เอ็กซ์แซค: เครื่องมือการบีบอัดและการค้นคืนเอกสารเอ็กซ์เอ็มแอล
XZaQ: An XML Zipping and Querying Tool

สิรินธร จียาศักดิ์[1]* และ ชาญยศ ปลื้มปิติวิริยะเวช[2]

[1]อาจารย์ สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยกรุงเทพ กรุงเทพฯ 10110
[2]ผู้ช่วยศาสตราจารย์ สาขาวิทยาการคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศและการสื่อสาร
มหาวิทยาลัยมหิดล จังหวัดนครปฐม 73170

บทคัดย่อ

งานวิจัย XZAQ นำเสนอวิธีสำหรับการบีบอัดและการค้นคืนข้อมูลเอ็กซ์เอ็มแอลที่ผ่านการบีบอัดแล้วด้วยวิธีเชิงไวยากรณ์ทำให้สามารถบีบอัดเอกสารและค้นคืนข้อมูลได้โดยไม่จำเป็นต้องคลายเอกสารทั้งหมดก่อน XZaQ ใช้วิธีบีบอัดแบบ Non-homomorphic เพื่อแยกโครงสร้างและข้อมูลของเอกสารเอ็กซ์เอ็มแอลออกจากกัน XZaQ เข้ารหัสโครงสร้างเอกสารพร้อมทั้งเชื่อมโยงความสัมพันธ์ระหว่างโครงสร้างและข้อมูลของเอกสารด้วยวิธีเชิงไวยากรณ์ XZaQ สามารถสนับสนุนการค้นคืนข้อมูลบนเอกสารที่ผ่านการบีบอัดแล้วด้วยการคลายข้อมูลเพียงบางส่วนเท่านั้น จากผลการทดลอง พบว่า การบีบอัดเอกสารเอ็กซ์เอ็มแอลด้วยวิธี Non-homomorphic และวิธีการเชิงไวยากรณ์ทำให้เอกสารมีขนาดเล็กลงโดยเฉลี่ย 70% เมื่อเปรียบเทียบกับขนาดเอกสารก่อนการบีบอัด และยังสามารถบีบอัดเอกสารได้ขนาดเล็กกว่า XGRIND ประมาณ 10% งานวิจัย XZaQ ออกแบบการเก็บข้อมูลเพื่อรองรับสำหรับการค้นคืนข้อมูล (Query) ได้ทั้งแบบ Simple Query และ Complex Query ด้วยเงื่อนไขชนิด Exact-match หรือ Range Query ได้ในขณะที่ XGRIND สามารถบอกได้เพียงว่าพบหรือไม่พบข้อมูลในการค้นคืนเท่านั้น

Abstract

This paper proposes a new method for XML compression, called XML Zipping and Querying (XZAQ), which can compress a regular XML file and query the compressed XML file without doing full decompression. XZAQ is based on non-homomorphic method, which does not preserve an interleaving between an element structure and data content. XZAQ is implemented by using grammar-based method to encode an element structure and to maintain links between the element structure and the data content. XZAQ can partially decompress a block of compressed data in order to support querying of XML data. From the experiments, XZAQ had an average compression ratio of 70% smaller than original size. It also provided better compression ratio of 10% less than XGRIND. Moreover, XZAQ could support either exact-match or range query in both simple and complex query types whereas XGRIND could provide only found or not found match.

คำสำคัญ : เอ็กซ์เอ็มแอล การบีบอัด การค้นคืน การคลายเอกสารเพียงบางส่วน

Keywords : XML, Compression, Querying, Partial Decompression

* *ผู้นิพนธ์และประสานงานไปรษณีย์อิเล็กทรอนิกส์ sirinthorn.c@bu.ac.th โทร. 08 1839 3846*

1. บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

XML (eXtensible Markup Language) ได้กลายมาเป็นมาตรฐานสำหรับการแลกเปลี่ยนข้อมูล และจัดเก็บข้อมูลบนเว็บแอพพลิเคชันที่ได้รับความนิยมในยุคปัจจุบัน XML เป็นภาษาที่สามารถนำมาประยุกต์ใช้ได้กับเทคโนโลยีสารสนเทศที่นิยมในปัจจุบัน ไม่ว่าจะเป็น โทรศัพท์มือถือ หรือคอมพิวเตอร์มือถือ รวมถึงการแลกเปลี่ยนข้อมูลข่าวสารบนเครือข่ายอินเทอร์เน็ต เนื่องจาก XML มีคุณลักษณะในการอธิบายความหมายของข้อมูล และมีความยืดหยุ่นในการใช้งาน โดย XML ใช้แท็กสำหรับการอธิบายความหมายของข้อมูล โดยผู้ใช้สามารถกำหนดชื่อแท็กที่สื่อถึงความหมายของข้อมูลในเอกสารนั้นได้เองจึงทำให้มีความยืดหยุ่น และเอกสารที่ถูกสร้างขึ้นเข้าใจได้ง่าย ดังนั้น จึงส่งผลต่อการเข้าถึงข้อมูลบนเอกสาร XML ทำให้สามารถเข้าถึงข้อมูลได้ง่ายขึ้นโดยสามารถเข้าถึงข้อมูลหรือค้นคืนข้อมูลบนเอกสาร XML ได้โดยตรงแต่การใช้แท็กเข้ามาอธิบายความหมายของข้อมูลส่งผลให้ขนาดของเอกสาร XML มีขนาดใหญ่ขึ้น เนื่องจากการใช้แท็กที่ซ้ำซ้อนกันจำนวนมากเพื่ออธิบายความหมายของข้อมูลประเภทเดียวกันแต่มีความหมายแตกต่างกัน ส่งผลกระทบโดยตรงทำให้ขนาดของเอกสารเอ็กซ์เอ็มแอลใหญ่ขึ้นเมื่อเทียบกับขนาดข้อมูลจริง และขนาดที่เพิ่มมากขึ้นนี้เองจึงส่งผลให้สิ้นเปลืองเนื้อที่ในการจัดเก็บเอกสาร สิ้นเปลืองแบนด์วิธ รวมถึงเวลาในการรับส่งเอกสารในการแลกเปลี่ยนเอกสารผ่านบนระบบเครือข่าย

จากปัญหาที่กล่าวข้างต้นการลดขนาดเอกสาร XML จึงมีความจำเป็นอย่างยิ่ง เมื่อขนาดเล็กลงทำให้การใช้พื้นที่จัดเก็บแบนด์วิธ รวมถึงเวลาในการแลกเปลี่ยนข้อมูลบนเครือข่ายก็ลดลง ดังนั้น ผู้วิจัยจึงตระหนักถึงความจำเป็นที่จะพัฒนาเครื่องมือ XZaQ Compressor สำหรับบีบอัดเอกสาร XML โดยเฉพาะเพื่อเพิ่มประสิทธิภาพในการบีบอัดได้ดีกว่าการบีบอัดข้อมูลทั่วไป XZaQ ใช้วิธีการเข้ารหัสเชิงไวยากรณ์ส่วนที่เป็นโครงสร้างของเอกสาร รวมถึงเครื่องมือ Query Processor ที่สามารถค้นคืนข้อมูลได้โดยไม่ต้องคลายเอกสารทั้งหมดก่อนเพื่อลดเวลาที่ใช้สำหรับการเข้าถึงข้อมูลบนเอกสาร XML ที่มีขนาดใหญ่

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อศึกษาและพัฒนาอัลกอริทึมสำหรับสร้างเครื่องมือในการบีบอัดและคลายเอกสารเอ็กซ์เอ็มแอล โดยใช้วิธีการเข้ารหัสเชิงไวยากรณ์

1.2.2 เพื่อศึกษาและออกแบบรูปแบบการเก็บข้อมูลที่ผ่านการบีบอัด ให้สามารถรองรับการค้นคืนข้อมูลบนเอกสารที่ผ่านการบีบอัดแล้วโดยคลายเอกสารเพียงบางส่วน

1.3 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1.3.1 เอ็กซ์เอ็มแอล (XML: eXtensible Markup Language)

ปัจจุบันภาษา XML ได้กลายมาเป็นมาตรฐานสำหรับการแลกเปลี่ยนข้อมูลข่าวสารระหว่างเว็บแอ็พพลิเคชัน เนื่องจาก XML มีความยืดหยุ่นโดยผู้ใช้สามารถกำหนดแท็กได้เอง พร้อมกันนี้แท็กที่ใช้ทั้งยังสามารถอธิบายความหมายของข้อมูลภายในเอกสารได้ ซึ่งแท็กจะคล้ายกับภาษา HTML (Hypertext Markup Language) ซึ่งเป็นภาษามาตรฐานสำหรับเขียนเว็บเพจในอดีตโดยแท็ก

ในภาษา HTML ใช้สำหรับจัดรูปแบบที่จะแสดงผลข้อมูลบนเว็บบราว์เซอร์เท่านั้น ในทางกลับกันแท็กในภาษา XML นั้นใช้สำหรับอธิบายความหมายของข้อมูลภายในเอกสารแทนที่จะใช้สำหรับจัดรูปแบบการแสดงผลอย่างภาษา HTML ดังนั้นหากใช้เอกสาร XML ในการแลกเปลี่ยนข้อมูลระหว่างเว็บแอ็พพลิเคชันนั่นก็หมายถึงรายละเอียดข้อมูลภายในว่ามีรายละเอียดอะไรอยู่ภายในเว็บหรือภายในเอกสารนั้น ซึ่งความหมายของข้อมูลนี้สามารถนำข้อมูลไปประมวลผลเพื่อนำไปวิเคราะห์และใช้ประโยชน์ได้ต่อไป

ส่วนประกอบข้อมูลพื้นฐาน เรียกว่า อิลิเมนต์ (Element) แต่ละอิลิเมนต์ประกอบด้วย แท็กเปิด (Start Tag) ข้อมูล (Content Data) และแท็กปิด (End Tag) โดยแต่ละเอกสารเอ็กซ์เอ็มแอลจะมีแท็กที่เป็นรูทอิลิเมนต์ (Root Element) เพียงแท็กเดียวเท่านั้นส่วนแท็กอื่น ๆ ที่อยู่ภายใต้รูทอิลิเมนต์จะมีจำนวนได้มากกว่าหนึ่งแท็กขึ้นไป โดยในแต่ละอิลิเมนต์สามารถมีแอททริบิวต์ (Attribute) ตั้งแต่หนึ่งแอททริบิวต์ขึ้นไป เพื่ออธิบายคุณสมบัติแต่ละอิลิเมนต์ได้หรือภายในอิลิเมนต์จะไม่มีแอททริบิวต์เลยด้วยการใช้แท็กอธิบายความหมายของข้อมูลนี้เองทำให้เอกสารเอ็กซ์เอ็มแอลมีความยืดหยุ่นต่อการใช้งานสูง อีกทั้งผู้ใช้สามารถกำหนดแท็กเพื่ออธิบายความหมายข้อมูลได้เอง

1.3.2 เอ็กซ์เอ็มแอลพาสเซอร์ (XML Parser)

การเข้าถึงเอกสารเอ็กซ์เอ็มแอลสามารถกระทำได้โดยผ่านเครื่องมือสำหรับวิเคราะห์เอกสารซึ่งเรียกว่า เอ็กซ์เอ็มแอลพาสเซอร์ โดยพาสเซอร์แบ่งเป็น 2 ชนิด ได้แก่ DOM Parser และ SAX Parser

DOM Parser

ดึงข้อมูลจากเอกสารโดยใช้วิธีแบบโครงสร้างต้นไม้ (Tree-based Approach) เริ่มแรกดอมพาสเซอร์จะอ่านข้อมูลทั้งเอกสารมาเก็บไว้ในหน่วยความจำทั้งหมดก่อนแล้ววิเคราะห์เอกสารเพื่อจำแนกโครงสร้างเอกสารผ่านดอมทรี (DOM Tree) โดยดอมทรีจะประกอบไปด้วยโหนด (Node) ที่เชื่อมโยงกัน

SAX Parser

ย่อมาจาก Simple API for XML ใช้วิธีการแบบลำดับเหตุการณ์ (Event-based Approach) ซึ่งวิธีการนี้จะไม่นำเอกสารมาเก็บไว้ในหน่วยความจำก่อนเหมือนของ DOM แต่จะวิเคราะห์และเข้าถึงข้อมูลเอ็กซ์เอ็มแอลตามเหตุการณ์ทีละเหตุการณ์ไปเรื่อย ๆ ก่อนกว่าข้อมูลจะหมด โดยวิธีการนี้นอกจากจะใช้เนื้อที่ของหน่วยความจำที่น้อยกว่ายังสามารถเข้าถึงข้อมูลได้เร็วเมื่อเทียบกับวิธีการของดอมทรี โดย SAX จะมีตัวควบคุมเหตุการณ์ที่เรียกว่า Event-handler เพื่อจำแนกเหตุการณ์ที่เกิดขึ้นภายในเอกสาร ดังนั้น ผู้วิจัยจึงเลือกใช้ SAX Parser เป็นเครื่องมือสำหรับวิเคราะห์ข้อมูลเอ็กซ์เอ็มเอลเพื่อเป็นพื้นฐานสำคัญสำหรับการบีบอัดเอกสารเอ็กซ์เอ็มแอลต่อไป

1.3.3 งานวิจัยที่เกี่ยวข้อง

ปัจจุบันเครื่องมือสำหรับบีบอัดเฉพาะสำหรับเอกสารเอ็กซ์เอ็มแอลได้ถูกพัฒนาขึ้นโดยสามารถจำแนกตามลักษณะโครงสร้างข้อมูล ความสามารถในการค้นคืนเอกสารเอ็กซ์เอ็มแอลที่ผ่านการบีบอัดแล้ว ดังสรุปในตาราง ดังนี้

ตารางที่ 1 การจำแนกประเภทงานวิจัยที่เกี่ยวกับการบีบอัด XML

Data Organization Scheme	Query Capability	
	Unqueriable	*Queriable*
Homomorphic	-	XGRIND, XPRESS
Non-homomorphic	XMILL, XPACK	XQueC, XQzip, ***XZAQ***

จากตารางที่ 1 ตารางสรุปเครื่องมือสำหรับบีบอัดเอกสารเอ็กซ์เอ็มแอลทั้งแบบที่รองรับการค้นคืนได้ (Queriable XML Compressors) และเครื่องมือที่สามารถบีบอัดได้เพียงอย่างเดียว (Unqueriable XML Compressors) เครื่องมือประเภท Unqueirable นี้ไม่รองรับการค้นคืนเอกสารที่ผ่านการบีบอัดแล้ว ดังนั้น จะต้องคลายเอกสารทั้งหมดก่อนจึงสามารถค้นคืนเอกสารได้ แต่สำหรับประเภทนี้จะมุ่งเน้นไปที่การบีบอัดเอกสารให้ได้มากที่สุด XMILL (Liefke and Suciu, 2000) เป็นงานวิจัยแรกที่พัฒนาสำหรับการบีบอัดเอกสาร XML โดยเฉพาะและเป็นต้นแบบงานวิจัยอื่น ๆ XPACK (Mairieng and Pluempitiwiriyawej, 2003) งานวิจัยของไทยชิ้นแรกเกี่ยวกับการบีบอัดเอกสาร XML ซึ่งงานวิจัยทั้ง 2 ใช้วิธีการบีบอัดในลักษณะ Non-homomorphic โดยจะแยกระหว่างโครงสร้างของเอกสารและข้อมูลออกจากกันก่อนแล้วจึงหาเทคนิคเพื่อเข้ารหัสโครงสร้างของเอกสารก่อนบีบอัดโครงสร้างและข้อมูลแยกออกจากกันทั้งนี้เพื่อเพิ่มประสิทธิภาพการบีบอัดให้ได้สูงสุด

ส่วนเครื่องมือประเภทที่รองรับการค้นคืน (Queriable XML Compressors) เป็นเครื่องมือสำหรับบีบอัดเอกสารเอ็กซ์เอ็มแอล พร้อมทั้งรองรับการค้นคืนเอกสารที่ผ่านการบีบอัดแล้ว โดยเครื่องมือประเภทนี้จะรองรับการสอบถามเพื่อค้นคืนข้อมูลบนเอกสารเอ็กซ์เอ็มแอลได้โดยไม่ต้องคลายเอกสารทั้งหมดก่อนหรือคลายเอกสารเพียงบางส่วน สำหรับเครื่องมือประเภทนี้จะมุ่งเน้นไปที่การบีบอัดพร้อมทั้งคำนึงถึงการค้นคืนเอกสารที่ผ่านการบีบอัดแล้วทำให้สะดวกในการค้นคืนเอกสารเอ็กซ์เอ็มแอลที่มีขนาดใหญ่ สามารถแบ่งกลุ่มเครื่องมือตามลักษณะการบีบอัดแบบ Homomorphic ได้แก่ XGRIND (Tolani, 2002), XPRESS (Jun-Ki, 2003) โดยวิธีนี้จะคงโครงสร้างและข้อมูลของเอกสารไว้ด้วยกัน ทั้งนี้เพื่อง่ายต่อการค้นคืนโดยไม่ต้องคลายเอกสารทั้งหมดก่อน ส่วนกลุ่มเครื่องมือที่ใช้วิธีบีบอัดลักษณะ Non-homomorphic ได้แก่ XCQ (Wai Yeung, 2003) XQzip (Cheng, 2004), XQueC (Arion, 2004) รวมถึงงานวิจัย XZaQ ก็จัดอยู่ในกลุ่มนี้ สำหรับการบีบอัดจะแยกโครงสร้างและข้อมูลออกจากกันบีบอัดแยกกันทั้งนี้เพื่อเพิ่มประสิทธิภาพของการบีบอัดให้ได้เอกสารที่มีขนาดเล็กมากที่สุดพร้อมทั้งรองรับการค้นคืนเอกสารได้

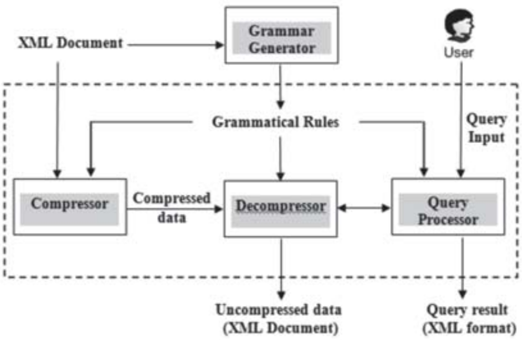
2. วิธีการทดลอง

คณะผู้วิจัยได้ศึกษาและพัฒนาอัลกอลิทึมเพื่อสร้างเครื่องมือสำหรับงานวิจัย XZaQ ดังรายละเอียดต่อไปนี้

2.1 เครื่องมือที่ใช้ในการวิจัย

การวิจัยครั้งนี้คณะผู้วิจัยได้สร้างเครื่องมือสำหรับทดลอง 4 ส่วน ได้แก่ เครื่องมือสำหรับวิเคราะห์และสร้างกฎไวยากรณ์ (Grammar Generator) เครื่องมือบีบอัดเอกสาร XML (XZaQ Compressor) เครื่องมือคลายเอกสาร XML (XZaQ Decompressor) และเครื่องมือค้นคืน

เอกสาร XML (Query Processor) ที่ผ่านการบีบอัดแล้วโดยคลายเอกสารเพียงบางส่วน (Partial Decompression) ดังแสดงในรูปที่ 1



รูปที่ 1 สถาปัตยกรรมหลักของ XZaQ

2.2 เครื่องมือช่วยในการวิเคราะห์และสร้างโครงสร้างเอกสารเอ็กซ์เอ็มแอล (Grammar Generator)

เครื่องมือ Grammar Generator นี้เป็นเครื่องมือสำหรับวิเคราะห์โครงสร้างของเอกสารเอ็กซ์เอ็มแอลเพื่อสร้างกฎไวยากรณ์สำหรับอธิบายความสัมพันธ์ระหว่างอิลิเมนต์และแอ็ตทริบิวท์โครงสร้างภายในของเอกสารเอ็กซ์เอ็มแอล กระบวนการหลัก คือ จะนำเอกสารเอ็กซ์เอ็มแอลมาแจกแจงให้อยู่ในรูปของโครงสร้างต้นไม้ (DOM Tree) แล้ววิเคราะห์ความสัมพันธ์ระหว่างอิมิเมนต์และแอตทริบิวท์ที่ได้จากการแจกแจงมาแล้วสรุปเป็นโครงสร้างของเอกสารโดยใช้กฎไวยากรณ์แทนความสัมพันธ์โครงสร้างและข้อมูลภายในที่เกิดขึ้น ซึ่งงานวิจัยนี้ได้ออกแบบไฟล์สำหรับจัดเก็บข้อมูลโดยแบ่งเป็น 5 ส่วนดังรูปที่ 2

Part1 : Part2 : Part3 : Part4 : Part5

รูปที่ 2 รูปแบบการจัดเก็บข้อมูลกฎไวยากรณ์ใน XZaQ

จากรูปที่ 2 แสดงรูปแบบการจัดเก็บข้อมูลกฎไวยากรณ์ที่ได้วิเคราะห์จากเอกสารเอ็กซ์เอ็มแอล มีทั้งหมด 5 ส่วน ดังนี้ คือ 1) Element Name ส่วนแสดงชื่อของอิลิเมนต์ 2) Rule Amount แสดงจำนวนกฎไวยากรณ์ของอิลิเมนต์นี้ 3) Rule Number แสดงหมายเลขกฎไวยากรณ์สำหรับอิลิเมนต์นี้ 4) Consequence Amount แสดงจำนวนความสัมพันธ์ของอิลิเมนต์ย่อยภายใต้อิลิเมนต์นี้ 5) Set of Consequence Numbers แสดงเซตของหมายเลขกฎไวยากรณ์ที่สอดคล้องกับจำนวนความสัมพันธ์ทั้งหมดที่เกิดขึ้นภายในเอกสารเอ็กซ์เอ็มแอล

```
<store>
 <book id="1">
    <type>romance</type>
    <title>love actually</title>
    <authors>
        <name>
            <first_name>Jenifer</first_name>
            <last_name> Groon </last_name>
        </name>
    </authors>
  </book>
  <book id="2">
    <type>cartoon</type>
    <title>Harry Potter </title>
    <authors>
        <name>
            <first_name> JK. </first_name>
            <last_name> Rolling </last_name>
        </name>
    </authors>
  </book>
</sotre>
```

รูปที่ 3 ตัวอย่างเอกสาร XML

จากรูปที่ 3 แสดงตัวอย่างโครงสร้างและข้อมูลภายในเอกสารเอ็กซ์เอ็มแอลเมื่อผ่านกระบวนการของ Grammar Generator จะได้ผลลัพธ์เป็นกฎไวยากรณ์ดังแสดงในตารางที่ 2

ตารางที่ 2 กฎไวยากรณ์ของเอกสาร XML

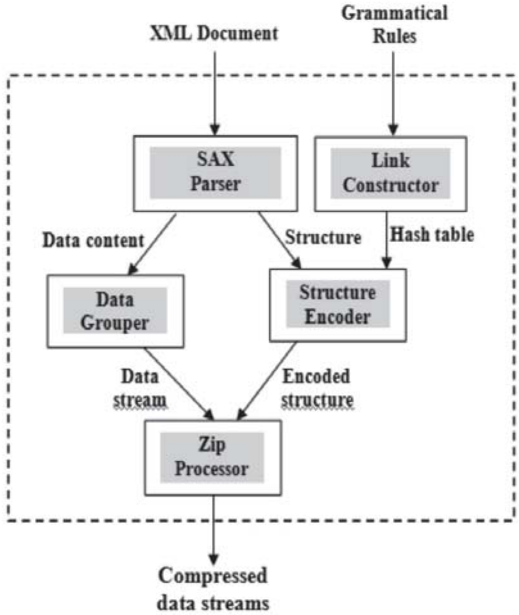
GRAMMAR RULES
store:1:1:1:2
book:1:2:4:3,4,5,6
@id:1:3:1:0
type:1:4:1:0
title:1:5:1:0
authors:1:6:1:7
name:1:7:2:8,9
first_name:1:8:1:0
last_name:1:9:1:0

จากตารางที่ 2 แสดงกฎไวยากรณ์ที่นำมาอธิบายความสัมพันธ์โครงสร้างข้อมูลที่มีอิลิเมนต์ย่อยภายใน เช่น อิลิเมนต์ book มีจำนวน 1 กฎไวยากรณ์ แทนด้วยกฎไวยากรณ์หมายเลข 2 โดยมีอิลิเมนต์ย่อยภายในจำนวน 4 อิลิเมนต์แทนด้วยกฎไวยากรณ์หมายเลข 3 4 5 และ 6 ลำดับเป็นต้น

สำหรับงานวิจัย XZaQ จึงได้นำกฎไวยากรณ์นี้เพื่อเข้ารหัสโครงสร้างข้อมูลและนำมาใช้สำหรับอธิบายความสัมพันธ์ระหว่างโครงสร้างของอิลิเมนต์และแอ็ตทริบิวท์สำหรับการค้นคืนข้อมูลต่อไป

2.3 เครื่องมือบีบอัดเอกสารเอ็กซ์เอ็มแอล (XZaQ Compressor)

สำหรับเครื่องมือบีบอัดเอกสาร XZaQ ใช้วิธีแยกโครงสร้างของเอกสารออกจากข้อมูล เรียกว่า non-homomorphic การเข้ารหัสโครงสร้างข้อมูลใช้วิธีเชิงไวยากรณ์ (Grammar-based Method) โดยกฎไวยากรณ์ใช้ตัวเลขแทนกฎไวยากรณ์ เรียกว่า หมายเลขกฎไวยากรณ์ (Grammar Rules) แล้วนำหมายเลขกฎไวยากรณ์เหล่านั้นมาแทนโครงสร้างของอิลิเมนต์และแอ็ตทริบิวท์ต่าง ๆ ที่ปรากฏภายในเอกสาร สำหรับการบีบอัดข้อมูลภายในเอกสารเอ็กซ์เอ็มแอล XZaQ ใช้วิธีจัดกลุ่มข้อมูลตามพาร์ธ (Path) โดยข้อมูลที่อยู่ในพาร์ธเดียวกันจะถูกจัดกลุ่มไว้รวมกันเป็นสายของข้อมูลเดียวกันเพื่อเข้าสู่กระบวนการบีบอัดเอกสารต่อไป



รูปที่ 4 โครงสร้างภายใน XZaQ Compressor

จากรูปที่ 4 แสดงโครงสร้างภายในของ XZaQ Compressor มีทั้งหมด 5 ส่วนประกอบ ได้แก่ Link Constructor, SAX Parser, Structure Encoder, Data Grouper และ Zip Processor โดยแต่ละส่วนประกอบมีรายละเอียดกระบวนการทำงาน ดังต่อไปนี้

Link Constructor

สำหรับ Link Constructor จะนำข้อมูลกฎไวยากรณ์ที่ได้วิเคราะห์มาขยายความหมายออกมาเป็นลักษณะของตาราง จำนวน 2 ตาราง ได้แก่ ตารางอิลิเมนต์ (Element Table) จะแสดงความสัมพันธ์ระหว่างชื่อของอิลิเมนต์/แอ็ตทริบิวท์

กับหมายเลขของกฎไวยากรณ์ ส่วนตารางลำดับความสัมพันธ์ (Consequence Table) จะแสดงความสัมพันธ์ระหว่างอิลิเมนต์และแอ็ตทริบิวท์ย่อยที่เกิดขึ้นภายในโครงสร้างของเอกสาร

ตารางที่ 3 Element Table

Element	Rule no.
store	1
book	2
@id	3
type	4
title	5
authors	6
name	7
first_name	8
last_name	9

ตารางที่ 4 Consequence Table

Rule no.	Consequence
1	[2]
2	[3, 4, 5, 6]
3	[0]
4	[0]
5	[0]
6	[7]
7	[8, 9]
8	[0]
9	[0]

จากตารางที่ 3 ตารางอิลิเมนต์จะถูกนำไปใช้สำหรับการเข้ารหัสโครงสร้าง และตารางที่ 4 ตารางลำดับความสัมพันธ์ถูกนำไปใช้ในกระบวนการอื่น ๆ ไม่ว่าจะเป็นการบีบอัด การคลาย และการค้นคืนเอกสารต่อไป

SAX Parser

เครื่องมือสำหรับวิเคราะห์เพื่อแจกแจงระหว่างโครงสร้างของเอกสารออกจากข้อมูลโครงสร้างเอกสารจะถูกเปลี่ยนให้อยู่ในรูปแบบของโทเคน (Token) โดยจะตัดส่วนที่เป็นสัญลักษณ์ของแท็กเปิดเหลือเพียงชื่อของอิลิเมนต์หรือชื่อของแอตทริบิวส์ ส่วนแท็กปิดจะเหลือเพียงสัญลักษณ์ “/” สำหรับส่วนที่เป็นข้อมูลจะใช้สัญลักษณ์ “#” แทนตำแหน่งของข้อมูลภายในเอกสารสายโครงสร้างที่ได้จากการวิเคราะห์นี้ คือ *“store, book, @id, #, type, #, /, title, #, /, authors, name, rst_name, #, /, last_name, #, /, /, book, @id, #, /, type, #, /, title, #, /, authors, name, first_name, #, /, last_name, #, /, /, /, ”* โดยสายโครงสร้างนี้จะถูกส่งไปยัง Structure Decoder ต่อไป

สำหรับข้อมูลของเอกสาร SAX Parser แจกแจงตามความสัมพันธ์ของโครงสร้างซึ่งสำหรับงานวิจัยนี้ เรียกว่า พาร์ธ (Path) โดยความสัมพันธ์เริ่มตั้งแต่ รูท อิลิเมนต์เรื่อยมาจนถึงอิลิเมนต์ที่เก็บข้อมูล เช่น พาร์ธ “store/book/@id” หมายถึง ข้อมูลของแอ็ตทริบิวท์ id ทั้งหมด หรือ พาร์ธ “store/book/type” หมายถึง ข้อมูลของอิลิเมนต์ type ทั้งหมด เป็นต้น แต่สำหรับ SAX Parser จะแจกแจงชนิดของพาร์ธทั้งหมดที่เกิดขึ้นภายในเอกสารโดยจะยังไม่จัดกลุ่ม ทั้งนี้เพื่อที่จะรองรับการจัดกลุ่มของข้อมูลที่มีความหมายเหมือนกันไว้ด้วยกันในส่วนกระบวนการของ Data Grouper ต่อไป

Structure Encoder

การเข้ารหัสโครงสร้างเอกสาร Structure Encoder เข้ารหัสด้วยการใช้กฎเชิงไวยากรณ์ โดยนำสายโครงสร้างจากกระบวนการ SAX Parser ที่ประกอบด้วยชื่อของอิลิเมนต์และแอตทริบิวส์มาเข้ารหัสด้วยหมายเลขกฎไวยากรณ์ ซึ่งหมายเลขกฎไวยากรณ์ถูกวิเคราะห์และแจกแจงรายละเอียด

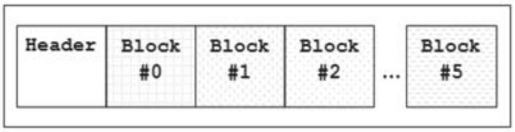
ไว้ในตารางอิลิเมนต์ และตารางลำดับ ดังแสดงไว้ข้างต้น ผลลัพธ์จากการเข้ารหัส คือ โครงสร้างเอกสารที่ผ่านการเข้ารหัสด้วยกฎไวยากรณ์แล้ว คือ *"1, 2, 3, 4, /, 5, /, 6, 7, 8, /, 9, /, /, 2, 3, 4, /, 5, /, 6, 7, 8, /, 9, /, /, /,"* เรียกว่า Structure Encoded โดยจะถูกส่งไปยังกระบวนการบีบอัดข้อมูลต่อไป

Data Grouper

เมื่อข้อมูลถูกแจกแจงตามลักษณะความสัมพันธ์หรือพาร์ธข้อมูลจากกระบวนการ SAX Parser แล้ว Data Grouper จะจัดกลุ่มข้อมูลที่อยู่ภายใต้พาร์ธเดียวกันไว้ในสายข้อมูลเดียวกันทั้งนี้เนื่องจากข้อมูลที่อยู่ภายใต้พาร์ธเดียวกันนั้นย่อมมีคุณลักษณะของข้อมูลที่คล้ายกันเพื่อประสิทธิภาพของการบีบอัดข้อมูลที่มีประสิทธิภาพยิ่งขึ้น

Zip Processor

เครื่องมือสำหรับบีบอัดข้อมูลนี้จะบีบอัดโครงสร้างเอกสารที่เข้ารหัสแล้วและบีบอัดข้อมูลในแต่ละสายข้อมูลที่ผ่านการจัดกลุ่มแต่ละกลุ่มแยกออกจากกันด้วย gzip แล้วนำโครงสร้างและข้อมูลที่บีบอัดแล้วเก็บไว้ในบล็อกข้อมูลการบีบอัด (Compressed Data Block) โดยเรียงตั้งแต่บล็อกหมายเลข 0 หมายเลข 1 หมายเลข 2 ไปเรื่อย ๆ จนครบทุกสายข้อมูล ดังรูปที่ 5



รูปที่ 5 รูปแบบการเก็บบล็อกข้อมูลภายหลังการบีบอัด

จากรูปที่ 5 แสดงการเก็บข้อมูลภายหลังจากการบีบอัดในลักษณะของบล็อกแยกกันเพื่อสนับสนุนการคลายเอกสารเพียงบางส่วน (Partial Decompression) ทำให้ XZaQ สามารถค้นคืน (Query) ข้อมูลที่ผ่านการบีบอัดข้อมูลแล้วโดยการคลายเอกสารเฉพาะส่วนของข้อมูลที่ต้องการค้นคืนเท่านั้นไม่จำเป็นต้องคลายเอกสารทั้งหมดก่อน

2.4 เครื่องมือคลายเอกสารเอ็กซ์เอ็มแอล (XZaQ Decompressor)

การคลายเอกสารของงานวิจัย XZaQ มีกระบวนการที่เป็นส่วนกลับของกระบวนการบีบอัดเอกสารซึ่งประกอบด้วย 3 กระบวนการ ได้แก่ 1) การคลายเอกสารที่ผ่านการบีบอัดแล้วเป็นบล็อกข้อมูล จะใช้เครื่องมือ เรียกว่า Block Decompressor เพื่อการคลายเอกสารส่วนของบล็อกข้อมูลโครงสร้างและส่วนของบล็อกข้อมูลเอกสารทั้งหมดแยกจากกัน แล้วนำส่วนของโครงสร้างเข้าสู่กระบวนการถอดรหัสโครงสร้างถัดไป 2) การถอดรหัสโครงสร้างใช้เครื่องมือ Structure Decoder โดยใช้กฎไวยากรณ์สำหรับถอดรหัสโครงสร้างด้วยการเปลี่ยนจากตัวเลขของหมายเลขกฎไวยากรณ์แทนด้วยชื่อของอิลิเมนต์หรือชื่อของแอตทริบิวส์เดิม จากนั้นนำโครงสร้างที่ถอดรหัสและบล็อกข้อมูลที่คลายแล้วเข้าสู่กระบวนการ สร้างเอกสารเอ็กซ์เอ็มแอลต่อไป 3) การสร้างเอกสารเอ็กซ์เอ็มแอลที่ผ่านการบีบอัดขั้นตอนนี้ใช้เครื่องมือ Document Generator เพื่อสร้างเอกสารเอ็กซ์เอ็มแอล (Uncompressed Document) ซึ่งผลลัพธ์จากการสร้างเอกสารนี้อยู่ในรูปแบบที่เรียกว่า Semantically Lossless ซึ่งหมายถึง เอกสารที่สร้างขึ้นจากกระบวนการคลายเอกสารจะยังคงความหมายเหมือนเดิมทุกประการ

เมื่อเปรียบเทียบกับเอกสาร เอ็กซ์เอ็มแอลก่อนถูกบีบอัดหรือเอกสารต้นฉบับ

2.5 เครื่องมือสำหรับค้นคืนเอกสารเอ็กซ์เอ็มแอล (Query Processor)

งานวิจัย XZaQ นี้จึงได้ออกแบบการบีบอัดเอกสารเอ็กซ์เอ็มแอลเพื่อหลีกเลี่ยงการคลายเอกสารเอ็กซ์เอ็มแอลทั้งหมด (Fully Decomresssion) ทั้งนี้เนื่องจากการคลายเอกสารทั้งหมดนอกจากต้องใช้จำนวนหน่วยความจำเพื่อรองรับข้อมูลจำนวนมาก อีกทั้งต้องใช้เวลามากสำหรับค้นคืนเอกสารที่มีขนาดใหญ่เมื่อเทียบกับเอกสารที่ต้องการค้นคืนเฉพาะเพียงบางส่วน ดังนั้น เครื่องมือสำหรับการค้นคืนเอกสารในงานวิจัย XZaQ รองรับการค้นคืนบนเอกสารเอ็กซ์เอ็มแอลที่ผ่านการบีบอัดแล้วโดยการคลายเอกสารเพียงบางส่วนโดยอาศัยเทคนิคการบีบอัดข้อมูลภายในเอกสารเอ็กซ์เอ็มแอลที่มีความหมายเดียวกันหรือพาร์ธเหมือนกันไว้ในบล็อกข้อมูลเดียวกันทำให้สามารถที่จะเข้าถึงบล็อกข้อมูลเพื่อการค้นคืนข้อมูลดังกล่าวได้โดยตรง

เครื่องมือสำหรับการค้นคืนเอกสาร (Query Processor) ของงานวิจัย XZaQ ใช้ภาษาสำหรับการค้นคืนเอกสารที่เรียกว่า XPath การค้นคืนเอกสาร ด้วย XPath สามารถรองรับข้อคำถาม (Query) ได้ทั้งแบบ Simple Query และ Complex Query โดย Simple Query หมายถึง ข้อคำถามที่ต้องระบุอิลิเมนต์หรือแอ็ตทริบิวท์ของพาร์ธทั้งหมด เช่น หากต้องการค้นคืนข้อมูลชื่อจริงต้องระบุพาร์ธ พาร์ธ คือ *bookstore/book/ authors/ author/ rst_name* และ Complex Query หมายถึง ข้อคำถามอาจจะระบุอิลิเมนต์หรือแอ็ตทริบิวท์ของพาร์ธไม่ครบถ้วนสามารถละข้อมูลบางส่วนได้ เช่น ต้องการค้นคืนข้อมูลชื่อจริง การระบุพาร์ธอาจเป็น *//firstname* หรือ */bookstore/book//first_name* การไม่ระบุอิลิเมนต์หรือแอ็ตทริบิวท์ในข้อคำถามทำได้โดยใช้สัญลักษณ์ “//” แทน ดังนั้น เพื่อให้สามารถรองรับการค้นคืนข้อมูลที่เป็น Complex Query ได้ XZaQ จึงต้องออกแบบการจัดเก็บข้อมูลเพิ่มเติมเรียกว่า Node ID ดังจะอธิบายรายละเอียดถัดไป

การสร้างโหมดไอดี (Implementation of Node ID)

เพื่อรองรับการค้นคืนข้อมูล (Query) บนเอกสารที่ผ่านการบีบอัดแล้ว งานวิจัยนี้จึงได้ออกแบบการเก็บข้อมูลโดยมีชื่อเรียกว่า Node ID ซึ่ง Node ID เป็นข้อมูลที่ประกอบด้วยตัวเลขจำนวนเต็ม 2 จำนวน ได้แก่ Start Id และ End Id สำหรับ Start Id จะถูกกำหนดเมื่อพบแท็กเปิดหรือแอ็ตทริบิวที่ปรากฏอยู่ภายในเอกสารโดยจะเริ่มต้นกำหนดและนับตัวเลขของแท็กเปิดที่เป็นรูทแท็กลำดับแรกจากนั้นก็นับตัวเลขและกำหนดหมายเลขของแท็กเปิดที่ปรากฏอยู่ภายในเอกสารถัดไปเรื่อย ๆ สำหรับ End Id จะถูกกำหนดเมื่อพบแท็กปิดโดยกำหนดเป็นตัวเลขเดียวกับที่กำหนดไว้ในแท็กเปิดที่พบล่าสุดซึ่งข้อมูลของ Node ID นี้เสมือนการเก็บช่วงของอิลิเมนต์ย่อยที่เกิดขึ้นภายในเอกสารทำให้สามารถรองรับการค้นคืนข้อมูลที่เป็นอิลิเมนต์ย่อย (Subtree) หรือ Complex Type ได้ พร้อมทั้งข้อมูลนี้จะถูกนำไปสร้างเป็นเอกสารเอ็กซ์เอ็มแอล ซึ่งเป็นผลลัพธ์จากการค้นคืน

สำหรับการค้นคืนเอกสารเริ่มโดยคำถามที่ต้องการค้นคืน (Query Input) เข้าสู่ 1) กระบวนการ

แจกแจงข้อคำถาม (Query Parsing) เมื่อวิเคราะห์แล้วจะส่งข้อมูลไปยัง 2) กระบวนการประเมินผลเพื่อหาคำตอบของข้อคำถาม (Query Evaluating) โดยเมื่อประเมินได้ข้อมูลในส่วนที่ตรงกับข้อคำถามแล้วก็จะคลายข้อมูลในส่วนนั้นเพียงส่วนเดียว เพื่อส่งข้อมูลผลลัพธ์นี้ต่อไปยัง 3) กระบวนการสร้างผลลัพธ์ (Result Generating) โดยผลลัพธ์ที่ได้จากการค้นคืนนี้จะอยู่ในรูปของเอ็กซ์เอ็มแอล

การแจกแจงข้อคำถาม (Query Parsing)

Query Parser ทำหน้าที่วิเคราะห์ข้อคำถาม ซึ่งข้อคำถามจะใช้ภาษา XPath การค้นคืนของ XZaQ จะรองรับทั้งข้อคำถามอย่างง่าย (Simple Type Query) และข้อคำถามที่ซับซ้อน (Complex Type Query) โดยขั้นตอนการวิเคราะห์เริ่มจากการแจกแจงชนิดของข้อคำถามซึ่งในภาษา XPath จะใช้สัญลักษณ์ "//" แทนข้อคำถามที่ซับซ้อน และใช้สัญลักษณ์ "/" แทนข้อคำถามอย่างง่าย จากนั้นวิเคราะห์ถึงเงื่อนไขของข้อคำถามว่ามีหรือไม่ และขั้นตอนสุดท้าย คือ แจกแจงข้อคำถามออกเป็นโทเคน (Token) เพื่อนำเข้าสู่กระบวนการประเมินผลเพื่อหาคำตอบของข้อคำถามต่อไป

การประเมินผลข้อคำถาม (Query Evaluating)

กระบวนการนี้ถือเป็นกระบวนการหลักของ Query Processor ซึ่งมีเครื่องมือที่เรียกว่า Query Evaluator โดยจะนำโทเคนต่าง ๆ ที่ได้จาก Query Parser มาแทนด้วยหมายเลขของกฎไวยากรณ์ซึ่งได้กล่าวรายละเอียดไว้แล้วในตอนต้น เพื่อนำหมายเลขกฎที่ได้ไปพิจารณาในตารางลำดับ (Consequence Table) เพื่อตรวจสอบว่าหมายเลขกฎนี้เป็นกฎข้อสุดท้ายแล้วหรือไม่ หากเป็นกฎลำดับสุดท้ายแล้วจะนำข้อมูลพาร์ธของกฎนี้ไปยัง Block Decompressor เพื่อดึงข้อมูลเฉพาะกลุ่มนี้ออกมาคลายเท่านั้น (Partial Decompression) กรณีที่ไม่ได้เป็นกฎลำดับสุดท้ายจะต้องนำกฎไปตรวจสอบในตารางลำดับเพื่อหาลำดับของกฎไวยากรณ์ที่อยู่ถัดไป จนกระทั่งพบลำดับของกฎไวยากรณ์ที่เป็นกฎลำดับสุดท้ายจึงนำชื่อพาร์ธของกฎนี้ส่งต่อไปยัง Block Decompressor เพื่อค้นหาและคลายข้อมูลเฉพาะส่วนที่ต้องการออกมาเท่านั้นและจะนำข้อมูลนี้ส่งต่อไปยังกระบวนการสุดท้าย เพื่อแสดงผลการสอบถาม (Query) ในรูปแบบของเอกสารเอ็กซ์เอ็มแอล

สำหรับการค้นคืนเอกสารเอ็กซ์เอ็มแอลที่ผ่านการบีบอัดแล้วนั้น ในงานวิจัย XZaQ สามารถรองรับ Query ได้ 2 ประเภท คือ Simple Query และ Complex Query อย่างเช่น Exact-match Query และ Range Query

การสร้างผลลัพธ์ (Result Generating)

กระบวนการนี้ใช้เครื่องมือ Result Generator สำหรับสร้างผลลัพธ์จากการค้นคืนเอกสารที่ผ่านการบีบอัดแล้วในรูปของโครงสร้างต้นไม้แปลงให้อยู่ในรูปแบบของเอกสารเอ็กซ์เอ็มแอลโดยจะต้องประกอบด้วยแท็กเปิดและแท็กปิดตามลำดับ

```
<output><name><first_name> Jenifer
</first_name><last_name> Groon </last_name>
</name><name><first_name> Sirinthorn
</first_name><last_name>Cheyasak</last_name>
</name><name><first_name> JK. </first_name>
<last_name> Rolling </last_name> </name>
</output>
```

รูปที่ 6 แสดงผลลัพธ์จากการค้นคืนในรูปแบบเอ็กซ์เอ็มแอล

3. ผลการทดลองและวิจารณ์ผล

งานวิจัย XZaQ ได้ทดลองวัดประสิทธิภาพกับเครื่องมือ XMill และ XGrind ภายใต้สภาพแวดล้อมเดียวกันทั้งหมดของเครื่องมือบีบอัด เครื่องมือคลายเอกสาร และเครื่องมือการค้นคืนเอกสาร ได้แก่ อัตราส่วนของการบีบอัด (Compression Ratio) เวลาที่ใช้บีบอัดเอกสาร (Compression Time) เวลาที่ใช้คลายเอกสาร (Decompression Time) และสุดท้ายคือเวลาที่ใช้ค้นคืนข้อมูลบนเอกสารที่ผ่านการบีบอัดแล้ว (Query Response Time)

1) Compression Ratio (CR) คือ การหาอัตราส่วนระหว่างขนาดของเอกสารที่ผ่านการบีบอัดแล้วต่อขนาดของเอกสารเอ็กซ์เอ็มแอลต้นฉบับด้วยสูตร

$$CR = \left(1 - \left[\frac{\text{Size of compression document}}{\text{Size of original document}}\right]\right) \times 100$$

2) (De)compression Time คือ เวลาเฉลี่ยที่ใช้ไปสำหรับการบีบอัด และการคลายเอกสารเอ็กซ์เอ็มแอล โดยการทดลองจะทดสอบการบีบอัดและการคลายเอกสารทั้งหมด 8 ครั้งต่อเอกสาร ซึ่งการทดสอบทั้ง 8 ครั้งนี้จะตัดเวลาที่สูงที่สุดต่ำที่สุด ออกจากนั้นนำการทดสอบ 6 ครั้งที่เหลือหาค่าเฉลี่ยของเวลาที่ใช้สำหรับการบีบอัดและการคลายเอกสาร

3) Query Response Time (QRT) คือ เวลาที่ใช้สำหรับค้นคืนข้อมูล (Query) บนเอกสารที่ผ่านการบีบอัดแล้วสำหรับงานวิจัย XZaQ จะค้นคืนโดยการคลายข้อมูลเพียงบางส่วนเท่านั้น โดยนำผลการทดลองมาเปรียบเทียบกับงานวิจัย XGRIND

3.1 ข้อมูลสำหรับการทดลอง

งานวิจัยครั้งนี้เลือกใช้ชุดเอกสารเอ็กซ์เอ็มแอลสำหรับนำมาทดลองโดยชุดข้อมูลนี้ได้รับความนิยมและเป็นมาตรฐานเพื่อนำมาทดสอบประสิทธิภาพสำหรับงานวิจัยด้านเอ็กซ์เอ็มแอล

ตารางที่ 5 แสดงคุณสมบัติของเอกสารทั้ง 5 ชนิดสำหรับทดสอบ

เอกสาร	ขนาด (ไบท์)	ความลึก	จ.น. แท็ก	จ.น.แอ็ตทริบิวท์
Baseball	655,146	5	45	0
Shakespeare	937,964	6	17	0
DBLP	1,022,987	3	14	2
XMark	1,461,019	9	74	9
Catalog	10,815,037	9	50	3

จากตารางแสดงคุณลักษณะของเอกสารที่ใช้ทดสอบในงานวิจัย XZaQ ประกอบด้วย 1) ขนาด หมายถึง พื้นที่สำหรับใช้เก็บเอกสารเอ็กซ์เอ็มแอลมีหน่วยเป็นไบท์ 2) ความลึก หมายถึงจำนวนความซับซ้อนของอิลิเมนต์ซ้อนอิลิเมนต์ 3) จำนวนแท็กทั้งหมดไม่ซ้ำกันที่ปรากฏภายในเอกสาร 4) จำนวนแอ็ตทริบิวท์ หมายถึง จำนวนแอ็ตทริบิวท์ทั้งหมดไม่ซ้ำกันที่ปรากฏภายในเอกสาร โดยเอกสารเอ็กซ์เอ็มแอลทั้งหมดที่เลือกมาทดสอบเป็นเอกสารที่ถูกนำมาทดสอบในงานวิจัยต่าง ๆ ที่กล่าวมาข้างต้น ได้แก่ Baseball ฐานข้อมูลเกี่ยวกับสถิติของนักเบสบอลของแต่ละทีมในเมเจอร์ลีค Shakespeare ข้อมูลเกี่ยวกับละครของเช็คสเปียร์ DBLP ข้อมูลเกี่ยวกับนักวิจัยและงานวิจัยต่าง ๆ ที่ตีพิมพ์ในวารสาร XMark ข้อมูลที่สร้างขึ้นจาก XML Benchmark Project ซึ่งเป็นมาตรฐานที่ใช้สำหรับทดสอบประสิทธิภาพกันอย่างแพร่หลายในงานวิจัยต่าง ๆ ข้อมูลภายใน XMark

จะประกอบด้วยแท็ก แอ็ตทริบิวท์ จำนวนมาก มีโครงสร้างที่ซับซ้อน และข้อมูลที่บรรจุภายประกอบด้วยข้อความที่มีความยาวหลาย ๆ บรรทัด และ Catalog ข้อมูลเกี่ยวกับรายละเอียดสินค้ารวมถึงข้อมูลลูกค้าสำหรับสั่งซื้อ ซึ่งโครงสร้างข้อมูลจะมีความซับซ้อน และมีขนาดไฟล์ที่ใหญ่

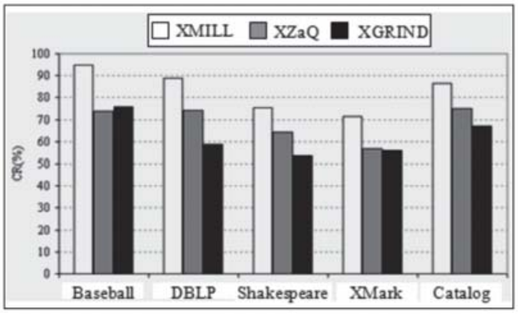
3.2 อัตราส่วนของการบีบอัด (Compression Ratio)

อัตราส่วนของการบีบอัด (CR) เป็นการทดสอบวัดประสิทธิภาพของเครื่องมือสำหรับการบีบอัดเอกสารเอ็กซ์เอ็มแอลของงานวิจัย XZaQ โดยใช้เอกสารเอ็กซ์เอ็มแอลดังตารางที่ 6 เป็นข้อมูลทดสอบสำหรับหาค่าของ CR เพื่อนำมาเปรียบเทียบกับงานวิจัย XMILL และ XGRIND ทั้งนี้เนื่องจากงานวิจัยทั้ง 2 นี้เป็นที่ยอมรับกันอย่างแพร่หลายและมีโปรแกรมสำหรับดาวน์โหลดเพื่อสามารถนำมาทดสอบได้

ตารางที่ 6 แสดงการเปรียบเทียบค่า CR

เอกสาร	Compression Ratio (CR)		
	XMILL	XZaQ	XGRIND
Baseball	95.01%	73.74%	75.79%
Shakespeare	75.48%	74.36%	58.59%
DBLP	88.98%	64.26%	53.71%
XMark	71.49%	56.81%	55.76%
Catalog	86.52%	74.97%	66.96%
ค่าเฉลี่ย	83.50%	68.83%	62.16%

จากตารางข้อมูล แสดงเปอร์เซ็นต์ของการบีบอัดเอกสารเอ็กซ์เอ็มแอลที่นำมาทดสอบโดยเปรียบเทียบระหว่าง XMILL XZaQ และ XGRIND พบว่า XMILL มีเปอร์เซ็นต์ของการบีบอัดมากที่สุด ประมาณ 83.50% ซึ่งมีความหมายว่าเมื่อนำเอกสารเอ็กซ์เอ็มแอลใด ๆ มาผ่านเครื่องมือบีบอัดจะทำให้ขนาดลดลง 83.50 เปอร์เซ็นต์ ดังนั้นขนาดของเอกสารที่บีบอัดแล้วจะเหลือเพียง 26.50 เปอร์เซ็นต์เท่านั้น เนื่องจาก XMILL ออกแบบเพื่อบีบอัดเอกสารเท่านั้นโดยไม่สามารถค้นคืนบนเอกสารที่ผ่านการบีบอัดแล้ว ได้แต่สำหรับ XZaQ และ XGRIND ออกแบบมาเพื่อรองรับการบีบอัดเอกสารพร้อมกับการค้นคืนเอกสารที่บีบอัดแล้ว โดย XZaQ บีบอัดได้โดยเฉลี่ย 68.63 เปอร์เซ็นต์ และ XGRIND บีบอัดเอกสารได้โดยเฉลี่ย 62.16 เปอร์เซ็นต์ตามลำดับ



รูปที่ 7 กราฟแสดงอัตราส่วนการบีบอัด (CR) ของเครื่องมือวิจัยต่อเอกสารทั้งหมดที่ใช้ทดสอบ

จากรูปที่ 7 กราฟแสดงค่าของ CR สำหรับเครื่องมือบีบอัดเอกสาร (Compressor) ทั้ง 3 งานวิจัย โดยใช้เอกสารเอ็กซ์เอ็มแอลสำหรับทดสอบดังแสดงในตารางที่ 5 พบว่า เครื่องมือบีบอัดเอกสารของงานวิจัย XZaQ สามารถบีบอัดเอกสารได้มากกว่าเครื่องมือของ XGRIND ดังนั้น งานวิจัย XZaQ จะบีบอัดเอกสารได้มากกว่า XGRIND ประมาณ 10% ยกเว้นเอกสาร Baseball ทั้งเนื่องจากเอกสารนี้มีข้อมูลส่วนใหญ่ประกอบไปด้วยข้อมูลชนิดตัวเลขจำนวนเต็มทั้งนี้เนื่องจาก XGRIND ใช้การบีบอัดเฉพาะสำหรับข้อมูลที่เป็น

ตัวเลขจึงส่งผลให้ค่าของ Compression Ratio สูงกว่า XZaQ ประมาณ 2%

3.3 ผลการเปรียบเทียบเวลาที่ใช้สำหรับการบีบอัดและการคลายเอกสาร (Compression Time/ Decompression Time)

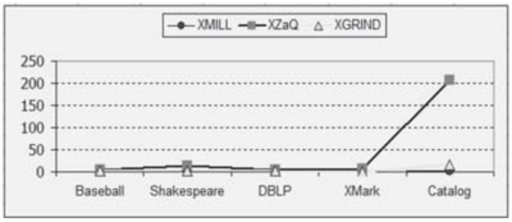
3.3.1 Compression Time

การทดสอบเวลาที่ใช้สำหรับการบีบอัดและการคลายเอกสารเอ็กซ์เอ็มแอลสำหรับทดสอบจะคำนวณเวลาเฉลี่ย 6 ครั้ง จากการรันเครื่องมือแต่ละชนิดทั้งหมดจำนวน 8 ครั้ง โดยตัดเวลาครั้งที่มากที่สุดและน้อยที่สุดออก

ตารางที่ 7 การเปรียบเทียบเวลาที่ใช้บีบอัดเอกสาร

เอกสาร	Compression Time (seconds)		
	XMILL	XZaQ	XGRIND
Baseball	0.122	4.886	1.302
Shakespeare	0.353	13.754	1.694
DBLP	0.133	4.108	1.293
XMark	0.340	5.913	0.479
Catalog	2.300	206.062	15.404

จากตารางการเปรียบเทียบผลการทดลองนี้พบว่า XZaQ ใช้เวลาสำหรับการบีบอัดข้อมูลมากที่สุด ทั้งนี้เนื่องจาก XZaQ พัฒนามาจากภาษา Java ในขณะที่ XMILL และ XGRIND พัฒนาจากภาษา C++ และจากผลการทดลองสามารถสรุปได้ว่าเวลาที่ใช้สำหรับบีบอัดจะแปรผันตามขนาดของเอกสาร และความซับซ้อนของโครงสร้างภายในเอกสาร ตัวอย่างเอกสาร Catalog มีขนาดจำนวนแท็กที่ไม่ซ้ำกัน และความลึกมากที่สุดเมื่อเทียบกับเอกสารทั้งหมด ทำให้ใช้เวลาสำหรับบีบอัดเอกสารมากที่สุดในทุก ๆ เครื่องมือที่ใช้ทดสอบเพื่อแสดงให้เห็นทิศทางของเอกสารและเวลาที่ใช้สำหรับบีบอัดได้อย่างชัดเจนทางผู้วิจัยจึงนำข้อมูลมาแสดงในลักษณะของกราฟ



รูปที่ 8 กราฟเส้นแสดงการเปรียบเทียบเวลาการบีบอัดข้อมูล

จากรูปที่ 8 พบว่า เวลาที่ใช้สำหรับการบีบอัดของเครื่องมือที่นำมาทดสอบในเอกสารทั้ง 5 แบบไปในทิศทางเดียวกัน หากเอกสารมีขนาดใหญ่และมีลักษณะโครงสร้างที่ซับซ้อนของเอกสารจะทำให้มีต่อเวลาที่ใช้บีบอัดมากขึ้นด้วย ดังนั้น จากการทดลองนี้จึงกล่าวได้ว่าเวลาที่ใช้สำหรับบีบอัดเอกสารแปรผันโดยตรงกับขนาดและลักษณะโครงสร้างที่ซับซ้อนของเอกสาร

3.3.2 Decompression Time

การคลายเอกสารเอ็กซ์เอ็มแอลจะคำนวณเวลาเฉลี่ย 6 ครั้ง จากการรันเครื่องมือแต่ละชนิดทั้งหมดจำนวน 8 ครั้ง โดยตัดเวลาครั้งที่มากที่สุดและน้อยที่สุดออกก่อน

ตารางที่ 8 การเปรียบเทียบเวลาที่ใช้คลายเอกสาร

เอกสาร	Decompression Time (seconds)		
	XMILL	XZaQ	XGRIND
Baseball	0.132	2.288	0.937
Shakespeare	0.066	2.953	1.819
DBLP	0.055	2.570	2.493
XMark	0.088	2.554	2.554
Catalog	0.596	30.861	6.302

จากตารางที่ 8 การเปรียบเทียบเวลาสำหรับคลายเอกสาร พบว่า การคลายเอกสารจะเป็นในทิศทางเดียวกับการบีบอัดเอกสาร กล่าวคือ เวลาที่ใช้สำหรับการคลายเอกสารจะแปรผันกับขนาดจำนวนของอิลิเมนต์ และโครงสร้างภายในเอกสารเอ็กซ์เอ็มแอลตัวอย่างเอกสาร Catalog ใช้เวลาสำหรับการคลายเอกสารมากที่สุดของทุกเครื่องมือที่ทดสอบ

3.4 ผลการทดสอบเวลาสำหรับการค้นคืนเอกสาร (Query Response Time: QRT)

งานวิจัย XZaQ ใช้ภาษา XPath สำหรับค้นคืนโดยสามารถรองรับการค้นคืนได้ 3 ประเภทหลัก ๆ ดังนี้ คือ

1) Simple Path Query หมายถึง การค้นคืนจำเป็นต้องระบุ path ทั้งหมดตั้งแต่อิลิเมนต์เริ่มต้นจนถึงอิลิเมนต์สุดท้ายของข้อมูลที่ต้องการ เช่น ต้องการค้นคืนข้อมูลของชื่อผู้แต่งหนังสือภายในเอกสาร ต้องระบุพาร์ธ */books/book/authors/author*

2) Partial Matching Path Expression หมายถึง การระบุอินพุตสำหรับค้นคืนเอกสารสามารถละเว้นชื่อของอิลิเมนต์ได้จำนวน 1 อิลิเมนต์ เช่น ต้องการค้นคืนข้อมูลของชื่อผู้แต่งหนังสือภายในเอกสาร จะต้องระบุ path ดังนี้ */books/book//author* จากตัวอย่างนี้ พบว่า อินพุตไม่ระบุอิลิเมนต์ชื่อ authors ไว้

3) Complicated Partial Matching Path Expression หมายถึง การระบุอินพุตสำหรับค้นคืนเอกสารสามารถละเว้นชื่อของอิลิเมนต์ได้มากกว่า 1 อิลิเมนต์ เช่น ต้องการค้นคืนข้อมูลของชื่อผู้แต่งหนังสือภายในเอกสาร จะต้องระบุ path เป็น */books//author* หรือ *//author* เป็นต้น จากตัวอย่างนี้ พบว่า อินพุตไม่ระบุอิลิเมนต์ชื่อ books book และ authors ไว้

สำหรับการค้นคืนทั้ง 3 ประเภทหลัก XZaQ สามารถรองรับการกำหนดเงื่อนไขได้ทั้งแบบ Exact-match Predicate และ Range Predicate ดังแสดงในตารางที่ 10 อินพุตการค้นคืน ดังนี้

ตารางที่ 9 แสดงตัวอย่างข้อคำถามที่ XZaQ สามารถรองรับสำหรับการค้นคืนโดยคลายข้อมูลเพียงบางส่วน

Query Name	Query Denition
P1	/Products/ProductItem/ProductID
P2	/Products/ProductItem[ProductID = 77]
P3	//ProductItem[ProductID >= 50 and ProductID <= 70]
F1	/FMDataBase/FCCAmRadio/LicenseNo
F2	/FMDataBase//Name
F3	//Name[LastName = JACKSON]
F4	/FMDataBase/FCCAmRadio[Record >= 249009 and Record <= 249015]
ST1	/staff/record/ID
ST2	//record[ID = 3128]
ST3	//record[ID >= 14500 and ID <= 14600]
SD1	/student/record/E_Name
SD2	//record[E_Name = Siriporn Chanprayoon]
SD3	//record[E_Name >= Siriporn Kamtonwong and E_Name <= Siriporn Kowaboot]

การทดลองสำหรับงานวิจัยในครั้งนี้ได้เปรียบเทียบกับเครื่องมือ XMILL และ XGRIND แต่สำหรับถูกออกแบบมาเพื่อการบีบอัดเอกสารให้

เล็กที่สุดเท่านั้นโดยไม่รองรับการค้นคืนข้อมูลบนเอกสารที่ผ่านการบีบอัดแล้วซึ่งหากต้องการค้นคืนจำเป็นต้องคลายเอกสารทั้งหมดก่อน ส่วนเครื่องมือ XGRIND สามารถค้นคืนข้อมูลบนเอกสารเอ็กซ์เอ็มแอลที่ผ่านการบีบอัดแล้วได้โดยตรงโดยไม่ต้องคลายเอกสารก่อน แต่ XGRIND นั้นสามารถรองรับการข้อคำถามได้เพียงการค้นคืน (Query) กรณีที่เป็น “found” และ “not found” หมายถึง การค้นหาข้อมูลใด ๆ ว่าพบหรือไม่พบในภายในเอกสารเท่านั้น

ตารางที่ 10 แสดงการเปรียบเทียบเวลาการค้นคืนเอกสารที่ผ่านการบีบอัดแล้ว

เอกสาร	Query Response Time (seconds)	
	XZaQ	XGRIND
Product	0.446	0.021
FCC	0.405	0.090
STF	0.513	0.025
STD	0.852	0.500

จากตารางที่ 10 แสดงการเปรียบเทียบเวลาที่ใช้สำหรับค้นคืนเอกสารที่ผ่านการบีบอัดระหว่าง XZaQ และ XGRIND พบว่า เวลาที่ใช้ในการค้นคืนของ XGRIND น้อยกว่าของ XZaQ ทั้งนี้เนื่องจาก XGRIND ใช้การเข้ารหัสแบบ Huffman-encoding ซึ่งสามารถรองรับการค้นคืนบนเอกสารโดยไม่ต้องผ่านการคลายเอกสารก่อน ส่วนงานวิจัย XZaQ ใช้เทคนิค gzip ซึ่งต้อง unzip เอกสารก่อนจึงจะสามารถค้นคืนเอกสารได้ แต่สำหรับการค้นคืนของมูลบนเอกสารที่ผ่านการบีบอัดแล้วของ XGRIND นั้นรองรับได้แต่เพียงกรณีที่พบและไม่พบข้อมูลเท่านั้น ในขณะที่ XZaQ ถูกออกแบบมาเพื่อให้สามารถรองรับการค้นคืนทั้งแบบ Simple Type และ Complex Type โดยกำหนดเงื่อนไขทั้งแบบ Exact-match และ Predicate-match ได้

4. สรุป

จากการวิจัย พบว่า การบีบอัดเอกสารเอ็กซ์เอ็มแอลด้วยวิธี Non-homomorphic และวิธีการเชิงไวยากรณ์ทำให้เอกสารมีขนาดเล็กลงโดยเฉลี่ย 70% เมื่อเปรียบเทียบกับขนาดเอกสารก่อนการบีบอัด XZaQ ยังสามารถบีบอัดเอกสารได้มีประสิทธิภาพดีกว่า 10% เมื่อเปรียบเทียบกับ XGRIND ซึ่งบีบอัดด้วยวิธี Homomorphic

งานวิจัย XZaQ ได้ออกแบบการเก็บข้อมูลด้วยโหนดไอดี (Node ID) ได้แก่ Start Id และ End Id เพื่อเก็บความสัมพันธ์ระหว่างอิลิเมนต์หลักและอิลิเมนต์ย่อยภายในโครงสร้างเอกสาร จากข้อมูลความสัมพันธ์ทำให้ XZaQ รองรับการค้นคืนข้อมูล (Query) ได้ทั้งแบบ Simple Type Query และ Complex Type Query ชนิด Exact-match Query หรือ Range Query ด้วยการคลายข้อมูลเพียงบางส่วนเท่านั้นไม่จำเป็นต้องคลายเอกสารทั้งหมดก่อน ในขณะที่ XGRIND สามารถค้นคืนได้เพียงว่า พบ (Found) หรือไม่พบ (Not fount) ข้อมูลในการค้นคืนเท่านั้น แต่ทั้งนี้งานวิจัย XZaQ ถูกพัฒนาขึ้นโดยใช้ภาษาจาวาเพื่อเพิ่มประสิทธิภาพเรื่องของเวลา ควรพัฒนาโดยใช้ภาษา C++ ส่วนประสิทธิภาพของการบีบอัดควรใช้เทคนิคเฉพาะที่เหมาะสมสำหรับข้อมูลแต่ละประเภท เช่น ข้อมูลตัวเลข ข้อมูลตัวอักษร เป็นต้น

5. กิตติกรรมประกาศ

งานวิจัยเรื่องนี้สำเร็จได้คณะผู้วิจัยต้องขอขอบพระคุณคณะวิทยาศาสตร์และเทคโนโลยี

สถาบันส่งเสริมการวิจัยและพัฒนานวัตกรรม มหาวิทยาลัยกรุงเทพ คณะเทคโนโลยีและการสื่อสาร มหาวิทยาลัยมหิดล ที่ช่วยเสนอแนะและอำนวยความสะดวกและให้การสนับสนุนตลอดการวิจัย

6. เอกสารอ้างอิง

Arion, A., Bonifati, A., Costa, G., D'Aguanno, S., Manolescu, I. and Pugliese, A. 2004. **XQueC: Efficient query evaluation over compressed XML data.** In proceedings of the 9th International Conference on Extending Database Technology (EDBT).

Cheng, James. and Wilfred, Ng. 2004. **XQzip: Querying Compressed XML Using Structural Indexing.** In proceedings of EDBT. Pages 219-236.

Harold, E.R. 2011. **Long Baseball Examples from the XML Bible.** [online]. Available from: http://www.cafeconleche.org/examples/baseball

Jun-Ki, Min. Myung-Jae, Park. and Chin-Wan, Chung. 2003. **XPRESS: A Queriable Compression for XML Data.** In proceedings of the ACM SIGMOD.

Liefke, H. and Suciu, D. 2000. **XMILL: an efficient compressor for XML data. In Proceedings of the SIGMOD Conference.** pp. 153-164.

Mairieng, Kesmanas. and Pluempitiwiriyawej, Charnyote. 2003. **XPACK: A Grammar-based XML Document Compression.** The Seventh National Computer Science and Engineering Conference (NCSEC), Chonburi. Thailand.

Salomon, David. 2000. **Data Compression: The Complete Reference.** 2nd Edition. New York: Springer.

Tolani, Pankaj M. and Haritsa, Jayant R. 2002. **XGRIND: A Query-friendly XML Compressor.** In 18th International Conference on Data Engineering (ICDE'02). Pages 225-234.

Wai Yeung, Lam., Wilfred Ng., Wood, Peter T. and Levene, Mark. 2003. **XCQ: XML Compression and Querying System.** Knowledge and Information Systems Volume 10. Number 4 (2006). Pages 421-452.

XML-benchmark project. 2011. **XML-benchmark: An XML Benchmark Project.** [online]. Available from: http://www.xml-benchmark.org/