

XMill: an Efficient Compressor for XML Data

Hartmut Liefke, Dan Suciu

Proceedings of the 2000 ACM SIGMOD international conference on Management of data

Presented by
Alex Baron

```
<Entry id="108_LYCES" class="STANDARD" mtype="PRT" seqlen="102">
  <AC>Q43495</AC>
  <Mod date="15-JUL-1999" Rel="38" type="Created"></Mod>
  <Mod date="15-JUL-1999" Rel="38" type="Last sequence update"></Mod>
  <Mod date="15-JUL-1999" Rel="38" type="Last annotation update"></Mod>
  <Descr>PROTEIN 108 PRECURSOR</Descr>
  <Species>Lycopersicon esculentum (Tomato)</Species>
  <Org>Eukaryota</Org> <Org>Viridiplantae</Org> ... <Org>Solanum</Org>
  <Ref num="1" pos="SEQUENCE FROM N.A">
    <Comment>STRAIN=CV. VF36</Comment>
    <Comment>TISSUE=ANTHER</Comment>
    <DB>MEDLINE</DB>
    <MedlineID>94143497</MedlineID>
    <Author>CHEN R</Author> <Author>SMITH A.G</Author>
    <Cite>Plant Physiol. 101:1413-1413(1993)</Cite>
  </Ref>
  <EMBL prim_id="Z14088" sec_id="CAA78466"></EMBL>
  <MENDEL prim_id="8853" sec_id="LYCes" status="1133"></MENDEL>
  <Keyword>Signal</Keyword>
  <Features>
    <SIGNAL from="1" to="30"> <Descr>POTENTIAL</Descr> </SIGNAL>
    <CHAIN from="31" to="102"> <Descr>PROTEIN 108</Descr> </CHAIN>
    <DISULFID from="41" to="77"> <Descr>BY SIMILARITY</Descr> </DISULFID>
    ...
  </Features>
</Entry>
```

XML

- Advantages
 - Self-describing
 - Plain text
 - Easy to edit
- Disadvantages
 - Verbose
 - Not-efficient

XMill (XDemill)

- Idea: Use XML tags to decide which compression algorithm to apply
- Compressors used:
 - zlib (gzip)
 - built-in data-specific
 - user-defined

weblog.dat: 15.9MB weblog.dat.gz: 1.6MB

```
202.239.238.16|GET / HTTP/1.0|text/html|200|1997/10/01-00:00:02|-|4478  
|-|-|http://www02.so-net.or.jp/|Mozilla/3.01 [ja] (Win95; I)
```



```
<apache:entry>  
  <apache:host>202.239.238.16</apache:host>  
  <apache:requestLine>GET / HTTP/1.0</apache:requestLine>  
  <apache:contentType>text/html</apache:contentType>  
  <apache:statusCode>200</apache:statusCode>  
  <apache:date>1997/10/01-00:00:02</apache:date>  
  <apache:byteCount>4478</apache:byteCount>  
  <apache:referer>http://www02.so-net.or.jp/</apache:referer>  
  <apache:userAgent>Mozilla/3.01 [ja] (Win95; I)</apache:userAgent>  
</apache:entry>
```

weblog.xml: 24.2MB weblog.xml.gz: 2.1MB weblog.xmi: 0.82MB

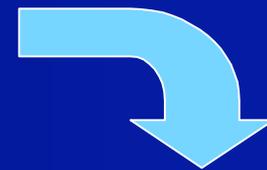
Three Principles

- Separate structure from content
 - XML tags
 - Attributes
 - Data
- Group related data items
 - Containers. E.g. <name> data items
- Apply semantic compressors
 - Data item types. E.g. numbers, IP's, etc.

Separating Structure from Content

- Start tags
 - Dictionary-encoded
- End tags
 - Replaced by token “/”
- Data values
 - Replaced with their container number

```
<Book> <Title lang="English"> Views </Title>
      <Author> Miller </Author>
      <Author> Tai </Author>
</Book>
```



```
Book = T1, Title = T2, @lang = T3, Author = T4
Structure = T1 T2 T3 C3 C4 / T4 C5 / T4 C5 / /
```

Grouping Data Values

- Each data value is uniquely assigned to one data container
- Language:
 - //Name
 - //#
 - //Title, //@lang
 - //Person/#
- Syntax:
 - xmill -p //Person/Title -p //(Name|Child) -p //# file.xml
- Default:
 - //#

Semantic Compressors

- Atomic semantic Compressors
- Combined Compressors
- User-defined Compressors

Atomic Semantic Compressors

Compressor	Description
t	default text compressor
u	compressor for positive integers
i	compressor for integers
u8	compressor for pos. integers < 256
di	delta compressor for integers
rl	run-length encoder
e	enumeration (dictionary) encoder
"..."	constant compressor

```
xmll -p //price=>i -p //state=>e file.xml
```

Combined Compressors

- Sequence Compressor: $seq(s1\ s2\ \dots)$
 - $Seq(u8\ \".\ " u8\ \".\ " u8\ \".\ " u8)$
- Alternated Compressor: $or(s1\ s2\ \dots)$
 - $Or(seq(u\ \ "-\ " u) u)$
- Repetition Compressor: $rep(d\ s)$
 - $Rep(\^, " e)$

Compressor	Description
t	default text compressor
u	compressor for positive integers
i	compressor for integers
u8	compressor for pos. integers < 256
di	delta compressor for integers
rl	run-length encoder
e	enumeration (dictionary) encoder
"..."	constant compressor

User-defined Compressors

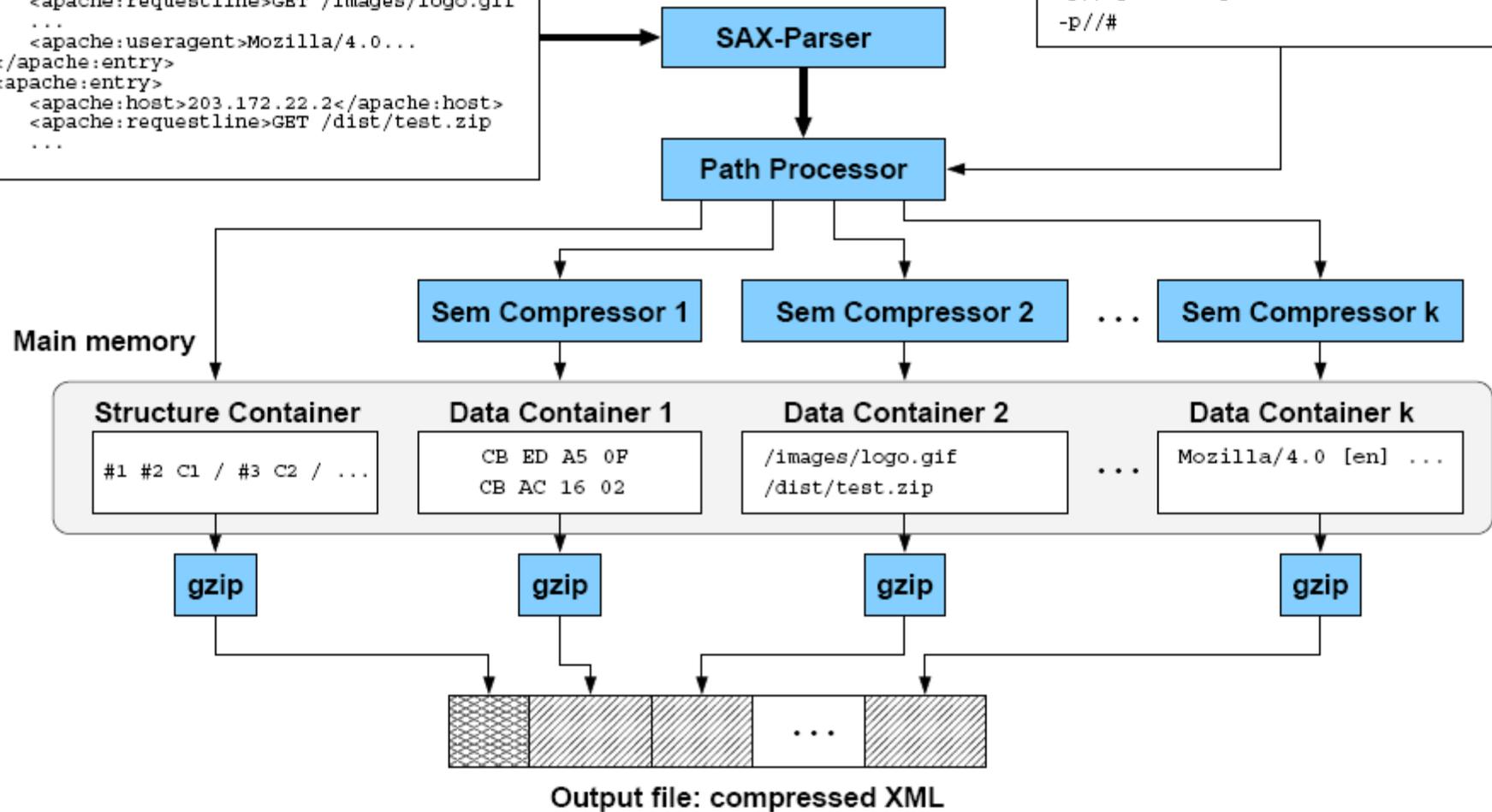
- Can be linked
- Need to conform to SCAPI (Semantic Compressor API)
- Make the tool application specific

Input file: XML

```
<apache:entry>
  <apache:host>203.237.165.15</apache:host>
  <apache:requestline>GET /images/logo.gif
  ...
  <apache:useragent>Mozilla/4.0...
</apache:entry>
<apache:entry>
  <apache:host>203.172.22.2</apache:host>
  <apache:requestline>GET /dist/test.zip
  ...
```

Command line: Container Expressions

```
-p//apache:host=>IP
-p//apache:requestline=>set("GET " t)
-p//#
```

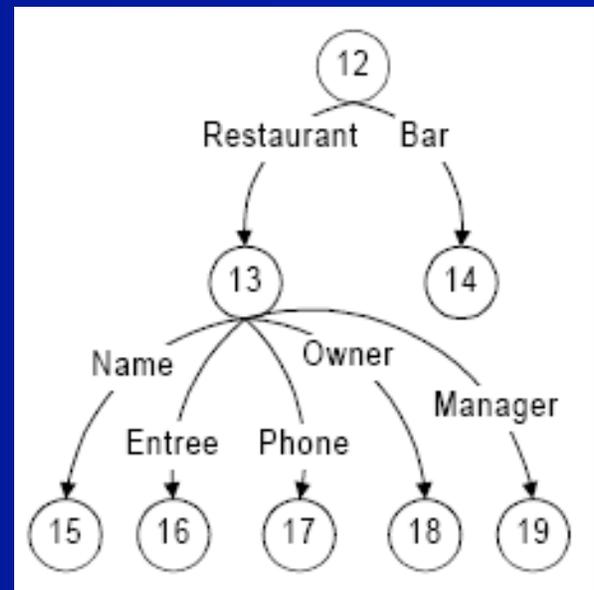
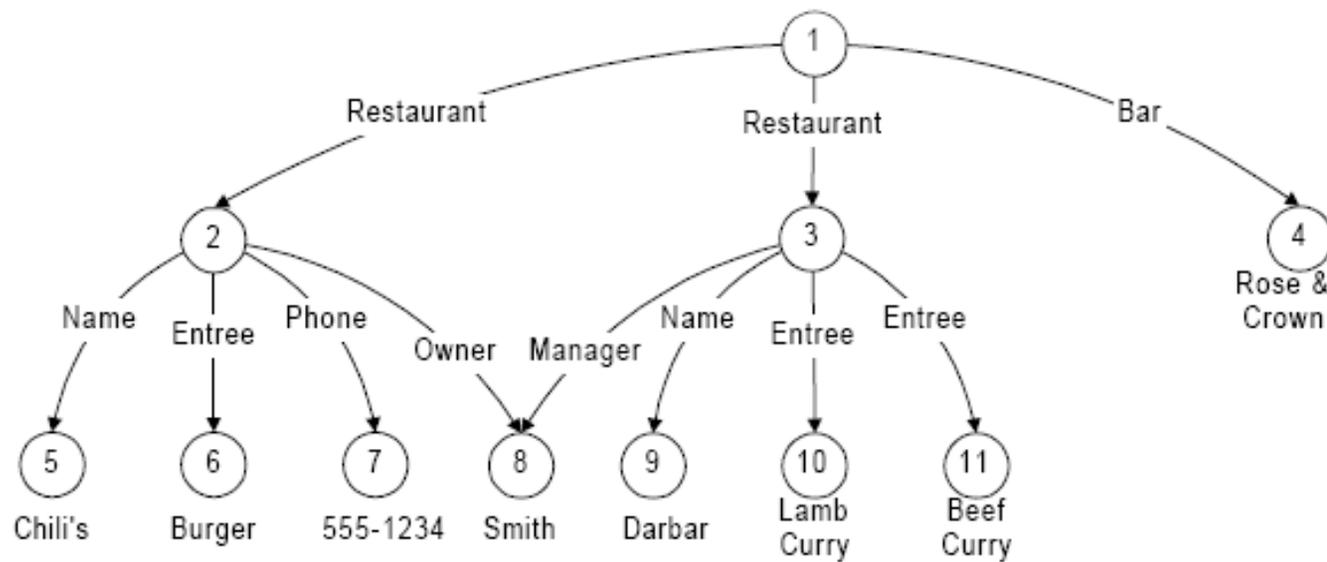


Path Processor

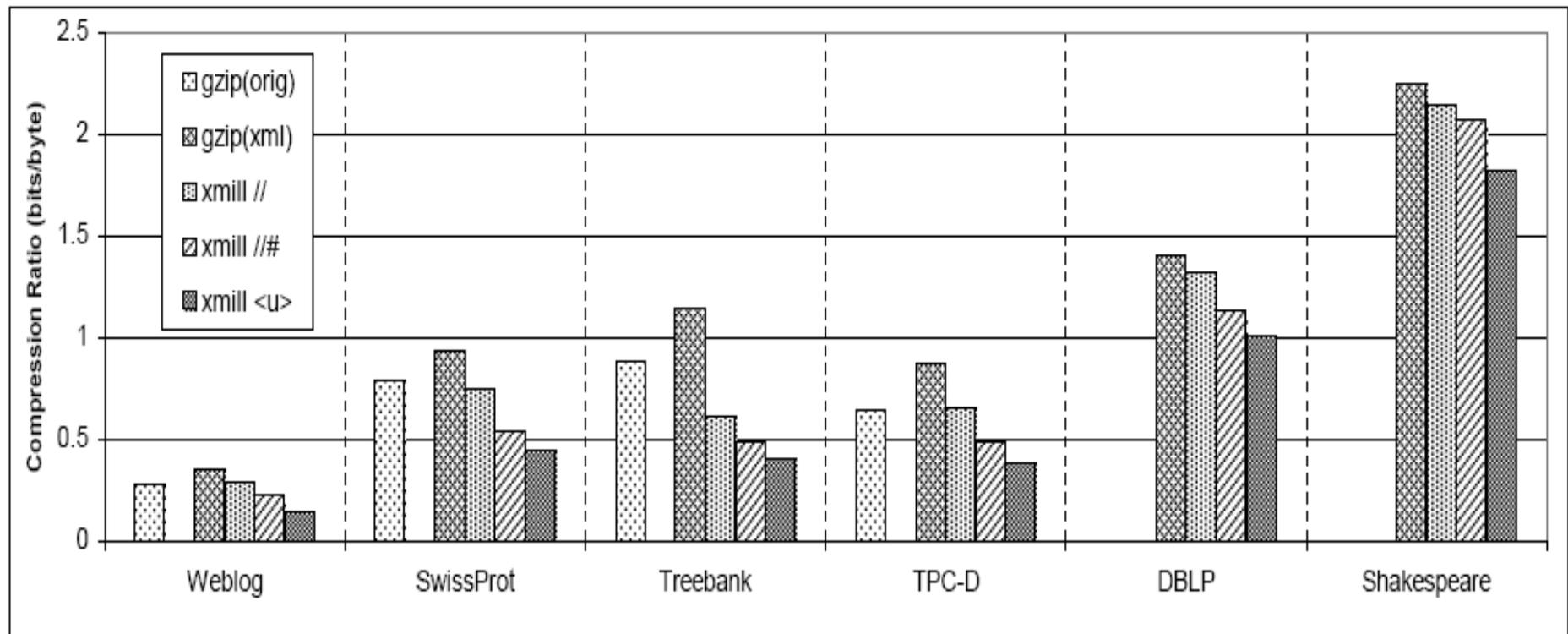
- Keeps track of the current path
- Evaluates container expressions
 - Regular Expressions
- Determines the semantic compressor for the value

Path Processors (2)

- Three approaches:
 1. Direct Evaluation of Regular Expressions
 - Use Deterministic Automata
 - Inefficient for multiple container expressions
 2. Evaluation using DataGuides
 - Construct a trie for all XML paths
 - Efficient except for irregular and deeply nested data
 3. Evaluation using Reversed DataGuides
 - Prune on the last few tags



Experimental Results



Compression Results

Limitations

- Not designed to work with query processor
- Not efficient for messages < 20KB
 - Overhead
 - Poor **gzip** compression

References

- Hartmut Liefke, Dan Suciu. 2000.
XMill: an Efficient Compressor for XML Data
- Roy Goldman, Jennifer Widom. 1997.
DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases

Information Theory

Claude Elwood Shannon

- Entropy:

$$H \stackrel{\text{def}}{=} p_1 \log \frac{1}{p_1} + \dots + p_n \log \frac{1}{p_n}$$

- Message of m symbols cannot be compressed to less than mH bits on average.
- Almost optimal compressors exist

Information Theory

Heterogeneous Sources

- XML data is heterogeneous
Source \mathcal{S} is a collection of $k+1$ sources S_0, S_1, \dots, S_k over alphabets A, B_1, \dots, B_k .
 $A = \{a_1, \dots, a_k\}$ (tags)

Messages are of the following form:

$x_1, y_1, x_2, y_2, \dots, x_m, y_m$

where x_1, \dots, x_m belong to A , and, whenever $x_j = a_i$, then the next symbol y_j belongs to B_i

Information Theory

XMill Performance

- The number of bits used is optimal on average:

$$S = \frac{1}{2} (H_0 + p_1 H_1 + \dots + p_k H_k).$$