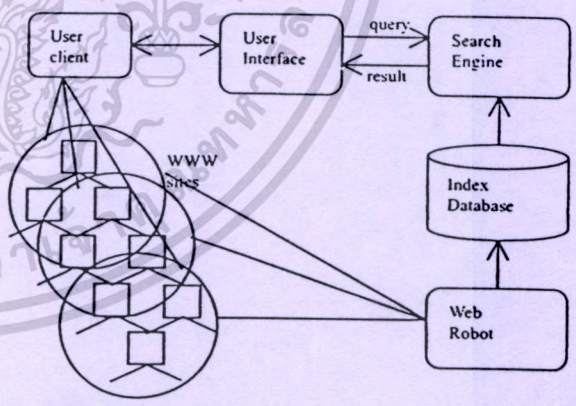


Figure 1. The architecture of Search engine

Index database has two types of database, inverted files, and lexicon, which is store web documents. Where inverted file is the file that for each word list the

files that contain that word, and lexicon is dictionary that over all words occurring in document. These inverted file and lexicon are built by Web robot in Back-end of search engine.

Simple Inverted file:

w1: docID, docID, docID, docID
w2: docID, docID, docID
w3: docID, docID, docID, docID, docID

Lexicon:

docID: w1, w2, w3, w4
docID: w1, w2, w4
docID: w2, w5, w6

Figure 2. Inverted file and Lexicon

Search engine back-end has function to traverse web graph, repeat follow link, store web, parse web document, extract new link, and update index database. Search engine back-end function can split into two parts, Gatherer for search and fetch web pages, and Indexer to index the web pages. Gatherer gets the start web page and extracts the link URL from the page check for redundancy, if not keep it in queue to fetch else discard it, this called crawling. Next the web page that fetched will send to Indexer. Then get next web pages from queue until no remain URL in queue.

Indexer receives web pages from gatherer then begins indexing process. From the retrieved web pages, it must discard html tag. Next step the high-frequency function words need to be eliminated. These comprise 40 to 50 percent of the content, as suggested, these words are poor discriminate and cannot possibly be used to identify document content. These words can be included in dictionary sometime called a negative dictionary or stop list. An example for stop list is shown in table 1. After discard stop words, next step is word stemming then identification of index terms and assign them to the documents of a collection.

Table 1. Excerpt from typical stop list. [3]

A	ALONG	ANOTHER
ABOUT	ALREADY	ANY
AFTER	ALSO	ANYHOW
AFTERWARDS	ALTHOUGH	ANYONE
AGAIN	ALWAYS	ANYTHING
AGAINST	AMONG	ANYWHERE
ALL	AMONGST	ARE
ALMOST	AN	AROUND
ALONE	AND	AS

3. Distributed Indexing Search engine

Internet and WWW have distributed and varied characteristic, but in gathering and indexing web of search engine is difference. Because of centralized system behavior of search engine back-end cause the efficiency of indexing and crawling decrease. In crawling step the web pages must be loaded to be index at search engine center. This produces the traffic load that not necessary and workload in indexing web at search engine a lot. To address this problem indexing process that distributed to web server host can reduce the workload and traffic load at search engine.

In this paper we break process of indexing into two parts, Preprocess, and Indexing map process to spread the workload. In preprocess html tag, and stop word in web pages will be discarded. The result of preprocess will send to index map process. On index map part inverted file and lexicon are build from result of preprocess. Result from indexing process is indexed data that will send to web robot inter face in center to update the index database.

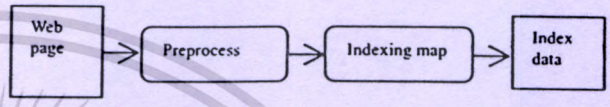


Figure 3. Breaking indexing process to preprocess and indexing map

By sending agent to web servers or nearest location we can reduce traffic communication between web servers and search engine. Indexer agent was sent to host and index web at host then sends result back to web robot Interface to update index database at center.

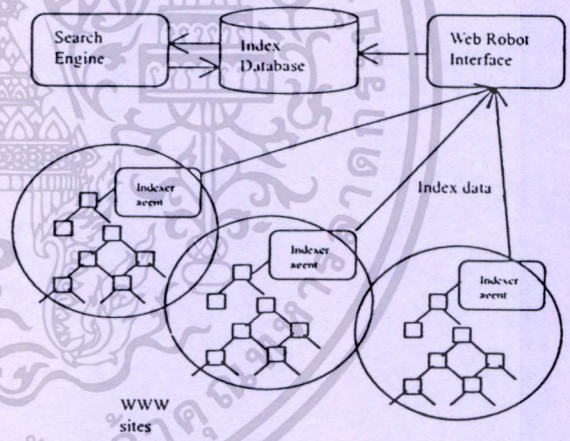


Figure 4. Distributed indexing Search engine.

In domain that has small number of web sites, one indexing agent cans response to build index data for all web sites in the domain. Communication over inter network can be reduced, and use local net work that have lower communication cost.

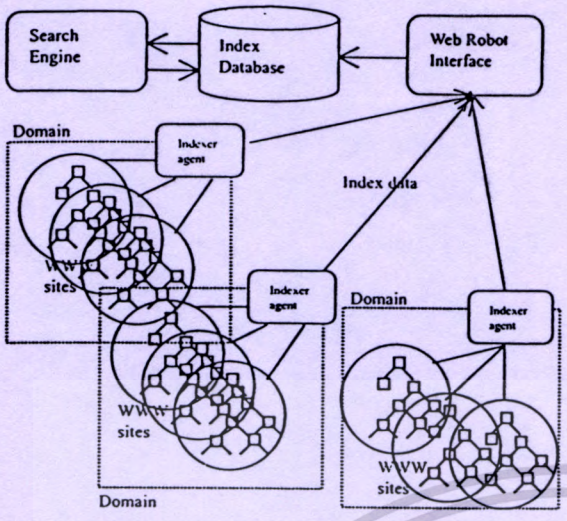


Figure 5. Distribute indexer agent into domain.

Next in large domain indexer agent can split into levels, preprocess, and index map process, and have one or more index map center in domain. Each sub level agent crawl hosts that they base on and do preprocess indexing of webs that they crawl. After that preprocess results of sub level preprocess agent send to sub index map agent that will collect the results and perform final indexing then send final index result to web robot interface to update index database at center.

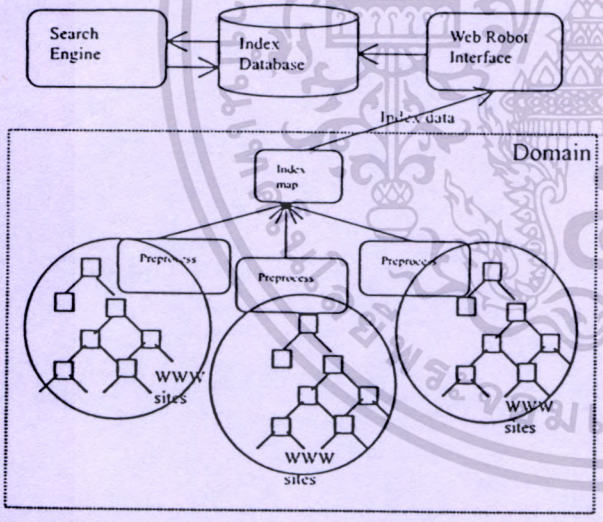


Figure 6. Split process to levels in domain

4. Analysis

Search engine is centralized fashion that work load in indexing process add up in search engine center when pages to index are increase. Distributed indexing search engine is difference. Workload at center is distributed to web servers then index web pages as local and send indexed data to update search engine database. This can reduce the workload in center, and network communication traffic. Workload that distributed to host still can split to reduce workload in host by setup a host that individual separate form web server, just for indexing the web in that domain, or split the indexing process to preprocess and indexing process.

In this section we analysis the effective of search engine and distributed indexing about workload, communication traffic and disk space usage.

4.1 Workload

Search engine crawl and fetch web pages from web servers then index them. Workload in indexing will add up in central, and reduces speed of indexing and update search engine database. Distributed indexing search engine is difference, it distributes indexing process to web servers or a host established separated individual for indexing function. The number of web pages is broken into small parts, and distributes in many hosts. Workload at central is reduced and distributes then speed up indexing web pages and updating search engine database. Because of distributing indexing process each indexer works concurrent at the same time and don't need to wait another, indexing process and update search engine database are expeditiously.

Consider about workload as equation (1). When input are raised, the workload are raised too.

$$W_i = P_i f(S_i) \quad (1)$$

W_i = work load at i
S_i = size of input at i
P_i = constant of process at i

Workload is increase when input increase.

4.2 Communication traffic

Search engine fetch web pages form web servers across network and index them at search engine center. This operation to bring about communication traffic that can present as in equation (2).

$$T = \sum_{i=1}^{n} F_i S_I + \text{Communication overhead} \quad (2)$$

T = total data transfer of search engine
F_i = the number of html files at host i
S_i = average html files size at host i

When consider about size of web pages and size of indexed data in table 2 from experiment that we setup five

hosts that hold web pages have size about 50 megabytes. We found that size of indexed web page can reduced about 50% of original size of web pages, as show in table 2.

Table 2 space average of html and indexed data from experimental.

Host	Size of files (MB.)	
	Html	Indexed
Host 1	50.1	25
Host 2	50.0	19
Host 3	51.2	20
Host 4	49.8	27
Host 5	50.2	25

If we can distribute indexing process to web hosts we can reduce communication traffic between web hosts and search engine center, because web pages were indexed and size of the index data is reduced from it source. As in equation (3) show size of data that transfer between distributed indexing agent and search engine center.

$$T_{index} = \sum_{i=1}^{n} F_i S_i R_i + \text{Communication overhead} \quad (3)$$

$$R_i = I_i/S_i \quad (4)$$

T_{index} = total data transfer of index
R_i = ratio of index per original file size at host i
I_i = index size at host i

From table 2 ratio of index per original file size equal 0.5 or reduced about 50% then total data transfer reduced about 50% plus communication overhead.

To examine cost of transferring data across network, communication between hosts in local network cheaper than inter network communication. Distance and size of data is factor that we must consider. In equation (5) presents that cost of communication in factor of size and distance.

$$C_{ij} = S_{ij}\, g(d_{ij}) \quad (5)$$

C_{ij} = cost of transferring data between i and j
S_{ij} = size of data transfer between i and j
d_{ij} = distance between i and j

From equation (3) size of data to transfer reduced about 50% cost of transferring data between host i and j in equation (5) is reduced 50% when distance between j *and* i is equal. In another way when size of data not changed when distance between i and j is reduced cost of data transferring is reduced. The tradeoff for reduced cost of transferring data is workload in the WWW hosts increased from indexing process that distributed to hosts.

4.3 Disk space

In distributed indexing search engine after indexing process. Result of process will send to next process or center of search engine to update search engine database. Result from this process is part of index database can be cached for later work such as resenting if have request form search engine center, or for searching in local.

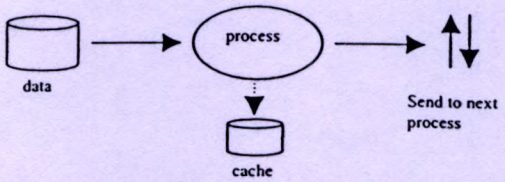


Figure 7. After indexing process result will send to next process and cache

From figure 7 size of disk cache can be presented as in equation (6).

$$D_i = \sum_{i=1}^{n} F_i S_i R_i \quad (6)$$

D_i = disk usage in caching at host i

From equation (6) for host that function to indexing web pages, size of indexed data of the web equal to disk usage for caching at that host.

5. Discussion and Conclusion

Internet search engine can be divided to three parts, front-end, index database, and back-end. In old fashion of search engine that use web robot to crawl and index web pages can bring about a lot of workload, and traffic communication to web robot. This can cause web robot effective in gathering and indexing process.

Distributed indexing search engine can resolve this problem by distribute indexing process to hosts and index web pages at same host or closer to source of web page and concurrently. Advantage of distributing indexing to host near by source of web pages is lower cost of transferring data to indexing process. Another advantage of distributed indexing is concurrent of indexing process that can be done by parallel processing of other indexers rapidly update of search engine database. The inferiority of distributed indexing process to WWW hosts is workload form indexing process is added up in WWW hosts. This idea of distributed indexing search engine can apply to use in organizations that have to cooperative and share information on web pages and distributed like group of university in Thailand.

Factors that to optimize distribute search engine data base are about three factors workload, communication traffic, disk space usage. How we can optimize these three factors and make distributed indexing search engine process faster and cheaper in operation still in study.

6. References

[1] Yuwono, B., and Lee, D.L., "WISE: A World Wide Web resource database system", IEEE Transaction on Knowledge and Data Engineering 8 (4), 1996, pp. 548-554.
[2] David Pallman., "Programming Bots, Spiders, and Intelligent Agents in Microsoft Visual C++," Microsoft Press, 1999.
[3] Salton, G., and McGill, M., "Introduction Modern Information Retrieval," McGraw-Hill, New York NY, 1983.
[4] C. Mic Bowman.; Peter B. Danzig., Darren R.Hardy ., "Harvest: A Scalable, Customizable Discovery and Access System" ,Technical Report CU-CS-732-94 Department of Computer Science University of Colorado-Boulder, Aug 1995.
[5] Brian Brewington., Robert Gray., Karsuhiro Moizumi., David Kotz., George Cybenko., Daniela Rus., "Mobile agents in distributed information retrieval," Department of Computer Science / Thayer School of Engineering Dartmonth College, 1999.

ประวัติผู้เขียน

ชื่อ - นามสกุล	โกศล เตือนวีระเดช
วันเดือนปีเกิด	2 มกราคม 2520
สถานที่เกิด	นครศรีธรรมราช
ประวัติการศึกษา	2541 คณะวิทยาศาสตร์ สาขาวิทยาการสารสนเทศ มหาวิทยาลัยศรีนครินทรวิโรฒประสานมิตร