Inferring Dependencies Among Web Services with Predictive and Statistical Analysis of System Logs

Ethem Utku Aktas[1,2], Mehmet Cagri Calpur[1,2], Umit Ulkem Yildirim[1], and Emrah Yıldırım[1]

[1] Softtech A.S.
Research and Development Center
Tuzla Piri Reis Cad. 62, 34947 Istanbul, Turkey
[2] Sabanci University
Orta Mahalle, 34956 Tuzla, Istanbul, Turkey
{utku.aktas, cagri.calpur, umit.yildirim, emrah.yildirim}@softtech.com.tr

Abstract. Software system behaviour analysis is a challenging research problem in software engineering. The main reason for this is the lack of real data from large industrial systems. Softtech Inc. is a subsidiary of a large private bank in Turkey and this study is aimed to analyse mapping the services architecture and the software system health of a particular department at Softtech using specific software web service logs. The services that are the subject of this study consist of 196 web services related to credit and credit card application transactions from various channels. While these processes are related to similar applications, they call various web services that perform different operations in the background. Related services account for 2 million logs daily. We have conducted empirical and statistical analysis on the data, in order to infer the correlations and dependencies among the observed web services. Hypothetically, we think there are 3 types of dependencies between the web services. In our experiments, we used average response times and the number of times web services are called at specific time intervals as input data. The results suggest that they can be used for inferring that there is a dependency between two web services. In this preliminary work for dependency inference from unstructured web services' log data, we have utilized simple statistical analysis tools to derive important insight about the collection of services under our observation. The results have encouraged us to carry on with a more detailed analysis approach to further advance our research efforts.

Keywords: Statistical Analysis · Predictive Analytics · Dependency Inference · Log Analysis · Software Service Management · Software Reliability · Software Architecture

Sistem Kütüklerinin Tahminsel ve İstatistiksel Analizi ile Web Servisleri Arasındaki İlişkilerin Çıkarımı

Ethem Utku Aktaş[1,2], Mehmet Çağrı Çalpur[1,2], Ümit Ülkem Yıldırım[1], and Emrah Yıldırım[1]

[1] Softtech A.S.
Research and Development Center
Tuzla Piri Reis Cad. 62, 34947 Istanbul, Turkey
[2] Sabanci University
Orta Mahalle, 34956 Tuzla, Istanbul, Turkey
{utku.aktas, cagri.calpur, umit.yildirim, emrah.yildirim}@softtech.com.tr

Özet. Yazılım davranış analizi önemli yazılım mühendisliği konularındandır ve endüstriyel sistemlerden elde edilen gerçek veri eksikliği nedeniyle incelenmesi zor bir problemdir. Bu çalışmada Türkiye'nin en büyük özel bankalarından İş Bankası'nın yazılım geliştirme ve bilgi işlem alanlarında hizmet veren iştiraki Softtech A.Ş.'nin bir departmanının geliştirdiği ve yönettiği web servislerinin logları ile web servisleri arası bağımlılık analizi ve servis mimarisi çıkarımı yapmayı hedefliyoruz. Çalışmamıza konu olan servisler çeşitli kanallardan gelen kredi ve kredi kartı başvuru işlemleri ile ilgili 196 web servisinden oluşmaktadır. Bu işlemler benzer uygulamalarla ilgili olmakla birlikte, arka planda farklı işlemler yapan çeşitli web servisleri çağırmaktadırlar. İlgili servisler günlük 2 milyon kütük kaydı oluşturmaktadır. Elde ettiğimiz veriyi üreten servisler arasındaki korelasyon ilişkisi çıkarımlarında bulunabilmek için ampirik ve istatistiksel analiz yöntemleri kullanılmıştır. Varsayımsal olarak, web servisleri arasında 3 tür bağımlılık olduğunu düşünüyoruz. Deneylerimizde, belli zaman aralıklarındaki ortalama yanıt süreleri ve web servislerin kaç kere çağrıldıkları girdi olarak kullanılmıştır. Sonuçlar bu iki bilginin, web servisleri arasında bir bağımlılık olduğu sonucuna varmak için kullanılabileceği öngörüsünü ortaya koymaktadır. Bu öncü çalışmamızda istatistiksel analiz yöntemleri kullanarak topladığımız web servisi verisinden değerli bilgiler çıkarmayı başardık, bu sonuçlar bizi araştırmamızı daha ileri götürmek konusunda da teşvik edicidir.

Anahtar Kelimeler: İstatistiksel Analiz · Tahminleme Analitiği · Bağımlılık Çıkarımı · Kütük Analizi · Yazılım Servis Yönetimi · Yazılım Güvenilirliği · Yazılım Mimarisi

1 Introduction

Software systems evolve and as they evolve, their complexity increases. As a result it gets harder to monitor them. However, reliability of such systems is a key factor for customer satisfaction. One of the fundamental assumptions in monitoring such complex software systems is that there exist patterns in the behavior of executions. Many previous studies support this assumption [1-8].

Recent trends in software architecture such as microservices, magnifies the problem of maintaining large number of services. The maintenance problem is really complex for large companies that rely on hundreds or even thousands of web services. Documentation and records for ownership of the architecture may be neglected. In this study, we propose web service dependency inference techniques to map large scale software systems.

One of the problems in studying this research area is the lack of real data from such complex systems. Execution data such as system logs need to be collected and analysed in these systems and we conjecture that it is possible to find out dependency relations between services by observing the log data [1-8].

In this preliminary study, our aim is to conduct a statistical analysis using system logs of specific web services and find dependency relations among these services. First, the architecture of the system under observation is presented. Then, structure of the log data and the procedure to collect and preprocess this data is explained. The methodology for the analysis of the data and the results are given next. Finally, comments and conclusions on the study are presented.

2 Related Work

Lin and Siewiorek collected error log data from 13 file servers for a 22 month period [5]. They conducted a trend analysis and proposed a heuristic method to predict failures.

Hellerstein, Zhang and Shahabuddin proposed a predictive approach to detect the probability of a threshold value to be violated for a production web server and the occurrence time for the violation [2]. In addition to modelling the non-stationary behaviour of metrics, they model stationary, time-serial dependencies.

Weiss modelled a genetic based machine learning system to predict rare events by identifying predictive temporal and sequential patterns within data [8].

Vilalta et al. reported three case studies for failure prediction [6]: Long term prediction of performance variables such as disk utilization, short term prediction of abnormal behaviour such as threshold violations and short term prediction of system events such as router failure. As a result, they show that predictive algorithms perform successful results in the estimation of performance variables and prediction of critical events.

Vilalta and Ma described an approach to detect patterns in event sequences [7]. They aimed to predict the occurrence of target events such as computer attacks on host networks.

Levy and Chillarege conducted a case study to develop an approach for early warnings of failure [3]. One of their findings in this study showed that the overall counts of alarms rise which foretell an impending failure.

Liang et al. investigated the characteristics of failure events, the correlation between these events and non-fatal events [4].

Fu and Zu developed a model to cluster failure events based on their correlations and predict their future occurrences [1].

3 Breakdown of System and Service Architecture

The system architecture consists of four layers: Top layer is the user interface layer. Second layer is for load balancing the transaction requests from the interface layer (IIS layer). Third layer contains the web services processing requests. Final layer is the database layer (Figure 1).

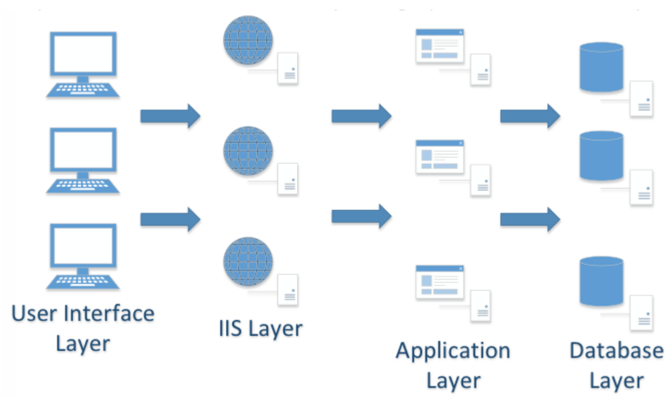

Fig. 1. General system architecture.

Tellers at branches send requests using Desktop/web applications or customers send requests via alternative distribution channels (such as internet, ATM, etc.).

3.1 Data Source Layer

The data is gathered from the round trip times of second layer and the third layer, in other words, load balancing layer and web-service layer. It includes the time stamp, physical server name (IIS server), name of the web-service called, response time and the contents of the request and the response.

4 Feature Engineering

Data preparation is the crucial part of statistics and machine learning processes. In the following section, the structure of the log data, how they are selected and pre-processed are explained.

4.1 Structure of Logs

The research data is a subset of the original log output that is gathered from the system. The information which had been used for our research was extracted from the original logs, which are web-service name, time stamp, number of calls, response time, system and business faults.

4.2 Feature Selection

The features for machine learning methods are prepared from the number of calls data of the logs for specific time intervals. The number of calls of a web service is an important indicator for showing the healthiness of the service. Increasing or decreasing number of calls may be indicating an upcoming problem in the system. Each row of the data set contains the number of calls for the same time slot of previous weeks, starting from 8 weeks prior to the prediction date. Since banking operations are mostly conducted on weekdays and the web services under observation are mostly used by the workforce in the main operations headquarters and branches, the dataset is formed using only weekday information.

The features for statistical analysis are also prepared by aggregating the log data. Both the number of calls and the response times for specific time intervals are used for statistical analysis. The number of times the web service is called at specific time intervals may show the dependency between the web services. For example, if the number of times a service is called increases and at the same time that of another web service also increases, this may be the result of a dependency between the services. The details of the statistical analysis are given section 5.2.

4.3 Preprocessing of Log Data

Log Data Aggregation. The amount of data generated for all the banking transactions would overwhelm any data processing system. The solution to this problem is by applying data aggregation techniques to the log data. Five minute periods of log data is aggregated into count of calls for that web-service, mean and standard deviation of the response time, and count of system faults and business faults.

If no service call occurs in that time period, no data aggregation can be done and so there would be missing data for those time periods. For such periods, data having zero counts, averages, etc. are inserted afterwards, so as to ensure the continuity of the time series.

5 Experiments and Results

The research consists of application of machine learning techniques on the aggregate log data to predict future response times and statistical analysis of the data for inferring the web service dependency mapping.

5.1 Predictive Analytics

Linear Regression, Support Vector Machines (SVM) and Random Forest techniques have been employed for the prediction of number of calls at specific time intervals from the aggregated data previously explained in *Feature Selection* section. Average error, which is the average percentage of the difference between the true value and the predicted number of calls of the web service for the related time period to the true value, is used as the performance metric. Random Forest application is the best performing machine learning technique, SVM come in second place with the average error value almost doubled and Linear Regression is the worst performing algorithm. The average error results are in the Table 1.

Algorithm	**Average Error (%)**
Random Forest	55.05
SVM	89.06
Linear Regression	114.37

Table 1. Average Error Values of number of calls for the prediction algorithms used in the research.

5.2 Statistical Analysis

The data consists of response times and number of calls for 196 web services, our aim is to uncover the relationship between these web services. In order to gain insight on the subject we have employed statistical methods and analysed correlations between the services' behaviour.

Empirical Analysis The purpose of empirical analysis of the data was to conduct fast analysis of the data to filter out obvious unrelated services. We have created pairs from each service and calculated Mean Percentage Similarity(MPS) for each pair. MPS values are calculated using the number of counts of the web services during specific time periods and these values are expected to show us the similarities between the web services. The error is based solely on web service activity, which means if there is a call for the web service there should be log data generated for that time window in our log files. Since the data is from one of the biggest banks and approximately 2 million log records are generated, there is a very high chance many of the services get calls in the same aggregate window. Therefore we have filtered the MPS values to search for 99% and above similarity for the empirical correlation observation. There are 196 pairs with 99% and higher MPS value, Table 2 shows some examples for the MPS based correlation. The evaluation shows that we can observe the dependency relations for these services as they represent the 3 dependency types that we have derived from the actual system.

Service	Observed Correlated Services	MPS
saveCreditCardApplicationDetail	saveCommunicationDetail	0.9999
	saveFinancialDetail	0.9995
	saveDocumentList	0.9995
commitApplication	saveCreditCardApplicationDetail	0.9966
	updateCommunicationInfo	0.9961
	saveFinancialDetail	0.9968

Table 2. Correlated web services sample with 99% MPS.

Correlation Analysis The second correlation type is based on the *Pearson Correlation*. The variables are the average response times for each aggregation window for the services under observation. In time series analysis, a correlated event may occur some time after the original event, such a delay is called lag and application of lag to the correlation analysis may yield interesting outcomes, which are normally hidden.

Pearson analysis was conducted on the data with 0 lag value. Because, in this work, the nature of our data collection strategy (aggregation of log data in 5 minute intervals) introduces a pseudo-lag property to our data set. The correlation is about the response times and the resulting *R values* suggest that both positive and negative correlations can be observed for the response times of the web services. We have filtered the Pearson Correlation results to observe only the strongest correlations with 95% or higher for positive correlation and negative correlation results.

Service	Observed Correlated Services	Pearson's R
saveCreditCardApplicationDetail	saveCommunicationDetail	0.9979
	saveFinancialDetail	0.9832
	saveDocumentList	Below 95%

Table 3. Pearson correlation samples.

Table 3 includes sample values out of 1020 correlation rules surpassing the 95% filter. Information in Table 2 and 3 summarizes the results and shows that empirical analysis produced rules are compatible to the Pearson Correlation.

Dependency Types Finding dependencies between web services is considered to be useful to find the possible sources of problems and the effected services when these problems occur. The architect of the systems-under-evaluation analysed the correlation rules and we have conjectured 3 types of dependency relations. The first type of relation is *Dependency on the Data Source*, where two unrelated services competing for the same database resource (Figure 2).

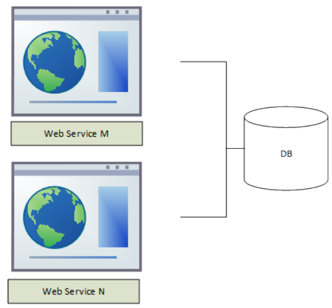

Fig. 2. Dependency on the Data Source.

The second type of relation is the *Hierarchical Dependency*, where a web service is called in another executing web service. Apart from the rest of the execution time for calling service, the response time is increased by the response time of the inner web service (Figure 3).

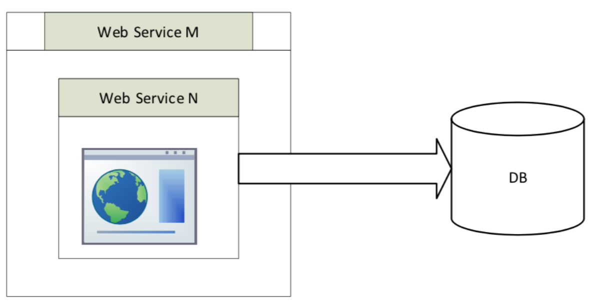

Fig. 3. Hierarchical Dependency.

The third type of relation is the *Serial Execution Dependency*, where a web service call follows another web service call in a procedure. We conjecture that negative Pearson Correlations of service calls occur based on this kind of dependency. If the first called web service becomes unresponsive and the response time increases, the second web service would not execute (Figure 4).

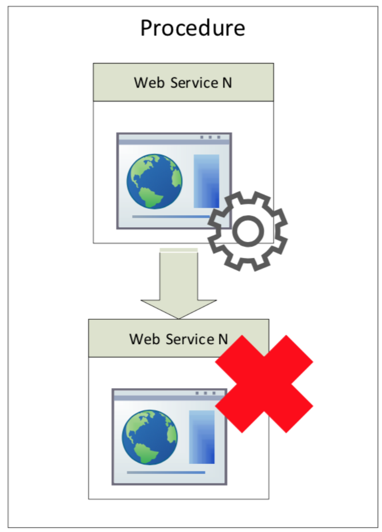

Fig. 4. Serial Execution Dependency.

The dependencies given above are the hypothesis that we think there are between the web services. And the selected correlation results of *R values* most possibly verify our hypothesis with our current knowledge of the related web services. However, the results should be verified by increasing the granularity of the data used.

6 Conclusion and Future Work

The results of the research demonstrates the inadequacy of the data aggregation method, because it is very likely that the real behaviour of the web services are lost. For further research effort, we will try to improve the granularity of the data to be able to re-apply machine learning methods to the response time predictions. Response time prediction and anomaly detection methods are very important for large organizations utilizing numerous system for their operations. Such research is important for companies with strict service level agreements and availability goals.

The statistical analysis methods generated some rules, which provides insight about the system-under-observation. By observing the system architecture, we think there are 3 types of dependencies between the web services. The results suggest that "average response times" and "the number of times web services are called at specific time intervals" can be used for inferring that there is a dependency between two web services. As a future research direction, we will use the correlation results as a foundation and we will conduct software system dependency mapping research.

References

1. Fu, S., Xu, C.Z.: Quantifying temporal and spatial correlation of failure events for proactive management. In: Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on. pp. 175–184. IEEE (2007)
2. Hellerstein, J.L., Zhang, F., Shahabuddin, P.: An approach to predictive detection for service management. In: Integrated Network Management, 1999. Distributed Management for the Networked Millennium. Proceedings of the Sixth IFIP/IEEE International Symposium on. pp. 309–322. IEEE (1999)
3. Levy, D., Chillarege, R.: Early warning of failures through alarm analysis a case study in telecom voice mail systems. In: Software Reliability Engineering, 2003. ISSRE 2003. 14th International Symposium on. pp. 271–280. IEEE (2003)
4. Liang, Y., Zhang, Y., Sivasubramaniam, A., Jette, M., Sahoo, R.: Bluegene/l failure analysis and prediction models. In: Dependable Systems and Networks, 2006. DSN 2006. International Conference on. pp. 425–434. IEEE (2006)
5. Lin, T.T., Siewiorek, D.P.: Error log analysis: statistical modeling and heuristic trend analysis. IEEE Transactions on Reliability **39**(4), 419–432 (1990)
6. Vilalta, R., Apte, C.V., Hellerstein, J.L., Ma, S., Weiss, S.M.: Predictive algorithms in the management of computer systems. IBM Systems Journal **41**(3), 461–474 (2002)
7. Vilalta, R., Ma, S.: Predicting rare events in temporal domains. In: null. p. 474. IEEE (2002)
8. Weiss, G.M.: Timeweaver: A genetic algorithm for identifying predictive patterns in sequences of events. In: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1. pp. 718–725. Morgan Kaufmann Publishers Inc. (1999)