Proceedings of the International Conference on Information Technologies (InfoTech-2008)
19th – 20th September 2008, Bulgaria
vol. 2

APPLICATION SERVER EVALUATION METHODS IN VARIOUS TASK CLASSES' DECISION MAKING

Janis Graudins
Software Engineering Department, Riga Technical University
e-mail(s): johnyk23@inbox.lv
Latvia

Abstract: The paper studies some aspects of an application server selection for various task classes. The multi tier application architecture has been illustrated and the definition of application server made. A set of assessment criteria is defined and two evaluation methods are used. Results of evaluation of eight application servers are considered as well as benefits and drawbacks of applied methods.
Key words: application server, analytical hierarchy process, multiattribute comparing method.

1. INTRODUCTION

For the last years most of the companies started to use different types of software systems to improve their efficiency and reduce expenses. Some of them needed more complex software, which cannot be implemented as a standalone application. To satisfy these needs a client-server application model with multi-layer architecture was developed. This model contains three or more layers (Fig. 1):

- client or consumer layer that usually includes simple, WEB or WEB services applications;
- server layer, which implements application business logic and is deployed on an application server;
- data layer, which provides application data. Can include databases, legacy applications or other systems of record (SOR).

The advantage of this model is application functionality replacement from the client to the server layer. This gives an independence from the client resources and allows increasing the application performance, scalability and functionality.

An application server is a kind of software which is placed on the server (middle) layer and connects client layer with data storage, realizing such services as server component management, fault tolerance, load balancing, transaction management, security management, server administrating tool and some other services (Gupta).

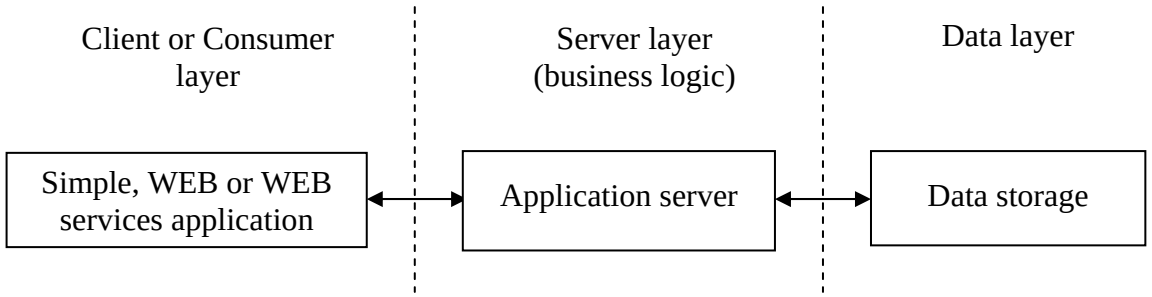

Fig.1. Classic 3-layer architecture

Furthermore, large enterprises, which have tens and hundreds of disparate systems, gradually start using service-oriented architecture (SOA). SOA advocates an approach in which a software component provides its functionality as a service that can be leveraged by other software components (Why ESB and SOA, 2006).

To integrate old and new, service-oriented architecture (SOA) needs an infrastructure that can connect any IT resource, whatever its technology or wherever it is deployed. To be flexible, it needs an infrastructure that can easily combine and reassemble services to meet changing requirements without disruption. And to be dependable, it needs an infrastructure that is robust and secure. Such infrastructure is provided by enterprise service bus (ESB).

SOA does not contradict with client-server architecture, but makes it more complex avoiding point to point integration (incorporating ESB as a hub). So, in case of SOA usage, ESB acts as a server consumer (client) and each system's application server acts as a service provider.

2. PROBLEM DEFINITION

There are tens or even hundreds of application servers available in the application server market, various editions and releases, and their number still grow. Each application server vendor has its own strategy and view on the processes in IT industry. It means that different servers provide different functionality depending on vendor view, have their strengths and weaknesses.

Furthermore, application servers can be based on different platforms supporting completely different standards.

From the other side each application server consumer (usually an enterprise) has its own requirements to a server. Requirements can vary heavily depending on project purpose, budget and other restrictions. Very often application server is selected to be

used for the whole enterprise (not just for one application) and making the right choice may significantly influence enterprise future.

To summarize, the appropriate application server selection is a key decision for the enterprise. So, providing guidelines and evaluation methods for server selection can significantly simplify the process of appropriate product acquisition.

3. PROBLEM SOLUTION

To help enterprise in appropriate server selection a set of comparing criteria was created. This list allows enterprise to asses candidate servers by each criterion and then using one or both of the provided evaluation methods to select best server for their needs.

However, each enterprise's requirements are unique and it means that significance of each criterion differs for different enterprises. Author has decided to classify all enterprises into groups by their overall requirements to the application server. Such classification allows providing of predefined significance values for each criterion in each group of requirements.

Eight application servers were selected for analysis, which take significant part on the application server market. List includes Borland Enterprise Server 5.2., IronFlare Orion 2.0.2., Oracle9i Application Server 9.0.2., BEA WebLogic Server 8.1 SP2, Windows Server 2003, IBM Websphere Application Server 5.1., JBoss 2.4.4. and Macromedia JRun 4 Service Pack 1a. Windows Server is based on .Net platform, while other servers on J2EE platform.

Two methods were used for application server evaluation: multiattribute comparing method and Analytic Hierarchy Process method (AHP). While, multiattribute method considered requirement groups, AHP method was used for overall comparison for a subset of application servers (including Borland, Oracle, BEA WebLogic, IBM WebSphere and JBoss)

3.1. Server evaluation criteria

48 evaluation criteria were selected for application server comparison, some of them from (Brebner et al., 2002) and (Graudins, 2004). To simplify server evaluation criteria were organized in 12 categories. However, some categories and separate criteria can be applied to specific application server platform only. Criteria categories, their description and platform appliance are shown in Table 1.

Table 1

Evaluation criteria categories

Category	***Criteria list***	***Description***	***Platform***
J2EE support	*Component support; Protocol support; Unit support; Services support; Technology version support*	*Server compliance to J2EE specification requirements*	*J2EE*
EJB container and bean support	*Container functionality; Session bean support; CMP additional opportunities; CMP entity finder methods; Message-driven bean support*	*Server components*	*J2EE*
WEB component support	*WEB server support; WEB container support; WEB services support*	*WEB support possibilities*	*J2EE, .Net*
Services	*Transaction support; Database support; Security; Naming service; Services flexibility*	*Common services support*	*J2EE, .Net*
Additional functionality	*Additional functionality; Integration opportunities; Other technology support*	*Additional functionality, supported extra services and technology, integration opportunities with extra tools*	*J2EE, .Net*
Scalability opportunities	*Clustering; Fault tolerance; Load balancing*	*Scalability*	*J2EE, .Net*
Development and Deployment	*Tool common functionality; Server component development; Server component deployment; System robustness in development time; Unit independence; Compatibility opportunities*	*Presence of tools for component development and deployment*	*J2EE, .Net*
System Management	*Server installation opportunities; Server and services administration; Server components administration and control; Debugging and logging*	*System administration and control*	*J2EE, .Net*
Product adaptation	*Putting into operation opportunities; System required resources and platforms; Documentation, samples and examples; Product distribution level; Prices; Version releases*	*Putting into operation and support opportunities*	*J2EE, .Net*
Server performance	*Server performance; Performance costs*	*Server performance*	*J2EE, .Net*
Specific properties	*Intuitive understanding; Certificates, awards; Specific points*	*Criteria not included in previously described categories*	*J2EE, .Net*
.Net support	*Component support; Protocol support; Services support*	*.Net platform support*	*.Net*

3.2. Classification of software systems

Author has classified systems with similar requirements into five groups (Graudins and Zaitseva, 2005a):

- Large enterprise application development. In this group the requirements are distributed homogeneously, because this group requires server to support well primary functionality, additional functionality, performance, scalability opportunities, development and deployment opportunities. Server is used as single service.
- Large enterprise application development with requirements for additional functionality is similar to previous group, but more important is server additional functionality. Server is used only as a part of large platform. For example, portal with complex functionality services.
- Large enterprise application development with requirements for high performance and fault tolerance group. In this group banking or military system can be meant.
- Small and medium enterprise application development group. The most important requirements are low cost and good integration opportunities (for example, with freeware products to reduce costs).
- Specific cases. Group is used when server is used in cooperation with some defined system and this cooperation highly influences server performance and functionality (for example, Oracle server with Oracle DB).

3.3. Multiattribute method

Multiattribute comparing method (Keeney and Raiffa, 1976) takes into account each server's evaluation criterion. This method consists of five steps (Graudins and Zaitseva, 2005b).

Step 1. On the base of the system requirements categories of criteria are estimated, giving weights $\mathbf{w_i}$ for each of them. So, $\mathbf{W = \{ w_1, w_2 , \ldots, w_n \}}$, where **n** – number of categories, $\mathbf{w_i}$ - ***i***-category's weight coefficient (value from 0 till 1). The sum of $\mathbf{w_i}$ (i=1, 2, …, n) is equal to 1. Value $\mathbf{w_i}$ depends on overall category criteria importance for the system.

Step 2. Analogically criteria of each category are estimated: $\mathbf{z_i = \{ z_{i1}, z_{i2} , \ldots, z_{im} \}}$, where $\mathbf{z_{ij}}$ - ***i***-category's ***j***-criterion weight coefficient (value from 0 till 1), **m** – number of criteria in category ***i***. The sum of $\mathbf{z_{ij}}$ (i=1, 2, …, m) is equal to 1. Value $\mathbf{z_{ij}}$ depends on criterion importance in category.

Step 3. Each application server is estimated, assigning such value $\mathbf{tij(x_k)}$ to every criterion ***j*** of separate category ***i*** that corresponds to possibility to realize the system requirements. Each server is evaluated by each criterion with value in range of 0..1 and usually it is done by experts (0 – feature not supported, 1 – fully supported by

server) After all criteria of a server $\mathbf{x_k}$ are estimated it is possible to calculate server's evaluation $\boldsymbol{Vi(x_k)}$ for appropriate category $\boldsymbol{i}$ using formulae:

$$V_i(x_k) = \sum_{j=1}^{m} z_{ij}\, t_{ij}(x_k), \quad (1)$$

where $\mathbf{t_{ij}(x_k)}$ – server $\mathbf{x_k}$ $\boldsymbol{i}$-category's $\boldsymbol{j}$-criterion estimation; $\mathbf{z_{ij}}$ – $\boldsymbol{i}$-category's $\boldsymbol{j}$-criterion weight coefficient; $\boldsymbol{m}$ – number of criteria in category $\boldsymbol{i}$; $\boldsymbol{k}$ – number of examined servers.

Step 4. The overall evaluation of each server $\mathbf{x_k}$ is calculated using formulae:

$$V(x_k) = \sum_{i=1}^{n} w_i\, V_i(x_k), \quad (2)$$

where $\boldsymbol{V_i(x_k)}$ – server $\mathbf{x_k}$ $\boldsymbol{i}$-category's evaluation; $\mathbf{w_i}$ – $\boldsymbol{i}$-category's weight coefficient; $\mathbf{n}$ – number of categories.

Step 5. Each server has $\boldsymbol{V(x_k)}$ evaluation value for each of the requirement groups, so the best server is one which has best score for selected requirement group.

3.4. Analytic Hierarchy Process method

The basic idea of the Analytic Hierarchy Process method is to convert subjective assessments of relative importance to a set of overall scores or weights. This method consists of eight steps (Saaty, 1996).

Step 1. Goal definition. The goal can be best application server detection.

Step 2. Criteria hierarchy definition. All criteria need to be organized in hierarchical tree and then method is applied for each hierarchy level. Author has used only one level of criteria hierarchy, which included criteria categories only (considering set of criteria as a while).

Step 3. Alternative list selection, which includes list of five candidate application servers.

Step 4. For each pair of criteria, a pairwise comparison is done to detect the relative importance of the two. The responses can use the following nine-point scale expressing the intensity of the preference for one criterion versus another (1=equal importance, 3=moderate importance or preference of one over another, 5=strong or essential preference, 7=very strong or demonstrated importance, 9=extreme preference). The result is arranged in a pairwise comparison matrix **C** of size **m**X**m** (where **m** is number of criteria).

Step 5. Each criteria hierarchy level (bottom-up) matrix creation and matrix eigenvector computation. Matrix eigenvector is computed in three steps: pairwise matrix is raised to powers that are squared each time, the row sums are calculated and normalized, process repeats until difference between these sums in two consecutive calculations is smaller than a prescribed value.

Step 6. Determine the relative ranking of alternatives under each criterion by creating appropriate server pairwise matrixes and computing their eigenvectors.

Step 7. Best alternative detection by multiplying alternative evaluation matrix by criteria ranking matrix.

Step 8. Benefit to cost ratio calculation.

4. EXPERIMENTAL RESULTS

All criteria weights and application server evaluations by each criterion were determined on the base of comprehensive application server and its documentation analysis, server usage in real IT projects. Independent expert viewpoint were considered as well as Internet reports and sources (for example, for server performance).

The results of using multiattribute method are shown in Table 2.

Table 2

Multiattribute method evaluation results

Requirement group\ Server	***Large enterprise application requirements***	***Large enterprise application development with requirements for additional functionality***	***Large enterprise application development with requirements for performance and fault tolerance***	***Small and medium enterprise application requirements***	***Specific cases***
Borland	*0.7899*	*0.7654*	*0.7660*	*0.7435*	*0.5704*
Orion	*0.6861*	*0.5965*	*0.6520*	*0.7430*	*0.4600*
Oracle	*0.8760*	*0.8972*	*0.8773*	*0.7449*	*0.9007*
WebLogic	*0.8683*	*0.8017*	*0.8918*	*0.7601*	*0.6904*
WebSphere	*0.8410*	*0.8229*	*0.8485*	*0.6790*	*0.8919*
JBoss	*0.4997*	*0.4138*	*0.3896*	*0.6697*	*0.3596*
JRun	*0.7423*	*0.6770*	*0.7298*	*0.8143*	*0.6410*
Windows	*0.8656*	*0.8868*	*0.8793*	*0.7414*	*0.8575*

Analytic Hierachy Process method evaluation results are shown in Table 3.

Table 3

AHP method evaluation results

	Benefit normalized value	***Cost, $***	***Cost normalized value***	***Benefit to cost ratio***
Borland	*0.20167165*	*12000*	*0.150943*	*1.336075*
Oracle	*0.20183257*	*20000*	*0.251572*	*0.802284*
Weblogic	*0.23238141*	*17000*	*0.213836*	*1.086725*
WebSphere	*0.23223661*	*30000*	*0.377358*	*0.615427*
JBoss	*0.13197683*	*500*	*0.006289*	*20.98432*

5. CONCLUSION

Application server selection analysis has led to the following conclusions:

- described methods can be useful for IT companies, to reduce expenses of application server selection and usage;
- developed estimation methods are flexible, so criteria list can be extended accordingly application servers evolution.

Comparison of AHP with multiattribute method has led to the following conclusion:

- AHP uses more natural approach (quantitative criteria compared directly), 0.0 – 1.0 scale not required;
- AHP is more flexible in work with criteria hierarchy (supports multiple level hierarchy);
- AHP becomes much complex if multiple criteria used on the same level.

REFERENCES

Brebner P., S. Chen, I. Gorton, Gosper J, L. Hu, A. Liu, D. Palmer and S. Ran (2002). *Evaluating J2EE Application Servers*, 105 p., CSIRO Middleware Technology Evaluation Series.

Graudins, J (2004). Comparing analysis of Java application servers, *Scientific proceedings of Riga Technical University*, **vol. 5**, p. 118 – 125.

Graudins J., L. Zaitseva (2005a). Application Server Selection for Definite Systems' Class, *Proceedings of the 19th International Conference on Systems for Automation of Engineering and Research*, p. 172 – 176.

Graudins J., L. Zaitseva (2005b). Application Server Evaluation Method, *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech'2005)*, p. IIIB.6-1 – IIIB.6-6

Gupta, R. H.. Explore the World of Application Servers. [Online document], Java-Boutique, Available at HTTP: http://javaboutique.internet.com/articles/AppServers/index.html

Keeney R. L., H. Raiffa (1976). *Decisions with Multiple Objectives: Preferences and Value Trade-offs*, 236. New York: Wiley.

Saaty T. (1996) *Multicriteria Decision Making. The Analytic Hierarchy Process*, 479 p., RWS Publications.

Why ESB and SOA (2006). [Online document], Available at HTTP: http://labs.jboss.com/file-access/default/members/jbossesb/freezone/resources/whitepapers/WhyESB.pdf