IMPROVING THE NETWORK TRANSMISSION COST OF DIFFERENTIATED WEB SERVICES

IRFAN AWAN[1] and MUHAMMAD YOUNAS[2]

[1]*Department of Computing, University of Bradford, Bradford, UK*
i.awan@scm.brad.ac.uk
[2]*Department of Computing, Oxford Brookes University, Oxford, UK*
m.younas@ieee.org

Abstract: This paper investigates into the transmission cost of web services related messages which is affected by network latency. Web services enable seamless interaction and integration of ebusiness applications. Web services contain a collection of operations so as to interact with outside world over the Internet through XML messaging. Though XML effectively describe message related information and is fairly human readable, it badly affects the performance of Web services in terms of transmission cost, processing cost, and so on. This paper aims to minimize network latency of message communication of Web services by employing pre-emptive resume scheduling. Fundamental principle of this approach is the provision of preferential treatment to some messages as compared to others. This approach assigns different priorities to distinct classes of messages given the fact that some messages may tolerate longer delays than others. For instance, shorter messages may be given higher priority than longer messages, or the Web service provider may give higher priority to the messages of paying subscribers.

Keywords: web services, xml messages, preemptive scheduling, performance.

1. INTRODUCTION

Web services are built on standard protocols and technologies such as HTTP, XML (Extensbile Markup Language), SOAP (Simple Object Access Protocol), WSDL (Web Service Description Langauge), and UDDI (Universal Description, Discovery and Integration) [4, 5, 6]. Web services are employed in various applications such as booking a flight, weather forecast, or buying books. Companies are increasingly deploying Web services to meet their business requirements. For instance, a Danish bank uses Web services to integrate its diverse systems into a single software infrastructure [13] — which includes the interaction and integration of the constituent systems which are used to provide different financial services to customers and business, selling mortgages, insurance, and pension plans.

Though Web services greatly facilitate the interaction and integration of heterogeneous Web-based systems, their performance is impoverished. This is mainly due to the fact that Web services mainly rely on XML-based SOAP messages. XML provides detailed description of SOAP messages. However, the design of such messages creates serious performance issues for Web services such as network transmission and processing costs. This paper investigates into the transmission cost of Web services related XML messages. The aim is to minimize the network latency of message communication of Web services by employing the priority scheduling mechanism [7, 9, 8]. Fundamental principle of this approach is the provision of preferential treatment to some messages as compared to others. We adopt pre-emptive resume scheduling mechanism which assigns different priorities to distinct classes of messages given the fact that some messages may tolerate longer delays than others. For example, shorter messages may be given higher priority than longer messages, or the Web service provider may give higher priority to the messages of paying subscribers.

The remainder of this paper is structured as follows. Section 2 provides background information on the related technologies of Web services and the network mechanisms. Section 3 explores the performance issue of Web services through the analysis of the related work. Section 4 presents the proposed approach. Section 5 illustrates experimental results. Section 6 concludes the paper.

2. BACKGROUND

In this section we provide the basic definitions and concepts used in the remainder of this paper. First, we provide a brief overview of web services. Second, we illustrate the principle of maximum entropy and a generalised distribution.

2.1 Web Services

The major technologies and protocols on which Web services are built include HTTP, XML, SOAP, WSDL, and UDDI [5, 6]. These technologies and protocols are organized into different layers of network, messaging, service description, service publication and service discovery. Figure 1 represents a generalised architecture of Web

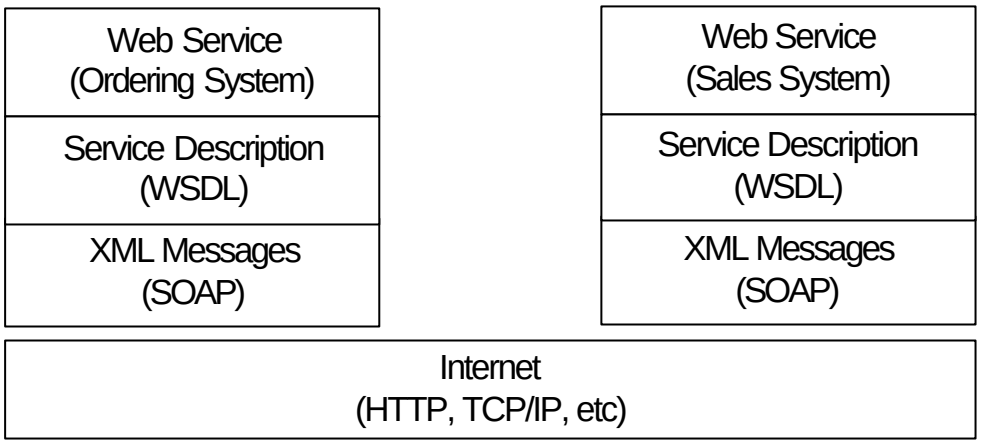

Figure 1. Web Services Architecture

services; showing communication between *sales* and *ordering* systems of a business application.

The lowest layer of Web services is the network. Web services that are publicly available on the Internet use commonly deployed network protocols such as TCP/IP, HTTP, FTP, and IIOP. HTTP is commonly used protocol in Web services.

In Web services, messages are communicated between participating systems (e.g., sales and ordering systems) using XML-based SOAP protocol. SOAP provides enveloping mechanism so as to communicate document-based messages. A SOAP message is an XML document. It comprises different parts such as SOAP envelope, a SOAP header, and a SOAP body. SOAP envelope is the top element of the XML document, which represents the message. Header is used to add features to a SOAP message. Features are added to the message in a decentralized manner without prior agreement between the participating systems concerning the message. SOAP body is a container for the information which is sent from the sender to the receiver of the message. SOAP messages are extensible thus they can be customised according to the application needs.

Figure 2 presents a simple SOAP message of a Web service. This example is adapted from a Shopping Basket Web Service (http://www.myweb-services.com/server/) which is listed at XMETHODS web site http://www.xmethods.net/. XMETHODS publicly lists the available Web services such as shopping baskets, weather, mobile SMS and so on. The SOAP message in Figure 2 represents the information which is to be sent to the Web service (http://www.myweb-services.com/server/) so as to modify required items in a shopping basket using the Web. This message represents the customer ID and its data type <CustomerGUID xsi:type="xs:string"></CustomerGUID>, the session related information <SessionID xsi:type="xs:string"></SessionID>, the item ID <ItemID xsi:type="xs:string"></ItemID> and the quantity of the items <ItemQtity xsi:type="xs:int"></ItemQtity>.

WSDL facilitates the process of service description. Each service provider (e.g., sales system) uses WSDL in order to define the details of the services it provides. Through WSDL services are defined as collections of network endpoints, or ports [6]. In order to define services WSDL document uses different elements such as *types* (used for data type definition); *message* (typed definition of the data); *operation* (describes an action which is supported by the respective service); *binding* (defines a protocol and data format specification for a particular port type); *port* (specifies an address for a binding); *service* (aggregates a set of related ports).

UDDI (such as XMETHODS) is used by the service requester (e.g., ordering system) and service provider (e.g., sales system) in order to publish and search for services. UDDI uses WSDL documents to publish details of the services and also facilitates the searching of services. UDDI is considered as yellow pages.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:n="http://www.myweb-services.com/xsd/"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Body soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <n:ModifySB>
      <CustomerGUID xsi:type="xs:string"></CustomerGUID>
      <SessionID xsi:type="xs:string"></SessionID>
      <ItemID xsi:type="xs:string"></ItemID>
      <ItemQtity xsi:type="xs:int"></ItemQtity>
    </n:ModifySB>
  </soap:Body>
</soap:Envelope>
```

Figure 2: XML-based SOAP Message

2.2 The Principle of ME (Maximum Entropy)

The principle of ME [14,15] provides a self-consistent method of inference for characterising an unknown but true probability distribution, subject to known (or known to exist) mean value constraints. The ME solution can be expressed in terms of a normalising constant and a product of Lagrangian coefficients corresponding to the constraints. In an information theoretic context [14], the ME solution corresponds to the maximum disorder of system states and, thus, is considered to be the least biased distribution estimate of all solutions that satisfy the system's constraints. In sampling terms, it has been shown [15] that, given the imposed constraints, the ME solution can be experimentally realised in overwhelmingly more ways than any other distribution. More details on entropy maximisation and its applications can be found in [16].

2.3 The GE (Generalised Exponential) Distribution

The GE distribution is an interevent-time distribution of the form

$$F(t) = P(W \le t) = 1 - te^{-st}, t \ge 0,$$
$$t = 2/(C^2 + 1),$$
$$s = tn,$$

where W is a mixed-type random variable (rv) of the interevent-time, whilst $(1/\nu, C^2)$ are the mean and Squared Coefficient of Variation (SCV) of rv W. The GE distribution is versatile, possessing pseudo-memory less properties which makes the solution of many GE-type queueing systems analytically tractable.

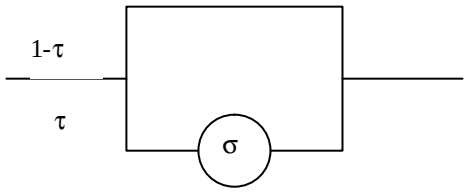

Figure 3: The GE distribution

3. RELATED WORK

Web services platform contains a collection of operations to enable its interaction between different systems through XML-based SOAP messaging. XML messaging severely affects the performance of Web services in different ways. These include: transmission cost, processing cost, message parsing, marshalling/un-marshalling process, and so on [1, 2, 3, 4]. For instance, the longer the XML messages the longer the transmission time over the network. B. Domanski [1] identifies that a typical XML message takes 10 to 20 times longer than the binary representation of the same message. Further, processing time of XML messages also contribute to the lower performance of Web services. XML messages are processed in different phases such as parsing, schema validation, binding, and transformation [1, 2].

Existing research proposes different strategies in order to improve the performance of Web services. G. Chafle et al [12] propose a decentralisation approach in order to efficiently orchestrate composite web services. Composite web service is built using different component web services. For instance, component web services such as flight booking, hotel reservation, and car rental can be orchestrated into a composite web service. The proposed decentralisation approach improves performance by increasing the throughput and reducing the response time. However, it does not consider the underlying network aspects and the prioritisation of web services. Moreover, [1, 2, 3] propose reduction in the size of XML messages, optimizing XML validation, monitoring application availability and performance.

Further, Yahia et al [4] propose a new Web-Services architecture in order to improve the performance of large scale XML data exchange. This new architecture is based on extending the existing WSDL of Web services. It fragments XML documents both at the service requesting (source) and service providing (target) systems. The source can specify XML document fragment that it is willing to produce while the target can specify XML document fragments that it is willing to consume. Such fragmentation minimizes unnecessary operations in the XML parsing, validation, etc. This results in the optimization of the XML data exchange process. This approach relieves target system from undoing the work that source system did to assemble XML documents in order to map them into its own data structure (i.e. to reduce un-marshalling operations). This approach significantly improves the performance of XML data exchange in web services.

The above approaches contribute to the performance optimization of Web services. However, they give no attention to the network latency and its affect on the performance of Web services. Current systems treat all the XML messages equivalently. Neither the network nor the end systems typically prioritize traffic for XML messages. However, there are cases where multiple level of service would be desirable. Not all Web services are equally important to the service requestor or to the service provider, and some services may be treated differently. For example, a Web service provider may want to give priority to the users based on their subscription status such as paying and non-paying subscribers. Thus priority should be given to the paying subscriber so as to retain them. Another example is to priorities shorter XML messages over longer XML messages so as to enable efficient data communication.

The next presents a new approach of differentiated scheduling that treats different XML messages differently depending on the size of request. The aim is to improve the performance of Web services by reducing the network latency associated with XML messages. To the best of our knowledge, this approach is the first effort towards performance optimization of Web services using differentiated scheduling mechanism.

4. THE PROPOSED APPROACH

This section presents the proposed approach of differentiated scheduling in Web services. The proposed approach aims to improve the performance of Web services which are characterised by distinct types of messages:

- A Web service provider may want to give high priority to the paying subscribers. For instance, a SOAP message (as in Fig.2) will be served faster if it belongs to a paying subscriber.

- A Web service provider may prioritise shorter XML messages over longer XML messages so as to enable efficient data communication. For example, large organisations such as AT&T use large volume of data to support daily operations. According to [4], the AT&T usage data from telephony network exceeds 60GB per day.

Within context of the above messages, it is envisaged that message delay in Web services can significantly be reduced thereby improving the performance of Web services. In order to reduce message queuing delays we take into account the priority scheduling mechanisms of active networks. Fundamental principle of these mechanisms is the provision of preferential treatment to some messages as compared to others. These mechanisms assign different priorities to distinct classes of messages in order to determine the order of service among them. These mechanisms are based on the fact that some messages may tolerate longer delays than others.

One of the useful priority scheduling mechanisms is the preemptive resume scheduling (PR) [9, 8, 10]. In the proposed approach, PR is employed at each network node involved in the processing of SOAP messages of Web services. According to PR, the arriving high priority message preempts the low priority message being processed. The preempted message resumes its processing soon after the high priority message is processed. In PR mechanism each node in the network is equipped with a finite capacity buffer that stores the incoming messages. The total time that a message spends in the node is the sum of the waiting time and the processing time.

Waiting time for each message is the sum of processing times for all the messages in front of it.

Employment of PR reduces the queuing delays at the network nodes involved in the processing of Web services. In order to calculate the queuing delay each network node is modelled as a queuing system with finite capacity. The arriving external traffic at each node is bursty as messages from various Web services can arrive simultaneously. This has been modelled using Compound Poisson Process (CPP). Each node may have multiple processors and hence can execute various messages simultaneously. This concurrent execution has been modelled using a generalised exponential (GE) distribution. Based on such information, each node has been analysed as a GE/GE/1/N queuing system with preemptive resume scheduling discipline to give preferential treatment to arriving messages. This analytical solution provides closed form expression to calculate the queuing delay at each network node.

Analysis of a GE/GE/1/N/PR Queue

Consider a stable single server GE/GE/1/N queue under a priority preemptive resume scheduling discipline and R (>1) multiple classes of messages. For each class, i (i=1,2,…,R), let λ_i be the mean arrival rate, C_{ai}^2 be the inter-arrival time squared coefficient of variation (SCV), μ_i be the mean service rate and C_{si}^2 be the service time SCV. Let at any given time, n_i ($0 \le n_i \le N$), $\sum_{i=1}^{R} n_i \le N$, be the number of class i messages in the queue (waiting and/or receiving service), $S=(n_1,n_2,\ldots,n_R)$ be a joint queue state and T be the set of all feasible states S. The form of the state probability distribution $P(S)$, $\{S \in T\}$ of a GE/GE/1/N/PR priority queue, can be characterized by maximizing the entropy functional,

$$H(P) = -\sum_{S \in T} P(S) \log P(S)$$

This is subject to prior information expressed in terms of the normalization and, for each class i (i=1,2,…,R), the marginal constraints of server utilization, U_i ($0<U_i< 1$), busy server probability θ_i ($0 < \theta_i < 1$) with $n_i>0$, mean queue length, L_i ($U_i \le L_i < N$) and conditional full buffer state probability, given that a class i message is in service, ϕ_i ($0 < \phi_i < 1$), satisfying the flow balance equations, namely

$$l_i(1-p_i) = m_i U_i, i = 1,2,\ldots,R,$$

where π_i is the marginal blocking probability that an arriving message of class i finds N messages in the queue. By employing Lagrange's method of undetermined multipliers and after some manipulation, the probability distribution of messages can be expressed by

$$P(S) = \frac{1}{Z} g_i x_i x_i^{n_i} y_i^{f_i(S)} \left(\prod_{j=i+1}^{R} x_j^{n_j} x_j^{h_j(S)} \right), i = 1,\ldots, R,$$

where Z is the normalizing constant, $\{g_i,\xi_i,x_i, y_i\}$ are the Lagrangian coefficients corresponding to constraints $\{U_i, \theta_i, L_i,\phi_i\}$, respectively and $\{h_i(S),f_i(S)\}$ are suitably defined auxiliary functions [9]. Utilizing this product-form solution, the closed-form expressions for basic performance metrics such as mean marginal and aggregate delays, Q_i and Q, respectively, can be obtained (c.f., [9]). In particular, the mean delays can be clearly determined (via Little's Law) by

$$Q_i = \frac{L_i}{\hat{l}_i}$$

where $\hat{l} = (1-p_i)$ is the mean effective arrival rate of class i messages and

$$Q = \sum_{i=1}^{R} \frac{\hat{l}_i}{\hat{l}} Q_i, \qquad \hat{l} = \sum_{i=1}^{R} \hat{l}_i.$$

5. EXPERIMENTAL RESULTS

Various experiments have been conducted based on analytical model for the PR and FCFS [11] service disciplines. We investigate two data services with different average sizes such as 62.5 KByte and 1MByte. These data services represent, for example, two typical XML documents [4]. We calculate the mean response time for both services treated under PR scheduling mechanism as well as under fair sharing scheme (FCFS). Results show that mean response time for high priority XML documents increases more rapidly by increasing the traffic load under fair sharing scheme as compared to PR mechanism (c.f., Fig. 4).

In the case of low priority documents the results are more interesting. Mean response time for low priority documents show better results when treated under fair share mechanism as compared to PR mechanism (c.f., Fig. 5). This is because the low priority documents are pre-empted by the high priority documents under PR mechanism which results in more queuing time.

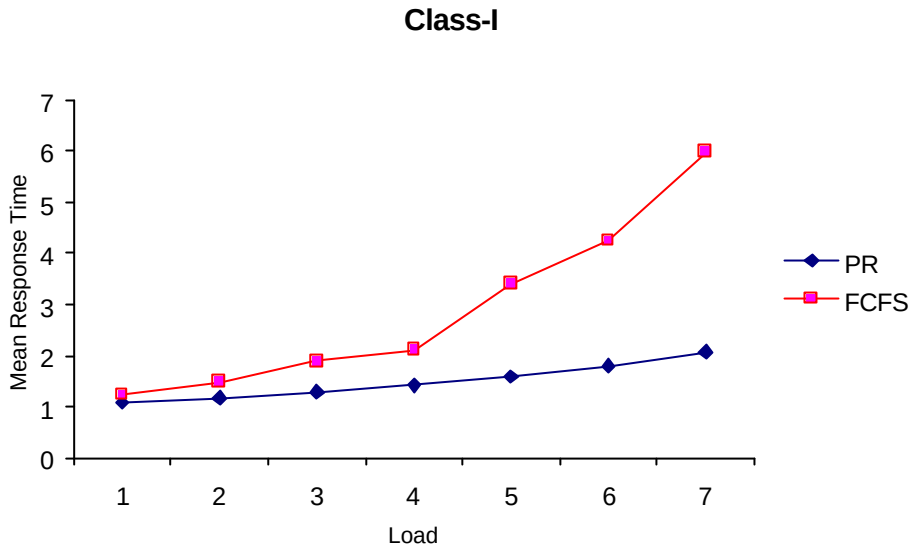

Figure 4. Mean response time against traffic load for class-I

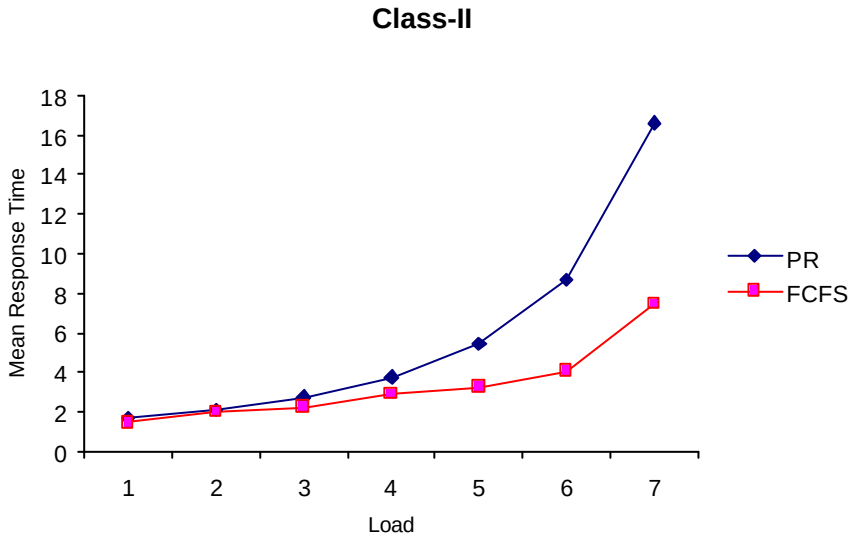

Figure 5. Mean response time against traffic load for class-II

6. CONCLUSION

We investigated into the performance aspects of Web services. Our investigation is motivated by the fact that current research pays little attention to the performance aspects of Web services. We presented a new approach which applies networking techniques to the Web services in order to improve their performance. Our investigation was mainly concerned with the optimisation of the transmission cost of XML documents in Web services. Web services exchange information in XML document format using SOAP protocol. XML provides detailed description of SOAP messages. However, the size and design of such messages creates serious performance issues for Web services such as network transmission and processing costs.

Our approach applies PR mechanism in order to minimize the network latency of XML documents in Web services. PR assigns different priorities to distinct classes of messages given the fact that some messages may tolerate longer delays than others. The proposed PR approach is tested through various experiments using analytical model. We consider two types of XML documents with varying sizes. We computed their mean transmission cost using the proposed PR mechanism and also FCFS mechanism. Experimental results show that mean response time for high priority XML documents increases more rapidly by increasing the traffic load under fair sharing scheme as compared to PR mechanism. In the case of low priority documents mean response time for low priority documents show better results when treated under fair share mechanism as compared to PR mechanism

References

1. B. Domanski: An Introduction to Web Services and Performance Issues. Z Journal December 2003 http://www.zjournal.com/issue.asp?IssueId=52
2. H. Adams: Web services performance considerations, Part 1. 17 February 2004 http://www-106.ibm.com/developerworks/library/ws-best9/
3. H. Adams: Web services performance considerations, Part 2. 2 March 2004 http://www-106.ibm.com/developerworks/webservices/library/ws-best10/
4. S.A-Yahia, Y. Kotidis: A Web-Services Architecture for Efficient XML Data Exchange. ICDE 2003
5. H. Kreger: Web Services Conceptual Architecture (WSCA 1.0). May 2001.
6. Web Services Activity. http://www.w3.org/2002/ws/
7. I. Awan, M. Younas: Analytical Modelling of Priority Commit Protocol for Reliable Web Applications. 19th ACM Symposium on Applied Computing (SAC04), March 2004, Nicosia, Cyprus
8. I.Awan, D. Kouvatsos: Approximate Analysis of Arbitrary QNMs with Space and Service Priorities. Performance Analysis of ATM Networks, Kluwer Academic Publishers, 1999, pp. 497-521
9. I.Awan, D. Kouvatsos: Approximate Analysis of Arbitrary QNMs with HoL Priorities, CBS Buffer Management Scheme and RS-RD Blocking. Proceeding of 18th UKPEW, Glasgow, UK, July 2002, pp. 15-26
10. A.C.Williams and R.A.Bhandiwad: A Generating Function Approach to Queueing Network Analysis of Multiprogrammed Computers. Networks Vol. 6, pp. 1-22, 1976.
11. S.G.Denazis: Queueing Network Model with Blocking and Multiple Job Classes. Ph.D. Thesis, University of Bradford, 1993.
12. G.Chafle, S.Chandra, V.Mann, M.G. Nanda: Decentralised Orchestration of Composite Web Services. Proceedings of the 13th International Conference on World Wide Web, (WWW 2004), ACM, New York, NY, USA, May 17-20, 2004, pp, 134-143
13. Danske Bank: Microsoft Web Services Case Studies. http://msdn.microsoft.com/webservices/understanding/casestudies/default.aspx
14. E.T.Jaynes, Information Theory and Statistical Mechanics, Phys. Rev, 106, (1957), pp. 620-630.
15. E.T.Jaynes, Information Theory and Statistical Mechanics, II Phys. Rev} 108, (1957), pp. 171-190.
16. D.D.Kouvatsos, Entropy Maximisation and Queueing Network Models, Annals of Operation Research, 48, (1994), pp. 63-126.

.

Biographies

Irfan Awan received his PhD degree ('97) in Performance Analysis of Queueing Network Models with priorities and blocking from the University of Bradford – UK. He is a senior lecturer in the Department of Computing, University of Bradford which he joned in 1999. He is member of the Network and Performance Engineering Research Group and co-tutor for the MSc Mobile Computing Course. Dr. Awan's research has mainly focussed on developing cost effective analytical models for measuring the performance of complex queueing networks with finite capacities and priorities. He has produced over 70 publications and edited proceedings of the 20th UK Performance Engneering Workshop. He has also authored several special issues of the international journals and is a member of various programme committees and steering committees for International conferences. He is a member of ILT and BCS.

Muhammad Younas is a Senior Lecturer in Computer Science in the Department of Computing, Oxford Brookes University, Oxford, UK. He received his PhD degree in Computer Science from the University of Sheffield, UK, in 2001. His research interests include Web and database technologies, transaction processing, agent technology, and mobile computing. He has published more than 40 research papers in international journals and conferences. He has also edited three books. He is the guest editor for various international journals. He has been involved in the organizing and program committees on a number of international conferences and workshops.
.