

事例報告：XMLスキーマ検証処理の性能改善

常盤茂伸[†]、田中秀宜[†]、石坂明彦[†]、
高野真一[†]、魚住栄市^{††}

[†]NTTソフトウェア株式会社

^{††}株式会社クロスリンク・コンサルティング

本稿ではXMLスキーマを用いたXML文書の妥当性検証処理の性能改善事例を報告する。本検証で用いたアプリケーションプログラムはXMLパーサを介しており重い処理である。XMLパーサで生成されるスキーマ解析結果をキャッシュする方式を提案し、これにより性能改善を図ることが可能であることを検証する。本稿で紹介する性能改善事例は、要求されたシステム性能要件を満たすための設計方式として活用できることを期待する。

Case report : The performance improvement of the XML schema verification processing

Shigenobu Tokiwa[†], Hidetaka Tanaka[†], Akihiko Ishizaka[†],
Shinich Takano[†], and Eiichi Uozumi^{††}

[†]NTT Software Corporation

^{††}CrossLink Consulting, Inc

The performance improvement case of the validity verification processing of the XML document which a XML schema was used for is reported in this paper. It is here through XML parser, and the application program used in this verification is heavy management. The form that cash does the result of a schema analysis that it is formed with XML parser is proposed, and it is verified that it is possible that performance improvement is planned by this. A performance improvement case to introduce in this paper expects that it can make use of it as a design form to satisfy demanded system performance necessary conditions.

1. はじめに

近年、XML文書[1]を取り扱うWebシステムの様々な適用事例が増えている。またXML文書の処理性能・信頼性・セキュリティなどに関する課題が多く、それに対する研究報告の事例が数多く存在する。[2][3]

本稿ではXMLスキーマを用いたXML文書の妥当性検証処理の処理性能に着目する。XMLパーサを介して妥当性検証を行う場合、アプリケーションプログラムは重い処理になる。そこで、XMLパーサで生成されるスキーマ解析結果をキャッシュする方式を提案し、これにより性能改善を図ることが可能であることを検証する。

以降の章の構成は次の通りである。まず、第2章では検証目的として妥当性検証処理の性能改善方法を提案する。第3章では検証方法と検証項目を説明し、第4章では測定結果を述べる。最後に第5章で考察と今後の課題についてまとめる。

2. 検証目的

本章では、妥当性検証処理の性能改善方法を提案する。

XML文書の妥当性検証を行う場合、XMLパーサでXMLスキーマ文書を解析処理し、その結果を参照してXMLインスタンスの妥当性検証を行う。これを以後「非キャッシュ

モデル」と表記する。この方式の場合、XML 文書の妥当性検証を行うたびに XML スキーマ文書の解析処理を必要とする。

そこで妥当性検証の性能改善方法として次の2つの方式を提案する。

第1はXMLパーサの解析処理で生成されるスキーマ文書解析結果のオブジェクトをメモリ上に常駐させキャッシュヒットさせる方式で、これを以後「メモリキャッシュモデル」と表記する。

第2はXMLパーサの解析処理で生成されたスキーマ文書解析結果のオブジェクトを一度ファイルに書き出し、妥当性検証時にファイルから再読み込みさせて解析処理を省略する方式で、これを以後「ディスクキャッシュモデル」と表記する。

この方式によりXMLパーサの解析結果をメモリ上またはディスク上にあらかじめキャッシュしておく、キャッシュの対象とするスキーマ文書の解析処理がスキップされるために処理時間の短縮が期待できる。

従って本稿の検証目的は、XMLパーサの解析処理で生成されるスキーマ文書解析結果をキャッシュすることにより処理時間の短縮を確認することである。また、従来の妥当性検証処理との比較により、提案した性能改善方法の効果および影響について明確にする。

3. 検証方法

本章では、検証方法と検証項目を説明する。

まず、本稿での検証環境は Web/AP サーバ上に J2EE プラットフォームを構築し、アプリケーションプログラムに妥当性検証処理を実装した。検証環境構成を図 3-1 に示す。

次に、第2章で述べた検証モデルを次節に述べる。

Web/APサーバ構成

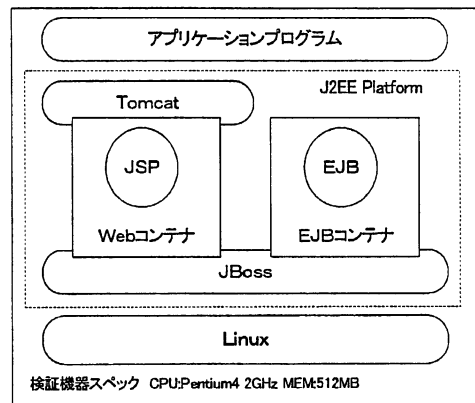


図 3-1 検証環境構成

3. 1 非キャッシュモデル

非キャッシュモデルは、XML 文書の妥当性検証時のたびに XML スキーマ文書の解析処理を行う。XMLパーサが生成したスキーマ文書解析結果のオブジェクトは処理終了後にメモリ上から削除する。(図 3.1-1)

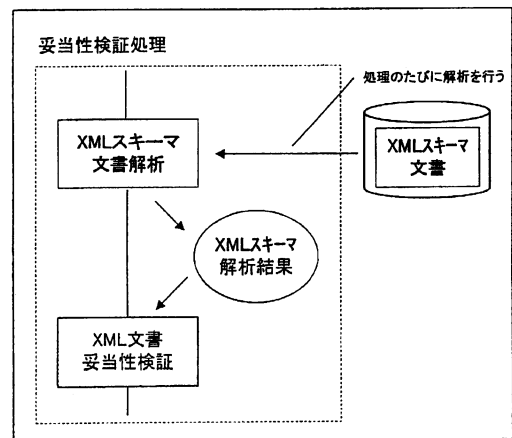


図 3.1-1 非キャッシュモデル

3. 2 メモリキャッシュモデル

メモリキャッシュモデルは、XML スキーマ文書の構造解析を予め処理しておき XMLパーサが生成したスキーマ文書解析結果の

オブジェクトをメモリ上に常駐させる。この処理タイミングを J2EE サーバ起動時に処理を行う。

XML 文書の妥当性検証時にはスキーマ文書解析結果オブジェクトがメモリ上に配置されているため、XML スキーマ文書解析がスキップすることになり、処理動作はキャッシュヒットする。(図 3.2-1)

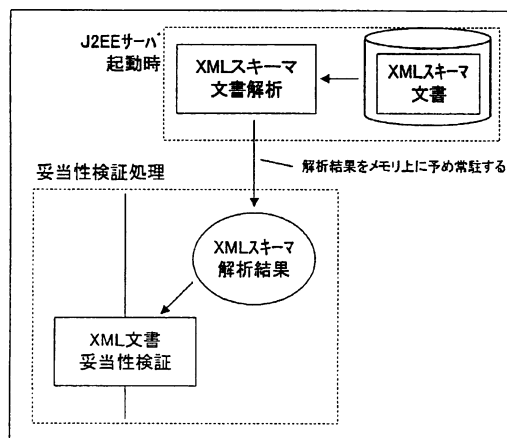


図 3.2-1 メモリキャッシュモデル

3. 3 ディスクキャッシュモデル

ディスクキャッシュモデルは、XML スキーマ文書の構造解析を予め処理しておき XML パーサが生成したスキーマ文書解析結果をオブジェクト形式でファイルに書き出す。この処理タイミングを J2EE サーバ起動時に処理を行う。

XML 文書の妥当性検証時にはスキーマ文書解析結果オブジェクトをファイルから読み込みメモリ上へ展開を行った後に、XML 文書の妥当性検証を行う。スキーマ文書解析結果オブジェクトがファイル上に格納されているため、XML スキーマ文書解析がスキップすることになる。但し、メモリ展開時にオーバーヘッドが発生する。(図 3.3-1)

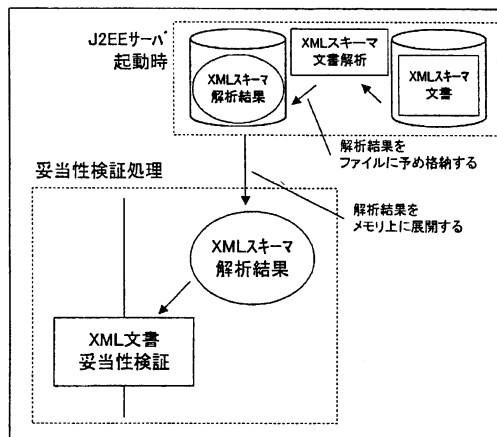


図 3.3-1 ディスクキャッシュモデル

3. 4 検証方法

検証方法は、アプリケーションプログラムで出力される処理時間、CPU 時間から前述した検証モデルの比較評価を行う。

XML 文書の妥当性検証を行う XML パーサの解析処理は、DOM(Document Object Model)方式でアプリケーションプログラムに実装した。

測定方法はXML文書をWebクライアントから送信し、APサーバ上のアプリケーションプログラムで単体走行をさせた場合の妥当性検証処理を測定する。

3. 5 検証項目

妥当性検証の処理時間は、XML スキーマ文書の構造、XML スキーマ文書のサイズ、およびXML文書のサイズに応じて変化することから、いくつかのXML スキーマ文書、XML 文書を用いて処理時間、CPU 時間を測定する。検証に用いた試験データを表 3.5-1 に示す。

表 3.5-1 試験データ

XML文書	XML文書 (kbyte)	element数	XMLスキーマ (kbyte)
A文書	102	1,849	202
B文書	746	10,622	1,677
C文書	1,436	17,669	2,049

4. 測定結果

本章では、測定結果を述べる。

検証結果を表 4-1～表 4-3 に示す。表 4-1 の処理時間結果より、XML パーサの解析処理で生成されるスキーマ文書解析結果をキャッシュすることにより処理時間が短縮されていることを確認できた。また、処理時間の改善率を表 4-2 に示すとおり、スキーマ文書解析結果をキャッシュすることにより一律に処理性能が向上している。

表 4-3 の CPU 時間結果より、CPU 時間の内訳は XML パーサの XML スキーマ文書の構造解析処理、もしくは XML 文書の妥当性検証処理に要する時間のほとんどを占めていることが分かり CPU バウンドの重い処理であるという裏付けが確認できた。

表 4-1 処理時間結果

XML文書	処理時間(ms)		
	メモリキャッシュ	ディスクキャッシュ	非キャッシュ
A文書	23	47	67
B文書	223	375	522
C文書	5,376	5,522	5,988

表 4-2 処理時間改善率

XML文書	処理時間改善率		
	メモリキャッシュ	ディスクキャッシュ	非キャッシュ
A文書	35%	69%	100%
B文書	43%	72%	100%
C文書	90%	92%	100%

表 4-3 CPU 時間結果

XML文書	CPU時間(ms)		
	メモリキャッシュ	ディスクキャッシュ	非キャッシュ
A文書	23	45	66
B文書	220	369	514
C文書	5,334	5,479	5,941

5. 考察、今後の課題

本章では、検証結果の考察と今後の課題について述べる。

第 4 章の測定結果より、検証目的に設定したスキーマ文書解析結果をキャッシュすることにより妥当性検証処理の処理時間短縮を確認できた。

メモリキャッシュモデルにおいて XML 文書サイズが小さい A 文書および B 文書では、従来の処理方式である非キャッシュモデルとの相対的な改善効果は 50% 以上になった。各検証モデルの処理動作から考察すると次の通りである。

ディスクキャッシュモデルは、メモリへの展開処理の分だけメモリキャッシュモデルに比べて処理時間が長い。そして XML パーサの解析処理がないためキャッシュレスモデルに比べて処理時間が短いことが要因であると測定結果より考えられる。

また、XML 文書サイズおよび XML スキーマサイズが大きい C 文書では、従来の処理方式である非キャッシュモデルとの相対的な改善効果はほとんど認められなかった。これは C 文書の element 数が多いために、メモリ、ディスク共にキャッシュを配置した場合でも XML 文書の妥当性検証処理時間がキャッシュの効果相殺してしまうと考えられる。

以上より、本稿で提案した XML パーサで生成されるスキーマ解析結果をキャッシュ

する方式をシステム設計に取り入れることで、システム性能要件を満たすための設計方式として活用できると考えられる。

最後に今後の課題について整理する。

本稿での性能検証では、単体走行をさせた場合の妥当性検証処理を評価した。システムの処理性能全体を検証するためには少なくとも2つの課題があると考えている。

1つはメモリ量の影響検証である。本稿で提案したスキーマ解析結果をメモリ上にキャッシュする場合、キャッシュ対象とするXMLスキーマ文書の数が増えるに従いJavaヒープ領域へのオブジェクトの格納量に注意しなければならない。使用可能なJavaヒープ領域が少なくなるとガーベジコレクションの発生頻度が高くなり処理性能への影響が顕著になるためである。

もう1つは高負荷処理時の検証である。本稿では単体走行時の処理時間の検証であったが、高負荷状態におけるスループットおよびリソース消費量（CPU使用率、メモリ使用率、ディスク使用率）の評価を行う必要があると考えている。

参考文献

- [1] W3C.ExtensibleMarkupLanguage(XML),
<<http://www.w3.org/>>,1996-2003
- [2] 朴大一, 遠山元道.XMLデータベースにおけるキャッシュ管理手法の提案, 電子情報通信学会,DEWS2005 5A-o3.
- [3] 井ノ口伸人, 吉川正俊.文書スキーマを利用したXMLファイル編成法,電子情報通信学会, DEWS2005 6B-i7.