AIM-HI: A framework for request routing in large-scale IT global service delivery

A. Khan
H. Jamjoom
J. Sun

In many network and IT (information technology) systems, users submit loosely defined (or "fuzzy") requests to obtain answers, solutions, or resources. Fuzzy requests are often presented in problem tickets and processed by an IT service management system. In such a system, problems are typically reported using vague user-generated descriptions of the symptoms (e.g., "my e-mail is not working"). Making use of the reported symptoms, the incident management system is then responsible for identifying the component causing the problem. An accurate and quick diagnosis from the fuzzy symptoms becomes critical for an efficient and timely resolution of the problem. In this paper, we propose a system for automated incident management using historical information (AIM-HI), a framework for autonomous routing of requests in large-scale IT global service delivery. AIM-HI incorporates historical request resolution information and frequency, together with queue bouncing trends to extrapolate algorithms for streamlining and automating the dispatch of requests or work among support groups and IT specialists. The simplicity and scalability of AIM-HI should lead to deployment in actual real operational systems in the future.

Introduction

In the global (worldwide) service delivery of IT (information technology), a user reporting a problem or failure will typically not know the actual root cause of the problem. Instead, a user provides a vague description of the symptoms in order to request service. Such "fuzzy" requests often require intervention by several IT specialists before the failed network or IT system resource is identified and the problem is resolved. The challenge of a timely resolution is amplified in an outsourced IT environment, where a service provider is often responsible for managing multiple large IT environments. In such an environment, thousands of fuzzy requests are generated every day with respect to diverse sets of issues related to numerous kinds of network, application, and middleware components. Support groups, organized around the managed infrastructures, need to correctly and quickly resolve such requests in order to meet various service-level agreements (SLAs).

A typical service request goes through two primary phases. During generation, each request is assigned a technology signature based on the reported problem symptoms. The signature reflects the likely failed system, component, and module. On the basis of the assigned signature, during routing or dispatching, a routing directory lookup identifies the support group best skilled to resolve the request. The request is then routed (through the incident management application) to the selected group. However, requests often are rerouted multiple times before being resolved, due to 1) the need for sequential interventions by multiple support groups and specialists and 2) misrouting. In this paper, we focus on misrouting, which is primarily caused by the coarse granularity of the assigned signature, in which each distinct signature may correspond to a diverse set of problems.

To help improve the request routing efficiency in IT global service delivery, we propose automated incident management using historical information (AIM-HI) that involves adaptively learning about both the generated requests and the specialists involved, and that applies the

0018-8646/09/$5.00 © 2009 IBM

derived knowledge to make more informed routing decisions for future requests. The AIM-HI system examines the assigned technology signature of each request and applies supervised learning algorithms [1] to evaluate the historical likelihood that the particular signature corresponds to specific problem symptoms. Prior performances of support groups are then analyzed to identify groups that have the required skill to resolve the diagnosed problem symptoms. In a related work, Shao et al. [2] apply a variable-order Markov model to determine potential resolver groups for a request.

However, identifying the correct support group does not guarantee that the request will be resolved in a single hop (i.e., redirection), since resolution efficiency of specialists varies, within the group, for each problem addressed by the group. For moderately sized groups, redirections of a request within a support group can add significant delays to the resolution of a request even when the request must go through a small finite sequence of groups, as in [2], before being resolved. In order to minimize the number of redirections, AIM-HI autonomously evaluates the performance of the specialists. The performance of a specialist is determined by her historical ability to handle a diverse set of problems and the efficiency with which she can resolve each type of problem. Preliminary evaluations suggest that AIM-HI can quickly route a new request to a potential support group and identify the appropriate specialist within the group to resolve the request.

This paper is organized as follows. First we provide a detailed quantitative overview of the outsourcing environment. Next, we describe the proposed AIM-HI framework in detail, and then we evaluate and analyze the performance of AIM-HI, and conclude the paper.

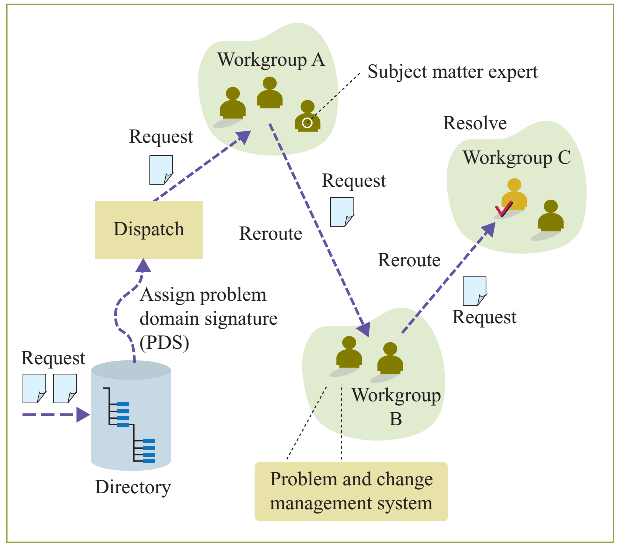

Figure 1

Overview of the global service delivery. (Republished with permission from Reference [3]; ©2008 IEEE.)

Quantitative overview of IT global delivery

The IT global service delivery we consider supports several hundred outsourced customer accounts that have generated more than 2 million service requests over the span of 10 months. Typically, for each supported technology (e.g., e-mail application or Linux** operating system), one or more workgroups services the associated requests. Each workgroup is assigned to one or more accounts and contains a variable number of subject matter experts (SMEs). A request goes through two primary phases: generation and dispatch. Regardless of whether the request is customer or machine generated, what is typically reported is the symptom, not the root cause of the problem.

When a request is customer generated, request routing is further complicated because problem symptoms are conveyed by both technical and nontechnical users. As mentioned in the previous section, based on the reported (and perhaps vague) problem symptoms, each request is assigned a technology or problem domain signature (PDS) that provides a hierarchical classification of the problem symptoms. For instance, the signature identifies the likely troubled system, component, module, and so forth.

At the dispatch phase, the primary challenge is to identify the correct workgroup that can solve the problem. Once the signature (PDS) is created, a potential workgroup is chosen from the routing directory, and the request is forwarded to the selected group. As mentioned, since symptoms may fail to capture the root cause of a problem, with the exception of simple issues (e.g., password reset), requests are often rerouted within and across workgroups before being resolved, as shown in **Figure 1**. The workgroup that resolves the problem is referred as the *resolver group*. **Table 1** provides a brief explanation of key terms used in this paper.

Given the large amount of requests available, we present results for four of the accounts that are representative of the complete set. The four accounts in **Table 2** are chosen such that they not only have a large number of requests, but also exhibit diversity in terms of the types of requests generated and the number of workgroups and SMEs responsible to service those requests.

Request routing efficiency

We begin with an analysis of the efficiency of the existing routing system. This also serves as a baseline for comparing proposed improvements with respect to the

Table 1 Terminology table.

Term	*Description*
Problem domain signature (PDS)	Technology signature associated with a request. This provides a hierarchical classification of the problem symptoms.
SMEs	Subject matter experts.
Workgroup	A group of SMEs that are in charge of a service domain.
Dispatcher	A person or system that assigns the requests to a workgroup.
Resolver group	A final workgroup that resolved the problem.

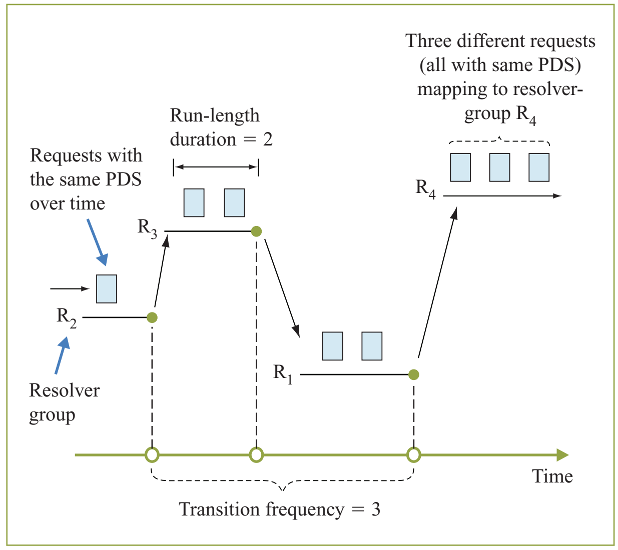

Figure 2

An example of transition frequency and run-length duration. The *y*-axis is used to denote different resolver groups.

existing system. The routing efficiency of a request is measured in terms of the number of redirections, or hops, required for request resolution. The routing path is typically captured by the problem management system and can be easily retraced.

The analysis suggests that only 20% of the requests in account A are resolved in a single hop, compared to 56%, 70%, and 63% for accounts B, C, and D, respectively. Single-hop resolutions are those typically handled by level 1 support (e.g., password resets). In general, the percentage of single-hop resolutions in an account is a measure of 1) the complexity of the requests serviced and 2) the scope for improving the routing efficiency. For example, account C has a large number of simple requests, probably associated with issues mostly handled by level 1 support. On the other hand, account A provides a large opportunity for improvement since only 20% of its requests are resolved in a single hop by the existing routing system.

Technology signatures mapping

Since a request is routed on the basis of its assigned signature (i.e., PDS), analyzing the mapping from PDS to resolver group will provide key insights into the causes of misrouting. First, we determine how well a particular PDS captures specific problem symptoms. This can be inferred from tracking the number of distinct resolver groups that have resolved requests assigned this distinct PDS. Intuitively, if a PDS maps to multiple resolver groups during its life cycle (the observed period of several months), then there is a clear ambiguity about the correct resolver group for requests that are assigned such a PDS. Our analysis shows that 55% of the signatures in accounts A and C map to multiple resolver groups compared to 24% and 27% for accounts B and D, respectively. Interestingly, a PDS may map up to 25, 30, 10, and 60 different resolver groups in accounts A, B, C, and D, respectively.

Table 2 Account summary.

Account	*Number of SMEs*	*Number of workgroups*
A	1,836	171
B	443	95
C	353	118
D	1,092	261

The mapping cardinality of a PDS does not capture the temporal variation of its mapping to a resolver group. More specifically, for a PDS, the mapping cardinality does not determine 1) the duration of each mapping and 2) the transition frequency between mappings. Both parameters are critical for building an efficient time-evolving request routing system, described later in the AIM-HI overview section.

The mapping duration of a PDS to a particular resolver group is captured by the *run length* of the mapping. This run length is the number of consecutive requests with the same PDS being resolved by the same group. As an example, in **Figure 2**, the run lengths of mappings for a PDS with resolver groups R_2, R_3, R_1, and R_4 are 1, 2, 2, and 3, respectively. Intuitively, a longer duration per mapping often reflects higher routing accuracy and fewer organizational changes. Our analysis of the requests shows that 30%, 15%, 8%, and 18% of the PDSs map to the same resolver group for a duration of only one run

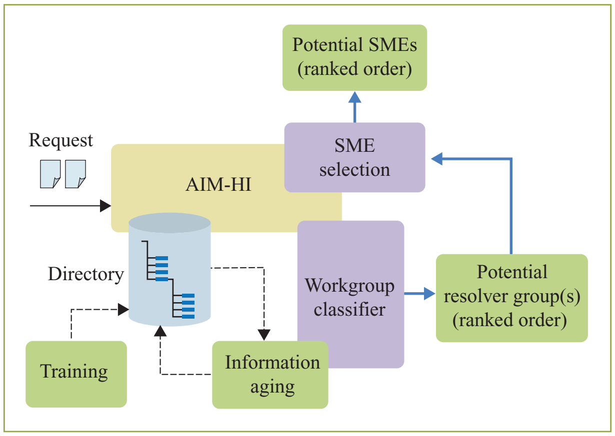

Figure 3

The AIM-HI architecture.

length in accounts A, B, C, and D, respectively. This implies a heavy oscillation in the mapping.

In order to understand the degree of oscillation in mapping, we determine the frequency of such transitions. In the illustrated example in Figure 2, the PDS undergoes three transitions in its mapping to resolver groups. In general, a higher transition frequency reveals a more unpredictable mapping from a PDS. The analyses of the accounts suggest that 35% of the PDSs in accounts A and C undergo two or more transitions. In contrast, approximately 50% of the PDSs exhibit two or more transitions in accounts B and D.

Workgroup compositions

Thus far, we have analyzed the dataset at the workgroup level. In reality, a workgroup consists of a large number of SMEs (as shown in Table 2), and requests often are redirected multiple times within a group before being resolved or forwarded to another workgroup. Furthermore, workgroups usually share SMEs, as an SME may belong to multiple workgroups. The assignment of an SME to a workgroup is based on his role as well as skill level and breadth of expertise. A more skillful SME may have the required skill to resolve requests originating from multiple problem domains and, thus, can contribute to multiple groups. Our analysis suggests 9%, 14%, 24%, and 25% of the SMEs work for more than one group for accounts A, B, C, and D, respectively.

AIM-HI overview

As mentioned, AIM-HI provides a framework for autonomous routing of requests in IT global service delivery. AIM-HI adaptively learns from historical routing information to build an efficient route prediction model for future requests. AIM-HI applies various supervised learning algorithms to determine the historical likelihood of a group to resolve requests exhibiting specific symptoms. AIM-HI also analyzes past performances of SMEs in handling requests in order to make more informed decisions about selecting SMEs for future requests. The architecture of AIM-HI is presented in **Figure 3**. The details of AIM-HI are presented in the following subsections.

Resolver-group selection

For each newly arrived request, AIM-HI identifies a set of workgroups that can potentially resolve the request. Since a request is routed on the basis of its assigned PDS, analyzing historical resolver groups for a PDS can potentially lead to more informed routing decisions for similar requests in the future. In order to evaluate historical mappings between a PDS and resolver groups, AIM-HI applies several supervised learning techniques [1], for which each new request is first classified using the current classifier, followed by a learning phase to update the current classifier with the mapping information in the recently examined request. In order to classify the categorical features of a PDS, we choose the following classification methods: maximum likelihood estimate (MLE), naïve Bayes, and decision tree. The classifiers are chosen according to their complexity, efficiency, and interpretability. In this work, our goal is not to perform a comprehensive evaluation of different learning algorithms. Rather, we are interested in comparing the performances of simpler, computationally less-expensive learning algorithms such as MLE and naïve Bayes with an algorithm such as decision tree that has more dynamic characteristics to potentially better classify categorical features.

MLE [4, 5] is chosen from a family of classifiers that provide simplicity and efficiency during both learning and classification phases. However, MLE may require a large training set to build an efficient classifier for each distinct PDS.

On the other hand, a naïve Bayes [6, 7] classifier requires a smaller training set, which is useful for an outsourcing environment that experiences rapid increases in newly supported accounts, typically with little historical data. An obvious drawback associated with the classifier is the assumption about the independence of each attribute in the PDS.

Finally, decision tree is one of the most widely applied data-mining algorithms for classifying categorical data [8]. It has dynamic properties such as powerful function approximation, and it facilitates easy interpretation of the classification results. The tree structure provides a useful balance between model complexity and accuracy, although construction of the tree is usually quite complicated. We consider a standard *C4.5 decision tree* [9] algorithm that has a very useful combination of error sampling rate and response time [10].

Since a dynamic classification performs an online update of the classifier with each newly served request, a critical design issue involves the amount of historical mapping information that needs to be considered for the update. Maintaining portions of the history is desirable for two reasons. First, as discussed in the quantitative overview of IT global delivery section, the mapping between a PDS and resolver groups can have a one-to-many relationship; thus, historical mappings can be used to filter noise, such as infrequent routing errors. Second, if a PDS has a historical mapping to only a single resolver group (i.e., a one-to-one mapping), then discarding such a PDS will lead to loss of information about a stable PDS-to-resolver-group mapping. We consider two mechanisms to determine the amount of historical mapping information that needs to be maintained: sliding window and exponential forgetting.

Sliding window
A sliding-window-based classifier is constructed by tracking the PDS-to-resolver-groups mapping of the recently observed requests. Specifically, a classifier C_n is updated with the mapping information from the most recent W requests $\{r_{n-W+1}, \ldots, r_n\}$, where r_n is the most recently serviced request. Mapping information of prior requests beyond W is discarded, and each of the mappings within the W request is given an equal weight during the update of C_n. By only considering requests that arrive within W, useful mapping information from other prior requests is ignored.

Exponential forgetting
As an alternative, exponential forgetting ages information with time, and new requests are given higher weights than the older ones during the update cycle of a classifier. As a result, all of the historical mapping vectors have certain effects when updating the classifier model, although the weight or importance of the information degrades with time.

SME selection
Once a set of potential resolver groups is selected, the goal is to identify efficient SMEs from the set to resolve the request. A random selection of SMEs tends to be inefficient, as SMEs within a group are not equally proficient in resolving each problem handled by the group. Moreover, a large number of random selections might be required to find the appropriate SME, even within a group of moderate size. Consequently, each redirection of the request adds delays to the resolution of the request. To avoid random selections of SMEs, AIM-HI maintains an efficiency score for each SME and chooses SMEs according to their scores.

The efficiencies of SMEs are determined by analyzing prior performances of the SMEs in resolving requests. From the past performances, AIM-HI determines for each SME: 1) historical likelihood to resolve a diverse set of problems and 2) resolution efficiency for each type of problem. The historical capability of an SME to resolve diverse types of problems is a reflection of the breadth of problem-solving skill of the SME. The breadth of problem-solving skills $B(S_k)$ of an SME S_k is defined as the number of requests with distinct PDSs handled by the SME. An SME who has worked on requests with diverse PDSs is likely to interact with numerous SMEs across various workgroups. In theory, such an SME is able to contribute to multiple workgroups, as shown by the analysis of workgroup compositions in the section on workgroup compositions. Therefore, SMEs who have resolved a more diverse set of problems are more likely to resolve a pending request with mislabeled or newer PDSs that might have been initially forwarded to the wrong workgroup.

For each of the diverse types of problems handled by an SME, AIM-HI determines the efficiency with which each such problem is resolved by the SME. Intuitively, an SME who is highly skilled with respect to resolving a particular problem is expected to require minimum time to resolve such a problem. We define each distinct PDS to correspond to a unique problem. For each distinct PDS_x, AIM-HI defines the problem resolution depth $D(S_k|PDS_x)$ of SME S_k as the expected number of redirections or hops required for a request, with signature PDS_x, to be resolved once it passes through S_k. An SME who is highly knowledgeable about particular problem symptoms will likely resolve such requests himself and have a very low $D(S_k)$ for that PDS. On the other hand, an SME who lacks in-depth knowledge about a particular PDS will have a much higher $D(S_k)$ for resolving requests with such a PDS, as the SME is likely to forward a larger portion of the requests to more knowledgeable SMEs. In its simplest form, $D(S_k|PDS_x)$ can be computed as

$$D(S_k|PDS_x) = \sum_{i=1}^{N} i \cdot P(i|S_k, PDS_x),$$

where i is the number of hops taken for a request with signature PDS_x to be resolved once it has passed S_k. $P(i|S_k, PDS_x)$ is the probability of taking i hops and is computed as follows:

$$P\left(i|S_k, PDS_x\right) = \frac{T(i|S_k, PDS_x)}{T(S_k|PDS_x)},$$

where $T(i|S_k, PDS_x)$ is the total instances of requests, with PDS_x, taking i hops from SME S_k to be resolved, and

$T(S_k|PDS_x)$ corresponds to the total number of requests with PDS_x handled by S_k.

On the basis of the derived performance metrics, AIM-HI determines an overall efficiency score $E(S_k|PDS_y)$ for SME S_k to resolve a new request with signature PDS_y

$$E\left(S_k|PDS_y\right) = \alpha \cdot N\left(\frac{1}{D(S_k|PDS_y)}\right) + (1-\alpha) \cdot N(B(S_k)),$$

where $N(1/(D(S_k|PDS_y)))$ and $N(B(S_k))$ are the normalized scores for the depth and breadth of historical problem-solving skills of S_k, and $0 \leq \alpha \leq 1$.

The efficiencies of the SMEs, in the workgroups selected in the section on resolver-group selection, are then determined on the basis of the desired selection criterion. If an SME is to be chosen based solely on his historical likelihood to efficiently resolve prior requests with PDS_y, then α should be maximized. Such selection can effectively reduce the request resolution time if prior requests with PDS_y have been historically resolved by one of the groups under consideration. On the other hand, an SME with expanded problem-solving skills is chosen by maximizing $(1-\alpha)$. The latter policy increases the resolution probability of requests with a misdiagnosed or previously unobserved PDS.

Once the efficiency scores of the probable SMEs are computed according to the desired policy, the request is routed to an SME with a high score. It is important to note that AIM-HI does not select the SME with the highest efficiency score in order to ensure that particular SMEs are not overloaded with requests. In future work, AIM-HI will take into consideration the request load or utilization of an SME to autonomously update the efficiency score of each SME.

Performance evaluation

The performance of AIM-HI is evaluated with respect to the entire sample size for each account. In creating our plots, we measure the performance of AIM-HI over non-overlapping 14-day intervals to provide a balance between observing time-varying trends in our evaluation and smoothing any anomalies that are typical in any actual dataset. Our results have suggested that naïve Bayes, decision tree, and MLE yield very similar performances in identifying a set of resolver groups for a new request. In general, the similarities in performance can be attributed to the fuzzy nature of the mapping between a PDS and resolver groups, where requests with the same PDS are resolved by as many as 60 different groups over time, as discussed in the section on technology signatures mapping. Due to the similarities in performances, we only present in details the performance of the dynamic MLE classifier.

Resolver-group selection

In evaluating AIM-HI, we first determine the accuracy of a dynamic MLE classifier in correctly predicting the resolver group for a new request. Due to the overlapping timeline of AIM-HI and the Markov-based routing model [2], a comparison of the performances of the two models has not been possible.

As presented in the section on resolver-group selection, AIM-HI uses two variations of dynamic supervised classification methods to identify a potential resolver group for the pending request. For the sliding window, we vary the window size W from 7 days to 91 days. The results in **Figure 4** depict a few interesting observations. First, a common dip in performance is noticeable during testing intervals 5 through 9. This is largely due to very few requests being served during that two-and-a-half month period. Since the classifier is updated based on a sliding window, mapping vectors corresponding to prior requests are discarded. As a result, a smaller window size performs worse than larger ones.

As a second observation, accounts A and C consistently exhibit low and high classification accuracies, respectively, if we ignore the intervals 5 through 9. This is largely due to the degree of stability in the mapping of the PDS in their respective accounts. As explained in the section on technology signatures mapping, in terms of number of consecutive requests, the longer a PDS maps to the same resolver group, the more predictable the mapping becomes, resulting in higher classification accuracy. Since 30% of the PDSs in account A do not have consecutive mappings to the same resolver group, the historical mapping vector of such a PDS becomes less useful in determining its next mapping, leading to inaccurate classification. On the other hand, in account C only 8% of the PDSs do not have consecutive mapping to the same resolver-group, thus resulting in a more predictable mapping.

As a third observation, the classification accuracy in account D shows a dip during the last few testing intervals. This is largely due to the arrival of requests with previously unseen feature vectors that are classified incorrectly due to the lack of historical mapping vectors. Overall, across the accounts, it can be inferred that a sliding window of only 7 days yields an accuracy similar to the other window sizes. This implies that historical mapping vectors may not need to be retained for longer periods, thus providing simplicity and savings in storage space and processing overhead.

For exponential forgetting, we vary the forgetting coefficient λ from 0.99 to 0.001 to analyze the effect of aging historical mapping information on classification accuracy. The results in **Figure 5** highlight several surprising results. First, during the testing intervals 5 through 9, the exponential forgetting classification yields

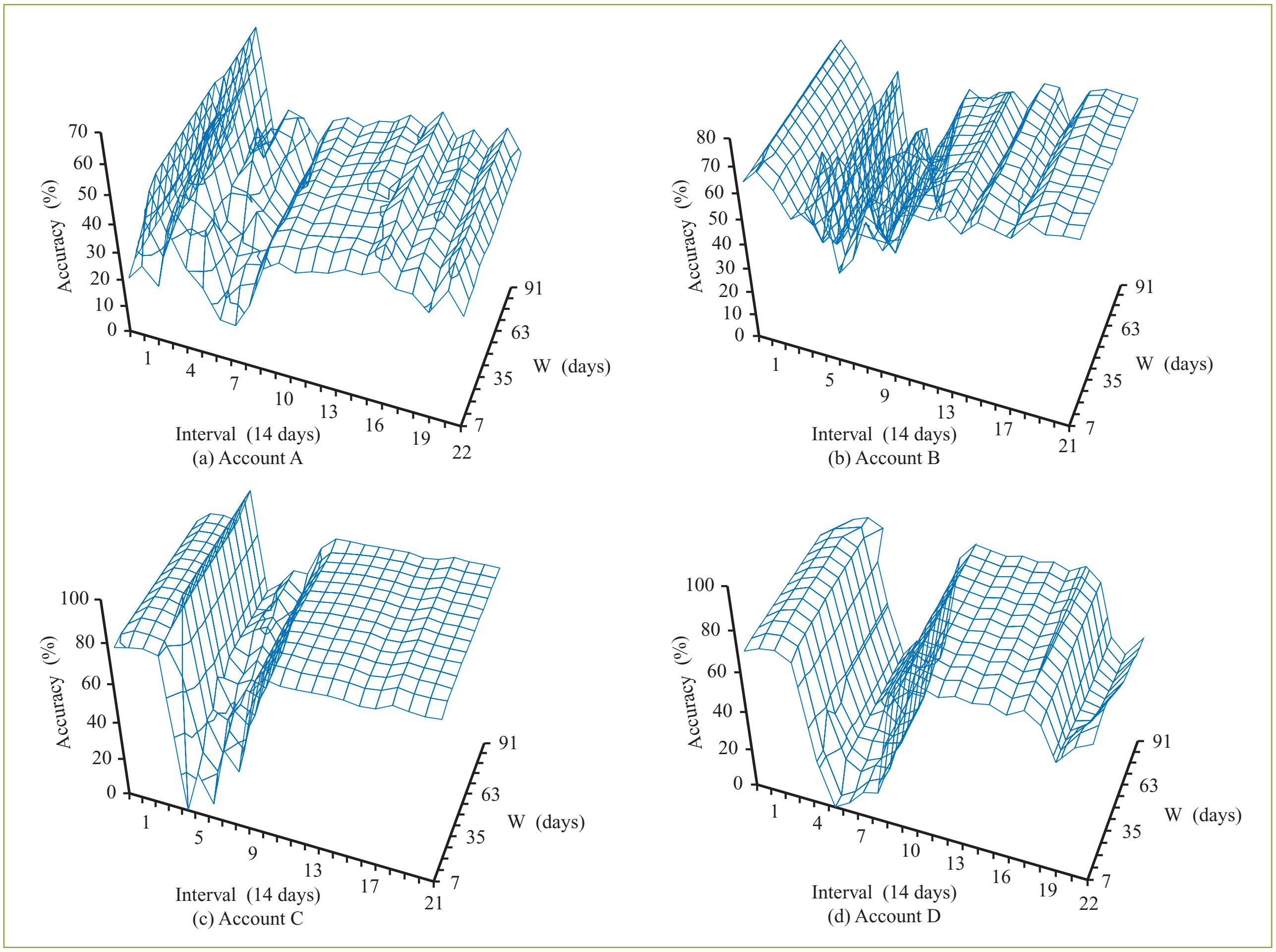

Figure 4

Accuracy of the sliding window classifier through time.

a much better accuracy than the sliding window. This is largely due to the capability of the exponential forgetting classification to retain and incorporate all of the historical mapping vectors when updating the classifier, and thus overcoming unavailability of sufficient requests at discrete intervals.

Second, one observes a trend similar to the sliding window case, as accounts A and C consistently exhibit low and high classification accuracies. This validates our prior observation that the behavior is due to the degree of stability in the mapping between a PDS/resolver-group pair within each account. The most surprising observation is that the accuracy acquired with a forgetting coefficient of 0.001 is similar to other aging variations across the accounts. This implies that classification schemes for this type of environment only need to "remember" the last correct classification. The finding also supports the results from the sliding window, where a window size of only 7 days yielded good accuracy.

Our analyses suggest that the leading cause of inaccurate classification is the instability in mappings from a PDS to resolver groups. Interestingly, one observes that a new request is usually associated with finding multiple potential resolver groups in the classifier. For instance, in accounts A, B, and D, more than 60% of the new requests found multiple mappings compared to 35% in account C. In order to utilize multiple mappings, for each new request, AIM-HI identifies a set of k resolver groups that has the most potential for resolving the request. The k groups are ordered according to their historical likelihood to resolve the request, as determined by the dynamic MLE classifier.

We limit k to 5 and use the dynamic MLE exponential forgetting classifier with $\lambda = 0.4$ to determine the top five workgroups for a pending request. The accuracy is

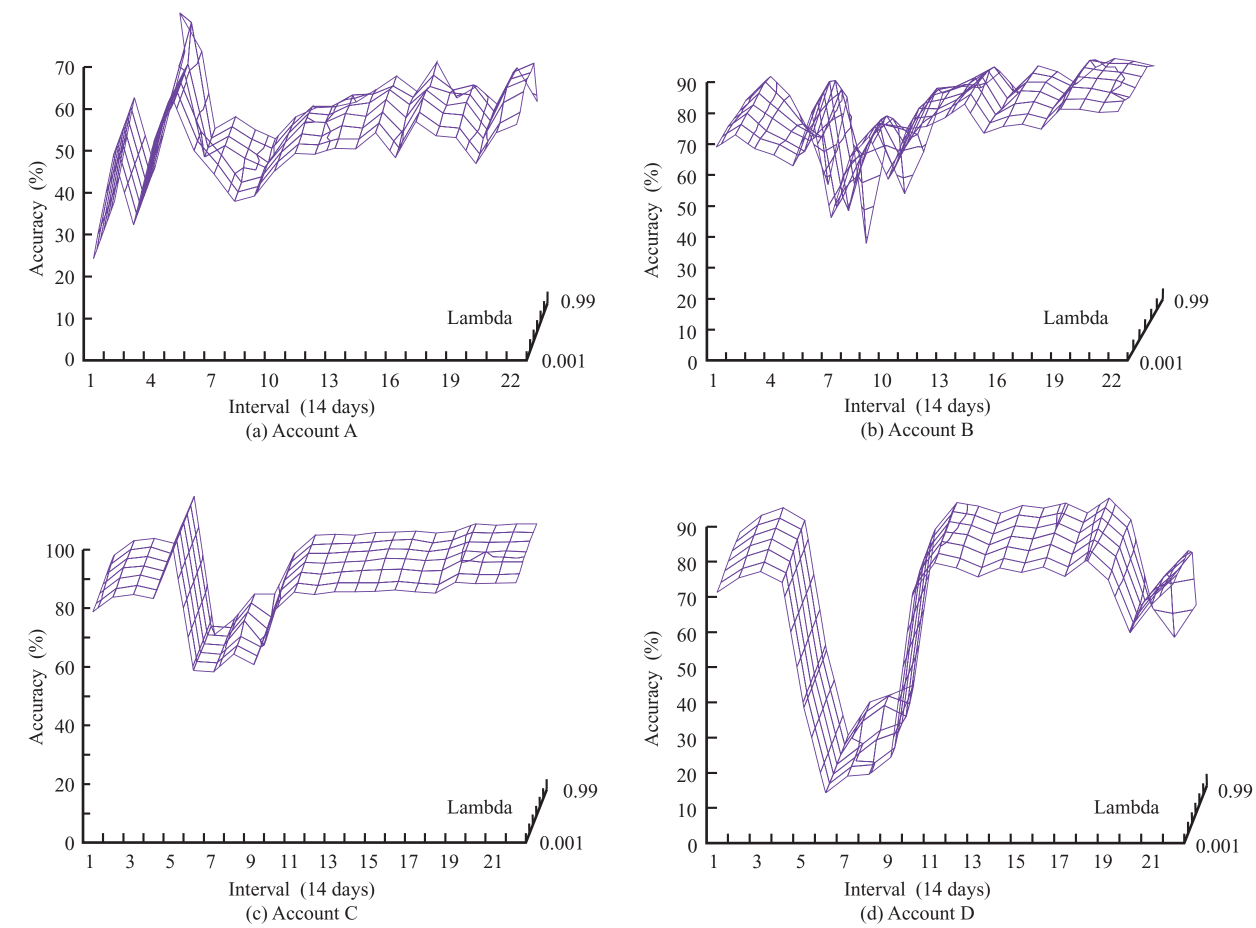

Figure 5

Accuracy of the exponential forgetting classifier through time.

measured cumulatively with respect to each workgroup to allow the best mapped resolver group to be selected first, followed by the second, and so forth. In **Figure 6**, we plot the improvement in accuracy with respect to a single or the top-matched workgroup selection. The results suggest that the MLE classifier can accurately identify a very small set of workgroups to resolve a request. However, one can notice a performance ceiling (i.e., limit) of the classifier, beyond which further prediction accuracy is not gained. For example, in accounts B, C, and D, there is no noticeable improvement in accuracy beyond choosing the top two potential workgroups. Prediction accuracy for requests in account A is also limited to the top four workgroups.

SME selection

From the set of potential resolver groups, an SME is chosen based on the efficiency score of the SME, as described in the section on SME selection. This leads to the following scenarios.

In case A, if the PDS of a new request matches a single resolver group, then AIM-HI selects an SME based solely on the resolution efficiency of the SME in resolving prior requests with the same PDS. By identifying an SME who is highly skilled in resolving the requests, a quicker resolution of the request is guaranteed since the request has already been routed to the appropriate workgroup.

In case B, if the PDS of a new request identifies multiple potential resolver groups, then AIM-HI determines the efficiency score for each SME in the matched groups and routes the request to the SME with the highest score. In this case, it is not obvious whether the efficiency of the SMEs should be computed based on their depth or breadth of knowledge. We propose to analyze in detail such situations in future works.

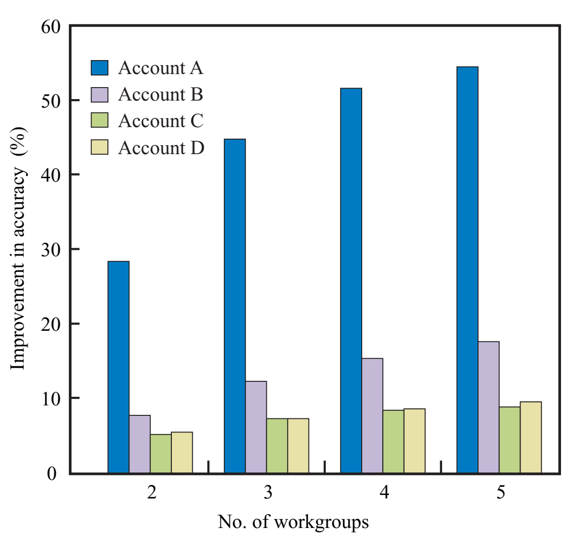

Figure 6

Cumulative improvement in accuracy with multihop routing.

In case C, if a previously unobserved PDS arrives that does not match any of the potential resolver groups, then AIM-HI selects an SME who has, in the past, worked on a highly diverse set of requests. The broader skills or knowledge of such SMEs can be explored in order to route the request to a potential resolver.

As mentioned, in all possible cases, AIM-HI avoids instantaneous selection of the SME with the highest efficiency score in order to avoid overloading an SME with requests. In addition, AIM-HI selects the next-hop SME with a degree of confidence that takes into consideration the number of the requests an SME has previously handled. Such a measure ensures that SMEs who have been randomly forwarded a problem (through misrouting) are not misleadingly assigned a high efficiency score for that particular type of problem.

Based on a preliminary evaluation, AIM-HI is found to attain dispatch accuracy as high as 91%. We are in the process of performing a more detailed evaluation and considering a limited pilot project in a production environment.

Conclusion

With the growing demand on IT outsourcing, studying the human aspect of request generation and resolution has become a key issue for scalable IT operation and management. In this paper, AIM-HI is proposed as a request routing framework that autonomously examines the trails (e.g., histories) of prior requests in order to build an efficient route prediction model for future requests. For each new request, AIM-HI first applies supervised learning algorithms to select a set of workgroups that can potentially resolve the request. From the selected groups, IT specialists can then be chosen on the basis of their derived efficiency scores. Our evaluations have shown that a dynamic MLE classifier can identify, with high accuracy, a small finite set of resolver groups for a pending request. Furthermore, the appropriately skilled specialists can also be rapidly identified from the selected groups with high precision.

We are currently performing more detailed evaluations of AIM-HI. Our future work will consider request loads and queuing delays with respect to specialists as well as the severity of the request in determining the efficiency score of the specialists. We will also evaluate AIM-HI for minimizing the resolution time of a request in addition to the number of redirections. Deployment of AIM-HI in an actual operational system is also being considered.

**Trademark, service mark, or registered trademark of Linus Torvalds in the United States, other countries, or both.

References

1. S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica* **31**, 249–268 (2007).
2. Q. Shao, Y. Chen, S. Tao, X. Yan, and N. Anerousis, "Efficient Ticket Routing by Resolution Sequence Mining," *Proceedings of 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'08)*, Las Vegas, NV, August 2008, pp. 605–613.
3. A. Khan and H. Jamjoom, "SOAR: Socially Aware Routing for Request Matching in Enterprise Environments," *Proceedings of IEEE International Conference on Services Computing (SCC)*, Honolulu, Hawaii, July 8–11, 2008, pp. 637–638.
4. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
5. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, 2009.
6. P. Domingos and M. J. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-one Loss," *Machine Learning* **29**, No. 2/3, 103–130 (1997).
7. D. J. Hand and K. Yu, "Idiot's Bayes: Not So Stupid After All?" *Intl. Statistical Rev*. **69**, No. 3, 385–398 (2001).
8. W. Buntine, "Learning Classification Trees," *Artificial Intelligence Frontiers in Statistics*, Chapman & Hall, 1993, pp. 182–201.
9. J. R. Quinla, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
10. L. Tjen-Sien, L. Wei-Yin, and S. Yu-Shan, "A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms," *Machine Learning* **40**, No. 3, 203–228 (2000).

Received December 17, 2008; accepted for publication February 17, 2009

Asheq Khan *Department of Computer Science and Engineering, University at Buffalo, 201 Bell Hall, Buffalo, New York 14260 (akhan6@cse.buffalo.edu)*. Mr. Khan is working toward a Ph.D. degree in computer science at the University at Buffalo, SUNY (State University of New York). He received his B.S. and M.S. degrees in computer engineering from University of Arkansas in 2002 and 2003, respectively. His research focuses on networking and covers topics in wireless mobile ad hoc networks, sensor networks, and social networks. Mr. Khan is a student member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM).

Hani Jamjoom *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (jamjoom@us.ibm.com)*. Dr. Jamjoom is a Research Manager at the IBM T. J. Watson Research Center, where he is focusing his research on the operational side of service science management and engineering (SSME), covering topics in data mining, social networking and collaboration, knowledge management, and system scalability. Dr. Jamjoom received a B.S. degree from the Rose-Hulman Institute of Technology in computer engineering and an M.Eng. degree from Cornell University in electrical engineering. He received his Ph.D. degree in computer science in 2004 from the University of Michigan, Ann Arbor, where he focused on quality-of-service architectures and the integration of controls in networks and operating systems to manage Internet services during overload scenarios.

Jimeng Sun *IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, New York 10532 (jimeng@us.ibm.com)*. Dr. Sun is a Research Staff Member at the IBM T. J. Watson Research Center. He received his B.S. and M.Phil. degrees in computer science from Hong Kong University of Science and Technology in 2002 and 2003, respectively. Dr. Sun obtained his M.S. and Ph.D. degrees in computer science from Carnegie Mellon University in 2006 and 2007, respectively. His research interests include data mining for streams and networks, databases, and service science. Dr. Sun has received his best research paper awards in ICDM (IEEE International Conference on Data Mining) 2008 and SDM (SIAM International Conference on Data Mining) 2007. He has published more than 30 refereed articles and a book chapter, has filed four patents, and has given four tutorials at technical conferences. Dr. Sun has served as a program committee member of SIGKDD (Special Interest Group on Knowledge Discovery and Data Mining), SDM, and CIKM (Conference on Information and Knowledge Management) and as a reviewer for TKDE (Transactions on Knowledge and Data Engineering), VLDB (Very Large Databases), and ICDE (International Conference on Data Engineering).