

Analysis of the Benchmark while Extracting Data from Database or XML File for Different Platforms

Ognian Nakov , Desislava Petrova

Abstract: The purpose of the research is the comparison between the extracting time of the Database Management System (DMS) and XML files for two different platforms - .NET Framework 1.1 and .NET Framework 2.0. The two latest versions of the Microsoft database server are used as Database Management System – SQL Server 2000 and SQL Server 2005. The extraction from the XML file is performed in two different ways – by SAX and DOM parser. The fact, that the typical for Java SAX parser is not realized in .NET, is taken into account, but with some approximation the consecutive reading of elements of the XML file can be used. A special application is developed for testing. The business layout of the application is presented by two web services, respectively working on ASP .NET 2.0 and on ASP .NET 1.1.

Keywords: XML, ASP.NET, .NET Framework, SQL Server, web applications, SAX, DOM parser

INTRODUCTION

The purpose of the research is the comparison between the extracting time of the Database Management System (DMS) and XML files for two different platforms - .NET Framework 1.1 and .NET Framework 2.0, using ASP .NET.

ASP .NET is development platform, supported by .NET Framework, suited for developing web applications.

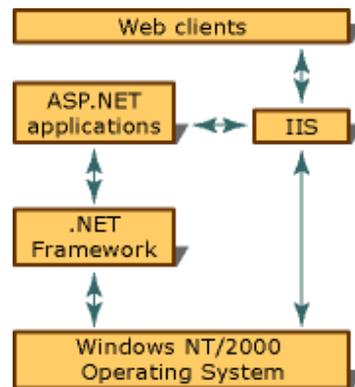


Figure 1

ASP .NET Execution Model is started, when a client sends a request for Web page for the first time. The process starts with sending of an HTTP GET request to the server from the client browser. The request is accepted, processed and directed by IIS to the necessary page, hosted on the server. ASP .NET parser interprets the code of the application, performs all necessary actions for the creating of the page. If the code hasn't been compiled to Dynamic Link Library (DLL), ASP .NET calls the compiler. The compiler compiles it into *Microsoft Intermediate Language* (MSIL). MSIL instructions are loaded and executed from The *Common Language Runtime* (CLR) environment. As a result, an HTML code is generated and sent back to the client browser.

Compiling of the script blocks to MSIL instructions leads to higher efficiency. The MSIL code is additionally compiled to machine code, using JIT compiler (just-in-time). JIT compiler compiles each code section when it is called, instead of compiling the whole application at once. As a result, .NET application starts faster. Once compiled, the code is saved until application is stopped. Therefore it is not necessary to compile the block of code each time it is called. Because of this process, it is expected, that the execution of the MSIL code will be nearly as fast as the execution of the machine code of the relevant platform. JIT compilation

supports early and late binding and handles the types during the execution. That's why MSIL code is more secure and insured against programmer's mistakes.

The ASP .NET 2.0 web application code, in contrast to ASP .NET 1.1, is not compiled to assembly during the development of the site, but during its first request. The compiled code is cached by the ASP .NET 2.0 environment and reused in next execution. If a file, that is responsible for the program logic, is changed, this cached compiled code is not valid anymore and the compilation is done again. This strategy ensures higher efficiency and easier development and support of the web applications. It is only necessary to change the determined file and save it. Changes take place immediately. The same strategy is actual for the ASPX application code. The difference between the two versions of the ASP .NET application can be seen on figure 2.

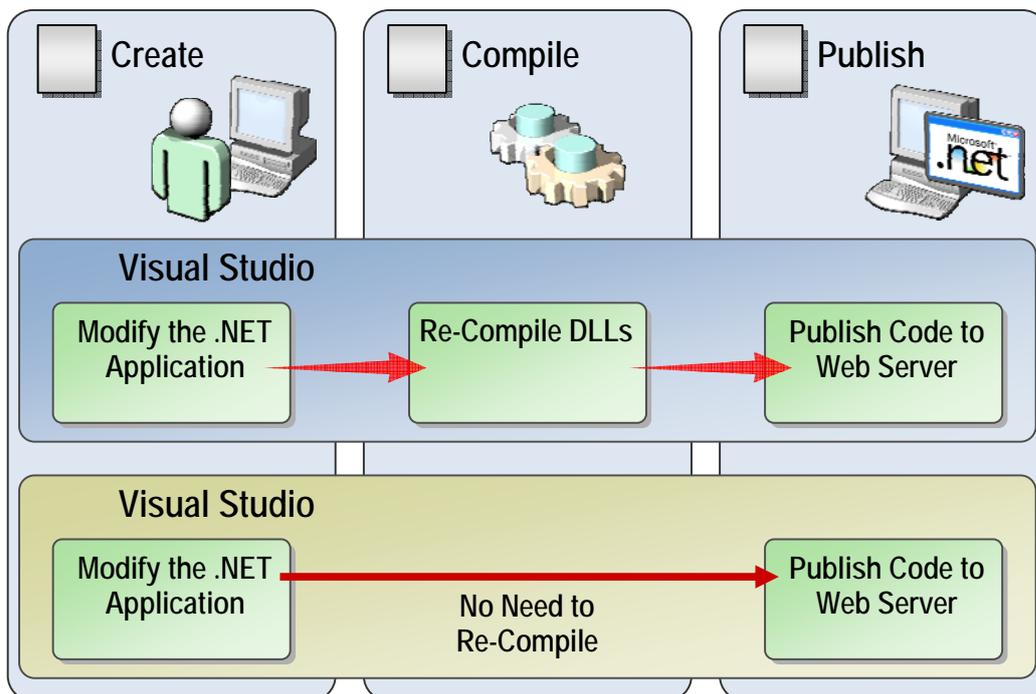


Figure 2

RESEARCH SETTINGS

The two latest versions of the Microsoft database server are used as Database Management System – SQL Server 2000 and SQL Server 2005.

In each server a table with test information is used. The number of records in each table is 20300. The structure of the tables is represented on figure 3.

StudentsTest				
Column Name	Data Type	Length	Allow Nulls	
StudentID	int	4		
FacultyNumber	nchar	14		
Fname	nvarchar	50		
Sname	nvarchar	50	✓	
Lname	nvarchar	50	✓	
Semester	int	4		
GroupNum	int	4		
IsValid	bit	1	✓	
DateInactiv	datetime	8	✓	

Figure 3

The XML file, used for the tests, contains the same information as the tables of the database. Each column from the tables is determined with one element from the XML file. The figure below illustrates an example of the representation of one record.

```

<?xml version="1.0" encoding="windows-1251" ?>
- <Students name="FKSU Students">
- <Student>
  <StudentID>32784</StudentID>
  <FacultyNumber>a21204000</FacultyNumber>
  <Fname>КИРИЛО</Fname>
  <Sname>ЕМИЛОВО</Sname>
  <Lname>ЧОБАНОВО</Lname>
  <Semester>1</Semester>
  <GroupNum>1</GroupNum>
  <IsValid>True</IsValid>
  <DateInactiv>07.1.2006 г. 20:28:54</DateInactiv>
</Student>

```

Figure 4

The extraction from the XML file is performed in two different ways – by SAX and DOM parser. The fact, that the typical for Java SAX parser is not realized in .NET, is taken into account, but with some approximation the consecutive reading of elements of the XML file can be used.

A special application is developed for testing. It can accumulate the results from the tests, so that an average value to be formed. Two tables, shown on figure 5, are used. One of the tables is a dictionary, containing definitions for each possible kind of measurements. The second contains the values from all measurements.



Figure 5

The business layout of the application is presented by two web services, respectively working on ASP .NET 2.0 and on ASP .NET 1.1. Figure 6 presents the application architecture.

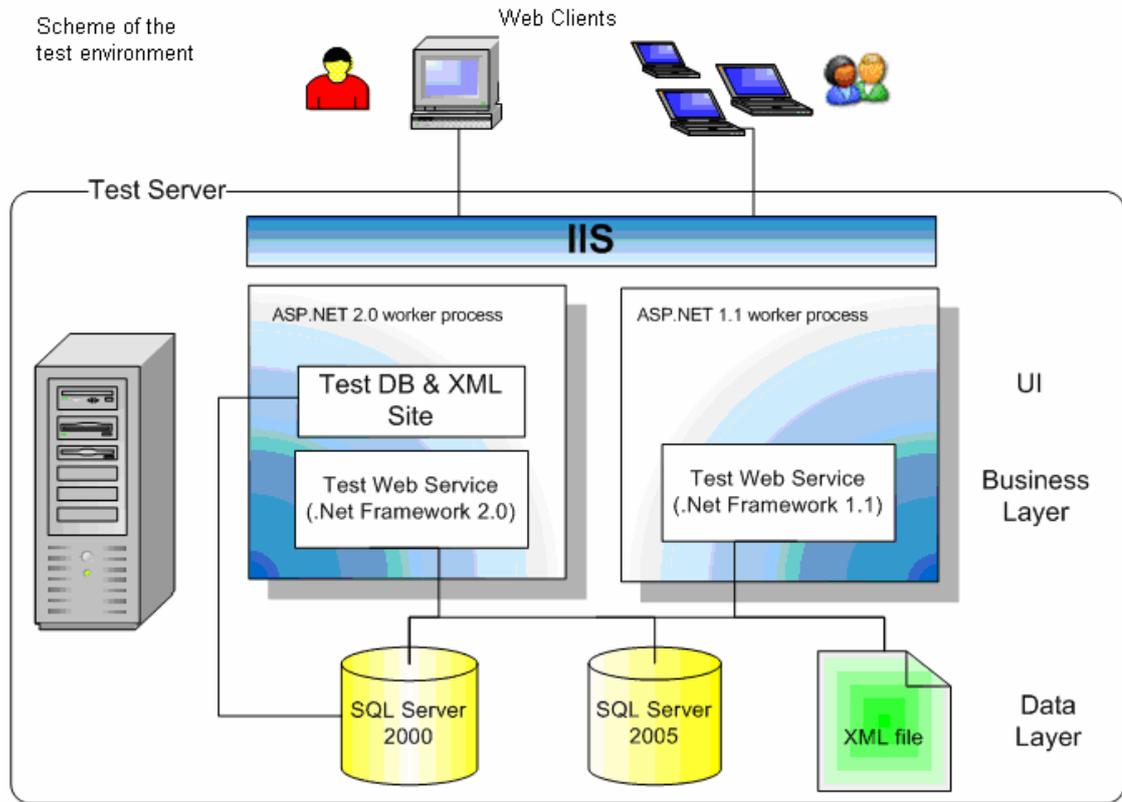


Figure 6

RESULTS

The results from the researches are summarized in the next chart:

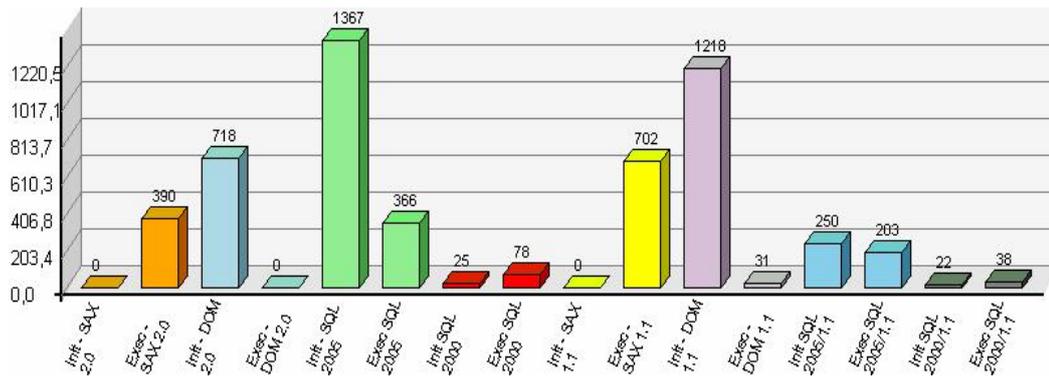


Figure 7

ANALYSYS AND CONCLUSIONS

Results show, that the minimum time for extracting of the test data is in the case of searching in a database system. In most cases the timings of the two SQL servers are equal and most probably the diversions come from differences in the transitory load of the test system. It also can be seen that the database react equally on the work with .NET Framework

1.0 and .NET Framework 2.0, which means both development environments are well optimized for work with the tested database servers.

Extracting of data from XML file is times slower. The tests are again accomplished on the two development environments - .NET Framework 1.0 and .NET Framework 2.0. The purpose of this test was to check if the announced from Microsoft increasing of the efficiency is real.

The two methods of using SAX parser and DOM parser can be applied in different scenarios. DOM parsers are appropriate for fewer amounts of data. If necessary the data can be loaded once and researched many times, because the whole XML document is loaded in the memory as DOM tree. Working this way is convenient – requests can be easily written using the XPATH language. But if this decision is chosen, the data will increase about ten times in the DOM tree. There are many cases, where the needed data is voluminous, so the variant of using DOM trees is excluded. It can be seen from the charts, that searching with DOM parser is even faster than the searching in database systems. The timings are below 1 ms, so they are visualized with 0 on the charts. It is also obvious that the time for preparation of the document in this case is more than the time for searching in database system.

The usage of SAX parsers is established on consecutive reading of the file till the information is found. So the searching time depends on the size of data, its structure and the searching algorithm. In our case we don't know where in the file is the requested information, so the whole file should be read. Additional time for loading the file is not necessary, but again the time for searching is at least 2-3 times more than the time when Database Management System is used. The whole document is not loaded in the operational memory, when using SAX parsers, so it is appropriate for working with huge arrays, if the operation is not real time or repeated very often.

In conclusion, undoubtedly, it is better to use Database Management Systems, because they have best data-extracting timings. The DMS have enlarging and flexible enough types system, especially the new SQL Server 2005. Huge advantages of the DMS are the entirety-control systems and assurance of the data, which are not available when working with XML files.

REFERENCES

- [1] Jeffrey Richter, Applied Microsoft .NET Framework Programming, SoftPress, 2002.
- [2] Angshuman Chakraborti, Microsoft .NET Framework Professional Projects, Duo Design OOD, 2003.
- [3] MOC for Managing a Microsoft Windows Server 2003 Environment, Microsoft Press, 2003.
- [4] MOC for Developing Microsoft ASP.NET 2.0 Web Applications, Microsoft Press, 2005.
- [5] Introducing Microsoft ASP.NET 2.0, Microsoft Press, 2005.
- [6] MSDN Library – Visual Studio .NET 2003.

ABOUT THE AUTHORS

Assoc.Prof. Ognian Nakov PhD, Technical University of Sofia, Phone: +359 2 965 3613, E-mail: nakov@tu-sofia.bg

Assistant Prof. Desislava Petrova, Technical University of Sofia, Phone: +359 2 965 2588, E-mail: d_petrova@tu-sofia.bg