Latency Prediction Using Aggregate Latency Profiles

Qiang Ye
University of Pittsburgh
Pittsburgh, PA
qye@mail.sis.pitt.edu

Avigdor Gal
Technion – Israel Institute of Technology
Haifa, Israel
avigal@ie.technion.ac.il

Vladimir Zadorozhny
University of Pittsburgh
Pittsburgh, PA
vladimir@sis.pitt.edu

Louiqa Raschid
University of Maryland
College Park, MD
louïqa@umiacs.umd.edu

ABSTRACT

A prominent challenge in the support of Wide Area Applications (WAA) involves their unpredictable behavior over a dynamic WAN that results in a considerable variability in access latency (end-to-end delay). Latency profiles capture the changing latencies that clients experience when accessing a server and can be utilized as a WAA monitoring and optimization tool. However, in the presence of hundreds of servers and tens of thousands of clients, managing millions of latency profiles cannot scale. In this paper we propose a method for scalable monitoring and performance prediction of WAA using aggregate latency profiles. We utilize Relevance Networks to computes and analyzes comprehensive pair-wise mutual information and correlation for the latency profiles, as well as to build and maintain aggregate latency profiles. We have conducted experiments demonstrating that there is a considerable amount of non-random associations between individual latency profiles. We empirically show that aggregating them improves the overall prediction power of distributed performance monitoring for WAA.

1. INTRODUCTION

Wide area applications (WAA) utilize a WAN infrastructure (e.g., the Internet) to connect a federation of hundreds of servers, typically content providers, with tens of thousands of clients. Servers provide services that may range from downloads of digital content to authentication sessions, involving two or more parties. It is expected that such applications must scale to tens of millions of resources.

WAA, while promising in their scope and impact, face significant challenges. A prominent challenge involves the unpredictable behavior of a dynamic WAN [20, 22] that results in a wide variability in access latency (end-to-end delay). In [19], we have proposed latency profiles as a conceptual modeling tool for the behavior of sources over a WAN. Latency profiles are time-dependent latency distributions that capture the changing latencies clients experience when accessing a server.

Latency profiles can be utilized as a WAA monitoring tool, gathering efficiently data on access and performance patterns. They can also serve to predict latencies that clients should expect in response to requests. Therefore, latency profiles can assist in personalizing services of WAA to client’s specific network capabilities, including available bandwidth and “distance” from server, in order to improve service delivery in a heterogeneous WAN environment. However, in the presence of hundreds of servers and tens of thousands of clients, managing millions of latency profiles cannot scale. Therefore, we explore in this paper a method for aggregating latency profiles. We propose the use of mutual information and correlation to define latency profile similarity, and use them further to aggregate similar latency profiles. We also propose an approach to analyze meaningful relationships among iLPs using Relevance Networks (RN) [6]. RN has been developed for functional genomic clustering to discover non-random associations between genes on the basis of their biological characteristics. In this paper we use RN for building and maintaining aggregate latency profiles. We have conducted experiments with latency profiles, demonstrating the feasibility of constructing and aggregating them. The experimental data was collected over the CNRI Handle testbed [23], an emerging IETF/IRTF standard that provides a global name service for use over WANs. We empirically show that there is a considerable amount of non-random associations between individual latency profiles. Our empirical analysis also indicate that the non-random associations between latency profiles improve overall prediction power of distributed performance monitoring. Therefore, the contribution of this paper is twofold. From a conceptual point of view, we introduce a useful tool for scalable management of WAA, in the form of aggregate latency profile. We also provide a technique, based on existing work in information theory, to successfully generate aggregate latency profiles.

The rest of the paper is organized as follows. The latency profile model and basic WAA performance monitoring architecture are presented in Section 1. We next detail our experiments with generating and aggregating latency profiles (Section 3). The paper is concluded with an overview of related work (Section 4) and directions for future research (Section 5).

2. LATENCY PROFILES: MODELING CLIENT-SERVER INTERACTION

In this section, we propose latency profiles as a conceptual model of the client-server interaction. To illustrate our approach, consider Figure 1 that describes WAA performance monitoring architecture. There are three types of nodes in this figure, namely clients, content servers, and performance monitors (PMs). Clients use PMs to
optimize their access to content servers. PMs maintain and estimate different performance measures for a group of client/server pairs with similar performance. In this paper we consider end-to-end latencies as the performance measure maintained by PMs. PMs maintain latency information in form of latency profiles. In related research we applied passive information gathering strategies for building individual latency profiles (iLPs) for client/server pairs. However, managing millions of individual latency profiles does not scale. Below we consider a method for aggregating latency profiles that improves scalability of the WAA monitoring.

The main idea of our approach is as follows. Suppose that a client/server pair \((c, s)\) does not have an associated iLP that can be directly used to optimize access from \(c\) to \(s\), or alternatively, the system does not have sufficient resources to continuously maintain such a profile. Assume further that there is a well-defined individual latency profile \(iLP_1\) associated with a client/server pair \((c_1, s_1)\). In addition we have latency profiles \(iLP_2\) and \(iLP_3\) associated with client/server pairs \((c, s_1)\) and \((c_1, s_1)\). If there is a non-random association between \(iLP_1\), \(iLP_2\) and \(iLP_3\), we argue that a reasonable estimate of access latency for \((c, s)\) can be obtained by grouping \(iLP_1\), \(iLP_2\) and \(iLP_3\) in an aggregate latency profile \(aLP\).

Our general approach is as follows. After constructing iLPs we aggregate them to improve the overall prediction power of PMs. The information maintained by PMs includes measured individual latency profiles and derived aggregate latency profiles \((aLPs)\). In this case, PMs form a hierarchy, or overlay network similar in spirit to control and measurement overlays in M-Coop architecture [21]. The number and placement of PMs should maximize scalability of performance monitoring and minimize uncertainty in latency estimation. The problem of designing PMs hierarchy is out of scope of this paper. Instead, we focus on constructing and utilizing \(aLPs\).

In the rest of this section we give a formal definition of \(iLP\) and elaborate on grouping of similar (non-randomly associated) \(iLPs\) in aggregate latency profiles. We start by defining individual latency profiles (Section 2.1). Next, we define aggregate latency profiles (Section 2.2) and introduce relevance networks as a tool for constructing aggregate latency profiles (Section 2.3). Finally, we show how aggregate latency profiles can be used for predicting latencies in Section 2.4.

### 2.1 Individual Latency Profiles

Given a client \(c\), a server \(s\), an object of size \(b\), and a temporal domain \(T\), an individual latency profile is a function \(iLP_{c,s} : T \times b \rightarrow \mathbb{R}^+ \cup \{TO\}\). \(iLP\) represents the end-to-end delay for a request from server \(s\) at time \(t\). \(TO\) represents timeouts. \(iLP_{c,s}\) comes in two flavors, similar to [11]. One flavor measures time-to-first, which depends on factors such as workload at the server and size of the requested object. The other flavor measures time-to-last, which has a greater dependency on network bounds.

Due to the stochastic nature of the network, \(iLP_{c,s}(t)\) is clearly a random variable, yet its specific representation can vary. Below assume \(iLP_{c,s}(t) = iLP_{c,s}\) for all \(t\), to be a discrete time-independent random variable, represented as an \((L, p)\) matrix where \([L] = [L_1, L_2, ..., L_n]\) is a row matrix of latencies and \([p] = [p_1, p_2, ..., p_n]\) is a row matrix of corresponding latency probabilities (\(\sum_{k=1}^{n} p_i = 1\)).

#### Example 1
As an example consider the following probability distributions corresponding to two individual latency profiles \((X\) and \(Y\) represent specific client/server pairs):

\[
LD_X = \begin{pmatrix}
1 & 0.5 \\
0.5 & 2
\end{pmatrix}, \quad LD_Y = \begin{pmatrix}
2 & 0.75 \\
0.35 & 3
\end{pmatrix}
\]

### 2.2 Aggregate Latency Profile

An aggregate latency profile \(aLP_{1,2}\) combines a set of \(n\) individual latency profiles \(iLP = \{iLP_{c_i,s_i}\}_{i=1}^{n}\). Constructing an \(aLP\) involves grouping \(iLPs\) with similar characteristics in order to improve overall latency prediction. Apparently, to achieve improvement in the prediction quality one has to aggregate only \(iLPs\) that are non-randomly associated with each other. Therefore, one should have a methodology to evaluate candidate \(iLPs\) for aggregation purposes. Below we introduce the concept of LP similarity that can be used to provide such evaluation.

We define a similarity function \(\Sigma : CS \times CS \times T \rightarrow SM\), where \(CS\) is the set of all possible client/server pairs, \(T\) is a set of finite time regions (possibly intervals), and \(SM\) will be discussed shortly. \(\Sigma\) is a function that measures, given two latency profiles, their similarity over \(\tau \in T\).

We next provide two specific measures of latency profile similarity, based on mutual information [10] and correlation [16]. To estimate the similarity of two \(iLPs\), we consider their joint behav-
ior described by a joint probability matrix \( [P(X,Y)] \). \([P(X,Y)]\) provides the probabilities of the joint occurrence of two latencies.

**Example 2.** Consider \( X \) and \( Y \), given in Example 1. Their joint probability distribution is given as:

\[
P(X,Y) = \begin{pmatrix} (1, 2) & (1, 3) & (2, 2) & (2, 3) \\ 0.5 & 0 & 0.25 & 0.25 \end{pmatrix}
\]

Mutual information between two random variables \( MI(X,Y) \) is defined as

\[
MI(X,Y) = \sum_{i,j} (p_{i,j} \log \frac{p_{i,j}}{p_i p_j})
\]

where \( p_{i,j}, p_i, p_j \) are joint and individual probabilities of the latencies \( X \) and \( Y \), respectively. A higher mutual information between two \( iLPs \) means that those \( iLPs \) are non-randomly associated. Conversely, a mutual information of zero means that the joint distribution of \( iLPs \) holds no more information than their individual distributions.

We define correlation between two random variables \( Corr(X,Y) \) as follows:

\[
Corr(X,Y) = \frac{1}{n-1} \sum_{i,j} \left( \frac{x_i - \bar{X}}{S_X} \right) \left( \frac{y_i - \bar{Y}}{S_Y} \right)
\]

The correlation coefficient as defined above measures the degree of linear association between two variables. A higher correlation between two \( iLPs \) can also indicate that those \( iLPs \) are non-randomly associated. In general, there is no straightforward relationship between correlation and \( MI \) [14]. While correlation captures linear dependence, mutual information is a general dependence measure.

**Example 3.** For the latency distributions of Example 1 and Example 2 we calculate mutual information \( MI(X,Y) \), as follows:

\[
p_{1,2} \log \frac{p_{1,2}}{p_1 p_2} + p_{1,3} \log \frac{p_{1,3}}{p_1 p_3} + p_{2,2} \log \frac{p_{2,2}}{p_2 p_2} + p_{2,3} \log \frac{p_{2,3}}{p_2 p_3} = 0.31
\]

As for correlation, \( \bar{X} = 1.5, \bar{Y} = 2.25, S_X = 0.58, S_Y = 0.5 \), and \( Corr(X,Y) = 0.57 \).

**2.3 Constructing Aggregate Latency Profiles using Relevance Networks**

We next propose an approach to analyze and visualize meaningful relationships among \( iLPs \) using Relevance Networks (RN) [6]. RN has been developed for functional genomic clustering to reveal non-random associations between genes on the basis of their biological characteristics. In this paper we apply RN in the context of WAA performance monitoring. In particular, we explore how RN can be used for building and maintaining aggregate latency profiles.

The RN-methodology is based on comparing pair-wise relationships (e.g., correlation and mutual information) for all \( iLP \) pairs. Consider a graph whose nodes represent \( iLPs \) and edges represent the relationships (associations) between them. Assume that we compute all pair-wise relationships. By choosing a relationship threshold and displaying only those edges with a relationship higher then the threshold, then, out of completely connected network of \( iLPs \), we extract clusters of \( iLPs \) whose relationship to each other is “stronger” than the threshold. Such clusters are called Relevance Networks. Observing how the threshold increase impacts characteristics of the Relevance Networks (e.g., number of edges and number of connected components), one can generate a set of \( iLP \) relationships and aggregate strongly related \( iLPs \). One outcome of this approach is that it provides us with a natural quality estimation of both \( aLPs \) (aggregate only above certain threshold), and the whole group of candidate LPs (sensitivity to threshold increase).

**Example 4.** Figure 2 provides an example of a relevance network, as was generated during the experiments. Each node is a client-server pair (e.g., pubs-qow) and an edge between two nodes represent a similarity above the network threshold. The RN in this example has two connected component (one of size 2 at the top of the figure and the other of size 8 at the right hand side of the figure).

To build a Relevance Network, one needs an input feed in the form of \( (iLP_1, iLP_2, Measure) \) and a threshold specification. The measure, in our case, is either the correlation or mutual information between \( iLP_1 \) and \( iLP_2 \).

In Section 3, we shall elaborate on our empirical results, using relevance networks.

**2.4 Latency Prediction Using Aggregate Latency Profiles**

After constructing an \( aLP \) from a set of \( iLPs \), we can improve prediction quality of an \( iLP \) using observations of \( iLPs \) from the same \( aLP \). To demonstrate that the meaningful relationships between profiles within an \( aLP \) discovered in the previous section can be used to improve the quality of latency prediction we will use latency estimations using conditional expectation (CE).

A well-known fact from estimation theory is that the expected value \( E(X) \) of a random variable \( X \) minimizes the expected value of the mean-square-error of estimation \( E((X - \text{est}_X)^2) \) [16]. Using an observation of a second random variable \( Y \) which is related to \( X \) in some way (e.g., \( Y \) is correlated with \( X \)), an optimal mean-square-error estimator of \( X \) given \( Y \) is the conditional expectation \( E(X|Y) \) of \( X \) given \( Y \) [16]:

\[
\text{est}_X = E(X|Y) = \sum_{x_i} (x_i p(x_i|y_i))
\]

where \( p(x_i|y_i) \) is the conditional probability of \( x_i \) given \( y_i \), which can be easily calculated from the joint probability distribution \( p_{x_i} \),

![Figure 2: Relevance network example](image-url)
using the following equation:

\[ p(x_i|y_1) = \frac{p(x_i, y_1)}{p(y_1)} \]

**EXAMPLE 5.** Using latency distribution from Example 1 and Example 2, we calculate the following conditional probability distribution:

\[
P(X|Y) = \begin{pmatrix} \frac{1}{2} & 1 & 1 & 2 & 2 & 2 \ \frac{0.67}{3} & 0.33 & 1 \end{pmatrix}
\]

Then,

\[ E(X|Y = 2) = 1 \times 0.67 + 2 \times 0.33 = 1.33 \]

\[ E(X|Y = 3) = 1 \times 0 + 2 \times 1 = 2 \]

It is obvious that conditional expectation based estimation outperform estimation based on simple expectation, which in this case would be \( E(X) = 1 \times 0.5 + 2 \times 0.5 = 1.5 \).

We assume that associated iLPs that form an aggregate latency profile can be efficiently used as related observations in CE based prediction of a given iLP. In the next section we provide an experimental validation of this assumption.

### 3. EXPERIMENTS

We are now ready to report our experiences with constructing aLPs. We explore two approaches for choosing aLP members, namely using maximum mutual information and using maximum correlation. In this section, we are interested in validating the following two hypotheses:

1. There is a considerable amount of non-random associations between iLPs in wide area environments.
2. Non-random associations between iLPs can be utilized to build aLPs that improve the overall prediction quality of distributed performance monitoring.

### 3.1 Experimental Methodology

In this section we report on the experimental methodology, including data preparation (Section 3.1.1) and the evaluation methodology (Section 3.1.2). The Relevance Network analysis and visualization was done using Tom Sawyer Graph Analyzing and Visualization Software [4].

#### 3.1.1 Data Preparation

The data was collected in December 2002, over the CNRI Handle testbed [23], utilizing a WAA accessing handles. The Handle protocol, an emerging IETF/IRTF standard, provides a global name service for use over WANs, providing a namespace, a name resolution service, and protocols for digital object location and access. The International Digital Object Identifier (DOI) Foundation (www.doi.org) and the community of publishers utilize Handle to facilitate the identification and exchange of intellectual property in the digital environment.

Based on an analysis of the DOI server logs, we determined the most popular content repositories for DOI data. The data is typically PDF files that are reachable via Handle resolution. We identified data objects of approximately similar size (between 70-100 KBytes) at these content servers.

For this experiment, our location of client and server sites was dictated by our access and ability to deploy multiple Handle clients within different subnetworks of a friendly AS. We studied the behavior of 10 servers and 22 clients, so that each AS that is represented in the experiment has two clients. Thus, there are \( 10 \times 22 = 220 \) Client-Server pairs altogether. Each Client-Server pair corresponds to one iLP. For any \( 2 \) iLPs we have performed the following data processing:

**Synchronization** Based on the latency timestamps in one iLP, we have identified the corresponding latency in another iLP, where correspondence is taken to be the latency with the closest timestamp. Two timestamps within one hour are considered to co-occur. It is worth noting that synchronization is not necessarily symmetric. That is, synchronizing \( iLP_1 \) with \( iLP_2 \) may result in different joint latency sample than that of synchronizing \( iLP_2 \) with \( iLP_1 \). To ensure the stability of our calculations, we repeated the same calculations by randomly permuting iLP orders. We defer the report of these experiments to an extended version of this work, yet suffice it to say at this time that we did not observe significant modifications among the permuted distributions.

**Normalization** We have generated latency logs of similar sizes, ranging from 970 to 1000 samples for each iLP. Normalization is aimed at ensuring comparable statistical measures.

**Similarity computation** For each synchronized and normalized latency log, we calculated mutual information (MI) and correlation, resulting in two similarity measures, \((iLP_1, iLP_2, \text{correlation})\) and \((iLP_1, iLP_2, \text{MI})\), for any two iLPs, \(iLP_1\) and \(iLP_2\).

#### 3.1.2 Evaluation methodology

We have evaluated the relative performance of mutual information and correlation in generating connected components in relevance networks. We have utilized four measures in our analysis, to be discussed shortly. All measures are provided as a function of the relevance network threshold.

**Associations** \( A(\text{th}) \). The number of edges that surpass the threshold. The number of associations ranges from 0 to the number of edges in a full graph, i.e., \( \frac{\text{th} \times (\text{th} - 1)}{2} \) for a graph with \( n \) nodes. The maximum number of associations is reached for a threshold of 0.

**Relevance networks** \( M(\text{th}) \). The number of connected components of size greater than 1. A connected component is defined as a subgraph in which any node is reachable from any other node. Therefore, the number of relevance networks ranges from 1 to \( \left\lfloor \frac{n}{2} \right\rfloor \) in a graph of size \( n \). At a threshold of 0 the graph contains a single relevance network.

**Participating nodes** \( P(\text{th}) \). The number of nodes that are associated. Connected components of size 1 are excluded from this measure, as they cannot serve in generating aLPs at the given threshold. The number of participating nodes ranges from 2 to \( n \) as long as \( A(\text{th}) > 0 \). Whenever \( A(\text{th}) = 0 \), \( P(\text{th}) = 0 \) as well. At a threshold of 0, there are \( n \) participating nodes.

**Connectivity** \( C(\text{th}) \). \( C(\text{th}) \) is computed as follows:

\[
C(\text{th}) = \begin{cases} \frac{A(\text{th})}{P(\text{th})} & A(\text{th}) > 0 \\ 0 & \text{otherwise} \end{cases}
\]

Capturing the strength of the relevance network. Whenever any pair within a relevance network has non-trivial associations \( A(\text{th}) = P(\text{th}) \times P(\text{th}) - 1 \), and thus \( C(\text{th}) = 1 \). This is, for example, the case, whenever the threshold is set to
0. On the other extreme, assume that $M(th) = 1$, yet any participating node is connected through a single association. Therefore,

$$C(th) = \frac{2n}{n(n-1)} = \frac{2}{n-1} \rightarrow n \rightarrow \infty 0$$

We use conditional expectation (see Section 2.4) to test the improvement in the quality of latency prediction.

3.2 Experimental Results

3.2.1 MI Relevance Networks

We have computed all pair-wise MI values, and set up the sequence of the MI thresholds between 0 and 1.5 with 0.25 increments. For each such threshold we modify the RN by discarding edges with $MI < th$. The relevance networks of each transformation correspond to the MI Relevance Networks with respect to a given threshold.

Figure 3 provides the changes to the four measures as a function of the threshold. As the threshold increases, the number and size of the relevance networks change. We observe a smaller number of associations and participating nodes, and larger number of smaller (down to two nodes) relevance networks. However, the number of relevance networks does not vary that much. In fact, up to $th = 0.4$ we observe only one relevance network despite considerable decrease of number of associations and participating nodes. With further increase of the threshold, the number of relevance networks varies between 1 and 3. Finally, we observe that as the threshold is increased from 0 to 0.5, the connectivity of the networks drops from 1 to 0.17 and then quickly increases in response to a smaller number of participants.

The main observation from this experiment is that with MI relevance networks there are a few dominating iLPs clusters, demonstrating a stable pattern in the network behavior.

Figure 4 provides a pictorial presentation of four MI Relevance Networks for threshold values of 0.9, 1.0, 1.1, and 1.2. In general, we observe grouping of iLPs either with common server (e.g., $ioi$ in the middle of the network), or with common client (e.g., $queen$ at the top and middle of the network). Client-based relevance networks seem to be more robust, surviving better increased thresholds.

3.2.2 Correlation Relevance Networks

We have computed pair-wise correlations for all iLPs and set up a sequence of correlation thresholds, ranging from 0 to 1 with a 0.05 increments. For each such threshold we modify the RN by discarding edges with $abs(Corr) < th$.

Figure 5 provides the changes to the four measures as a function of the threshold. Similarly to MI Relevance Networks, with increase of the threshold we observe a smaller number of associations and participating nodes, and larger number of relevance networks. With the threshold increasing from 0 to 0.4, the connectivity of the networks drops from 1 to 0.06 and then quickly increases with further decrease of the number of nodes (iLPs) participating in the Relevance Networks. The number of correlation relevance networks varies more than the counterpart MI relevance networks. It starts increasing even for low thresholds (unlike the MI RNs that remained unchanged for the initial increase in the threshold) and goes up to 11 for a threshold of 0.4. After that, it gradually reduces until reaching the value of 1 at a threshold of 0.95. One may conclude that a decrease of number of associations and participating nodes has more impact on correlation relevance networks comparing to MI relevance networks. Also, the number of iLP clusters is higher.

Figure 6 shows a set of correlation relevance networks for threshold values of 0.5, 0.6, 0.7, and 0.8. Similarly to MI relevance networks, we observe grouping of iLPs either with common server (e.g., $do$ in the middle of the network), or with common client (e.g., $queen$ at the top and middle of the network). The granularity of correlation relevance networks is finer. Unlike with MI relevance networks, clients and servers demonstrate a similar level of robustness in the face of a threshold increase.

To conclude, our experiments show that a considerable amount of non-random associations between latency profiles exist. In particular, the clustering around a single server or a single client seem to go hand in hand with results of earlier studies [13, 15]. When comparing MI RNs and Correlation RNs, it seems that the latter provides more useful aLPs than the former. With correlation we were able to achieve a reasonable number of aLPs, that contain most of the iLPs. For example, a threshold of 0.25 yields 6 aLPs, involving 114 out of 220 iLPs. For the same number of iLPs, MI provides a single relevance network, that with most likelihood will reduce the accuracy of prediction (see next section). On the other hand, MI provides at most 3 RNs, utilizing 56 iLPs, at the most. MI RNs provide a slightly higher level of connectivity, with a minimum of 0.13 as opposed to a minimum of 0.05 with correlation RNs. This may indicate a weaker inter-cluster connectivity.

3.2.3 Prediction

To demonstrate that the meaningful relationships between iLPs discovered in the previous section can be used to improve the quality of latency prediction we have applied conditional expectation to estimate latencies using observations of iLPs with different MI and correlations. Our hypothesis is that a higher MI value and a higher correlation indicate an improvement in prediction ability. We used MI and correlation relevance networks considered in the previous section to select two groups of iLPs as follows. First group ($G_{mi}$) includes four iLP pairs of increasing MI, while the second group ($G_{corr}$) includes iLP pairs of increasing correlation. Figure 7 shows quantile plots for relative error of prediction ($abs(x - x_{est})/x$) using conditional expectation as latency estimation. We plot the error graphs for higher and lower values of MI and correlation. The quality of prediction increases with increase of mutual information and correlation. This trend is more obvious for correlation then for MI. For higher values of MI ($> 0.9$) learning curves have some overlap in the area of smaller relative errors. However, the percentage of larger relative errors is much higher for lower values of MI and correlation.

To generalize this conclusion we plotted MI and correlation vs. the average relative error of prediction for complete MI and correlation relationship graphs (Figure 8). We observe a strong dependency of prediction quality on MI and correlation between latency profiles. We also observe that variability of the relative error is considerable. Figure 9 shows that major part of prediction errors (about 900 predictions) is in a good range of [0, 1]. However, more than 1000 prediction errors are large (above 3), and as we see from Figure 6, they can be as much as 75. Meanwhile, practically all of the large prediction errors spread over areas of low MI ($< 0.4$) and low correlation ($< 0.2$).

An important observation to be concluded from this discussion is that by aggregating non-randomly related latency profiles with higher values of MI and correlation assist in eliminating large prediction errors and in maintaining appropriate prediction quality.

4. RELATED WORK

We now consider related work in a variety of fields, reflecting
Figure 3: Characteristics of MI Relevance Networks as a Function of MI Threshold

Figure 4: Examples of MI Relevance Networks
Figure 5: Characteristics of correlation Relevance Networks as a Function of correlation Threshold

Figure 6: Examples of Correlation Relevance Networks
the models and techniques upon which we draw.

Commercial solutions have addressed performance issues for the WAN, e.g., Keynote and Appliance [1, 2]. They are typically based on proprietary technology and are not designed to be scalable to a federation of autonomous servers. Further, the focus of these products is to monitor traffic, bandwidth, or server utilization for (possibly limited number of) servers, so as to identify performance bottlenecks and improve or ensure quality of service for some specific applications or services. Clustering is a key to scalable construction and maintenance of performance monitoring, which is not an explicit objective of these systems.

Researchers involved with the UCBerkeley SPAND Project [22] have established techniques for shared passive information gathering, where performance data is gathered for all requests. Their techniques gathered large amounts of low level data and is not scalable to large numbers of clients and servers or to continuous monitoring. We present an approach for continuous performance monitoring that is scalable and will keep profiles up-to-date.

There has been research on route aggregation based on IP prefixes exchanged via the Border Gateway Protocol (BGP) as well as research to exploit BGP information for intelligent routing and to monitor and predict performance. Research on clustering of clients is reported in [13, 15]. Our work differs from these works in that we are interested in non-random associations that do not necessarily evolve from physical proximity.

Previous work in the area of digital resource caching, Web caching, and replication has also been very successful in improving client’s access to resources. [3, 5, 7, 8, 9, 18, 17, 12, 24]. However, such research has not typically focussed on learning latency distributions for numerous client-server pairs.

In prior research, we developed a catalog of latency distributions, between a specific client and an Internet accessible WebSource [25], based on WebPT - an online learning tool [11]. We identified observable characteristics (feature vectors) that are reflective of the particular source’s latency distribution for a specific client. Such features include the significance of the Time of Day and the Day of the Week, as well as significance of noise (variance) on both network and server workloads. This work extends the previous work in generating aggregate latency profiles that scale well in a wide area application environment.

5. CONCLUSION

We have presented the concept of a latency profile as a method for capturing the behavior of sources over a WAN. We also propose aggregate latency profiles as a scalable methodology for utilizing latency profiles. Mutual information and correlations are compared in their ability to explore useful aggregate latency profiles. Our experiments show that in general correlation serves better is generat-
ing aggregate latency profiles and in predicting latencies.
We plan on implementing our methods in a prototype, allowing
the generation of aggregate latency profiles and testing them out in
retrieving documents based on handle information. We are going to
use more advanced prediction techniques such as Neural Networks
and Web Prediction Tool [25], to fully utilize prediction power of
aggregate latency profiles.

There are many open topics left for future research. In particular,
we are interested in investigating the best methodology for setting
aggregate latency profiles. For example, what should be a sufficient
sample of an individual latency profile, to decide on which cluster it
should join? Also, a methodology for tuning clusters once formed
seems to be crucial for estimating the quality of prediction over
extended periods of time.

Acknowledgments
The authors acknowledge the valuable insights provided by Galit
Shmueli.

6. REFERENCES
caching and zipf-like distributions: Evidence and
networks: Functional genomic clustering using pairwise
entropy measurements. In Proc. Pacific Symposium on
algorithms. Proc. USENIX Symposium on Internet
[8] V. Cate. Alex - a global filesystem. Proc. USENIX File
improve freshness. In Proceedings of the ACM-SIGMOD
conference on Management of Data (SIGMOD), pages
Learning response time for websources using query feedback
and application in query optimization. VLDB Journal,
algorithm. Fifth International Web Caching and Content
[14] W. Li. Mutual information functions versus correlation
[15] Z. Mao, C. Cranor, F. Douglas, M. Rabinovich,
O. Spatscheck, and J. Wang. A precise and efficient
evaluation of the proximity between web clients and their
local dns servers, 2002.
[16] W. Mendenhall and T. Sincich. Statistics for Engineering and
architecture for a global web hosting service. Proceedings of
update propagation in epidemic replicated databases. In
P.M.G. Apers, M. Bouzeghoub, and G. Gardarin, editors,
Advances in Database Technology - EDBT'96, 5th
International Conference on Extending Database
Technology, Avignon, France, March 25-29, 1996,
Proceedings, volume 1057 of Lecture Notes in Computer
profiles: Performance monitoring for wide area applications.
In Proceedings of the Third IEEE Workshop on Internet
Applications (WIAPP '03), San Jose, CA, June 2003.
congestion of flows via end-to-end measurement.
infrastructure for network measurement. The Third IEEE
architecture for adaptive applications. In Proceedings of
Validating an access cost model for wide area applications.
Proceedings of the International Conference on Cooperative