Performance Analysis of Selected Frameworks for an Integration Platform Development

T. GÓRSKI
gorski@wat.edu.pl, tomasz.gorski@rightsolution.pl

Institute of Computer and Information Systems
Faculty of Cybernetics, Military University of Technology
Kaliskiego Str. 2, 00-908 Warsaw, Poland

The article demonstrates how to analyze the performance of an integration platform created using different frameworks. The paper contains a set of integration platform performance metrics. In the manuscript a justification of the frameworks selection is included as well. The article contains a description of a business case, which was implemented on the integration platform with selected frameworks. The article also encompasses the description of the test environment and an analysis of performance test results.

Keywords: integration platform, performance, information system development.

1. Introduction

The introduction of new services or improvement of existing services in companies involves the development of new systems, often made in technologies newer than existing systems. This approach results in the creation of multiple applications running within the same organization with similar functionality and storing similar or even the same data. This raises the costs of maintaining IT systems and complicates the implementation of business processes run by many branches of the same organization. Factors have caused the demand for technologies and solutions that enable integration of existing systems and applications, as well as the reuse of already created functionality [2], [8], [11], [12], [13]. Another problem is how to manage a large number of functionally related information systems as part of an enterprise architecture [7]. There are many concepts and the integration of information systems, lately service-oriented architecture (called Service-Oriented Architecture – SOA) [5] is in use. The central element in such solutions is a bus service (Enterprise Service Bus – ESB) [3]. The market offers a variety of frameworks, allowing the construction of the service bus, which is part of an integration platform, consistent with the approach of SOA [4], [9]. Crucial is also the approach to design the integration platform [1], [6]. These solutions differ from each other not only at the cost of purchase, but also by the ease of integrating the solution development and hardware needs of the system. It becomes essential to choose a framework for the integration platform development by providing the required level of performance for the systems adopted by considered organization.

2. Integration Platform Parameters

The two main aspects are considered when making
a decision on choosing a framework to build the platform:

- technical parameters of constructing the platform environment, its performance and ease of management
- price of framework for platform construction (license and the required equipment) and the cost of its maintenance.

The most important element when choosing a framework to build an integration platform, is to estimate the current and anticipated future requirements. The basic ones are:

- platform throughput P_p – the number of possible messages to serve on a platform that is sent between integrated systems in time unit
- number of integrated systems N – number of systems connected to the integration platform, which interchange messages
- the number of services exposed from system u_i – the number of services from each of the integrated systems exposed on the integration platform, $i = \overline{1, N}$
- maintaining code that performs the integration on the platform – time to implement solutions on the platform, the cost of system extension.

One of the key integration platform's properties is its performance. Before choosing a framework to build an integration platform, the following issues should be considered:

- the estimated traffic on the platform R_p – the number of messages exchanged per unit of time among each of the integrated systems

$$R_p = \sum_{i}^{N}\sum_{j}^{N} l_{ij} \tag{1}$$

where: l_{ij} – number of messages sent from system i to j

- hardware requirements – resources for each of the frameworks to support the operation of the platform.

Helpful in making the right choice may prove to be a clearly defined set of performance measures of the integration platform. Such measures can include:

- the amount of RAM allocated by an integration platform
- percentage usage of CPU(s)
- the number of transactions supported per unit of time
- the number of messages processed on the platform per unit time.

Hardware requirements can be described by the following indicators:

- the minimum required disk space
- the minimum amount of RAM allocated for the platform
- the maximum amount of RAM allocated for the platform
- the minimum number of processors needed to run the platform
- the maximum number of processors, on which the platform may run.

3. Frameworks Selection

At present, on the market there are many frameworks to build an integration platform. Two of them were chosen for comparison:

- webMethods – the product is delivered by Software AG
- Mule – product is supplied by MuleSoft.

Selected frameworks differ significantly in terms of:

- hardware requirements – the first of them requires far more hardware resources
- price for the license – the first product is an expensive solution and Mule is an OpenSource product
- ease of installation and configuration
- design and implementation of business processes.

Selected frameworks, though very different, allow the implementation of the business case on an integration platform.

4. Business Case to Deploy on an Integration Platform

The main task of the integration platform is to provide communication between the systems connected to it, both synchronous and asynchronous. The implementation of this assumption also covers the issue of transformation of messages. Next, the design and integration issues of the integration platform construction, was presented for selected frameworks. In the considered business case it was assumed that three systems integrate their activities:

- PS CRM – PeopleSoft CRM
- CBD – Central database of debtors
- MGC – Module generator of contracts.

Integration of the systems under consideration is required due to the implementation of the following use cases of information systems:

- prepare contract – the case realizes the functionality of sending orders to prepare the contract for the client to be sent for signing
- lock account.

The use case "Prepare contract" needs asynchronous communication between the PS CRM system and the platform and between platform and system MGC (Figure 1).

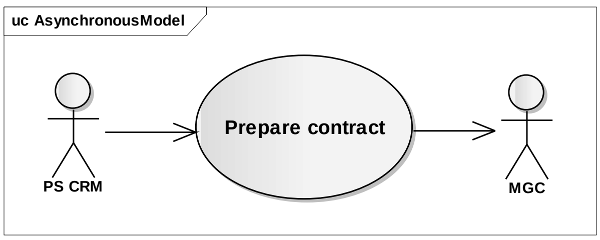

Figure 1. Asynchronous communication between PS CRM and MGC systems

The use case "Lock account" needs synchronous communication between the CBD system, the integration platform and between the platform and system PS CRM (Figure 2).

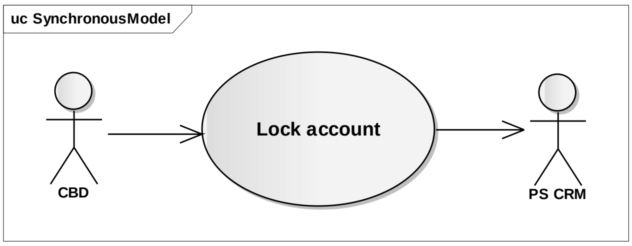

Figure 2. Synchronous communication between CBD and PS CRM systems

5. Description of Messages Required by Use Cases

The use case ”Prepare contract” requires the implementation of asynchronous communication between systems PS CRM and MGC. Message sent from the PS CRM contains a request for contract generation for the customer. Table 1 shows the description of fields in the message.

Table 1. Message from PS CRM fields descriptions

No	Field name	Field business description
1	WCY_GEN_CONTRACT_REQ_MSG	Root element – request
2	Requested	Request Id in PS CRM system
3	clientId	Client Id registered in PS CRM system
4	contractType	Contract type in PS CRM
5	requestSystem	Name of source system
6	destinationSystem	Name of destination system
7	orderSubType	Order type
8	orderStatus	Status
9	orderDate	Date of order dispatch for service

Message sent from the integration platform to the MGC system includes a request to generate the contract for the customer. This message is consistent with the definition of the MGC interface. This message has been prepared on the basis of the input message from the PS CRM system and the logic contained in the implementation of services. Table 2 shows description of fields in the message.

Table 2. Message from MGU fields descriptions

No	Field name	Field business description
1	WCY_WM_GEN_CONTRACT_REQ_MSG	Root element – request
2	Requested	Request Id in PS CRM system
3	clientId	Client Id registered in PS CRM system
4	templateId	Number of document type in MGC system
4	contractType	Contract type in PS CRM
5	requestSystem	Name of source system
6	orderSubType	Order type
7	orderStatus	Status
8	orderDate	Date of order dispatch for service
9	Result	Root element – processing result on EAI layer
10	errorCode	Error code – ”0” when successful processing
11	errorDescription	Error description – „” when successful processing

The use case ”Lock account” requires the implementation of synchronous communication between systems PS CRM and CBD. The CBD System initiates message flow on the platform and instructs PS CRM system to lock the customer’s account due to debts. CBD invokes this service by using HTTP, and gets a response about inserting an appropriate message to the MQSeries queue of the PS CRM system. Table 3 shows the description of fields in the message.

Table 3. Message from CBD fields descriptions

No	Field name	Field business description
1	customerId	Client number in CBD system
2	serviceName	Service name
3	serviceId	Detailed code of client’s service
4	orderReason	Request sending of lock account reason

Message sent from an integration platform to the PS CRM system includes a request to lock the customer's account in the PS CRM system. The message contains details about the type of service, the client and the period for which service is to be blocked. Table 4 shows the description of the message.

Table 4. Message from platform to PS CRM fields descriptions

No	Field name	Field business description
1	WCY_WM_CUSTOMER_SERVICE_REQ_MSG	Root element – request
2	lockTime	Lock time period for account
3	orderDate	Date of order dispatch for service
4	customerId	Client number in CBD system
5	serviceId	Detailed code of client's service
6	orderReason	Request sending of lock account reason
7	serviceName	Name of blocked service
8	businessErrorCode	Business error code in CBD

The role of the platform in the implementation of use cases is both the appropriate control of the flow of messages and their structure adaptation so that the target system can properly receive and process them.

6. Use Cases Realization with webMethods

The implementation of use cases with webMethods framework encompasses the following steps:

- Creation webMethods packages of channels and adapters for external systems and the enterprise layer
- Creation and transfer of file ".bindings" to the server webMethods containing definition of connection and aliases of the MQ Series queues
- Start-up of IntegrationServer instance and configuration of webMethods Broker
- Defining of connections to the MQ Series queues
- Create JMS notifiers for channels, which take messages from the queue for further processing
- Creation of services, which receive messages from channels and process deployment on the enterprise layer
- Creation of services responsible for the proper messages processing
- Creation of services of external systems' adapter.

Figure 3 presents an UML sequence diagram showing the design of the use case "Prepare contract".

Activities carried out on the integration platform are implemented using the services created for the project or provided by the platform. Figure 4 presents an UML sequence diagram showing the design of the use case "Lock account".

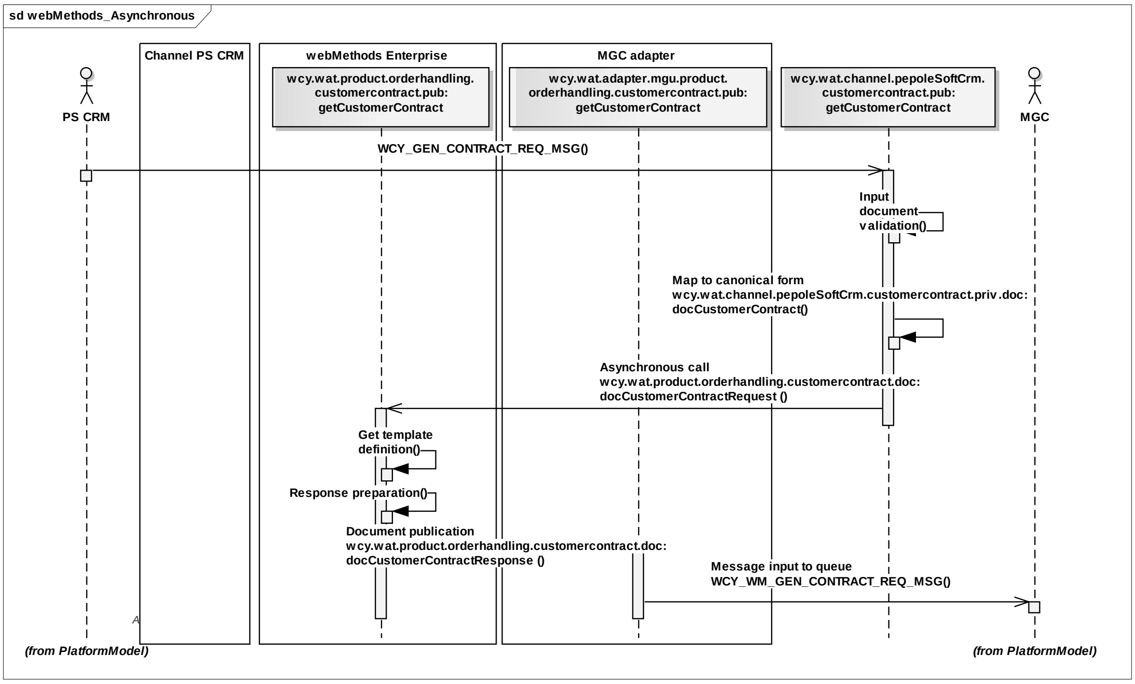

Figure 3. Realization of use case "Prepare contract" with webMethods framework

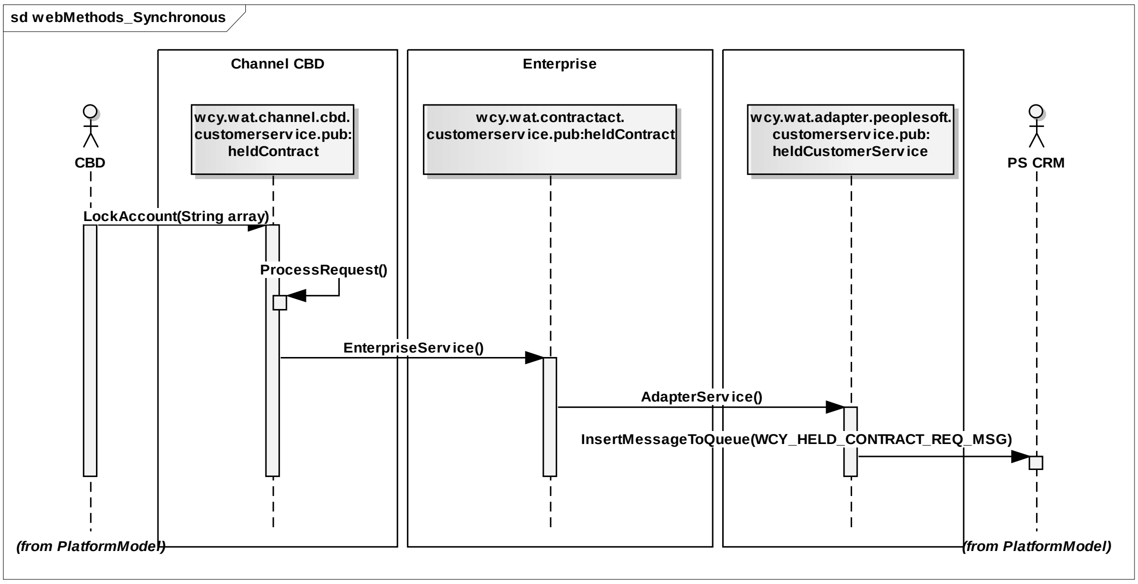

Figure 4. Realization of use case ”Lock account” with webMethods framework

7. Use Cases Realization with Mule

In the realization of use cases with the Mule framework, a key element was to define a connector for an IBM MQ Series queue. After definition of the connector, you can both read and write messages to the MQ Series queue. Figure 5 presents an UML sequence diagram showing the design of the use case ”Prepare contract”. Activities carried out on the platform are implemented using the built-in mechanisms of the platform.

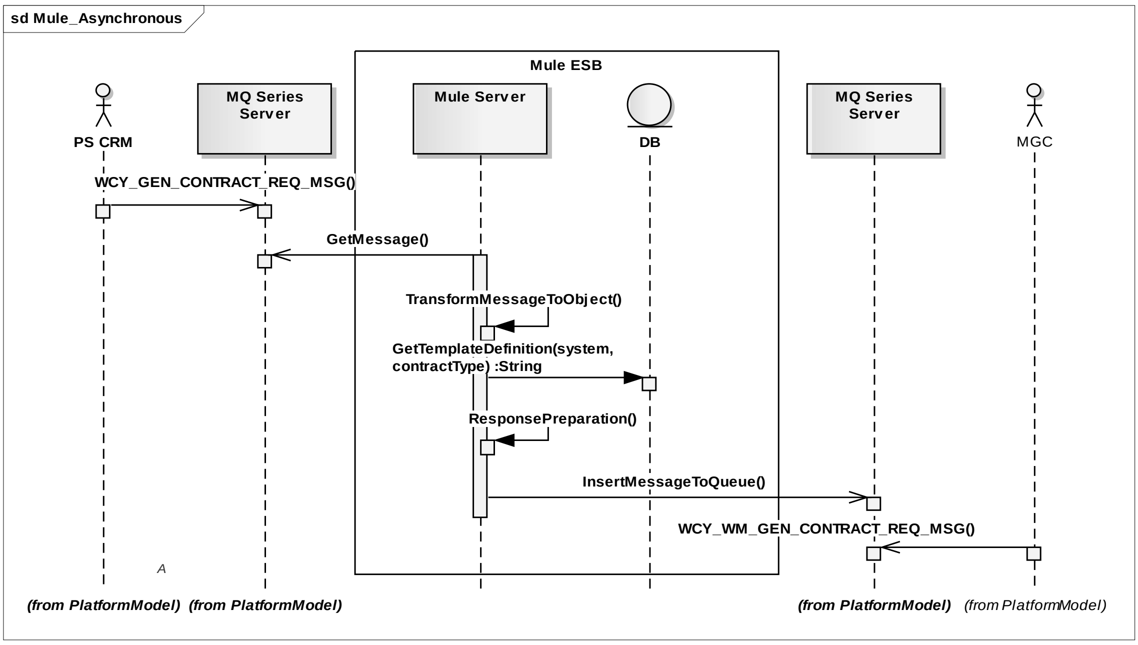

Figure 5. Realization of use case ”Prepare contract” with the Mule framework

Figure 6 presents an UML sequence diagram showing the design of the use case ”Lock account”. Activities carried out on the integration platform are implemented using the services created for the project or provided by the platform. All services in the realization are called synchronously.

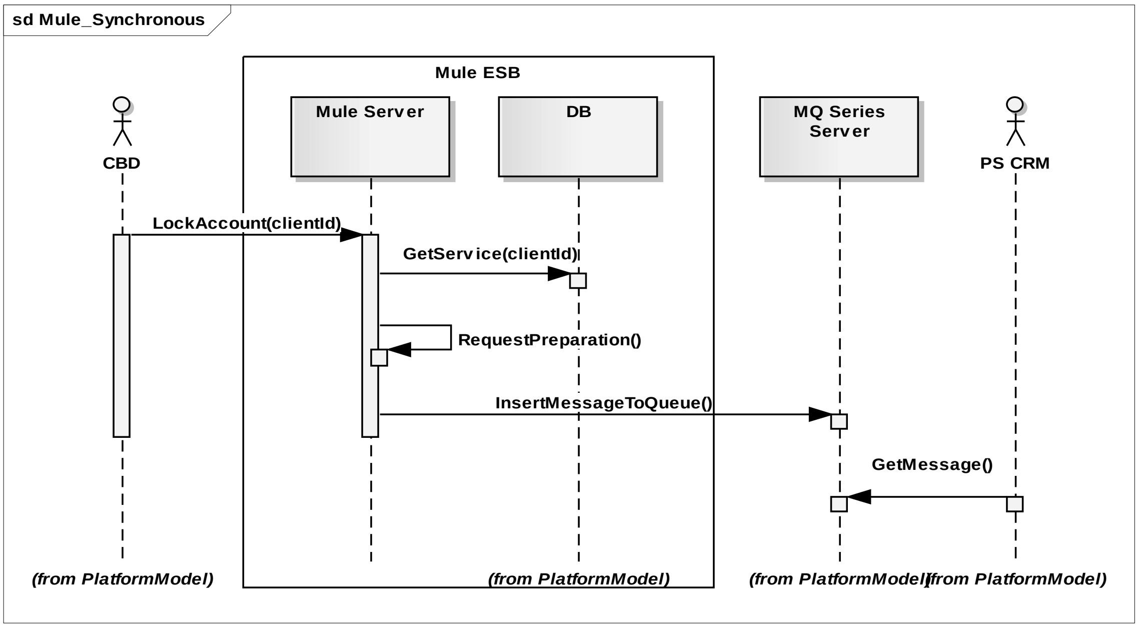

Figure 6. Realization of use case ”Lock account” with the Mule framework

8. The Test Environment

Each of the selected frameworks was installed on the same test environment. Both frameworks were installed on the same hardware server with the following characteristics: operating system – Ubuntu 8.04.3 LTS; RAM – 4GB, 4GB of memory paging (swap); Hard Disk Drive – 160GB; Processor – 8 x Intel (R) Pentium (R) D CPU 2.80GHz.

In addition, the following software was installed on the hardware server: database: Oracle Database 10g Release 10.2.0.1.0 – 64bit; Java: JRE 1.5, JDK 1.5.0_16; frameworks: Mule, webMethods; Queues Server IBM MQSeries.

In order to generate communication between systems a Java application was created. The application is able to simulate the synchronous and asynchronous communication between considered systems and the integration platform. For realization of asynchronous communication the application is responsible for:

- Preparation of the input message for processing
- Placement of the message in the MQSeries queue
- Waiting for the arrival of the message into queue of the target system
- Verification of the correctness of the received message
- Report generation of processing time.

To ensure synchronous communication, the application invokes request of the service by protocol http and checks the result sent in response. In Mule framework service is exposed as a web service whereas in webMethods service is invoked using the framework’s API.

9. Performance Research Results Analysis

Series of performance tests, for both synchronous and asynchronous communications, was carried out for selected frameworks with using, described above, test application. Next are presented collected values of performance metrics of the integration platform. Platform’s performance was tested with different levels of load. Number of messages simultaneously generated on the platform to perform was following: 10, 100, 500, 1000, and 3000. It was important to verify platform performance in response to simultaneous service group of messages (e.g.: invoice generation in a company at the end of month). Table 5 shows the results for integration platform built with webMethods. Table 6 shows the results for an integration platform built with Mule.

Table 5. Performance tests results for the webMethods framework

Synchronous communication				Asynchronous communication		
Number of messages	Processing time [s]	Percentage usage of CPU [%]	Mean RAM usage [KB]	Processing time [s]	Percentage usage of CPU [%]	Mean RAM usage [KB]
10	1.2	1.0	499244	8	0.7	456917
100	8	1.4	499454	35	1.3	454305
500	70	2.9	496996	170	2.4	454206
1000	146	3.6	497265	369	3.5	451039
3000	390	3.8	526863	1378	3.6	461379
Minimum amount of RAM allocated for the platform				256MB		
Maximum amount of RAM allocated for the platform				512MB		
RAM memory allocated by platform				366MB		

Table 6. Performance tests results for the Mule framework

Synchronous communication				Asynchronous communication		
Number of messages	Processing time [s]	Percentage usage of CPU [%]	Mean RAM usage [KB]	Processing time [s]	Percentage usage of CPU [%]	Mean RAM usage [KB]
10	1.5	1.2	701246	7	1.2	602229
100	15.2	1.3	701301	81	1.2	602947
500	120	1.35	732602	450	2.0	605987
1000	313	1.4	797405	938	2.1	605919
3000	926	1.52	833682	3733	2.3	623982
Minimum amount of RAM allocated for the platform				None		
Maximum amount of RAM allocated for the platform				None		
RAM memory allocated by platform				41.8 MB		

Processing time of the specified number of messages simultaneously generated on the integration platform was analyzed. Figure 7 shows the processing time dependence of a certain number of messages for both frameworks using synchronous communication. Figure 8 shows the processing time dependence of a certain number of messages for both frameworks using asynchronous communication.

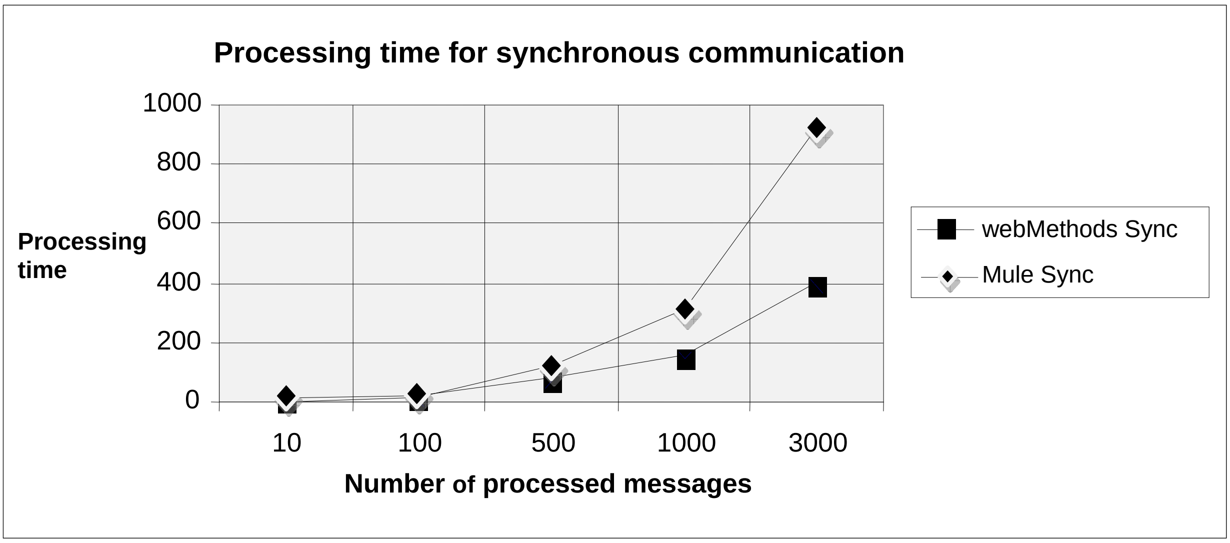

Figure 7. Processing time simultaneously generated a number of messages for synchronous communication

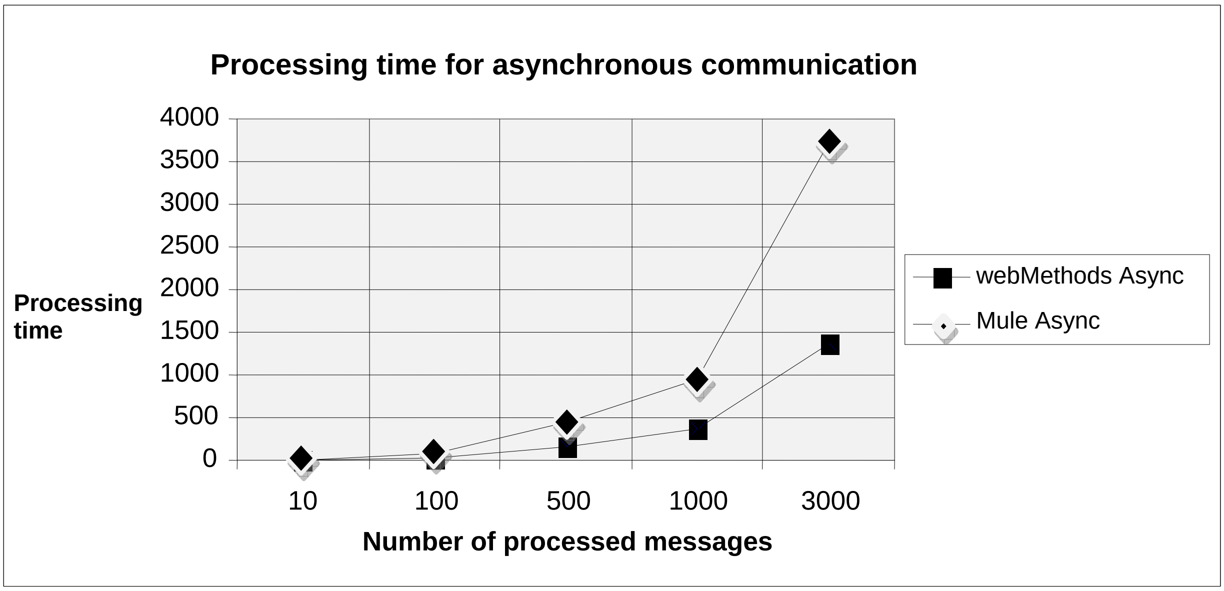

Figure 8. Processing time simultaneously generated a number of messages for asynchronous communication

An analysis of the results shows that the platform built with webMethods, not only sends messages about 2-3 times faster, but also copes far better with the increasing communication load. Response times remain practically at the same level. For example, for asynchronous communication for 1,000 and 3,000 messages response times remain within 0.5 second. In the case of the Mule framework, response time for the platformwhich processes 3,000 messages in relation to the response time for the platform that processes 1,000 messages is approximately 266% (from 1 second to 2.66 seconds). By contrast, a predominance of Mule is the consumption of resources. The platform built using Mule needs almost 9 times less RAM for its operation.

10. Summary

The article demonstrates how to analyze integration platform performance constructed using different frameworks. Two frameworks were examined: Mule and webMethods. It is important that by knowing the estimated traffic on the integration platform, you can consciously choose a solution matching the actual needs for integration platform development. In addition, performance requirements can then be verified with the issues of costs to be incurred for the construction of the platform in terms of hardware and licensing requirements. Based on the survey and collected values of metrics, it can be concluded that each of the compared platforms have its strengths and weaknesses. Mule is a framework for small and medium-sized projects. This framework has small hardware requirements and handles well basic types of messages. However, using WebMethods high-performance integration platforms can be built. Integration solutions generally have imposed requirements on response times. Framework webMethods is intended rather for large companies, not only because of its good performance, but primarily in terms of hardware requirements and prices. Performance analysis of communication modes showed that response times are much shorter for both frameworks in case of synchronous communication. However, asynchronous communication through message queues unloading mechanisms, allows for a load distribution system over time and eliminating the possibility of a critical system overload.

11. Bibliography

[1] N. Bieberstein, *Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap*, IBM Press, 2005.

[2] *Business Process Modeling Notation (BPMN) 1.1*, OMG, http://www.omg.org/spec/BPMN/1.1/PDF, 2008.

[3] D. Chappell, *Enterprise Service Bus*, O'Reilly, 2004.

[4] R. Credle, *Patterns: SOA Design Using WebSphere Message Broker and WebSphere ESB*, IBM Press, 2007.

[5] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, 2005.

[6] T. Górski, „Zwinność i dyscyplina w podnoszeniu efektywności zespołów projektowych", *Biuletyn Instytutu Systemów Informatycznych*, No. 6, 2010,

[7] T. Górski, „Metoda modelowania architektury korporacyjnej z zastosowaniem UML", in: *Wstęp do architektury korporacyjnej*, B. Szafrański, A. Sobczak (ed.), WAT, 2009.

[8] JSR-000208 Java Business Integration 1.0, Sun Microsystems, http://jcp.org/ aboutJava /communityprocess/final/jsr208/index.html, 2005.

[9] M. Keen, *Patterns: SOA with an Enterprise Service Bus in WebSphere Application Server V6*, IBM Press, 2005.

[10] M. Keen, *Patterns: Implementing an SOA Using an Enterprise Service Bus*, IBM Press, 2004.

[11] UDDI Version 3.0.2, OASIS 2004, http://uddi.org/pubs/uddi-v3.0.2-20041019 pdf, 2004.

[12] Web Services Business Process Execution Language Version 2.0, OASIS 2007, http://docs.oasis-open.org/wsbpel/2.0 /OS/wsbpel-v2.0-OS.doc, 2007.

[13] Web Services Description Language (WSDL) Version 2.0, W3C 2007 http://www.w3.org/TR/wsdl20/, 2007.

Badanie wydajności wybranych środowisk budowy platformy integracyjnej

T. GÓRSKI

W artykule przedstawiono sposób badania wydajności platformy integracyjnej utworzonej przy wykorzystaniu różnych środowisk. Artykuł zawiera opis metryk wydajności platformy integracyjnej. Analizie poddano także zakres narzędzi wspierających projektowanie rozpatrywanych rozwiązań. Artykuł zawiera opis przypadku biznesowego implementowanego na platformie integracyjnej. Przedstawiono w nim także konfigurację środowiska testowego oraz dokonano analizy wyników badań wydajnościowych.

Słowa kluczowe: platforma integracyjna, wydajność, projektowanie systemów informatycznych.