การสำรวจการบีบอัดข้อมูลเอ็กซ์เอ็มแอล

A Survey of XML Data Compression

ประพันธ์ เลขาโสภณ
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น
Email: jingxth@yahoo.com

กานดา รุณนะพงศา
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น
Email: krunapon@kku.ac.th

บทคัดย่อ

ข้อมูลบนเว็บที่แลกเปลี่ยนกันในปัจจุบันส่วนใหญ่นิยมแสดงในรูปแบบของเอกสาร XML (Extensible Markup Language) ซึ่งโดยทั่วไปเอกสาร XML มีขนาดใหญ่เมื่อเทียบกับขนาดของข้อมูลจริงในเอกสาร เนื่องจากมีการใช้แท็กที่ซ้ำกันในการอธิบายข้อมูลประเภทเดียวกันที่มีรายละเอียดต่างกัน ในบทความนี้ได้สำรวจถึงเทคนิคที่ใช้ในการบีบอัดเอกสาร XML ถึงแม้ว่าแต่ละเทคนิคที่ใช้ในการบีบอัดเอกสาร XML มีวิธีการที่แตกต่างกันซึ่งพบว่าทุกเทคนิคมีความจำเป็นต้องใช้ข้อมูลเกี่ยวข้องกับโครงสร้างของเอกสารทั้งสิ้น โดยการกำหนดกฎไวยากรณ์จากการวิเคราะห์โครงสร้างเอกสาร จากผลการสำรวจนั้นแต่ละเทคนิคมีจุดเด่นและจุดด้อยที่แตกต่างกันและทำให้ความสามารถในการใช้งานแตกต่างกันอีกด้วย

คำสำคัญ การบีบอัดข้อมูล, XML, กฎไวยากรณ์

1. บทนำ

ปัจจุบัน XML (Extensible Markup Language) [6] ได้เข้ามามีบทบาทและเป็นมาตรฐานในการแลกเปลี่ยนข้อมูล เนื่องจากการสร้างเอกสารอาจมีโครงสร้างที่หลากหลาย และเมื่อมีการเปลี่ยนแปลงข้อมูลเกิดขึ้นย่อมต้องมีการเปลี่ยนแปลงโครงสร้างตามไปด้วย ซึ่งจะมีความยุ่งยากในการแก้ไขสำหรับแอปพลิเคชันขนาดใหญ่ ด้วยเหตุนี้ XML จึงมีบทบาทมากขึ้น ซึ่ง XML มีความสามารถในการอธิบายความหมายของข้อมูลและมีความยืดหยุ่นในการใช้งาน การนำ XML มาใช้งานสามารถทำได้โดยการใช้แท็กเป็นตัวกำกับและการตั้งชื่อแท็กที่สื่อถึงความหมายของข้อมูล ทำให้เอกสารที่ถูกสร้างขึ้นเข้าใจได้ง่าย จึงเป็นส่วนสำคัญที่ทำให้การเข้าถึงข้อมูลได้ง่ายขึ้น แต่จากการที่มีการใช้แท็กเข้ามาช่วยในการสื่อถึงความหมายทำให้เกิดการบันทึกแท็ก ชนิดเดียวกันบ่อยครั้งในเอกสาร

```
<Book>
   <Author>
      <Name> Pissamai </Name>
   </Author>
   <Author>
      <Name> Porntip </Name>
   </Author>
</Book>
```

รูปที่ 1 ตัวอย่างเอกสาร XML

จากรูปที่ 1 จะเห็นได้ว่ามีการบันทึกข้อมูลประเภทเดียวกันหลายครั้ง ซึ่งในแต่ละครั้งจะมีรายละเอียด

แตกต่างกัน ด้วยเหตุนี้ ขนาดของเอกสารจึงมีขนาดใหญ่เมื่อเทียบกับขนาดข้อมูลจริงภายในเอกสารนั้น ส่งผลให้สิ้นเปลืองเนื้อที่หากต้องการจัดเก็บเอกสารและสิ้นเปลืองเวลาในการรับส่ง หากต้องการแลกเปลี่ยนข้อมูลระหว่างองค์กร ผ่านระบบเครือข่าย

จากปัญหาในเรื่องขนาดของข้อมูลแนวทางที่สามารถนำมาใช้ในแก้ปัญหาได้คือการบีบอัดข้อมูล XML (XML data compression) เพื่อลดขนาดของเอกสารซึ่งเป็นการบีบอัดข้อมูล XML โดยเฉพาะทำให้สามารถเพิ่มประสิทธิภาพในการบีบอัดได้ดีกว่าการใช้วิธีการบีบอัดข้อมูลทั่วไป

ในการให้รายละเอียดของโครงสร้างเอกสาร ส่วนใหญ่จะใช้ DTD หรือ XML Schema ซึ่งรายละเอียดของโครงสร้างเอกสาร XML มักจะถูกอธิบายด้วย DTD (Document Type Definition) [1] ถ้าหากมีการส่งข้อมูล XML โดยมี DTD เป็นส่วนเพิ่มเติม XML จะทำให้การแลกเปลี่ยนข้อมูลง่ายขึ้นเนื่องจากมีการกำหนดว่า แท็กอะไรที่ควรมี และความสัมพันธ์ระหว่างข้อมูลประเภทต่างๆ ควรจะเป็นอย่างไร แต่ในการสร้าง DTD มีรายละเอียดและขั้นตอนที่ยุ่งยากพอสมควร ทำให้การสร้างเอกสารมีความยุ่งยาก บ่อยครั้งที่มีเอกสารซึ่งประกอบด้วยเนื้อหาที่มีชนิดต่างกัน และต้องการตรวจสอบชนิดของข้อมูล (Datatypes) [10] เหล่านี้ได้ แต่ DTD ไม่ได้ถูกออกแบบมาเพื่อตรวจสอบชนิดของข้อมูลเหล่านี้หรือการตรวจสอบขอบเขตของค่า (Value) ดังนั้น XML Schema [8,9] จึงถูกสร้างขึ้นมาเพื่อการแก้ปัญหาเหล่านี้ XML Schema ต่างจาก DTD ตรงที่ DTD มีรูปประโยค (Syntax) เป็นของตัวเอง ส่วน XML Schema นั้นถูกเขียนขึ้นโดยใช้ไวยากรณ์ของภาษา XML นอกจากการจัดสร้างข้อมูลที่ DTD นำเสนอแล้ว XML Schema ยังช่วยกำหนดชนิดของข้อมูลโดยใช้เนมสเปซ (Namespace) [7] และกำหนดช่วงค่าของแอตทริบิวต์ (Attribute) และอิลิเมนต์ (Element)

ตัวอย่าง DTD ของเอกสาร XML จากรูปที่ 1

<!ELEMENT Book (Author+)>

<!ELEMENT Author (Name)>

<!ELEMENT Name (#PCDATA)>

จากตัวอย่างข้างต้นในลิสต์ลำดับ จะประกอบไปด้วยการประกาศให้ <Book> เป็นอิลิเมนต์ที่มี Author อย่างน้อยหนึ่ง Author โดยอิลิเมนต์ Author นั้นมีข้อมูลของอิลิเมนต์ Name และอิลิเมนต์ Name เก็บข้อมูลตัวอักษรซึ่งถูกประกาศเนื้อหาเป็น PCDATA (Parsed Character Data)

เทคนิคในการบีบอัดข้อมูล XML ที่มีอยู่ในปัจจุบันนั้น จะใช้ XML Schema เป็นกฎเกณฑ์และข้อกำหนดในการทำการบีบอัดข้อมูล XML และในบทความนี้ในหัวข้อที่ 2 จะได้นำเสนอกระบวนการบีบอัดแบบต่างๆ ที่มีอยู่ในปัจจุบันซึ่งได้แก่ XMILL, XPRESS และ XPACK โดยที่ทั้ง 3 วิธีดังกล่าวจะได้นำเสนอภาพรวมถึงข้อเด่นข้อด้อย และแนว ทางในการพัฒนาเทคนิคใหม่ๆ ในการบีบอัดข้อมูล XML จะนำเสนอในหัวข้อที่ 3 และบทสรุปในหัวข้อที่ 4

2. งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่กล่าวถึงการบีบอัดข้อมูล XML ในปัจจุบันนั้นมีอยู่ 3 วิธีได้แก่ XMILL, XPRESS และ XPACK ซึ่งทั้ง 3 เทคนิคมุ่งเน้นในการลดขนาดของข้อมูลที่เป็น XML ให้เล็กลง โดยแต่ละวิธีจะใช้เทคนิคที่แตกต่างกันไป ดังที่จะได้เสนอต่อไปนี้

2.1 XMILL [2]

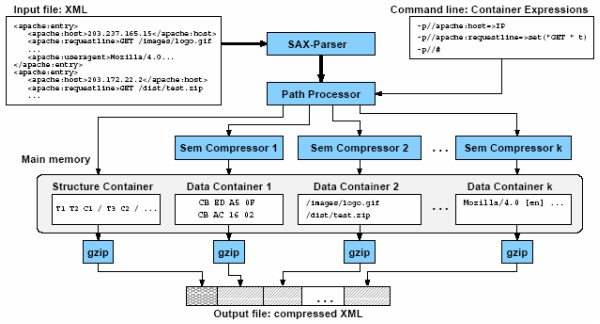
เป็นเทคนิคแรกที่ได้ทำการบีบอัดเอกสาร XML เพื่อให้มีขนาดเล็กลงโดยได้ใช้ zlib เป็นไลเบอรี่ในตัว XMILL ซึ่งเป็นตัวเดียวกันกับที่ใช้ใน GZip โดยใช้หลักการ 3 ข้อ คือ 1) แยกโครงสร้างออกจากข้อมูล 2) จัดกลุ่มให้กับข้อมูลต่างๆ 3) นำข้อมูลที่ได้เข้าสู่กระบวนการสร้างความสัมพันธ์ทางกลุ่มคำกับสัญลักษณ์ ซึ่งในการบีบอัด

ข้อมูลโดยวิธีนี้จะใช้ SAX Parser เป็นตัวจัดการกับข้อมูล XML เพื่อนำข้อมูลที่ได้เข้าสู่กระบวนการบีบอัดของ XMILL โดยที่หลักการดังกล่าวมีรายละเอียดดังนี้

2.1.1 แยกโครงสร้างออกจากข้อมูล เป็นการแยกโครงสร้างที่ประกอบไปด้วยแท็ก และแอตทริบิวต์ออกจากส่วนที่เป็นข้อมูล

2.1.2 จัดกลุ่มให้กับข้อมูลต่างๆ ที่ได้จากการแยกโครงสร้างออกจากข้อมูล โดยให้ข้อมูลที่เป็นแบบเดียวกันจัดอยู่ในกลุ่มเดียวกัน ส่วนข้อมูลจะเรียงกันไปต่อจากแท็กที่ได้จัดกลุ่มแล้ว

2.1.3 ข้อมูลที่ได้จะเข้าสู่กระบวนการสร้างความสัมพันธ์ทางกลุ่มคำกับสัญลักษณ์ เพื่อให้สามารถอ้างถึงกลุ่มคำที่ได้จัดแยกแล้ว และสามารถบอกความหมายของกลุ่มคำนั้นๆได้



รูปที่ 2 ผังการทำงานของ XMILL

ตัวอย่างในการทำงานของ XMILL

เอกสาร XML ต้นฉบับ

<Book>

<Author>

<Name> Pissamai </Name>

</Author>

<Author>

<Name> Porntip </Name>

</Author>

</Book>

ทำการแยกแท็กและข้อมูลออกจากกัน

Book = T1, Author = T2, Name = T3

C1=Pissamai, C2=Porntip

โครงสร้างที่สมบูรณ์

Structure = T1 T2 T3 C1 / T2 T3 C2 //

จากตัวอย่าง XMILL จะทำการจัดการโครงสร้างของเอกสารใหม่ก่อนการบีบอัดและบีบอัดเสร็จแล้ว จะเห็นได้ว่าโครงสร้างเล็กลงจากต้นฉบับ แต่เรายังไม่สามารถเข้าใจโครงสร้างที่ทำขึ้นมาใหม่ได้ ต้องอาศัยการขยายข้อมูลที่ถูกบีบอัดนี้ให้เป็นข้อมูลเดิมก่อนจึงจะสามารถเข้าใจข้อมูลภาย ในได้

2.2 XPRESS [4]

เป็นเทคนิคการบีบอัดข้อมูล XML ที่สนับสนุนการค้นหา (Query) ข้อมูลที่ถูกบีบอัดแล้วได้โดยใช้การเข้ารหัสที่เรียกว่า reverse arithmetic encoding ซึ่งวิธีการนี้ขึ้นอยู่กับชนิดของข้อมูลโดยจะมีหลักการทำงานคือ 1) วิเคราะห์ข้อมูล XML 2) เข้ารหัสข้อมูล XML 3) กระบวนการ ค้นหา ข้อมูลที่ถูกบีบอัดแล้ว (Query Processing)

2.2.1 ตัววิเคราะห์ข้อมูล XML (XML Analyzer) อัลกอริทึมของตัววิเคราะห์ข้อมูล XML จะสร้าง Hash table ที่เรียกว่า Elemhash ซึ่งจะเก็บข้อมูลต่างๆอันได้แก่ ชนิดของข้อมูล, ความถี่ในการใช้งานเป็นต้น ค่าดังกล่าวจะถูกจัดเก็บให้อยู่ในช่วง Interval เรียกค่านี้ว่า Statistics collector ซึ่งจะคอยนับจำนวนของอิลิเมนต์ที่เกิดขึ้น อย่างไรก็ตามถ้าแท็กนั้นเป็นอิลิเมนต์ระดับที่สูงกว่า เช่น root element จะเป็นการยากที่จะกำหนดช่วงเส้นทางของข้อมูล จึงจำเป็นต้องใช้หลักทางคณิตศาสตร์ชั้นสูง (high precision floating arithmetic)

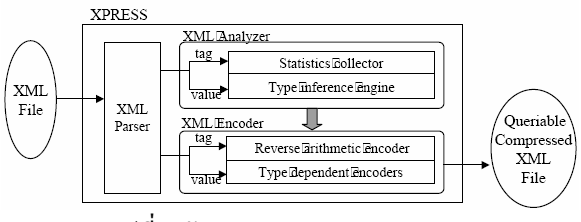
2.2.2 เข้ารหัสข้อมูล XML (XML Encoder) ในการเข้ารหัสข้อมูลจะทำการเปลี่ยนรูปให้เป็นไบนารี ซึ่งในแต่ละช่วงของการเข้ารหัสจะแทนด้วย u8 ใช้ 7 บิต

จำนวน 1 ไบต์, u16 ใช้ 15 บิต จำนวน 2 ไบต์และ u32 ใช้ 31 บิต จำนวน 4 ไบต์ เมื่อเข้ารหัสแล้วจึงทำการบีบอัดข้อมูลต่อไป ดังตารางที่ 1

ตารางที่ 1 Data Encoders

Encoder	Description
u8	encoder for integers where max-min < 2^7
u16	encoder for integers where< 2^7+1 < max-min < 2^{15}
u32	encoder for integers where < 2^{15}+1 <max-min< 2^{31}
f32	encoder for floating values
dict8	dictionary encoder of textual data

2.2.3 กระบวนการค้นหาข้อมูลที่ถูกบีบอัดแล้ว (Query Processing) เมื่อข้อมูลถูกบีบอัดโดย XPRESS แล้วกระบวนการ ค้นหา (Query) จะทำให้ชื่อที่มีความยาวสั้นลงแล้วทำการแปลงรูปไปเป็นลำดับช่วง ซึ่งโดยทั่วไปแล้ว หนึ่งลำดับช่วงจะเท่ากับเส้นทางของชื่อที่ทำให้สั้นลงแล้ว หรือเป็นการลดความยาวของชื่อให้สั้นลงไปนั่นเอง เมื่อต้องการค้นหา (Query) ข้อมูลจะทำการค้นหาจาก Hash table ซึ่งจะเก็บช่วงของข้อมูลนั้นๆ ที่ได้ทำการบีบอัดแล้ว เพื่อให้ได้ช่วงของข้อมูลที่ตรงกับการค้นหา (Query) นั้น



รูปที่ 3 ผังการทำงานของ XPRESS

ตัวอย่างการทำงานของ XPRESS

เอกสาร XML ต้นฉบับ

<Book>

<Author>

<Name> Pissamai </Name>

</Author>

<Author>

<Name> Porntip </Name>

</Author>

</Book>

ทำการแยกโครงสร้างแท็กออกจากข้อมูล

Book=T1, Author=T2, Name=T3, *C1=Pissamai, C2=Porntip*

โครงสร้างใหม่ที่ได้

Non-homomorphic

X0

T1

T2 T3 C1 / /

T2 T3 C2 / /

/

X1

encode(Pissamai)

encode(Porntip)

Homomorphic

T1

T2

T3 encode(Pissamai) /

/

T2

T3 encode(Porntip) /

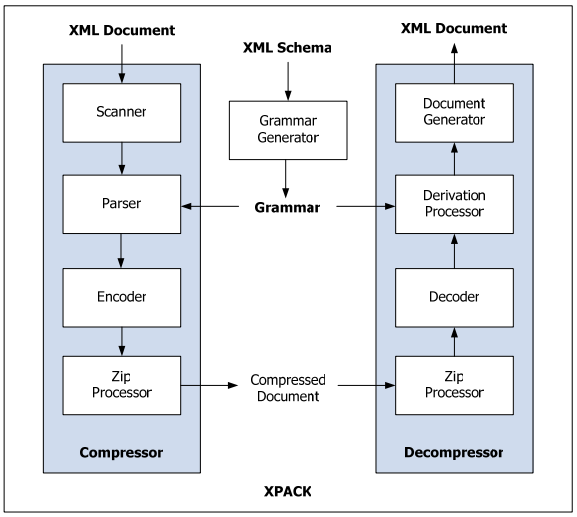
/

จากตัวอย่างจะสามารถแบ่งการบีบอัดเป็น 2 แบบคือ แบบแยกการบีบอัดแท็กและข้อมูลออกจากกัน (Non-homomorphic) และแบบรวมส่วนที่เป็นแท็กและข้อมูลไว้ด้วยกัน (Homomorphic) แล้วทำการบีบอัด ซึ่งทั้ง 2 แบบให้ผลที่ไม่แตกต่างกันมากนัก จะเห็นได้ว่าขนาด

ของโครงสร้างใหม่ที่ได้เล็กลง และข้อมูลยังดูเข้าใจยากอยู่ และไม่มีตัวขยายข้อมูลที่บีบอัดแล้ว

2.3 XPACK [3]

เป็นการบีบอัดเอกสาร XML ด้วยวิธีการเชิงไวยากรณ์ ในหัวข้อนี้นำเสนอภาพรวมส่วนประกอบของเทคนิคที่เรียก ว่า XPACK ซึ่งใช้วิธีการเชิงไวยากรณ์ในการบีบอัดและการขยายเอกสาร XML ส่วนประกอบหลักของ XPACK คือ 1) Grammar Generator ทำหน้าที่ในการสร้างกฎไวยากรณ์ 2) Compressor ทำหน้าที่บีบอัดเอกสาร และ 3) Decompressor ทำหน้าที่ขยายเอกสารที่ผ่านการบีบอัด ซึ่งส่วนประกอบต่างๆมีรายละเอียดดังปรากฏในรูปที่ 4



รูปที่ 4 ผังการทำงานของ XPACK

2.3.1 ตัวสร้างกฎไวยากรณ์ (Grammar Generator) มีหน้าที่ในการสร้างกฎไวยากรณ์เพื่อใช้ในการบีบอัดเอกสาร โดยการวิเคราะห์คำอธิบายโครงสร้างเอกสาร XML ภาษาที่ใช้ในการอธิบายโครงสร้างดังกล่าวได้แก่ XML Schema

2.3.2 ตัวบีบอัดข้อมูล (Compressor) ทำหน้าที่ในการบีบอัดเอกสารโดยประกอบไปด้วยส่วนประกอบย่อย 4 ส่วนด้วยกันคือ Scanner, Parser, Encoder และ Zip Processor ดังรูปที่ 4 โดยการทำงานภายใน Compressor เริ่มต้นที่ Scanner รับเอกสาร XML เข้ามาเพื่อเปลี่ยนให้เป็นของเครื่องหมาย (Token) ที่ใช้ในกฎไวยากรณ์ จากนั้น Parser ทำการวิเคราะห์ด้วยวิธีการเชิงไวยากรณ์เพื่อให้ได้ผลลัพธ์ออกมาเป็น Parser Tree ต่อมาจึงนำ Parser Tree มาเข้ารหัสให้อยู่ในรูปของแถวลำดับด้วย Encoder ซึ่งผลลัพธ์ที่ได้จะนำไปทำการบีบอัดในขั้นตอนสุดท้ายที่ Zip Processor โดยทำการบีบอัดข้อมูลด้วยอัลกอริทึมการบีบอัดข้อมูลทั่วไป ผลลัพธ์จากการบีบอัดคือเอกสารที่ได้รับการบีบอัด (Compressed Document) ซึ่งสามารถนำไปจัดเก็บ หรือแลกเปลี่ยนระหว่างองค์กรผ่านเครือข่ายต่อไป

2.3.3) ตัวขยายข้อมูลที่ถูกบีบอัด (Decompressor) เมื่อต้องการใช้เอกสารที่ได้รับการบีบอัดแล้ว จำเป็นต้องทำการขยายเอกสารก่อนโดยใช้ Decompressor ซึ่งประกอบไปด้วย 4 ส่วนย่อย คือ Unzip Processor, Decoder, Derivation Processor และ Document Generator ดังรูปที่ 4 Decompressor มีการทำงานโดยเริ่มที่ Unzip Processor ทำการขยายเอกสารที่ผ่านการบีบอัดให้อยู่ในรูปของข้อมูลที่เข้ารหัส จากนั้น Decode ทำการถอดรหัสข้อมูลให้อยู่ในรูปของ Parse Tree เพื่อให้ Derivation Processor ใช้วิธีการเชิงไวยากรณ์ให้ได้มา ซึ่งสัญลักษณ์ที่จะใช้ในการสร้างเอกสาร สุดท้าย Document Generator ทำการสร้างเอกสาร XML ที่มีความหมายของข้อมูลภายในเหมือนกับเอกสารต้นฉบับ

การจัดโครงสร้างไวยากรณ์ของ XPACK เป็นดังนี้

<t> ::= t # / d (a, v)

โดยที่

<t> แทนรูปแบบการจัดเรียงข้อมูลในเอกสาร XML (*ไม่ใช่แท็ก!!*)

t แทนแท็กเปิด (Open Tag)

แทนข้อมูลภายในแท็ก

/ แทนแท็กปิด (Close Tag)

d แทนประเภทข้อมูลภายในแท็ก (Element Data Type)

a แทนชื่อแอตทริบิวส์ (Attribute Name)

v แทนประเภทข้อมูลของแอตทริบิวส์
(Attribute Data Type)

จากหลักการจัดโครงสร้างไวยากรณ์ข้างต้นนี้ เมื่อมีเอกสารที่จะทำการบีบอัดข้อมูล XPACK จะทำการจัดโครงสร้างใหม่ได้จากตัวอย่างต่อไปนี้

ตัวอย่างการทำงานของ XPACK

เอกสาร XML ต้นฉบับ

```
<Book>
        <Author>
                <Name> Pissamai </Name>
        </Author>
        <Author>
                <Name> Porntip </Name>
        </Author>
</Book>
```

สร้างกฎไวยากรณ์

1 <Book> ::= Book <Author><Name> /

2 <Author> ::= Author <Name> /

3 <Name> ::= Name # / string

โครงสร้างใหม่ที่ได้

Book Author Name Pissamai / Author Name Porntip/ /

จากตัวอย่างจะเห็นได้ว่ามีการนำเอกสาร XML ต้นฉบับมาทำการสร้างกฎไวยากรณ์ขึ้นมาแล้ว จากนั้นจึงทำการจัดรูปแบบโครงสร้าง ที่สอดคล้องกับกฎไวยากรณ์ที่ได้กำหนดขึ้นเมื่อเปรียบเทียบกับโครงสร้างเดิมจะเห็นได้ว่ามีขนาดที่เล็กลง แต่อย่างไรก็ตามยังต้องมีการขยายข้อมูลที่ถูกบีบอัดออกมาโดยอ้างอิงกับกฎไวยากรณ์นั้นๆ ให้เป็นเอกสารต้นฉบับก่อน จึงจะสามารถเข้าใจข้อมูลภายในได้

3. ภาพรวมของการบีบอัดข้อมูล XML และแนวทางในการพัฒนาการบีบอัด

จากการทำงานของการบีบอัดข้อมูล XML ทั้ง 3 เทคนิคนั้นในแต่ละเทคนิคจะมีทั้งจุดเด่นและจุดด้อยอยู่ ซึ่งสามารถจำแนกส่วนที่สำคัญๆได้ดังนี้

XMILL ใช้เทคนิคในการแยกโครงสร้างของเอกสารโดยการแยกส่วนที่เป็นแท็กออกจากส่วนที่เป็นข้อมูล ซึ่งทำให้ง่ายในการจำแนกชนิดของข้อมูล จากนั้นในกระบวนการบีบอัดได้มีการใช้ zlib เป็นไลบารีซึ่งไลบารีดังกล่าวเป็นส่วนหนึ่งที่ใช้ในโปรแกรม GZip ที่มีความสามารถในการบีบอัด และสามารถขยายข้อมูลที่ถูกบีบอัดได้ แต่อย่างไรก็ตาม XMILL ยังมีข้อที่ควรปรับปรุงบางประการคือ ในการบีบอัดด้วยเทคนิคนี้ยังต้องอาศัยการใช้ Schema เป็นตัวสร้างไวยากรณ์ก่อนการบีบอัดข้อมูล อีกทั้งข้อมูลที่ถูกบีบอัดด้วยเทคนิคนี้ยังไม่สามารถค้นหา (Query) ข้อมูลที่ถูกบีบอัดอยู่ได้

XPRESS ใช้เทคนิคในการแยกโครงสร้างก่อนการบีบอัด โดยแยกส่วนที่เป็นแท็กและส่วนที่เป็นข้อมูลออกจากกันโดยใช้ hash table เป็นที่เก็บตัวบ่งชี้ข้อมูลต่างๆ จากนั้นจะใช้ XML Parser ร่วมในกระบวนการบีบอัดและสามารถค้นหาข้อมูลที่ถูกบีบอัดแล้วได้ ซึ่งเป็นส่วนที่สำคัญของเทคนิคนี้ แต่อย่างไรก็ตาม XPRESS ยังคงมีข้อปรับปรุงบางประการคือ การใช้การบีบอัดด้วยเทคนิคนี้ยังคงใช้ Schema เป็นตัวสร้างไวยากรณ์ก่อนการบีบอัดและข้อมูลที่ถูกบีบอัดด้วยเทคนิคนี้ยังไม่มีตัวขยาย ข้อมูลที่ถูกบีบอัดให้เป็นข้อมูลเดิมได้

XPACK การบีบอัดด้วยเทคนิคนี้จะมีตัวสร้างกฎไวยากรณ์ขึ้นมา ทำให้ง่ายในการจัดการกับเอกสารก่อนการบีบอัดข้อมูล อีกทั้งยังสามารถขยายเอกสาร XML ที่ได้

ทำการบีบอัดแล้วให้สามารถเป็นเอกสาร XML ปกติได้ แต่ถึงอย่างไรก็ตาม XPACK ยังคงมีข้อที่ควรปรับปรุงคือ จะต้องใช้ สกีม่า (Schema) เพื่อสร้างกฎไวยากรณ์เป็นหลัก ซึ่งเป็นส่วนสำคัญของเทคนิคนี้ อีกทั้งยังไม่สามารถค้นหา (Query) ข้อมูลที่ถูกบีบอัดอยู่ได้

จากภาพรวมของเทคนิคการบีบอัดข้อมูล XML จะสังเกตเห็นได้ว่าทุกเทคนิคจะทำการแยกส่วนที่เป็น แท็ก และส่วนที่เป็นข้อมูลออกจากกัน เพื่อง่ายต่อการจำแนกชนิดของข้อมูล ส่วนการบีบอัดข้อมูลจะมีความแตกต่างกันไป โดยที่วิธี XMILL จะใช้ zlib ที่เป็นไลบารีของ GZip และเมื่อพัฒนาแล้วจะบีบอัดข้อมูลได้ดีกว่า GZip โดยตรง ส่วน XPRESS จะใช้ XML Parser ซึ่งให้ผลในการบีบอัดดีกว่า XMILL [4] และวิธี XPACK จะใช้ Zip Processor ซึ่งไม่ได้บอกว่าเป็นแบบใดในการบีบอัดข้อมูลและบทความที่อ้างถึงได้กล่าวว่า สามารถบีบอัดข้อมูลได้ดีกว่า XMILL [3] จากการเปรียบเทียบข้อมูลดังกล่าวจะเห็นได้ว่า XMILL จะเป็นตัวเปรียบเทียบประสิทธิภาพในการบีบอัดข้อมูล ที่ทำให้ข้อมูลลดลง แต่ยังไม่มีบทความที่รายงานผล เปรียบเทียบประสิทธิภาพของการบีบอัดข้อมูลระหว่างเทคนิค XPRESS และเทคนิค XPACK

โดยสรุปแล้วทั้งสามเทคนิคนี้ จะให้ความสำคัญกับการลดขนาดของข้อมูลเป็นหลักและจะใช้ Schema เป็นตัวกำหนดไวยากรณ์ให้กับข้อมูลที่จะทำการบีบอัด ซึ่งในการนำไปใช้งานจริงนั้น การที่ผู้ใช้ปลายทางสามารถถอดรหัสที่ถูกบีบอัดออกมาให้เป็นข้อมูลเดิมได้นั้น จำเป็นจะต้องใช้ตัวขยายข้อมูลจาก วิธีการที่ถูกบีบอัดจากต้นทางนั้นๆ ทำให้ไม่สะดวกกับการใช้งานซึ่งอาจเป็นข้อจำกัดในการนำไปใช้งานกับกลุ่มคนทั่วไป และนอกจากนั้นเอกสารบางชนิดมีการเปลี่ยนแปลงอยู่ตลอดเวลา ทำให้โครงสร้างเอกสารต้องมีการเปลี่ยนแปลงและสกีม่า (Schema) เปลี่ยนแปลงตามไปด้วย ดังนั้นควรจะมีเทคนิคในการบีบอัดข้อมูล XML โดยไม่ใช้ Schema เพื่อลดข้อจำกัดดังกล่าวได้

4. บทสรุป

การบีบอัดข้อมูล XML มีความสำคัญเพราะ XML เป็นภาษาที่มีข้อดีอยู่หลายประการ ซึ่งปัจจุบันได้มีการใช้กันอย่างแพร่หลาย แต่ข้อเสียของ XML ก็มีเช่นกัน โดยปกติแล้วเอกสารภาษา XML จะมีขนาดใหญ่กว่าเอกสารของเท็กซ์ไฟล์ (Text File) เนื่องจากเอกสารภาษา XML มักจะมีการใช้แท็กกำกับความหมายของข้อมูล การส่งข้อมูลโดยใช้ภาษา XML ทำให้เกิดความต้องการเน็ตเวิร์กแบนด์วิดธ์ (Network Bandwidth) ขนาดใหญ่เนื่องจากขนาดของไฟล์ วิธีหนึ่งที่จะช่วยลดเน็ตเวิร์กแบนด์วิดธ์ในการส่งข้อมูล คือการบีบอัดข้อมูลให้มีขนาดเล็กลง อีกทั้งยังช่วยให้ใช้พื้นที่ในการเก็บข้อมูลลดลงตามไปด้วย จากงานวิจัยที่ผ่านมาเกี่ยวกับการบีบอัดข้อมูล XML นั้นได้ทำการบีบอัดข้อมูลให้มีขนาดเล็กลง ซึ่งในแต่ละวิธีจะใช้ สกีม่า (Schema) เป็นตัวกำหนดไวยากรณ์เป็นของตัวเอง ซึ่งยากต่อการนำไปใช้งานทั่วๆไป โดยเทคนิคที่ควรมีคือการบีบอัดข้อมูลไม่จำเป็นต้องใช้สกีม่า(Schema) เพื่อให้ง่ายต่อการใช้งานทั่วๆไปซึ่งไม่ต้องมีตัวกำหนดไวยากรณ์แบบใดแบบหนึ่งโดยเฉพาะ และสามารถที่จะค้นหา (Query) ข้อมูลที่ถูกบีบอัดได้ เพื่อที่จะได้ไม่เสียเวลาในการขยายข้อมูล

เอกสารอ้างอิง

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

[2] 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.

[3] K. Maireang and C. Pleurmpitiwiriyavach. “XPACK: A Grammar-based XML Document Compression”, *In Proceeding of NCSEC2003 the7th*

National Computer Science and Engineering Conference, October 28-30, 2003.

[4] 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.

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

[6] W3C. "Extensible Markup Language (XML) 1.1", Feb 4, 2004, Available at http://www.w3.org/TR/2004/REC-xml11-20040204/.

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

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

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

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