Association for Information Systems
AIS Electronic Library (AISeL)

ICEB 2003 Proceedings

International Conference on Electronic Business (ICEB)

Winter 12-9-2003

# Efficient Prediction of Quality of Service for Data Mining Web Services

Shonali Krishnaswamy

Seng Wai Loke

Arkady Zaslavsky

Follow this and additional works at: https://aisel.aisnet.org/iceb2003

This material is brought to you by the International Conference on Electronic Business (ICEB) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICEB 2003 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

# Efficient Prediction of Quality of Service for Data Mining Web Services

Shonali Krishnaswamy, Seng Wai Loke and Arkady Zaslavsky
School of Computer Science and Software Engineering
Monash University
Australia
{Shonali.Krishnaswamy, Seng.Loke, Arkady.Zaslasvky}@infotech.monash.edu.au

## Abstract

Apriori estimation of quality of service (QoS) levels is a significant issue in web services since Service Level Agreements (SLAs) need to specify and adhere to such estimates. In this paper we present novel cost formulae for estimating the end-to-end response time for distributed data mining (DDM) web services.

## 1. Introduction

Umesh Dayal [2] predicted that "…data analysis and mining functions themselves will be offered as business intelligence e-services that accept operational data from clients and return models or rules". The growing number of Web Service Providers (WSPs) like digiMine™ (http://www.digimine.com) and Information Discovery™ (http://www.datamine.aa.psiweb.com) who offer commercial data mining web services are testimony to this statement [8, 11, 4]. The increasing focus on data mining web services can be attributed to the recognition of data mining as an important technology in aiding strategic decision making coupled with the commercial viability of the WSP paradigm. The option of Internet-delivery of data mining services is emerging as attractive for small to medium range organisations, which are the most constrained by the high cost of data mining software, and consequently, stand to benefit by paying for software usage without having to incur the costs associated with buying, setting-up and training. While the primary focus of the commercial arena has been the delivery of data mining as an web service, there is an emerging research focus on providing data mining models as services on the Internet [14]. Thus rather than the hosting of a data mining system and delivering the results as a service, the aim is to be able to sell data mining models, which can be bought, for example, by start-up organisations operating in a given vertical domain.

In general, the operational cornerstone for WSPs is the contractual agreement between the client and the service provider, known as the Service Level Agreement (SLA). The SLA is a legally binding document and specifies the contractual obligations of the WSP with respect to the guaranteed level of service and the penalties associated with failure to comply with the contract. In the specific context of WSPs, which are governed by SLAs, QoS metrics have a direct bearing on the success in monetary terms. Therefore, the ability to accurately predict the level of service that can be guaranteed is of immense value to WSPs. It is this context that motivates [13] to state quality of service has "…a lot to do with a cost-benefit analysis and prediction for SLAs". The need for such predictive estimates of the quality of service that can be ensured is also clearly indicated by qualitative studies presented in [1] which surveyed user behaviour and the criteria users applied to assessing network services and pricing. One of the outcomes of this study was that "…users value predictive feedback over feedback concerned with current statistics". While predicting the quality of service a priori is an important requirement for WSPs as discussed in [1, 13], it has not been addressed by the web services community. This can be attributed to several reasons. The focus has been on system level metrics that are the key to load balancing and resource utilisation and not on application centric metrics such as the waiting time, probability of successful completion and end-to-end application response time [16, 19]. It is difficult to formalise the semantics of application centric metrics since characteristics vary from application to application. Prediction is challenging in a dynamic environment such as the Internet, where WSPs operate. This challenge is considerably increased by the consequences and pitfalls of inaccurate estimates, because of the direct relationship between prediction accuracy and revenue/loss.

In summary, WSPs require the development of techniques to estimate the quality of service that can be ensured by the service provider. This need is driven by SLAs, which have a direct bearing on the revenue of WSPs and is increasingly being stated as an issue that must be addressed [1, 13]. In this paper we present techniques for estimating the response time of data mining application services.

## 2. Response Time of Distributed Data Mining

In this section, we present techniques for estimating the end-to-end response time in data mining web services. At a conceptual level, the response times for different scenarios is as illustrated in the example scenario illustrated in figure 1. The client has two data sources ("Data 1" and "Data 2") and two computational resources ("Server 1" and "Server 3") that it can make available for mining. One of the datasets (Data 1) is located on Server 2 that is not available for mining. The service provider has three high-performance servers. The three servers on the service provider's side may be geographically distributed, thereby making the

communication costs variable.

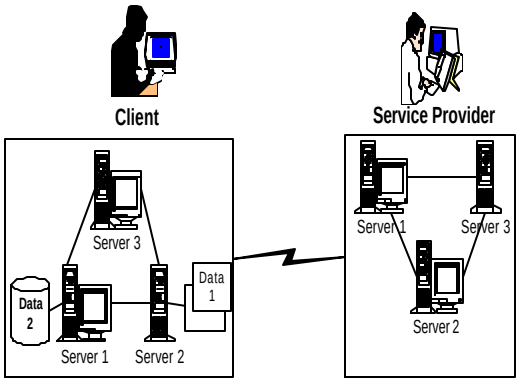


**Figure 1 Data Mining Web services**

There are three possible options that the client can choose:

1. The mining should be done locally using only the client's computational resources. In this case a mobile agent based approach will have to be employed in order to perform the task at the client's site.
2. The mining should be done remotely using only the service provider's computational resources. In this case, a client-server approach will have to be used and the data has to be transferred to the service provider's servers.
3. The client has no preference for the location. In this case, a combination approach of mixing the client-server and the mobile agent models can be used.

The estimation requires the identification of the cost components and formalisation of a cost model for response time in the context of distributed data mining (DDM) web services. We now formalise the cost components of the DDM response time and present estimation techniques for these cost components.

## 2.1 Cost Components of the DDM Response Time

In this section we specify the different cost components of the response time in distributed data mining web services. The response time of a task in a distributed data mining web service broadly consists of three components: *communication, computation* and *knowledge integration.*

*Communication:* The communication time is largely dependent on the operational model. It varies depending on whether the task is performed using a client-server approach or using mobile agents. In the client-server model the communication time is principally the time taken to transfer data from distributed servers to a high performance machine where the data mining is performed. In the mobile agent model, the communication time revolves around the time taken to transfer mobile agents carrying data mining software to remote datasets and the time taken to transfer results from remote locations for integration.

*Computation:* This is the time taken to perform data mining on the data sets and is a core factor irrespective of the operational model.

*Knowledge Integration:* This is the time taken to integrate the results from the distributed datasets.

The response time for distributed data mining is as follows:

$$T = t_{dm} + t_{com} + t_{ki} \quad (1)$$

In Eq. 1 above, T is the response time, $t_{dm}$ is the time taken to perform data mining, $t_{com}$ is the time involved in communication and $t_{ki}$ is the time taken to perform knowledge integration. The modelling and estimation of the knowledge integration ($t_{ki}$) variable is dependent on the size and contents of the results obtained from the distributed datasets. Given that the primary objective of data mining is to discover hitherto unknown patterns in the data [3], we consider the a priori estimation of the time taken to perform knowledge integration to be outside the scope of this paper (since knowledge integration depends on the characteristics of the results of the data mining process). Having identified the cost components of the DDM response time, we now formalise the overall estimation cost for different models and scenarios.

## 2.2 Cost Matrix for Representing the Composite Response Time

We now present a cost matrix for computing the composite DDM response time estimates for different strategies. The response time for distributed data mining as presented in Eq. 1 consists of three components including communication (due to either transfer of mobile agents and/or transfer of data), computation (performing data mining) and knowledge integration. The following discussion focuses on the communication and computation components and does not consider the knowledge integration component. The strategy is to compute estimates for the individual cost components and then uses the estimates to determine the overall response times for different strategies. The cost matrix to calculate the composite response time for different DDM strategies is denoted by *CM* and is represented as a two-dimensional *mxn* matrix, where *m* is the number of available servers and *n* is the number of datasets. A fundamental feature of the cost matrix that makes it applicable for both the mobile agent and client-server models is that we incorporate location information on the datasets and servers.

The elements of the cost matrix represent the estimated response time and are defined as follows:

1. Let *m* be the number of servers.
2. Let $S = \{S_1, S_2, \ldots, S_m\}$ be the set of servers. A server $S_j$ can either be located at the service provider's site or at the client's site. Therefore, let $S^{SP}$ be the set of servers located at the service providers site and let $S^C$ be the

set of servers located at the client's site. The following properties are true for the sets $S$, $S^{SP}$, $S^{C}$.

- $S^{SP} \cup S^{C} = S$; the set of servers available at the client's site and the set of servers available at the service provider's site summarily constitute the total set of available servers. The obvious corollaries are $S^{SP} \subseteq S$ and $S^{C} \subseteq S$.
- $S^{SP} \cap S^{C} = f$ ; thus a server either belongs to the client or the service provider. It cannot belong to both.
- $S^{SP} = f$ is valid and indicates that the client is not willing to ship the data across and $S^{C} = f$ is also valid and indicates that the client's computational resources are unavailable or inadequate.

In order to specify the location of a server we use the following notation: $S_j \in S^{SP}$ and $S_l \in S^{C}$ where $l, j = 1, 2, \ldots, m$. This distinction is necessary for specification of how the response time has to be estimated in the cost matrix.

3. Let $n$ be the number of datasets and let $DS = \{ds(1), ds(2), \ldots, ds(n)\}$ represent the labelling of the datasets.
4. Let $ds(i)_{S_j}$ represent the location of a dataset labelled $ds(i)$, $i=1, 2, \ldots, n$ at server $S_j$, where $ds(i)$ Î $DS$ and $j= 1, 2, \ldots, m$. Thus datasets are uniquely identified and multiple datasets at locations can be represented.

Let $cm_{ij} \in CM$ be the estimated response time for taking a dataset located at the server $j$ and mining it at the server $i$, where $1 \le i \le m$ and $1 \le j \le n$. The value of $cm_{ij}$ is computed as follows:

$$cm_{ij} = \begin{cases} MA_{S_X \to S_i} + TR^{ds(k)_{S_j}}_{S_j \to S_i} + W_{S_i} + DM^{ds(k)_{S_j}}_{S_i}, i \neq j, S_X \in S^{SP}, S_j \in S^{C}, S_i \in S^{C} \\ MA_{S_X \to S_i} + W_{S_i} + DM^{ds(k)_{S_j}}_{S_i}, i = j, S_X \in S^{SP}, S_j \in S^{C}, S_i \in S^{C} \\ TR^{ds(k)_{S_j}}_{S_j \to S_i} + W_{S_j} + DM^{ds(k)_{S_j}}_{S_i}, i \neq j, S_i \in S^{C}, S_i \in S^{SP} \end{cases}$$

In the above equation:

- $MA_{S_X \to S_i}$ is the time to transfer a mobile data mining agent from server $S_X$ (which is a server of the service provider) to server $S_i$
- $TR^{ds(k)_{S_j}}_{S_j \to S_i}$ is the time to transfer a dataset $ds(k)_{S_j}$ located at server $S_j$ to server $S_i$
- $DM^{ds(k)_{S_j}}_{S_i}$ is the time to mine dataset $ds(k)_{S_j}$ located originally at server $S_j$ at server $S_i$
- $W_{S_i}$ is the wait time at server $S_i$ required for the completion of previous tasks and is generally more significant when $S_i \in S^{SP}$ (i.e. server $S_i$ is located at the service provider's site).

As presented above there are three formulae for estimating the response time for different scenarios in the cost matrix. The first one is for the case where the server $S_i$ where the mining is to be performed is at the client's site but does not contain the dataset $ds(k)_{S_j}$ (which as indicated is located at the server $S_j$). Hence there is a need to transfer the data from its original location $S_j$ to the server $S_i$ to perform the data mining. Further, the client's site would not have the data mining software and a mobile agent needs to be transferred from the service provider's site (represented as $S_X$). The second formula is for the case where the server where the mining is to be performed is at the client's site and the dataset is located on the same server (i.e. $i = j$). In this case, the mobile agent needs to be transferred but there is no need to transfer the data. The third formula is for the case where the server where the mining is to be performed is located at the service provider's site (i.e. $S_i \in S^{SP}$ ) and therefore the data has to be shipped across from the client's site to perform mining. The three formulae map to the three scenarios outlined earlier, namely, that of mining at the client's site, mining at the service provider's site and using both sites. The cost matrix has been designed to determine the response time for mining the datasets at the different available servers. The location of the servers and the datasets determine the cost formula that has to be applied for computing the response time. We now present cost formulae for estimating the individual cost components.

## 3. Estimating Individual Cost Components of Response Time

As discussed, we now present strategies for estimating the individual components of the DDM response time.

### 3.1 Estimating the Communication Cost

The communication cost in the DDM process varies depending on whether the client-server strategy is followed or the mobile agent model is used.

#### 3.1.1 Mobile Agent Model

In general, the mobile agent model for DDM involves dispatching mobile agents carrying the mining algorithms to the locations of the data to perform data mining. Thus, the model is characterised by a set of mobile agents traversing the relevant data servers to perform mining. This can be expressed as $m$ mobile agents traversing $n$ servers (that contain datasets). In the context of data mining web services, the mobile agent model is applied when the client specifically requires the task to be performed using the client's computational resources (e.g. where the client does not want the data to leave the site). This model can also be used in cases where the client has no preference, but applying this model results in better response time. Therefore it can be seen that the mobile agent option primarily involves the servers located at the client's site. In order to estimate the transfer times for mobile agents,

consider the following:

Let $N$ be the total number of servers at the clients site.

Let $n$ be the number of servers where the mining is performed.

Let $S^C$ represent the $n$ servers in question i.e. $S^C = \{S_1, S_2, \ldots, S_n\}$.

Let $m$ be the number of mobile agents dispatched from the service provider's central server *CES* to perform mining at the set of servers $S^C$.

Let $t_{ma}(x, y) = MA_{x \to y}$ refers to the time taken by the agent ma to travel from node x to node y.

In order to compute the estimates in the cost matrix, we need to estimate the transfer times of the mobile agents from *CES* to the servers of the set $S^C$ (i.e. we need to estimate $MA_{CES \to S^C}$).

Given that there are $n$ servers that have datasets for mining and $m$ agents, there are three possible alternatives within this scenario and they are:

1. $m = n$, where the number of mobile agents is equal to the number of servers. This implies that one data mining agent is sent to each server involved in the distributed data mining task.
2. $m < n$, where the number of mobile agents is less than the number of data servers. The implication of having fewer agents than servers is that some agents traverse more than one server. This option is not used often, but is modelled to allow for cases where there is an imposed ordering in the traversal. For example, a scenario where an agent must first visit server $r$ to obtain background knowledge necessary to perform data mining at server $s$. This knowledge can be in the form of a concept hierarchy [5] or a database schema that defines the attributes of the database located at server $s$. In such cases, it is necessary to have one agent traverse several servers.
3. $m > n$, which we do not explicitly consider since this is in effect equivalent to the case *1* above with respect to travel time where there is a mobile agent available per server. That is, having modelled the travel time for a single mobile agent from one server to another it is implicit that we can estimate the travel time for several agents that have to be sent between the same servers. A possible scenario for sending more than one agent to a server may be if there are several datasets that need to be mined at the server and different agents are required to process the various datasets.

Each of the above cases has a cost function and the cost models for estimating the response time. However, in this paper, we focus on the scenario of equal number of mobile agents and data servers, since this is the most common.

This case, where data mining from different distributed data servers is performed in parallel and there is one mobile agent per server (i.e. m=n). The algorithm used across the different data servers can be uniform or varied. The service provider dispatches a mobile agent encapsulating the data mining algorithm (with the relevant parameters) to each data server participating in the distributed data mining task. In order to derive the cost function for the general case involving $n$ data servers and $n$ data mining mobile agents (since $m=n$), we first formulate the cost function for the case where there is one data server and one data mining agent. Let us consider the case where data mining has to be performed at the $i^{th}$ server $S_i \in S^C, 1 \le i \le n$. The overall cost function for the response time to perform distributed data mining involving the $i^{th}$ data server is computed from the formulae presented in section 3.2.3 and consists of the time to transfer the mobile agent to the server, transfer the data to the server (if the data is not originally located at the server) and perform the mining subject to any Wait time imposed. Each of these components have to be estimated in order to derive the overall estimated response time. In this section we focus on estimating the communication time for the transfer of a single data mining mobile agent dmAgent from server *CES* to server $S_i$, that is $t_{dmAgent}(CES, S_i)$.

The time taken for a mobile agent to travel depends on the following factors: the size of the agent, the bandwidth between servers and the latency between the servers. The travel time is proportional to the size of the agent and is inversely proportional to the bandwidth (i.e. the time taken increases as the agent size increases and decreases as the bandwidth increases). The latency is the delay, typically expressed in Round Trip Time (RTT) and is added to obtain the transfer time. The latency has higher impact on the transfer time, when the amount of data that is transferred is small, whereas the bandwidth has a higher impact when the amount of data is large. This can be expressed as follows:

$$_{dmAgent}(CES, S_i) \propto \text{size(dmAgent)} \quad (2)$$

$$t_{dmAgent}(CES, S_i) \propto 1 / (\text{bandwidth between } CES, S_i) \quad (3)$$

From (2) and (3):

$$t_{dmAgent}(CES, S_i) = \lambda(CES, S_i) + \text{size(dmAgent)} / \beta(CES, S_i) \quad (4)$$

In the above Equation 4, $\lambda$ is the latency and $\beta$ is the bandwidth. While bandwidth and latency can be measured, we need to determine the size of the dmAgent. In [StS97] the size of an agent is given by the following triple:

size of an agent = < Agent State, Agent Code, Agent Data>

where, *Agent State* is the execution state of the agent, *Agent Code* is the program that is encapsulated within the agent that performs the agent's functionality and *Agent Data* is the data that the agent carries (either as a result of some computation performed at a remote location or the additional parameters that the agent code requires). On adapting the above representation to express the size of the data mining agent (dmAgent), we now have,

size(dmAgent) = <dmAgent state, data mining algorithm, input parameters>

We now extend the cost estimate for the general case characterised by $n$ mobile agents and $n$ distributed servers. Thus, $n$ mobile agents encapsulating the respective mining algorithms and parameters are dispatched concurrently. Mining is performed at each of the sites in parallel and the results are returned to the central server. Thus, the transfer time per server from Eq. 4 above, is $t_{dmAgent}(CES, S_j)$, $1 \le j \le$

*n*. The total transfer time is the time taken by the mobile agent that requires the longest individual transfer time. In case the agent is required to travel back to the server, the estimated travel time is computes as $t_{dmAgent}(S_j, CES)$, $1 \le j \le n$.

We have modelled the scenarios for using mobile agents to perform distributed data mining and have presented cost formulae to estimate the mobile agent transfer times for each case. This addresses the *a priori* estimation of the communication mobile agent transfer time component in the cost matrix. The next section focuses on estimating the data transfer time, which is the second communication component.

### 3.1.2 Estimating Data Transfer Time

The transfer of data in the hybrid DDM model can be attributed to two reasons. Firstly, the hybrid model integrates both the client-server and mobile agent models. In the client-server model the data is transferred from the client's site to the servers of the service provider. Secondly, data transfer occurs when the data resides on a server that is not available for mining (e.g. because lacks the necessary computational resources or it is dedicated for some other tasks) and has to be transferred to another server at the client's site. Typically, the transfer times in the first case would be more significant than the second where the transfer might occur within an organisation's intranet. In this section, we present the cost formulae for estimating data transfer times $TR_{S_j \to S_i}^{ds(k)_{S_j}}$ for transferring a dataset $ds(k)_{S_j}$ located at server $S_j$ to server $S_i$. Server $S_j$ is at the client's site and $S_i$ may be at either the client's site or the service provider's site.

The cost formulae for transferring datasets are primarily dependent on the size of the data, the bandwidth between the servers and the latency or delay. Let $ds(k)_{S_j}$ represent the dataset that is located $S_j \in S^C$ and has to be transferred to $S_i$ (where $S_i \in S^{SP}$ or $S_i \in S^C$). The data transfer time $t_{ds}(S_j, S_i) = TR_{S_j \to S_i}^{ds(k)_{S_j}}$ can be estimated as follows in equation 5:

$t_{ds}(S_j, S_i) = \lambda(S_j, S_i) + \text{size}(ds(k)) / \beta(S_j, S_i)$ In the above equation 5, $\lambda$ is the latency and $\beta$ is the bandwidth between the servers $S_j$ and $S_i$. Since typical dataset sizes tend to be very large, the effect of the bandwidth will be greater than the effect of the latency of the data transfer time.

We have modelled the cost formulae for the data transfer component of the DDM response time. The cost formulae presented are applicable to DDM systems irrespective of their architectural model and are unique in that, we propose to mathematically model the communication costs in distributed data mining. Further, the cost models developed also take into account the different cases and options within each model, thereby facilitating comparison of different sub options within a given model. There has not been a comparison of communication times between the client-server and mobile agent models with respect to distributed data mining. The cost formulae developed in this section support the explicit comparison of the communication time between the client-server and mobile agent models for distributed data mining. Further, while previous costing techniques in distributed data mining have implicitly considered the communication cost in terms of the number of datasets [PaS01] or have assigned the cost in terms of bandwidth [TuG00], our approach is based on taking into account several factors including bandwidth, latency, dataset sizes and mobile agent sizes. Finally, the cost formulae proposed and developed in this chapter allow *a priori* estimation of the communication components in the cost matrix. Having modelled the response time for the communication component, we now present our technique for estimating the response time of the computational component (i.e. the cost of performing data mining).

## 3.2 Estimating the Data Mining Cost

In order to estimate accurately the computation cost of data mining tasks we have proposed and developed a novel rough sets based algorithm for application run time estimation [6, 9, 10]. It must be noted that considerations of space do not allow us to present an explanation of this algorithm here, however we present experimental results of applying this technique for estimating the run times of data mining tasks.

# 4. Experimental Results and Analysis

The viability of using these cost estimates depends on the accuracy of the estimation techniques. Thus, the estimated communication times and predicted data mining task run times must be close to actual run time for these cost formulae to form the basis for effective optimisation.

We have proposed a model for a priori estimation of the DDM response time by estimating the individual components. These estimates are used in the cost matrix to represent the overall response time for different strategies. As discussed, communication is an important factor in the response time for distributed data mining. We developed cost formulae for estimating the transfer times for both the models for performing distributed data mining, namely, mobile agents and client-server. In the context of data mining web services, the mobile agent model maps to the case where the task is performed at the client's site and the client-server model maps to the case where the task is performed at the service provider's site. In this section, we present experimental results of our cost formulae for estimating the transfer time for both data and mobile agents.

## 4.1 Estimating Mobile Agent Transfer Time

In this section, we present the estimated and actual transfer times for the data mining mobile agents implemented in our DDM system Distributed Agent-based Mining Environment (DAME) [7]. The mobile agents were developed using the Aglets™ SDK [12] and are used to carry the data mining software to remote data servers. The agents are provided with an itinerary of destinations and respective tasks to be performed in each server that it visits. The agent carries the data mining software as a serialised object. We note that while we use Aglets for experimental and implementation purposes, our cost formulae for estimating the transfer times of mobile data mining agents are not specific to any toolkit. The experimental evaluation consisted of estimating the transfer times for mobile data mining agents and comparing it with the actual transfer times to determine the mean error. The experiments were conducted using four distributed machines with the three machines connected through high speed communication links (two machines via a 100 Mbits link and one machine via a 10 Mbits link) and one machine connected via a 28.8 Kbits modem link. Two machines were located on one campus of Monash University and one machine was located in another campus and was on a different domain. The size of the information carried by the mobile agent was different in each run by varying the software that it carried and the tasks that it was required to perform. The comparative difference in the estimated and the actual results obtained from the tests are shown in figure 2.

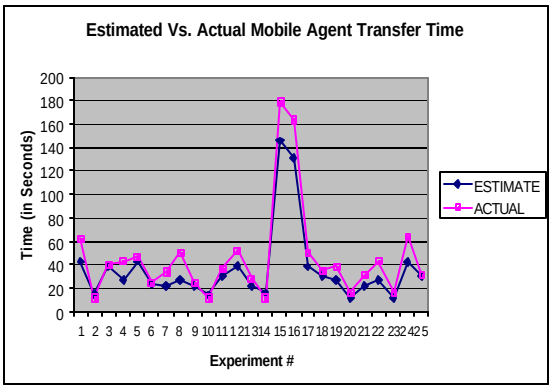


Figure 2 Estimation Accuracy for Mobile Agent Transfer Times

In summary the experimental scenario used mobile agents carrying algorithms and data varying from approximately 0.5 MB to 30 MB over connections where the highest bandwidth obtained was 7.3 Mbits/s and the lowest bandwidth was 24 Kbits. The latency was a factor only when the slow links were used. The mean error of the estimates is 10.84 sec, which shows that our strategy for estimating the transfer times is accurate. We obtained the highest accuracy for an error of 1 sec and the lowest accuracy for an error of 33 sec. It must be noted that the lowest accuracy was obtained when the network link was a 28.8 Kbits modem connection, which should be viewed in the context of comparing modem speeds with T1 speeds. Furthermore the mean error as a percentage of the mean run-times is 23.59 percent, which is a good indicator of accuracy.

## 4.2 Estimating Data Transfer Time

In this section we present experimental results for the accuracy of the cost formulae for estimating the data transfer times. The data transfer times are primarily dependent on the size of the data and the network characteristics such as bandwidth and latency. In the implementation of the DAME system the data transfer is done using FTP, which we note is also used among commercial data mining service providers such as digiMine™ (http://www.digimine.com). The experiments were conducted by measuring the latency and bandwidth in real time between the two servers involved in the data transfers and then transferring the datasets required for mining. We conducted experiments using file sizes varying from 5 MB to 65 MB. We used two different connections – a high speed link (where the average bandwidth ranged from 6.5 – 8.3 Mbits and the latency was negligible) and a modem link (where the average bandwidth ranged from 41-45 Kbits due to internal modem compression and the latency varied from 2 to 3 sec). The comparative results between the estimated and actual transfer times for the three different bandwidth and latency characteristics shown are illustrated in figures 3 and 4.

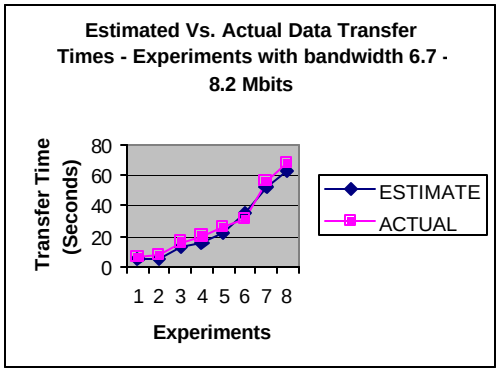


Figure 3 Experiments with Bandwidth Varying from 6.7 – 8.2 Mbits

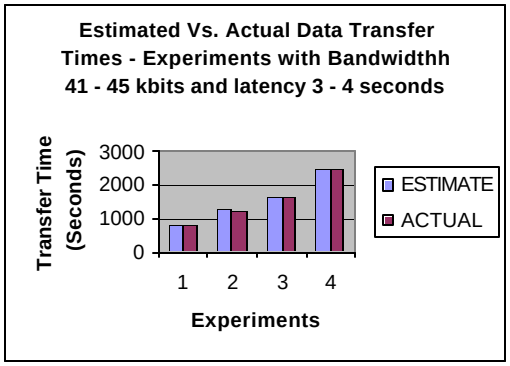


Figure 4 Experiments with Bandwidth Varying from 41 – 45 Kbits

The mean error for the experiments conducted using a high speed connection was 3.56 sec and the mean error as a percentage of the mean transfer time was 12.05 percent, which are both low indicating the high accuracy of the cost formulae. The mean error for the experiments conducted using a low speed link was 20.5 sec. However, the mean error as a percentage of the mean transfer time was 1.3 percent, which shows that the estimates are accurate. The combined mean error was 6.95 sec.

The difference in the mean error as a percentage of the mean transfer times between the two types of connections can be attributed to the fact that overall transfer times were very low with the high speed link. We have thus far presented experimental results validating the cost formulae for a priori estimation of the communication component of distributed data mining.

### 4.3 Estimating the Run Times of Data Mining Tasks

In this section, we present experimental results obtained by using our rough sets based application run-time estimation algorithm on data mining tasks. We compiled a history of data mining tasks by running several data mining algorithms on a network of distributed machines and recording information about the tasks and the environment. We executed several runs of data mining jobs by varying the parameters of the jobs such as the mining algorithm, the datasets, the sizes of the datasets, the dimensionality of the datasets and the machines on which the tasks were run. The algorithms used were from the WEKA package of data mining algorithms [18]. We generated several datasets of sizes varying from 1MB to 20MB. The data mining jobs were executed on four distributed machines with different physical configurations and operating systems. Three machines had Windows 2000 and one machine had Solaris 5.8. Two of the Windows machines were Pentium III with 833 Mhz processor and 512 MB memory, while the other was a Pentium II with 433 Mhz processor and 128 MB memory. The third machine was a Sun Sparc with 444 Mhz processor and 256 MB memory. The history provides the data for the estimation algorithm to predict the run time of a given data mining task. The rationale for building a history using a distributed network of nodes was two-fold. Firstly, we wanted to obtain a diverse history and test the estimation accuracy given a varied history. Secondly, it represents a realistic scenario where data mining web service providers and clients would typically operate using a distributed network of servers and would use the estimation for each node in the allocation of jobs.

For each data mining job, the following information was recorded in the history: the algorithm, the file name, the file size, the dimensionality of the data, the operating system, the version of the operating system, the IP address of the local host on which the job was run, the processor speed, the memory, the status of the server (whether it was dedicated or not), the start and end times of the job. The history was used to conduct experiments using the run-time estimation process described in chapter 4. We used histories with 100 and 150 records and each experimental run consisted of 20 tests. The performance accuracy is illustrated in figure 5, which presents the actual and estimated run-times from one of our experimental runs.

The mean error is approximately one minute and the error as a percentage of the actual run-times is 26.4%. The reduct that our algorithm selected as a similarity template included the following attributes: algorithm, file size, dimensionality and available memory. It must be noted that our application run-time estimation technique like other techniques that rely on historical data [15, 17] is limited by the initial need to collect a history.

We have presented experimental results that demonstrate the performance accuracy of our rough sets algorithm for estimating application run-times of data mining tasks. Thus far, the experimental evaluation has validated our technique for a priori estimation of the DDM response time. This facilitates service providers to commit to service levels that can be ensured in SLAs.

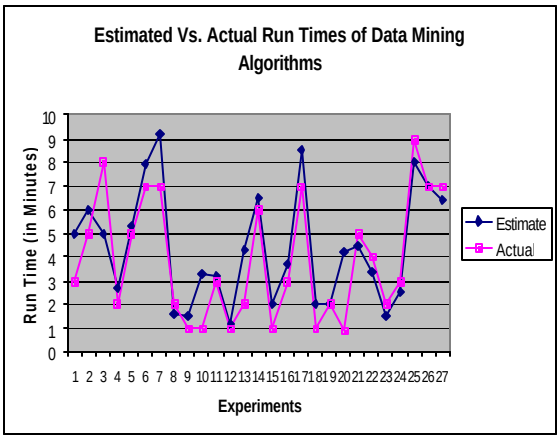


**Figure 5 Performance of Run-Time Estimation for Data Mining Tasks**

## 5. Conclusions and Future Work

This issue of estimating QoS in application services is important and one that needs to be addressed. This paper takes a first step in estimating the response time for data mining web services. The model presented in this paper is applicable to distributed, data intensive application services in general. The current focus is experimental evaluation of this work with other application services and focusing on additional QoS metrics.

## 6. Acknowledgements

This research is supported by the Australian Research Council.

## References


[1] Bouch, A., and Sasse, A., (2001), " *Why Value is Everything: A User-Centered Approach to Internet Quality of Service and Pricing*", Proceedings of the Ninth International Workshop on Quality of Service (IWQoS), Lecture Notes in Computer Science (LNCS) 2092, pp. 59-72.

[2] Dayal, U., (2001), "*Data Mining Meets E-Business: Opportunities and Challenges*", Proceedings of the First SIAM International Conference on Data Mining, Online Proceedings, Abstract of Keynote Available At:

http://www.siam.org/meetings/sdm01/html/k4.htm

[3] Fayyad, U, M., Piatetsky-Shapiro, G., and Smyth, P., (1996), "*From Knowledge Discovery to Data Mining an Overview*", Advances in Knowledge Discovery and Data Mining, (eds) Fayyad, Piatetsky-Shapiro, Smyth and Uthurusamy, AAAI / MIT Press, pp. 1-37.

[4] Harney, J., (2002), "*Big-Company Intelligence*", Enterprise Systems, June 2002, Available Online: http://www.esj.com/features/article.WSP ?EditorialsID=107

[5] Han, J., and Fu, Y., (1991), "*Exploration of the Power of Attribute-Oriented Induction in Data Mining*", Knowledge Discovery in Databases, (eds) G. Piatetsky-Shapiro and W.J. Frawley, AAAI/MIT Press, California, pp. 152-170.

[6] Krishnaswamy, S., Loke, S, W., and Zaslavsky, A., (2002), "*Supporting the Optimisation of Distributed Data Mining by Predicting Application Run Times*", Proceedings of the Fourth International Conference on Enterprise Information Systems (ICEIS 2002), April 3-6, Ciudad Real, Spain, pp. 374-381.

[7] Krishnaswamy, S., Loke, S, W., Zaslavsky, A., (2002), "*Towards Anytime Anywhere Data Mining Services*", Accepted for publication in the Proceedings of the Australasian Data Mining Workshop, To be held in conjunction with the 15th Australian Joint Conference on Artificial Intelligence (AI02), Canberra, Australia, 3rd December.

[8] Krishnaswamy, S., Zaslavsky, A., and Loke, S, W., (2001), "*Towards Data Mining Services on the Internet with a Multiple Service Provider Model: An XML Based Approach*", Journal of Electronic Commerce Research - Special issue on Electronic Commerce and Service Operations, Vol. 2, No. 3, August, pp. 20-56. Available Online:

http://www.csulb.edu/web/journals/jecr/issues/20013/paper2.pdf

[9] Krishnaswamy, S., Zaslasvky, A., and Loke, S, W., (2002), "*Predicting Application Run Times Using Rough Sets*", Ninth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2002), Annecy, France, July, IEEE Press, pp. 455-462.

[10] Krishnaswamy, S., Zaslavsky, A., and Loke, S, W., (2002), "*Techniques for Estimating the Computation and Communication Costs of Distributed Data Mining*", Proceedings of International Conference on Computational Science (ICCS2002) – Part I, Lecture Notes in Computer Science (LNCS) 2331, Springer Verlag. pp. 603-612.

[11] Krishnaswamy, S., Zaslavsky, A., and Loke, S, W., (2002), "*Internet Delivery of Distributed Data Mining Services: Architectures, Issues and Prospects*", Accepted as Book Chapter in: Architectural Issues of Web-enabled Electronic Business, IDEA Publishing Group.

[12] Lange, D., B., and Oshima, M., (1998), "*Programming and Deploying Java Mobile Agents with Aglets*", Addison-Wesley, 1998.

[13] Morsel, A., (2001), "*Metrics for the Internet Age: Quality of Experience and Quality of Business*", Hewlett-Packard Labs Technical Report HPL-2001-179, Available Online:

http://www.hpl.hp.com/techreports/2001/HPL-2001-179.html

[14] Sarawagi, S., and Nagaralu, S, H., (2000), "*Data Mining Models as Services on the Internet*", SIGKDD Explorations, Vol. 2, No. 1. Available Online:

http://www.acm.org/sigkdd/explorations/

[15] Smith, W., Foster, I., and Taylor, V., (1998), "*Predicting Application Run Times Using Historical Information*", Proceedings of the IPPS/SPDP'99 Workshop on Job Scheduling Strategies for Parallel Processing, Lecture Notes in Computer Science (LNCS) 1459, Springer Verlag, pp. 122-142.

[16] Sahai, A., Ouyang, J., Machiraju, V., and Werster, K., (2001), "*BizQoS: Specifying and Gauranteeing Quality of Service for Web Services through Real Time Measurement and Adaptive Control*", Hewlett-Packard Labs Technical Report HPL-2001-96, Available Online:

http://www.hpl.hp.com/techreports/2001/HPL-2001-134.html

[17] Smith, W., Taylor, V., and Foster, I., (1999), "*Using Run-time Predictions to Estimate Queue Wait Times and Improve Scheduler Performance*", Lecture Notes in Computer Science (LNCS), 1659, Springer Verlag, pp.202-229.

[18] Witten, I, H., and Eibe, F., (1999), "*Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*", Morgan Kauffman.

[19] Wolter, K., and Moorsel, A., (2001), "*The Relationship between Quality of Service and Business Metrics: Monitoring, Notification and Optimization*", Hewlett-Packard Labs Technical Report HPL-2001-96. Available Online:

http://www.hpl.hp.com/techreports/2001/HPL-2001-96.html