Speculative Authorization

Pranab Kini
University of British Columbia, Vancouver, Canada
Email: pranabk@ece.ubc.ca

***Abstract*—In large scale distributed systems, arriving at authorization policy decision is complex and computationally expensive. This decreases the overall performance of the system due to the delays introduced in fetching objects. Performance could be improved if policy decisions were made in advance and objects prefetched. We introduce a model to study the behavior of subjects in systems and find relationships between objects to precompute authorization decision. Markov chains have been popular for predicting future events based on subject behavior. We believe that analysis of relationships and patterns in objects can add to predictive capability. Our initial results suggest an improvement of 4 percent on top of prediction capability exhibited by markov chains.**

I. Introduction

Modern access control architectures are based on request-response paradigm [Kar03] as shown in figure 1. In this architecture, the policy enforcement point (PEP) intercepts application request from subject and forwards it to policy decision point (PDP) as authorization request. PDP checks underlying authorization policy to compute authorization response. Authorization response indicates if the subject has access to requested resource.

In large scale distributed systems, PDP's are installed as dedicated authorization servers and serve many PEP's. This leads to separation of authorization and application logic which is advantageous. These systems handle interactions between thousands of subjects and objects defined by authorization policies. They define the access levels of various subjects on objects present in the system. As the system grows in size, authorization policies grow in number and complexity. Consider a system with $'M'$ subjects and $'N'$ objects. A naive authorization policy scheme would consist of $'MN'$ policy definitions. If a policy matrix is built, it would consist of $'M'$ rows and $'N'$ columns, each row indicating access level of a subject on all the objects in the system. In this scenario, whenever subject requests for any object, PDP has to arrive at a policy decision from the large matrix of authorization policies. In practice, it might have to retrieve information from various resources to arrive at a decision. Actual computation of response can be expensive. Thus when several subjects simultaneously make requests for objects, delays would be introduced by PDP in computing authorization responses. The humongous size of subjects and objects along with the complexity in making policy decisions could eventually decrease the overall performance of the system.

Subjects access objects in some pattern. For example, during end of term, students(subjects) usually access resources(objects) related to final exam, project submission, etc. Such patterns could be studied and analyzed to prefetch access levels of students on these resources. In other words, PDP could speculate future authorizations required by subjects based on their behavior. PDP can analyze patterns from history of requests made by subjects and use this information to prefetch the policy decisions for objects that would be requested by subjects in near future. Once policy decisions are precomputed, PDP pushes these decisions in PEP caches. Later, when subject requests for object whose policy decision has been prefetched, PEP gets the response from its cache and serves the subject instantaneously. This results in better performance provided PDP analyzes and speculates policy decisions effectively.

In this paper, we introduce a model in which PDP speculates future authorizations required by subjects. The analysis for prediction is based on history of accesses made by subjects in the past. As an initial step, file system has been considered for the purpose of speculation but we plan to generalize the idea to other systems. In our case, files are treated as objects and users accessing those files are our subjects. Files and objects would be interchanged to suite the context wherever required. The same strategy holds good for subjects and users. We obtained the history of files accesses from [Per03]. Two features were extracted from the log traces. The sequence in which the files were accessed by the users and directory structure of the file system. The sequence of events helps us to predict the behavior of users in the future using markov chains. Markov chains can predict only those events that have occurred in the past in a particular sequence. If any user accesses files that she has never accessed before, markov chains would fail to predict such events. Directory structure helps in predicting such unseen instances. Files are usually placed in directories and sub-directories with a definite pattern. Files in the same sub directory have stronger correlation than files separated apart in different sub-directories. In this work, we use this feature for developing relationship between objects. Other possible ideas have been put forth in 'discussion' section. Initial results suggests an improvement of 4 percent on top of predictive capability exhibited by markov chains.

The idea of prefetching has been studied extensively in World Wide Web(WWW) domain. There are differences in prefetching policy decisions for objects as compared to prefetching web pages which are detailed in related work section.

The rest of the paper is organized as follows. The next section describes the approach used for modeling. Section III presents the technique followed for evaluation of model. In

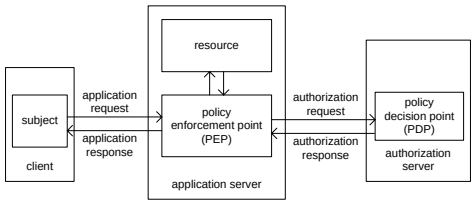

Fig. 1. PEP-PDP architecture

section IV we discuss our initial results. In section V, we explore various research papers that have been published in the area followed by discussion section. In section VII, we draw conclusions from the obtained results and discuss future work.

II. Approach

We develop our approach on the fact that PDP analyzes past requests made by subjects from log traces. These requests help in extracting features for any system. We build our model w.r.t. file system as we analyzed and speculated future file accesses. The model can be generalized easily by analyzing log traces of particular systems. We extract two features i.e. the sequence of requests and the location of files in the directory structure. Markov chains have been used to record the sequence in which files were accessed. Even though it is not important to predict the exact sequence of events for file accesses, knowledge about sequence provides information on user behavior. We consider a bigram model which gives us information about the immediately accessed file given current file while maintaining the simplicity of markov chains. Markov chains predict those behavior that have been observed in the past. To built training set based on markov chains, we need large history of accesses. The success of markov models depends on any subject making similar sequence of requests repeatedly. Markov chains cannot predict behavior that has not taken place in the past. To solve this problem, we extract second feature from the file system. This feature is the location of file in the directory structure. In any file system, files in a particular sub-directory are related to each other. This relationship is usually stronger than the files that are separated far apart in the directory structure. We have incorporated this information in our model to predict those events that have not been observed in the past. In the next two subsection we describe the two aspects of our model followed by the technique in which we add the information obtained from these sub-parts.

A. Markov Chains

We built a first order markov model(bigram model) from the log information. First order markov chains are given as follows

$$P(X_j|X_i) \tag{1}$$

The above equation gives the probability that file X_j was accessed immediately after X_i. We are interested in calculating this quantity from the logs. For this purpose, we calculate the number of times X_j was accessed immediately after X_i.Let us call this quantity $n(i \rightarrow j)$. In any file system having n files, there are $n-1$ possible files that can be accessed immediately after any given file. We calculate all such files that were accessed immediately X_i. Let us call this term as $n(i \rightarrow k)$ where $k = 1, 2, .., n-1$ Thus the probability of accessing file X_j after file X_i can be given by

$$P(X_j|X_i) = \frac{n(i \rightarrow j)}{\sum_{k=1}^{n-1} n(i \rightarrow k)} \tag{2}$$

A transition matrix ϕ is built based on the above equation. Each row of the transition matrix denotes the probability of all possible files accessed immediately after a particular file. The sum of all probabilities in any given row is equal to one. Each row can be considered independent of each other. Thus the likelihood function is

$$P(X_j|X_i, \Phi) = \prod_j \prod_i \phi_{j|i}^{n(i \rightarrow j)} \tag{3}$$

B. Distance Calculation

As mentioned before, markov chains can only predict those instances that have occurred in the past. In this section we build a model that uses relationship between two files for speculation. We consider the fact that files in the same subdirectory are closely related to each other. Files in different subdirectories will have less relationship as compared to ones in the same subdirectory. This relationship is associated to distance metric between two files. Distance between 2 files will be less if they are close to each other in the directory system. The further they are separated from each other in the directory, distance between them will increase accordingly.

The transition matrix in this case would have the same number of columns and rows as described for markov chains. We follow hierarchial bayesian model approach proposed by [CB95], to incorporate relationship i.e. 'distance' between files in our model. For this purpose, let us consider a Dirichlet prior on each row of above transition matrix. Thus the i^{th} row of transition matrix would have the prior given as

$$Dir(\alpha m_1, ..., \alpha mn - 1) = Dir(\alpha \mathbf{m}) \tag{4}$$

In the above equation, $\mathbf{m}$ is the prior mean satisfying the criteria $\sum_k m_k = 1$. These m_k's are inversely proportional to the distance metric between two files. α is the prior strength. Thus α_k becomes the prior probability of predicting file j given file i, based on the relationship between the two files. The posterior in our case would be given by

$$T_i \ (\alpha + N_i) \tag{5}$$

where $N_i = (N_{i1}, N_{i2}, ..., N_{in-1})$ is the vector that records the number of times we have transitioned out of state i to other states.

```
---O:0000002560---
O:0000002560 || T:1997/09/12-22:43:00 || U:/ || R:http://www.hyperreal.org/
O:0000002560 || T:1997/09/12-22:50:27 || U:/categories/software/ ||
R:http://www.hyperreal.org/music/machines/
O:0000002560 || T:1997/09/12-22:50:38 || U:/categories/software/Windows/ ||
R:http://www.hyperreal.org/music/machines/categories/software/
O:0000002560 || T:1997/09/12-22:50:47 ||
U:/categories/software/Windows/V909V03.TXT ||
R:http://www.hyperreal.org/music/machines/categories/software/Windows/
O:0000002560 || T:1997/09/12-22:51:06 || U:/categories/software/Windows/ ||
R:http://www.hyperreal.org/music/machines/categories/software/
O:0000002560 || T:1997/09/12-22:51:18 ||
U:/categories/software/Windows/ravemusc.txt ||
R:http://www.hyperreal.org/music/machines/categories/software/Windows/
```

Fig. 2. sample traces

C. Combination of submodels

The above two submodels II-A and II-B are combined as suggested in [CB95]. The combined model can be given by

$$P(j|i, \phi, \alpha\mathbf{m}) = m_i\lambda_i + (1 - \lambda_i)f_{j|i} \tag{6}$$

where

$$f_{j|i} = N_{j|i}/N_i \tag{7}$$

$$\lambda_i = \frac{\alpha}{\alpha + N_i} \tag{8}$$

The value of λ is not fixed manually but it gets set automatically based on the two transition matrices. In this model, we do not choose the prior randomly. Thus, we have an informative prior as against the one proposed in [CB95].

III. Evaluation

To evaluate our proposed model, we analyzed anonymized log traces provided by [Per03]. Sample trace can be found in figure 2. The log file contains accesses to music machines for a single day resulting in approximately 100 MB of data. The accesses are organized into paths. Paths are series of URLs requested from a particular machine. Paths also contain details about music files associated with accessed URLs. We extract information with respect to music files location and sequence in which files were accessed for our analysis. These logs do not distinguish among multiple subjects coming from the same source. It means that subjects accessing resources from same 'IP' address cannot be differentiated. Thus patterns were formulated with respect to specific 'IP' address and not subjects or in other words we consider each 'IP' address as subject and study its behavior. This assumption is reasonable to formulate patterns. However caching of pages at the site was disabled so that every page must be requested, even when revisited. This helps us boost our sequence count to enhance predictions based on bigram model.

Approximately 26000 entries were found in the log files. We separated this data into 2 sets viz. training set 90 percent and test set 10 percent. Data in the training set was used to build transition matrix for bigram model. Initially, all unique entries were found in the data. This number indicates the size of transition matrix. Next, we calculate the sequences in which files were accessed and update the transition matrix accordingly. For e.g., if path $'B'$ was accessed after path $'A'$, we update the count in the A^{th} row and B^{th} column of transition matrix. Eventually, each row in this transition matrix consists of total number of times any other file was accessed from the file represented by that row.

Unique entries in training sequence were also used to calculate separation of files in directory structure. This separation was associated with a distance metric. Closer the files in the directory structure, lesser is the distance between them. Files in subdirectories that are far apart from each other have greater distance metrics. A transition matrix was built to capture this second feature. Thus each row in this transition matrix consists of distances between a file with all possible files in the system. We believe that knowledge about distances gives us an informative dirichlet prior. Informative priors help in making better decisions when markov chains fail to predict efficiently. For the purpose of evaluation, the value of $'\alpha'$ required for dirichlet prior was chosen to be 30 as suggested in [SH05].

Test set was used to find the accuracy of our prediction. For each path in the test set, above two transition matrices were consulted to predict the next possible path. If there was popular sequence of event, bigram model helped in prediction. On the other hand, unseen events were predicted efficiently using the second feature on distance metric. We discuss the obtained results in the next section.

IV. Results

As mentioned before, paths from the test data were used to evaluate our model. We tested our predictions against the test set. We found how well our predictions fitted the test sequence. From equation 6, it is clear that value of λ lies between $0 - 1$. The value of λ can be interpreted as follows. Whenever a sequence (say '$A \rightarrow B$, where $'A'$ and $'B'$'are individual paths) is repeated several times in the training set, transition matrix formed by bigram model will have a higher count in the A^{th} row and B^{th} column. Thus the denominator in equation 8 will be much higher quantity than numerator resulting in smaller value of λ. In this case, bigram model provides sufficient evidence for speculation. On the other hand, when bigram model cannot find sequence in the history or sequence count is low, our algorithm depends on transition matrix built on distance metrics to make predictions. The value of λ is higher because of smaller value of 'sequence count'. In either case, our algorithm fetches 3 most likely files that could be accessed. We summarize initial results as follows:

1) When λ lies between 0 and 0.3, the prediction is approximately, 32 percent. This can be attributed to the higher sequence count in transition matrix of bigram model. Note that our algorithm gives priority to bigram models over distance metrics. Thus bigram models will have its dominance in this algorithm if the sequence count has a higher number even though distance between 2 files is comparatively small. To test the effectiveness of distance metrics on the algorithm, we ran our algorithms by nullifying the effect of bigram models. In this case, prediction solely depended on distance metrics. Results indicates predictions of 20 percent.

2) When λ is approximately 0.7, the prediction was 19 percent. There is a decrease in predictive capability of the model, but we believe that distance metric predicted events that could not be handled by bigram model.
3) When λ is 1, the prediction was 4 percent. This implies that distance metric solely contributed to 4 percent of prediction. Note that in this case, bigram model completely failed to predict the next access because the event wasn't seen before. On manual observation of some predictions, we found that files that were predicted were actually separated by large distances. This can be interpreted as follows. Files that are separated by smaller distances usually had a higher sequence count in transition matrix of bigram model. Our algorithm predicted such sequences from bigram model due to its dominance i.e. even though distance metrics contributed in prediction, its effect was nullified due to bigram model. When bigram models failed completely, distance metrics boosted prediction by additional 4 percent. Thus, this number can be considered as add-on to popular bigram model. When predictions cannot be handled by bigram model, distance metrics provide an additional 4 percent increase in prediction rate.

V. Related Work

To the best of our knowledge, the work on speculative authorization was first proposed back in 2005 [Bez05]. A technical report [Hil07] builds up in this direction. The preliminary results reported in this work demonstrates that prefetching future authorizations can improve the performance of the system. In this work, the author uses markov chains for speculations. In this work, results obtained for bigram model are same as that obtained by our model when bigram model dominates equation 8. As mentioned before, markov models can predict only those instances that have occurred in the past. Thus the model fails to predict unseen events. Markov models usually need sufficiently large traces to get trained and predict future authorizations. We tried to overcome these shortcomings in our work. Second feature that we choose tries to predict unseen events in our model. Apart from this, we have not encountered any research in the field of speculating authorization so far.

From distributed systems point of view, there exists a lot of literature in the area of predicting web pages [SYLZ00], [AKT08], [SH05], [DK04], [Pon06], [ZGY$^+$05], [Dav02]. This area of research can be related to our research on speculating authorizations. Having said so, there are differences between speculating web pages and policy decisions. Caches can store many more authorizations responses as compared to web pages because it requires very less memory. The challenge in speculating authorizations is to obtain those requests which are not present in the cache at any given instance but will be needed to satisfy the request of subjects at a future time. We list the conceptual and technical differences between the two below:

1) Usually, the number of web pages that a user can reach from a given web page is fixed. This possibility has been explored in [AKT08], [Pon06]. In our case, a subject can request for any object from the entire pool of objects in the system. This relatively increases the complexity of speculating authorizations because we need to consider all possible objects present in the system.
2) Sequence in which any user browses web pages receives much importance in predicting next access [DK04], [SYLZ00], [AKT08]. This feature is most commonly considered for two possible reasons. Given log traces, it is very easy to extract the sequence of accesses. Also, as mentioned in the first point, users can sequentially access web pages because of their interconnections. In case of speculating authorizations, we are more interested in knowing the future requests, not necessarily in particular sequence.
3) The concept of Support Vector Machines [BC00] has been used in [ZGY$^+$05], [AKT08]. This idea could be explored to evaluate its application in the field of speculative authorization in the future.
4) Davison [Dav02] presented a paper which shows that content inside a web page especially the one closer to any hyperlink could be used as a feature to predict the next web page. Extracting such features from objects i.e. files in our case, would increase the complexity of finding solution in finite time. Also, the paper assumes that exact structure and content of the web page is known before prediction. This assumption will not help in generalizing the model to other web pages.

VI. Discussion

Our goal in this project was to analyze patterns in PEP-PDP architecture that can help us speculate future authorizations. Features are extracted from log traces. Building markov chains from the available logs is most popular way of solving problems in prediction. We attribute this popularity to ease of getting information on sequences, once logs are available. Also, this is one of the most obvious features that could be extracted from any given logs.

Predictions based on markov chains has fundamental shortcomings:

1) To predict future accesses with confidence, one needs to have a large history of accesses. Confidence about future will depend on the number of times particular sequence occurred in the past. Markov chains fail to perform accurately when training sequence is short. Also, higher order markov chains cannot predict accuratey due to his reason [Hil07]
2) As mentioned before, markov chains can predict only those sequences that have occurred in the past if history of sequences is the only feature extracted from the logs.

To overcome these shortcomings, we tried to extract additional features from available log traces. Our preliminary work in this direction suggests that additional features help in predicting those events that have not been seen in the past. Our

algorithm depends on markov chains to predict sequences, but we found that extracting more features from the available logs improves prediction capability.

Our initial results cannot support the fact that additional features could take over markov chains and outperform prediction capability of markov chains. When we observe particular file, we extract 3 possible predictions based on distance metrics. Note that files in the same subdirectory have equal distance metrics. It is difficult to choose 3 best files from a set of files that are separated by the same distance. Other features would be needed to improve prediction capability. If we start fetching all files that are equidistant, cache will get overloaded. For testing purposes, we prefetched all possible files and predictions capability went up to 24 percent which is comparable to results obtained by bigram models. We are currently working on extracting other features that could result in grouping or clustering objects based on certain pattern that they exhibit.

VII. Conclusion and Future Work

In this paper, we present a model to speculate future authorizations in PEP-PDP paradigm. The model captured relationship between objects in addition to behavioral pattern exhibited by subjects. Initial results suggest that our model can predict those events that have not occurred in the past, which can be considered as our contribution in the field. We are exploring ideas to extract more features from log traces that form specific patterns which would improve predictive capability.

Acknowledgment

The author would like to thank Dr. Konstantin Beznosov and Dr. Kevin Murphy for their timely help. Also the author would like to thank 'LERSSE' members for their feedback from time to time.

References

[AKT08] Mamoun Awad, Latifur Khan, and Bhavani Thuraisingham. Predicting WWW surfing using multiple evidence combination. *VLDB Journal*, 13:401–417, 2008.

[BC00] Kristin Bennett and Colin Campbell. Support vector machines: hype or hallelujah? *ACM SIGKDD Explorations Newsletter*, 2:1–13, 2000.

[Bez05] Konstantin Beznosov. Flooding and recycling authorizations. In *Proceedings of the New Security Paradigms Workshop (NSPW'05)*, pages 67–72, Lake Arrowhead, CA, USA, 20-23 September 2005. ACM Press.

[CB95] MacKay D. J. C. and Peto L. C. B. A hierarchical dirichlet language model. *Natural Language Engineering*, 1(6):289–307, 1995.

[Dav02] Brian Davison. Predicting web actions from HTML content. In *Proc. of the Hypertext*, pages 159–167, 2002.

[DK04] M. Deshpande and G. Karypis. Selective Markov models for predicting web page accesses. *ACM transactions on Internet Technology*, 4(2):163–184, 2004.

[Hil07] Jeremy Hilliker. Speculative authorization. Technical Report LERSSE-TR-2007-01, LERSSE, Dept. of Elec. and Comp. Engineering, University of British Columbia, March 2007.

[Kar03] G. Karjoth. Access control with IBM Tivoli Access Manager. *ACM Transactions on Information and Systems Security*, 6(2):232–57, 2003.

[Per03] Mike Perkowitz. Adaptive web sites. http://www.cs.washington.edu/research/adaptive/download.html, 2003.

[Pon06] Alexander P. Pons. Object prefetching using semantic links. *SIGMIS Database*, 37(1):97–109, 2006.

[SH05] R. Sen and M. Hansen. Predicting web users' next access based on log data. *Journal of Computational and Graphical Statistics*, 12(1):1–13, 2005.

[SYLZ00] Zhong Su, Qiang Yang, Ye Lu, and Hongjiang Zhang. Whatnext: a prediction system for web requests using n-gramsequence models. In *Proc. of the First International Conference on Web Information Systems Engineering*, pages 214–221, 2000.

[ZGY+05] Zhili Zhang, Changgeng Guo, Shu Yu, DeYu Qi, and Songqian Long. Web prediction using online support vector machine. In *Proc. of the 17th IEEE International Conference on Tools with Artificial Intelligence(ICTAI'05)*, pages 214–221, 2005.