

CSV compaction to improve data-processing performance for large XML documents

Shigeru Yoshida, Hironori Yahagi, and Junichi Odagiri
Peripheral Systems Laboratories, Fujitsu Laboratories Ltd.
10-1 Morinosato-Wakamiya, Atsugi 243-0197, Japan
Tel: +81-46-250-8849, Fax: +81-46-248-3233

Email: {yoshida.shig-04, yahagi.hironori, odagiri.junichi }@jp.fujitsu.com

1. Introduction

XML (Extensible Markup language) is the global standardized and flexible electronic data expression form, expected to further spread through its varied and wide use. However, the XML form is much more redundant than the conventional CSV (Comma Separated Values) form, and the processing of large XML documents leads to excessively heavy CPU loads and huge memory requirements. The required CPU resource for data-processing is dependent on the number of elements in an XML document. To improve data-processing performance for XML documents in a record form, such as bills and customer lists (the data size tends to be large for business purposes), a reversible compaction method which reduces the number of apparent elements has been newly developed.

2. Method

User applications that handle XML documents usually access specific elements for processing, and rarely access all elements. Focusing on this feature, the method leaves the elements required for processing intact as they are and packs the contents of the other elements into one new element in a CSV form. Since the converted documents are XML even after compaction, they can be adapted to the user's applications with only slight modifications. The compaction is executed using XSLT [1] which is a format conversion of the basic function in an XML environment, such as Apache XML software and MSXML in Microsoft Internet Explorer6.0. To enable the user to easily perform the conversion, we let the user simply draw up an XML document, "compaction specification" which enumerates all elements in a record and specifies the elements to be compacted, allowing the XSL style sheets used for the conversion and reverse conversion to be automatically generated. This automatic generation is also executed using XSLT.

3. Evaluation

We implemented the XSL style sheets used to automatically generate the conversion and reverse conversion XSL style sheets [2], and evaluated the performance using Java XML Model Benchmark [3]. The improvement of data-processing performance was measured using test documents with the sizes ranging from 1 megabyte to 20 megabytes and the improvement of main memory requirements and parsing time of a standardized XML API software, DOM parser [4], were almost proportional to the reduction of the number of elements. The data size of an XML document was reduced by at least about 1/3 when all the elements in each record are packed into one element.

References

- [1] XSLT (Extensible Stylesheet Language Transformation) <http://www.w3.org/TR/xslt>
- [2] Fujitsu Labs, CSV compaction for XML, <http://www.labs.fujitsu.com/en/freesoft/csvc4xml>
- [3] Sosnoski Software Solution Inc., Java XML Model Benchmark, <http://sosnoski.com/opensource/xmlbench>
- [4] DOM (Document Object Model) <http://www.w3.org/DOM>