การบีบอัดข้อมูล XML โดยใช้วิธีจัดเรียงข้อมูลแบบกระชับ
XML Data Compression using Brevity Encoding

ประพันธ์ เลขาโสภณ กานดา รุณนะพงศา
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น
Email: superjing@gmail.com krunapon@kku.ac.th

บทคัดย่อ

ภาษา XMLได้วิวัฒนาการมาเป็นภาษามาตรฐานในการเสนอและแลกเปลี่ยนข้อมูลบนอินเตอร์เน็ตเนื่องจากภาษา XML เป็นภาษาที่ใช้ง่ายมีความยืดหยุ่น และไม่ขึ้นอยู่กับระบบปฏิบัติการของเครื่องคอมพิวเตอร์ แต่ว่าข้อมูลที่เป็นภาษา XML นั้นมักจะมีขนาดใหญ่และมีข้อมูลที่ซ้ำซ้อนอันเนื่องมาจากการใช้แท็กที่ชื่อซ้ำ ๆ ในการอธิบายข้อมูล เอกสาร XML จึงมีขนาดใหญ่ซึ่งทำให้ต้องการพื้นที่ในการจัดเก็บในปริมาณมากและใช้เวลานานในการส่งข้อมูล ดังนั้นงานวิจัยที่ต้องการจะบีบอัดข้อมูลในภาษา XML จึงมีความจำเป็นอย่างยิ่ง งานวิจัยที่ได้ทำมาแล้วนั้น ข้อมูลที่ถูกบีบอัดแล้วอาจจะอยู่ในรูปแบบของไบนารีหรือรูปแบบที่เป็นที่เข้าใจกับผู้พัฒนาเครื่องมือเท่านั้น

งานวิจัยนี้ได้นำเสนอวิธีบีบอัดข้อมูลแบบใหม่ชื่อว่า เอ็กซ์เบร็ววิตี้ (XBrevity) ซึ่งเป็นวิธีการที่ใช้ในการบีบอัดข้อมูลหรือคลายการบีบอัดข้อมูลโดยที่ข้อมูลที่ถูกบีบอัดสามารถเป็นที่เข้าใจได้กับคนทั่วไป การบีบอัดข้อมูลโดยที่เอกสารที่ถูกบีบอัดอยู่ในรูปแบบของภาษา XMLแบบย่อนั้นจะทำให้เอกสารนั้นยังคงข้อดีของภาษา XML ในขณะที่ขนาดของเอกสารเล็กลงข้อมูลที่ถูกบีบอัดอยู่ในรูปแบบของภาษา XML*

คำสำคัญ: เอ็กซ์เอ็มแอล, การบีบอัดข้อมูล

Abstract

XML becomes the standard language for data representation and exchange on the Internet because it is simple, flexible, and platform neutral. However, XML data is often large and verbose since it consists of many repeated tags which are used to self-describe data. The large size of data results in an excessive amount of space required for storing data on the disk space and that of time required for transmitting data over the network. Thus, it is necessary to find an effective compression technique for XML data. Previous compression techniques generate the compressed XML data that is in the binary format or in the proprietary format to only that XML compressor tool. Such format is often incomprehensible to others or difficult to automatically parse.

In this work, we propose XBrevity, an XML compressor which supports compressing and uncompressing XML data. XBrevity adopts a novel encoding method that has the compressed XML data in the XML format. Thus, the compressed XML data still preserve the advantage features of XML but the XML document has the smaller size.

Keywords: XML, Compression, Data exchange

1. บทนำ

ปัจจุบัน XML (Extensible Markup Language) [16] ได้เข้ามามีบทบาทและเป็นมาตรฐานในการแลกเปลี่ยนข้อมูล เนื่องจาก XML มีความสามารถในการอธิบายความหมายของข้อมูลและมีความยืดหยุ่นในการใช้งาน

*http://gear.kku.ac.th/~krunapon/research/xbrevity (จะเปิดให้ดาวน์โหลดได้ถ้าหากบทความได้รับการตีพิมพ์)

การนำ XML มาใช้งานสามารถทำได้โดยการใช้แท็กเป็นตัวกำกับและการตั้งชื่อแท็กที่สื่อถึงความหมายของข้อมูล ทำให้เอกสารที่ถูกสร้างขึ้นเข้าใจได้ง่าย จึงเป็นส่วนสำคัญที่ทำให้การเข้าถึงข้อมูลได้ง่ายขึ้น แต่จากการที่มีการใช้แท็กเข้ามาช่วยในการสื่อถึงความหมายทำให้เกิดการบันทึกแท็ก ชนิดเดียวกันบ่อยครั้งในเอกสาร

```
<?xml version="1.0"?>
<Book>
  <Author>
    <Name> Pissamai </Name>
  </Author>
  <Author>
    <Name> Porntip </Name>
  </Author>
</Book>
```

รูปที่ 1 ตัวอย่างเอกสาร XML

จากรูปที่ 1 จะเห็นได้ว่ามีการบันทึกข้อมูลประเภทเดียวกันหลายครั้ง ซึ่งในแต่ละครั้งจะมีรายละเอียดแตกต่างกัน ด้วยเหตุนี้ ขนาดของเอกสารจึงมีขนาดใหญ่เมื่อเทียบกับขนาดข้อมูลจริงภายในเอกสารนั้น ส่งผลให้สิ้นเปลืองเนื้อที่หากต้องการจัดเก็บเอกสารและสิ้นเปลืองเวลาในการรับส่ง หากต้องการแลกเปลี่ยนข้อมูลระหว่างองค์กร ผ่านระบบเครือข่าย

โดยทั่วไปแล้วข้อมูล XML จะเก็บอยู่ในรูปแบบของไฟล์ ดังนั้นการบีบอัดข้อมูล XML ที่จะมีการนำไปใช้กันได้จริงจะเป็นการบีบอัดไฟล์ XML เนื่องจากข้อมูล XML จำนวนมากจะเก็บไว้ในไฟล์ ดังนั้นการบีบอัดไฟล์ XML จึงมีความจำเป็นอย่างยิ่ง

ปัจจุบัน XML ได้ถูกนำมาใช้งานในหลายสาขาวิชาชีพ ไม่ว่าจะเป็นทางธุรกิจซึ่งมีการนำ ebXML (Electronic Business XML) [15] และ BPEL (Business Process Execution Language) [2] ไปใช้ด้านกราฟฟิกซึ่งมีการนำ SVG (Scalable Vector Graphics) [4] ไปใช้ หรือแม้กระทั่งด้านวิทยาศาสตร์ซึ่งมีการนำภาษา CML (Chemical Markup Language) [11] ไปใช้ ไม่ว่าจะเป็น BPEL, SVG, หรือ CML ต่างก็เป็นภาษา XML ประเภทหนึ่ง

นอกจากนี้แอพพลิเคชั่นอีกอันหนึ่งที่สำคัญของภาษา XML ซึ่งก็คือเว็บเซอร์วิส (Web Service) [21] เป็นซอฟต์แวร์ที่ได้นำเอามาใช้อย่างมากในปัจจุบัน และคาดว่าจะมีการใช้อย่างแพร่หลายมากขึ้นในอนาคต ในปัจจุบันได้มีการนำเว็บเซอร์วิสเอามาใช้ในการให้บริการผ่านอินเตอร์เน็ตโดยบริษัทชั้นนำ อาทิเช่น Yahoo Search Web Services [23], Google Web APIs [6], Amazon Web Services [1], และ eBay API [3] จุดเด่นของเทคโนโลยีเว็บเซอร์วิสคือ การที่มันทำให้โปรแกรมที่พัฒนาโดยภาษาและใช้แพลตฟอร์มที่แตกต่างกันสามารถติดต่อและทำงานร่วมกันได้โดยใช้ภาษา XML เป็นภาษาสื่อกลางในการแลกเปลี่ยนข้อมูล ฉะนั้นจะเห็นได้ว่าข้อมูล XML จะมีปริมาณเพิ่มมากขึ้นและขนาดของข้อมูลของ XML ที่มักจะมีขนาดใหญ่จะส่งผลกระทบต่อประสิทธิภาพการทำงานของแอพพลิเคชั่นที่ใช้ XML เป็นภาษาในการบันทึกข้อมูล

จากปัญหาในเรื่องขนาดของข้อมูลแนวทางที่สามารถนำมาใช้ในแก้ปัญหาได้คือการบีบอัดข้อมูล XML (XML data compression) เพื่อลดขนาดของเอกสารซึ่งเป็นการบีบอัดข้อมูล XML โดยเฉพาะทำให้สามารถเพิ่มประสิทธิภาพในการบีบอัดได้ดีกว่าการใช้วิธีการบีบอัดข้อมูลทั่วไป

ในบทความนี้ในหัวข้อที่ 2 จะได้นำเสนอกระบวนการบีบอัดแบบต่างๆ ที่มีอยู่ในปัจจุบันซึ่งได้แก่ XMill, XGrind, XPRESS และ XPACK จากนั้นหัวข้อที่ 3 จะมีเนื้อหาในส่วนของแนวทางในการพัฒนาเทคนิคใหม่ๆในการบีบอัดข้อมูล ผลการทดลองจะนำเสนอในหัวข้อที่ 4 และบทสรุปอยู่ในหัวข้อที่ 5

2. งานวิจัยที่เกี่ยวข้อง

โดยปกติแล้วในการบีบอัดไฟล์ข้อมูลทั่วไปนั่นจะนิยมใช้ gzip [5] เนื่องจากเป็นซอฟต์แวร์ที่นำไปใช้ได้โดยไม่ต้องเสียค่าใช้จ่าย อีกทั้งสามารถจะบีบอัดข้อมูลได้ดีและไม่จำเป็นจะต้องมีข้อมูลเกี่ยวกับโครงสร้างของเอกสาร แต่ว่าการบีบอัดไฟล์ XML โดยใช้ gzip นั้นมีข้อจำกัดที่ gzip ไม่สามารถตรวจสอบพบอีลิเมนต์ที่ซ้ำ ๆ ที่อาจจะไม่ได้อยู่ติดกัน

งานวิจัยที่กล่าวถึงการบีบอัดข้อมูล XML ในปัจจุบันนั้นมีอยู่หลายวิธีการได้แก่ XMill [8], XGrind [14], XPRESS [10] และ XPACK [9] ซึ่งมุ่งเน้นในการลดขนาดของข้อมูลที่เป็น XML ให้เล็กลง โดยแต่ละวิธีจะใช้เทคนิคที่แตกต่างกันไป

2.1 XMill

เป็นวิธีการแรกที่ได้มีการนำเสนอวิธีการบีบอัดเอกสาร XML โดยเริ่มจากการแยกส่วนแท็กซึ่งภายในประกอบไปด้วยอิลิเมนต์และแอตทริบิวต์ออกจากข้อมูลที่เป็นตัวอักขระ จากนั้นจะทำการจัดความสัมพันธ์ของกลุ่มข้อมูลเป็น containers โดยจัดให้ข้อมูลที่เป็นแบบเดียวกันอยู่ในกลุ่มเดียวกัน ในขั้นตอนต่อมาจะนำ containers แต่ละตัวมาทำการบีบอัดเข้าไป จากกันแล้วทำการบีบอัดทั้งหมดโดยอาศัย gzip [5] เพื่อให้ได้ข้อมูลที่ออกมาเป็นไฟล์เดียว

เนื่องจากการจัดกลุ่มข้อความต้องอาศัยความเข้าใจของความหมายของข้อมูลซึ่งขึ้นอยู่กับชนิดของแอพพลิเคชั่น ฉะนั้น XMill จึงมักต้องให้ผู้ใช้โปรแกรมพิจารณาความหมายของข้อมูล อีกทั้งผู้ใช้โปรแกรมไม่สามารถค้นหาข้อมูลที่ต้องการจากเอกสารที่ถูกบีบอัดแล้ว ถึงกระนั้นก็ตาม XMill ถือว่าเป็นบทความวิจัยแรก ๆ ที่ทำให้ผู้วิจัยทั้งหลายตระหนักถึงความสำคัญของปัญหาและวิธิการแก้ปัญหาในการบีบอัดข้อมูล XML ซึ่งได้มีผู้นำเทคนิคบางเทคนิคของ XMill ไปใช้ในการเก็บและค้นหาข้อมูล XML [12]

2.2 XGrind

XGrind มีข้อดีที่ XMill ซึ่งก็คือผู้ใช้ XGrind สามารถค้นหาข้อมูลได้จากเอกสารที่ถูกบีบอัด คุณสมบัตินี้เป็นผลมาจากการที่ข้อมูลที่ถูกบีบอัดแล้วยังคงมีโครงสร้างของเอกสารเดิม

แต่อย่างไรก็ตามผู้ใช้โปรแกรม XGrind ไม่สามารถตอบคำถามบางประเภทโดยไม่ได้คลายการบีบอัดของข้อมูล ตัวอย่างของคำถามที่ต้องมีการคลายการบีบอัดข้อมูลคือคำถามที่ถามช่วงของค่า (range query) หรือคำถามที่ถามหาข้อมูลค้นหาบางส่วนที่ตรงกับข้อมูลที่มีอยู่ (partial match)

นอกจากนี้ XGrind จะบีบอัดเฉพาะเอกสาร XML ที่มีเอกสาร DTD (Document Type Definition) ซึ่งเป็นเอกสารที่ระบุโครงสร้างของเอกสาร XML ซึ่งเอกสาร XML บางเอกสารอาจจะไม่มี DTD ก็ได้ ฉะนั้นผู้ใช้โปรแกรม XGrind จะต้องสร้าง DTD สำหรับเอกสาร XML ที่ยังไม่มี DTD

2.3 XPRESS

XPRESS มีข้อดีเช่นเดียวกับ XGrind ตรงที่สามารถจะค้นหาข้อมูลจากเอกสารที่ถูกบีบอัดแล้วแต่ว่า XPRESS ไม่ได้ใช้ DTD

XPRESS ได้นำเสนอแนวคิดใหม่ที่ใช้ reverse arithmetic encoding เป็นวิธีการในการจัดเรียงข้อมูลเพื่อให้การหาคำตอบสำหรับ XPath expressions เป็นไปได้อย่างมีประสิทธิภาพ นอกจากนี้ XPRESS ได้พัฒนาการหาชนิดของข้อมูลโดยไม่ต้องอาศัยข้อมูลอินพุตจากผู้ใช้โปรแกรม

แต่อย่างไรก็ตาม XPRESS มีข้อจำกัดที่ XPRESS จะไม่สามารถเข้าใจเอกสาร XML ที่มีการใช้ ID และ IDREF และไม่มีวิธีการขยายข้อมูลที่ถูกบีบอัดให้เป็นข้อมูล XML ปกติ

2.4 XPACK

เป็นการบีบอัดเอกสาร XML โดยใช้วิธีการเชิงไวยากรณ์ในการบีบอัดและการขยายเอกสาร XML

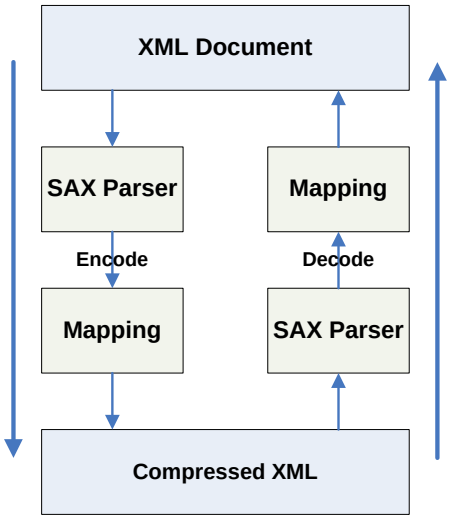
ส่วนประกอบหลักของ XPACK คือ Grammar Generator ทำหน้าที่ในการสร้างกฎ ไวยากรณ์ ส่วนที่สองคือ Compressor ทำหน้าที่บีบอัดเอกสาร และในส่วนสุดท้ายจะเป็น Decompressor ซึ่งทำหน้าที่ขยายเอกสารที่ผ่านการบีบอัดโดยอาศัยโครงสร้างเดิมของเอกสาร

ข้อจำกัดของ XPACK ก็คือโปรแกรมไม่สามารถบีบอัดจัดการกับเอกสาร XML ที่มีอิลีเมนต์เป็นแบบ mixed-content ซึ่งคืออิลีเมนต์ที่มีทั้งอีลิเมนต์และตัวอักขระอยู่ข้างในและผู้ใช้โปรแกรมไม่สามารถค้นหาข้อมูลจากเอกสาร XML ที่ถูกบีบอัดแล้ว

3. เทคนิคการบีบอัดแบบ XBrevity

ในเทคนิคการบีบอัดด้วย XBrevity นั้นจะประกอบไปด้วย ส่วนของการบีบอัดเอกสาร XML (Compressor) และการขยายเอกสาร (Decompressor) เพื่อทำให้เอกสารที่ถูกบีบอัดด้วย XBrevity กลับมาเป็นเอกสาร XML เดิมได้

โครงสร้างการทำงานของ XBrevity แสดงไว้ในรูปที่ 2



รูปที่ 2 แสดงโครงสร้างการทำงานของ XBrevity

3.1 โปรแกรมบีบอัดข้อมูล (Compressor)

มีหน้าที่ในการบีบอัดเอกสาร XML ซึ่งจะใช้ SAX Parser เป็นตัวอ่านวิเคราะห์เอกสารที่จะทำการบีบอัดเข้ามา ในระหว่างที่ทำการอ่านวิเคราะห์เอกสารอยู่นั้นจะทำการเข้ารหัสโดยการ Mapping เพื่อเปลี่ยนรูปแบบให้เป็นตัวอักษรแล้วเขียนค่าลงไปบนไฟล์ โดยการสร้างไฟล์ที่มีข้อมูลที่ถูกบีบอัดแล้ว โดยในเอกสารใหม่ที่ได้ประกอบไปด้วย <d>…</d> ซึ่งเป็นส่วนของข้อมูลที่ถูกบีบอัดแล้วกับ <m …/> ซึ่งเป็นส่วนของการอธิบายความหมายของข้อมูลภายในเอกสาร (Metadata) ซึ่งเอกสารที่ถูกบีบอัดนี้สามารถไปจัดเก็บหรือแลกเปลี่ยนข้อมูลระหว่างเครือ ข่ายได้ และสามารถเข้าใจได้ง่ายโดยที่ไม่ต้องขยายข้อมูลกลับไปเป็นต้นฉบับเดิม เนื่องจากอยู่ในรูปแบบของเอกสาร XML แล้วและมีตัวอธิบายข้อมูลอยู่ภายในให้หรือจะทำการขยายเอกสารให้เป็นเอกสารเดิมได้ ด้วยตัวขยายเอกสาร (Decompressor) ที่ออกแบบไว้

3.2 โปรแกรมคลายการบีบอัดข้อมูล (Decompressor)

เมื่อเอกสารที่ถูกบีบอัดด้วย XBrevity ต้องการขยายให้กลับมาเป็นเอกสาร XML ต้นฉบับสามารถทำได้โดยใช้ SAX Parser ที่ออกแบบมาเพื่อวิเคราะห์เอกสารโดยถอดรหัสเอกสารที่ถูกบีบอัดนั้นออกมา แล้วทำการ Mapping ค่าที่ได้ ในแท็ก <m …/> แล้วนำค่าที่อ่านได้ไปแทนที่ตัวแปรเดิมที่อยู่ภายใน <d>…</d> เพื่อให้กลับออกมาเป็นเอกสารต้นฉบับ

3.3 ตัวอย่างการทำงานของ XBrevity

ในงานวิจัยนี้ได้นำเสนอการบีบอัดข้อมูลโดยไม่ใช้ Schema ซึ่งแสดงตามรูปที่ 3 ซึ่งเป็นตัวอย่างของเอกสารที่ประกอบไปด้วย อิลิเมนต์ และแอตทริบิวส์ต่างๆ จะเห็นได้ว่ามีการใช้งานของแท็กที่ซ้ำๆกัน แต่ข้อมูลภายในแตกต่างกัน ทำให้โครงสร้างดูซับซ้อนและเอกสารมีขนาดใหญ่

```
<?xml version="1.0"?>
  <bib>
    <article>
      <authors>
        <author aid="a1" eid="e1">
          <name>Pissamai<nickname>Aom
          </nickname></name>
        </author>
        <author aid="a2">
          <name>Porntip</name>
        </author>
      </authors>
      <year>2005</year>
    </article>
    <article>
      <authors>
        <author aid="a3">
          <name>Thongchai</name>
        </author>
      </authors>
    </article>
  </bib>
```

รูปที่ 3 เอกสาร XML ตัวอย่าง A

ในการบีบอัดข้อมูลนั้นจะมีข้อมูลเมต้าดาต้า (Metadata) เพื่อใช้อ้างอิงอิลิเมนต์และแอตทริบิวส์ ซึ่งทำให้โครงสร้างที่มีอยู่เดิมสั้นลงทำให้ขนาดของเอกสารเล็กลงด้วย

```
<?xml version="1.0"?>
<c>
  <d>
    <e1e2e3 a4="a1" a5="e1" e6v="Pissamai"
 e7v="Aom"/>
    <xxe3 a4="a2" e6v="Porntip"/>
    <xe8 v="2005"/>
    <e1e2e3 a4="a3" e6v="Thongchai"/>
  </d>
  <m f="bib article authors author aid eid name nickname
year " b="0 1 2 3 a4 a5 6 7 8 "/>
</c>
```

รูปที่ 4 เอกสาร XML ตัวอย่าง A ที่ถูกบีบอัดแล้ว

โดยที่ **c** = การบีบอัด (Compressed)

d = ข้อมูล (Data)

m = ข้อมูลที่อธิบายข้อมูล (Metadata)

f = ชื่อเต็ม (Full name)

b = ชื่อย่อ (Brieviation name)

v = ค่าของข้อมูล (Value)

x = ใช้การเรียงซ้อนของอีลิเมนต์ที่พบก่อนหน้านี้

จากรูปที่ 4 ก่อนการบีบอัดข้อมูล (แท็ก **c**) จะทำการ mapping ข้อมูลต่างๆให้เป็นตัวแปร และทำให้ข้อมูลเมื่อบีบอัดแล้วจะเห็นได้ว่ามีขนาดเล็กลง ซึ่งข้อมูลจริงจะถูกบีบอัดอยู่ในแท็ก **d** แท็กที่ขึ้นต้นด้วย **e** เป็นแท็กที่เป็นชื่อย่อของอีลิเมนต์ ส่วนแท็กที่ขึ้นต้นด้วย **a** เป็นแท็กที่เป็นชื่อย่อของแอตตริบิวส์ **v** เป็นตัวอักษรที่ชี้ให้เห็นว่าจะมีการเก็บค่าที่อยู่ในอิลิเมนต์ ตัวอักษร x เป็นตัวอักษรที่บ่งบอกว่า อิลิเมนต์ปัจจุบันจะใช้การเรียงลำดับของการซ้อนกันของอิลิเมนต์ที่เป็นบรรพบุรุษ (ancestor elements) ของอิลิเมนต์ที่เพิ่งพบก่อนหน้านี้ อย่างเช่น < *e1e2e3 a4="a1" …/>* ตามด้วย <xxe3 …/>

นั้นบ่งบอกว่า <xxe3 …/> มีการเรียงลำดับอิลิเมนต์เป็น <e1e2e3 …/> ในขณะที่ <xe8 …/> นั้นมีการจัดเรียงลำดับอิลิเมนต์เป็น <e1e8 …/> ส่วน e0 ไม่จำเป็นต้องอยู่ในข้อมูลของการจัดเรียงเนื่องจากในแต่ละเอกสารมี root element มีเพียงตัวเดียว ดังนั้นในเอกสารที่ถูกบีบอัดจึงมี <e1e2e3 …/> แทนที่จะเป็น <e0e1e2e3 …/> จะเห็นได้ว่าโครงสร้างยังคงเป็นเอกสาร XML จากการบีบอัดด้วยวิธีการนี้จะเห็นได้ว่า เอกสารใหม่มีขนาดเล็กลงกว่าต้นฉบับ อีกทั้งยังเป็นเอกสารที่สามารถอ่านเข้าใจได้ง่าย และยังอยู่ในรูปแบบของเอกสาร XML ที่ถูกต้องตามหลักไวยากรณ์ของภาษา XML ด้วย (Well-formed XML)

เอกสาร XML อีกตัวอย่างหนึ่งเป็นเอกสารที่เก็บข้อมูลของบทความวิจัยต่างๆ เอกสารตัวอย่างในลักษณะนี้แสดงในรูปที่ 5

```
<?xml version="1.0"?>
<bib>
  <inproceedings>
    <authors>
      <author>N. Zhang</author>
      <author>V. Kacholia</author>
      <author>M. T. Ozsu</author>
    </authors>
    <title>A Succinct Physical Storage Scheme for Efficient Evaluation for Path Queries in XML</title>
    <booktitle>Proceedings of the IEEE International Conference on Data Engineering</booktitle>
    <month>March</month>
    <year>2004</year>
    <pages>55-65</pages>
  </inproceedings>
  <inproceedings>
    <authors>
      <author>B. B. Yao</author>
      <author>M. T. Ozsu</author>
      <author>N. Khandelwal</author>
    </authors>
    <title>XBench Benchmark and Performance Testing of XML DBMSs</title>
    <booktitle>Proceedings of the IEEE International Conference on Data Engineering</booktitle>
    <month>March</month>
    <year>2004</year>
    <pages>621-632</pages>
  </inproceedings>
</bib>
```

รูปที่ 5 เอกสาร XML ตัวอย่าง B

หลังจากใช้โปรแกรม XBrevity ทำการบีบอัดข้อมูลแล้วจะได้เอกสารที่ถูกบีบอัดดังแสดงในรูปที่ 6

```
<?xml version="1.0"?>
<c>
  <d>
    <e1e2e3 v="N. Zhang"/>
    <xxe3 v="V. Kacholia"/>
    <xxe3 v="M. T. Ozsu"/>
    <xe4 v="A Succinct Physical Storage Scheme for Efficient Evaluation for Path Queries in XML"/>
    <xe5 v="Proceedings of the IEEE International Conference on Data Engineering"/>
    <xe6 v="March"/>
    <xe7 v="2004"/>
    <xe8 v="55-65"/>
    <e1e2e3 v="B. B. Yao"/>
    <xxe3 v="M. T. Ozsu"/>
    <xxe3 v="N. Khandelwal"/>
    <xe4 v="XBench Benchmark and Performance Testing of XML DBMSs"/>
```

```
    <xe5 v="Proceedings of the IEEE International Conference on Data Engineering"/>
    <xe6 v="March"/>
    <xe7 v="2004"/>
    <xe8 v="621-632"/>
  </d>
  <m f="bib inproceedings authors author title booktitle month year pages " b="0 1 2 3 4 5 6 7 8 "/>
</c>
```

รูปที่ 6 เอกสาร XML ตัวอย่าง B ที่ถูกบีบอัดแล้ว

4. การทดลอง

ในการทดลองการบีบอัดข้อมูล ได้ทำการสร้างเอกสาร XML ตัวอย่างโดยสร้างจากแหล่งดาวน์โหลดตามข้อมูลที่อ้างอิงไว้หรืออาจใช้ XMark [13] ในการสร้างไฟล์ขึ้นมาก็ได้ ซึ่งขนาดเอกสารที่นำมาทดสอบจะมีค่าต่างๆกัน และนำเอกสารที่ได้นี้ไปทดสอบกับ gzip และ XMill เพื่อเปรียบเทียบประสิทธิภาพการบีบอัดข้อมูล และพัฒนาระบบการบีบอัด โดยการทดลองนี้ได้ทำบน Intel Pentium4 1.80GHz หน่วยความจำ 256 MB ระบบปฏิบัติการ Windows XP Professional with Service Pack 2 และใช้ภาษาจาวา (J2SE 5.0) ในการสร้างและพัฒนาระบบการบีบอัด แต่ในการทดลองนี้ไม่ได้ทดสอบร่วมกับ XGrind, XPESS และ XPACK เนื่องจาก XGrind เป็นโปรแกรมที่ใช้ทดสอบบน Linux เท่านั้น ส่วน XPRESS และ XPACK ไม่มีโปรแกรมให้ดาวน์โหลดเพื่อทดสอบ

4.1 การวัดประสิทธิภาพของการบีบอัด

ประสิทธิภาพของการบีบอัดคือ อัตราส่วนในการบีบอัด ซึ่งหาได้จาก

$$\text{อัตราส่วน} = 1 - \frac{\text{ขนาดของเอกสารที่บีบอัดแล้ว}}{\text{ขนาดของเอกสารต้นฉบับ}} \quad (1)$$

ตารางที่ 1 คุณลักษณะของเอกสารที่ใช้ในการทดสอบ

XML Files	Size (bytes)	Depth	Elements	Attributes
f1.xml[1]	669,309	7	46	0
f2.xml[2]	4,222,646	7	199	0
f3.xml[3]	1,166,916	6	15	1
f4.xml[4]	251,865	6	17	0
f5.xml[5]	3,021,921	3	12	0

จากข้อมูลในตารางที่ 1 f1.xml, f2.xml, …, f5.xml เป็นชื่ออ้างอิงถึงเอกสาร XML ที่ใช้ในการทดสอบซึ่งเอกสารเหล่านี้ผู้อ่านบทความสามารถดาวน์โหลดได้ตามเว็บไซต์ที่ได้ระบุไว้ นอกจากนี้ในตารางที่ 1 ยังมีข้อมูลซึ่งบ่งบอกชนิดของข้อมูลดังนี้ *Size* ขนาดของเอกสาร XML เป็นไบต์ (bytes) *Depth* ความลึกหรือจำนวนชั้นสูงสุดของการจัดเรียงอีลิเมนต์ในเอกสาร *Element*s คือจำนวนอิลิเมนต์ในเอกสารที่เป็นชื่อเดียวกัน *Attributes* คือจำนวนแอตทริบิวต์ในเอกสารที่เป็นชื่อเดียวกัน

เมื่อทำการบีบอัดในแต่ละวิธีแล้ว ผลที่ได้ปรากฏอยู่ในตารางที่ 2 ซึ่งแสดงขนาดของเอกสารที่ถูกบีบอัดกับเอกสารต้นฉบับ

ตารางที่ 2 การเปรียบเทียบขนาดของเอกสารที่บีบอัดแล้ว

XML Files	**Compressed File Size (bytes)**		
	gzip	**XMill**	**XBrevity**
f1.xml	66,752	35,528	505,238
f2.xml	1,002,187	700,137	3,103,734
f3.xml	114,361	83,618	897,109
f4.xml	68,000	64,184	226,246
f5.xml	127,509	74,439	1,868,296

จากตารางที่ 2 จะเห็นได้ว่าขนาดที่บีบอัดด้วย gzip และ XMill มีขนาดเล็กมากเมื่อเทียบกับ XBrevity ส่วนของแท็ก

[1]ดาวน์โหลดได้จาก http://www.ibiblio.org/xml/examples/1998statistics.xml

[2]ดาวน์โหลดได้จาก http://www.w3.org/XML/Binary/2005/03/test-data/Over100K/factbook.xml

[3]ไฟล์ใน /examples/shakespeare.xml ของ Xmill ดาวน์โหลดได้จาก http://www.research.att.com/sw/tools/xmill/

[4]ดาวน์โหลดได้จาก http://www.eda.org/pub/ibis/xml/sample1/format.xml

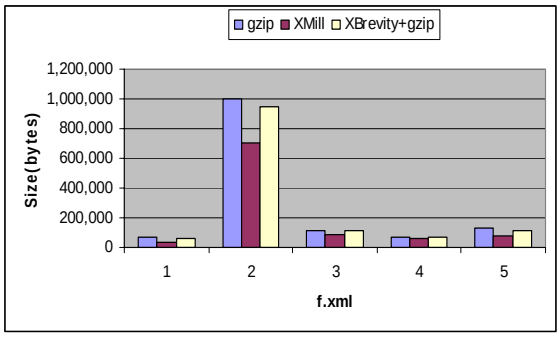
[5]ดาวน์โหลดได้จาก http://gnosis.cx/download/weblog.xml

เท่านั้น ซึ่งผู้วิจัยกำลังพัฒนาเพิ่มเติมเพื่อให้บีบอัดส่วนที่เป็นข้อความได้ด้วย

แต่อย่างก็ตามเมื่อเรานำเอา XBrevity ที่ได้ทำการจัดเรียงข้อมูลแบบกระชับแล้ว มาใช้ร่วมกับ gzip จะได้ผลการทดลองดังตารางที่ 3 และกราฟแสดงขนาดที่ได้จากการบีบอัดจากรูปที่ 7

ตารางที่ 3 การเปรียบเทียบขนาดเมื่อ XBrevity+gzip

XML Files	Compressed File Size (bytes)		
	gzip	**XMill**	**XBrevity+gzip**
f1.xml	66,752	35,528	63,612
f2.xml	1,002,187	700,137	951,463
f3.xml	114,361	83,618	109,905
f4.xml	68,000	64,184	66,815
f5.xml	127,509	74,439	115,960



รูปที่ 7 กราฟแสดงขนาดการบีบอัดเมื่อ XBrevity+gzip

จากการนำ XBrevity รวมกับ gzip ทำให้ประสิทธิภาพในการบีบอัดทำได้ดีขึ้นกว่าการใช้ XBrevity เพียงอย่างเดียว เนื่องจากว่าเมื่อ เอกสารที่ผ่านการจัดเรียงแบบกระชับด้วย XBrevity แล้วนั้นขนาดของเอกสารจะเล็กกว่าต้นฉบับเดิมเมื่อรวมกับ gzip จึงทำให้ขนาดที่บีบอัดได้เล็กกว่า gzip ซึ่งผลที่ได้แสดงไว้ในตารางที่ 3 และรูปที่ 7 แล้ว แต่เมื่อเทียบกับ XMill แล้วยังใหญ่กว่าเล็กน้อย

ตารางที่ 4 การเปรียบเทียบอัตราส่วนในการบีบอัด

XML Files	**Compression Ratio (%)**		
	gzip	**XMill**	**XBrevity+gzip**
f1.xml	90.0	94.7	90.5
f2.xml	76.3	83.4	77.5
f3.xml	90.2	92.8	90.6
f4.xml	73.0	74.5	73.5
f5.xml	95.8	97.5	96.2
Average	85.0	88.6	85.7

จากตารางที่ 4 จะเห็นได้ว่าเมื่อขนาดของ XBrevity ที่รวมกับ gzip มีนาดเล็กลงทำให้อัตราส่วนในการบีบอัดสูงขึ้นตามไปด้วย ดังนั้นเวลาที่ใช้ในการบีบอัดข้อมูลก็ลดลงด้วย ซึ่งผลที่ได้แสดงในตารางที่ 5

ตารางที่ 5 การเปรียบเทียบเวลาในการบีบอัด

XML Files	Compression Time (seconds)		
	gzip	**XMill**	**XBrevity+gzip**
f1.xml[1]	1.5	1	1.5
f2.xml[2]	3	2.5	2.9
f3.xml[3]	1	0.8	1
f4.xml[4]	0.6	0.3	0.5
f5.xml[5]	2.5	2	2.4

และเมื่อมีการนำไฟล์ที่ถูกบีบอัดในตารางที่ 5 มาทำการขยายออกเป็นเอกสารต้นฉบับจะได้เวลาที่ใช้ในการขยายผลที่ได้คล้ายคลึงไปในทางเดียวกันกับการบีบอัดเอกสาร แต่การขยายจะเร็วกว่าเล็กน้อย และในส่วน XBrevity ที่ใช้การบีบอัดรวมกับ gzip มีคุณสมบัติเดียวกันกับ gzip แต่ขนาดของไฟล์เล็กกว่าทำให้การขยายไฟล์จึงทำได้เร็วกว่า gzip

จากผลการทดลองที่ได้นั้น จะเห็นได้ถ้าเปรียบเทียบอัตราส่วนของการบีบอัดแล้ว XMill และ gzip มีค่าค่อนข้างสูง และระยะเวลาในการบีบอัดก็ใช้เวลาที่น้อย เมื่อเทียบกับ XBrevity เพราะ XBrevity บีบอัดเฉพาะส่วนของแท็กเท่านั้น แต่ถ้านำ XBrevity ที่ได้ทำการบีบอัดแล้วมาใช้ร่วมกับ gzip จะเห็นว่ามีขนาดของข้อมูลที่ลดลงมาค่อนข้างมาก รวมถึงเวลาที่ใช้ในการบีบอัดก็เร็วกว่าเดิม ซึ่งผลที่ได้ใกล้เคียงกับการใช้ gzip และ XMill และจะเห็นได้ว่าเวลาที่ใช้ในการขยายไฟล์ออกเป็นเอกสารต้นฉบับจะน้อยกว่าการบีบอัด ไม่ว่าจะใช้วิธีการใด

เนื่องจาก XBrevity ต้องเขียนข้อมูลที่อ่านได้ลงไปบนไฟล์ที่อยู่ในรูปแบบภาษา XML ด้วยจึงทำให้ใช้เวลานานอีกทั้งข้อมูลที่เขียนลงไปนั้นเป็นรูปย่อของเอกสาร แต่ทำให้ยังคงข้อดีของภาษา XML ไว้ อีกทั้งยังเป็นผลดีในแง่ของการอ่านทำความเข้าใจเอกสารที่บีบอัดแล้วได้ง่าย เพราะว่าในการนำไปใช้งานจริงนั้น การที่ผู้ใช้ปลายทางสามารถถอดรหัสที่ถูกบีบอัดออกมาให้เป็นข้อมูลเดิมได้นั้น จำเป็นจะต้องใช้ตัวขยายข้อมูลจากข้อมูลที่ถูกบีบอัดแล้ว ซึ่งอาจทำให้ไม่สะดวกกับการใช้งาน และเป็นข้อจำกัดในการนำไปใช้งานกับกลุ่มคนทั่วไป

5. สรุป

การบีบอัดข้อมูล XML มีความสำคัญเพราะ XML เป็นภาษาที่มีข้อดีอยู่หลายประการ ซึ่งปัจจุบันได้มีการใช้กันอย่างแพร่หลาย แต่ข้อจำกัดของ XML ก็มีเช่นกัน โดยปกติแล้วเอกสารภาษา XML จะมีขนาดใหญ่กว่าเอกสารของเท็กซ์ไฟล์ (Text File) เนื่องจากเอกสารภาษา XML มักจะมีการใช้แท็กกำกับความหมายของข้อมูล การส่งข้อมูลโดยใช้ภาษา XML ทำให้เกิดความต้องการเน็ตเวิร์กแบนด์วิดธ์ (Network Bandwidth) ขนาดใหญ่เนื่องจากขนาดของไฟล์ วิธีหนึ่งที่จะช่วยลดเน็ตเวิร์กแบนด์วิดธ์ในการส่งข้อมูล คือการบีบอัดข้อมูลให้มีขนาดเล็กลง อีกทั้งยังช่วยให้ใช้พื้นที่ในการเก็บข้อมูลลดลงตามไปด้วย จากงานวิจัยที่ผ่านมาเกี่ยวกับการบีบอัดข้อมูล XML นั้นได้ทำการบีบอัดข้อมูลให้มีขนาดเล็กลง ซึ่งในบางวิธีการจะต้องใช้ สกีม่า (Schema) เป็นตัวกำหนดไวยากรณ์เป็นของตัวเอง และบางวิธีการต้องขยายเอกสารให้เป็นต้นฉบับก่อนจึงจะสามารถใช้งานได้ ซึ่งยากต่อการนำไปใช้งานทั่วๆไป

เทคนิคที่ได้นำเสนอคือการบีบอัดข้อมูล XML โดยใช้วิธีจัดเรียงข้อมูลแบบกระชับ (XBrevity) ซึ่งเป็นเทคนิคการจัดเรียงแบบกระชับ เพื่อให้เอกสารมีขนาดเล็กลงจากเดิม และยังสามารถเข้าใจความหมายเอกสารที่ถูกบีบอัดได้ง่าย โดยไม่ต้องมีการขยายเอกสารให้เป็นต้นฉบับเดิมเพราะอยู่ในรูปแบบของภาษา XML อยู่แล้ว ทำให้ยังคงข้อดีที่มีอยู่ในภาษา XML อีกทั้งยังง่ายต่อการใช้งานทั่วๆไป ซึ่งไม่ต้องมีตัวกำหนดไวยากรณ์แบบใดแบบหนึ่งโดยเฉพาะ และสามารถที่จะค้นหา (Query) ข้อมูลที่ถูกบีบอัดได้ เพื่อที่จะได้ไม่เสียเวลาในการขยายข้อมูลอีก

ในงานวิจัยนี้เน้นความสำคัญในการบีบอัดข้อมูลในส่วนของโครงสร้างแท็ก แต่ตัวอักขระซึ่งเป็นข้อมูลภายในแท็กยังไม่ได้รับการบีบอัด ดังนั้นจึงยังสามารถพัฒนาประสิทธิภาพของเทคนิคการบีบอัดได้มากขึ้นไปอีกโดยการบีบอัดอักขระภายในแท็ก ซึ่งจะมีผลให้ขนาดของเอกสารลดลงด้วย

นอกจากนี้แนวทางและวิธีการนำเอาเอกสารที่ถูกบีบอัดแล้วซึ่งอยู่ในรูปแบบ XML ไปใช้เพื่อการค้นหาข้อมูลก็เป็นงานวิจัยที่น่าสนใจ ผลการทดลองจากการเปรียบเทียบเวลารวมจากเวลาในการบีบอัดข้อมูลและเวลาในการส่งเอกสาร XML ที่ถูกบีบอัดแล้วกับเวลาที่ใช้ในการส่งเอกสาร XML เดิม เป็นอีกผลการทดลองต่อเนื่องที่ควรจะทำ

เอกสารอ้างอิง

[1] Amazon. “Amazon Web Services”, Available at http://www.amazon.com/gp/aws/landing.html

[2] BEA Systems, IBM, Microsoft, SAP AG and Siebel Systems. "Business Process Execution Language for Web Services", Available at http://www-128.ibm.com/developerworks/library/specification/ws-bpel/

[3] eBay. "eBay API", Available at http://developer.ebay.com/common/api

[4] J. Ferraiolo, J. Fujisawa, D. Jackson. "Scalable Vector Graphics (SVG) 1.1 Specification", Available at http://www.w3.org/TR/SVG

[5] J.L. Gailly and M. Adler, "gzip : The compressor data", Available at http://www.gzip.org/

[6] Google. Google Web APIs (beta), Available at http://www.google.com/apis/

[7] D. Hunter, C. Cagle, D. Gibbons, N. Ozu, J. Pinnock and P. Spencer. "Beginning XML" , *Wrox Press*, 2002.

[8] H. Liefke and D. Suciu. "XMill: an Efficient Compressor for XML Data." , In *Proceeding of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 153-164, May 2000.

[9] K. Mairiang, and C. Pluempitiwiriyawej. "XPACK: A Grammar-based XML Document Compression", In *Proceeding of NCSEC2003 the7th National Computer Science and Engineering Conference*, October 28-30, 2003.

[10] J.-K. Min, M.-J. Park, and C.-W. Chung. "XPRESS: A Queriable Compression for XML Data." In *Proceeding of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 122-133, June 9-12, 2003.

[11] P. Murray-Rust and H. Rzepa. "Chemical Markup Language (CML)", Available at http://www.xml-cml.org

[12] K. Runapongsa and J. M. Patel, "Storing and Querying XML Data in Object-Relational DBMSs", EDBT Workshops 2002: 266-285

[13] A. R. Schmidt, F. Waas, M. L. Kersten, M. J. Carey, I. Manolescu, and R. Busse. "XMark: A Benchmark for XML Data Management", Proceedings of the International Conference on Very Large Data Bases (VLDB), pp. 974-985, Hong Kong, China, August 2002, Available at http://monetdb.cwi.nl/xml/index.html

[14] P. M. Tolani and J. R. Haritsa. "XGRIND: A Query-friendly XML Compressor." In *Proceedings of 18th International Conference on Databases Engineering*, February 2002.

[15] UN/CEFACT and OASIS. "ebXML: Enabling a Global Electronic Market", Available at http://www.ebxml.org/

[16] W3C. "Extensible Markup Language (XML) 1.0 (Third Edition)", Feb 4, 2004, Available at http://www.w3.org/TR/2004/REC-xml-20040204/.

[17] W3C. "Namespaces in XML 1.1", Feb 4, 2004, Available at http://www.w3.org/TR/xml-names11/.

[18] W3C. "XML Schema Part 0 : Primer " , May 2, 2001, Available at http://www.w3.org/TR/xmlschema-0/.

[19] W3C. "XML Schema Part 1 : Structures", May 2., 2001, Available at http://www.w3.org/TR/xmlschema-1/.

[20] W3C. "XML Schema Part 2 : Datatypes ", May 2, 2001, Available at http://www.w3.org/TR/xmlschema-2/

[21] W3C. "Web Services", Available at http://www.w3.org/2002/ws/

[22] W3C. "XQuery : An XML Query Language", Available at http://www.w3.org/XML/Query

[23] Yahoo. "Yahoo! Search Web Services", Available at http://developer.yahoo.net/