Integrated Analytics for Application Management using Stream Clustering and Semantics

M. Omair Shafiq
School of Information Technology, Carleton University, Ottawa, ON, Canada

Keywords: Semantics, Streaming Clustering, Integrated Analytics, Application Execution and Management.

Abstract: Large-scale software applications produce enormous amount of execution data in the form of logs which makes it challenging for managing execution of such applications. There have been several semantically enhanced analytical solutions proposed for enhanced monitoring and management of software applications. In this paper, author proposes a customized semantic model for representing application execution, and a scalable stream clustering based processing solution. The stream clustering based approach acts as key to combine all the other analytical solutions using the proposed customized semantic model for logs. The proposed approach works in an integrated manner that clusters log data that is produced, as a result of events occurring during execution, at a large-scale and in a continuous streaming manner for managing execution of software applications. The proposed solution utilizes semantics for better expressiveness of log events, other related data and analytical approaches, through stream clustering based integrated approach, to process logs that helps in enhancing the process of monitoring and management of software applications. This paper presents the customized semantic logging model for scalable stream clustering, algorithm design and discussion on scalable stream clustering based solution and its integration with other analytical solutions. The paper also presents experimentation, evaluation and demonstrates applicability of the proposed solution.

1 INTRODUCTION

Building analytical solutions is challenging but making different analytical solutions work together is even more challenging. Several analytical solutions are proposed that focus on processing and analyzing data in a particular manner. Different analytical solutions may have different strengths and speciality in analyzing data and could be beneficial in different aspects. Some analytical solutions are better in discovering different hidden correlations among different features in data. Other analytical solutions are better in categorizing data based on different features. With large and complex systems to be analyzed, multiple analytical solutions are often built to analyze data in such system from different aspects. This brings another challenge in making all the analytical solutions work together in a meaningful and integrated manner.

For example, in an earlier work of the author, a hybrid solution of semantically formalized logging with advanced analytical solutions for enhanced monitoring and management of software applications (Shafiq, 2014b) was proposed. The proposed solution was built using semantic models to be able to formally describe components as well as events descriptions in execution logs of software applications. Analytical solutions were then built to effectively process such semantically formalized logs. In this way, information available with higher level of explicitness and expressiveness was better utilized. Data described formally and with higher level of expressivity makes it easier for the analytical solutions to process and analyze such data to be able to have monitoring and management of execution of software applications in an enhanced and effective manner.

There are several possible analytical solutions that can be integrated together in a meaningful way to perform deep and extensive analysis in a collective manner. However, in order to perform integration of different analytical solutions, inputs and outputs of different analytical solutions have to be matched syntactically and semantically. In this paper, author shows how previously proposed semantically enhanced analytical techniques can be

Shafiq, M.
Integrated Analytics for Application Management using Stream Clustering and Semantics.
DOI: 10.5220/0006334802800287
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 1*, pages 280-287
ISBN: 978-989-758-247-9

integrated together in a meaningful and effective manner.

In (Shafiq, 2015) an Association Rule Mining based approach was proposed. It is based on Semantic extension of FP-Growth algorithm for effective ranking and adaptation of Web Services. The approach was hybrid, i.e., partially using semantic annotations to Web Services combined with semantically adapted FP-Growth for Association Rule Mining allows the pre-processing of requests for searching Web Services. It helps in improving Web Service selection experience from performance as well as precision perspectives. This approach takes a set of log events as an input, and outputs a set of association rules.

In (Shafiq, 2014b), a hybrid approach for enhanced and automated monitoring and management of applications was built by using Semantics with Bayesian Classification. Semantics were used to formalize and structure logs from application execution which are then utilized by Bayesian Classification to classify different types of possible issues, with classification extended from (Friedman, 1997). It helped in reducing the size of problem space for system and application administrators to focus on the problematic part of application rather than the whole application, at the time errors of faults occur. This approach takes a given set of log events as input, uses its Bayesian classification based learning mechanism to deduce the system state of the system as output.

In (Shafiq, 2014a), a social network based solution with Semantic Logs to handle missing values and incomplete data during execution of applications. The proposed solution is based on semantically formalized logging (Shafiq, 2014b) for recording execution of applications and later-on using it to deduce possibly new or hidden information by analysing such logs. Key elements in logs were identified and correlations were modelled into a social network analysis hexagon. It was further shown that how such correlations between different key elements of semantic logs can be used to deduce new and non-obvious correlations between other elements of semantic logs and then utilize this information in monitoring and management of applications. This approach takes a set of log events and uses the proposed social network analysis based solution to deduce any hidden or missing correlations.

The proposed solution in this paper aims to show how semantic logs can plan an important role in integrating all the three techniques together in a meaningful manner.

The integrated analytics solution is also required to handle incoming events from logs as a stream. Such incoming events can be large in number, large in velocity and may also have different variety. This makes the events from logs to be of the scale of big data. Therefore, our proposed solution also includes a stream clustering based overall integration approach for different analytical approaches. There could be several other ways to perform integration of all of the components together. However, in order to keep the proposed integrated analytics solution open and generic, stream clustering has been chosen as the best candidate for following reasons. First, it allows processing of incoming event logs in the form of a stream. Second, it performs categorization of incoming log events into different categories (i.e., clusters) which can be then used by other analytical approach to perform further analysis. Third, it is an unsupervised leaning approach and does not require prior knowledge of data for clustering and hence that makes it a good candidate for acting as a broker-style interface to process incoming data and categorize it into different clusters and make it available for further processing by other analytical approaches.

The rest of the paper is organized as follows. Section 2 presents related work in the area of stream clustering and monitoring and management of software applications. Section 3 presents proposed solution of stream clustering on semantic logs for integrated analytics. Section 4 presents experiments and discusses evaluation of results as well as compares it with that of existing solutions. Section 5 presents conclusions followed by references.

2 RELATED WORK

A number of related works have been studied and analyzed that are carried out in the areas of clustering of logs for different types of software systems or software code management repositories or monitoring and management of software execution. These research works range from monitoring and analysis of stand-alone applications to large-scale applications with multiple components, middleware-based solutions and service based systems. Brief discussions and analyses on some of the interested and related approaches is described as follows.

In (Vaarandi, 2003), clustering of log events is proposed based on different features of events in logs. Different clustering algorithms (Hand, 2001) and (Berkhin, 2002) have been used to cluster log events into different categories. Authors categorize different lines in log files as different objects and then use clustering algorithms to cluster different lines into different clusters. After the clusters of event types are been identified, different analysis techniques are further used for detecting temporal associations between event types. A clustering tool called SLCT (Simple Logfile Clustering Tool) has been built based on these analyses techniques. However, limitation of this approach is that authors do not make any attempt to structure or formalize data in logs. The solution build be authors mostly relies on unstructured and almost not expressive data.

In (Makanju, 2008), authors use logs from a network management software and perform clustering in order to have a better and meaningful view for system and network administrators. Authors believe that clustering that allow system and network administrators to view faulty parts of log data easily rather than being overwhelmed with a large amount of log data and then having to manually find out faults. Large amounts of log data with a lot of different and irrelevant information may make process of monitoring difficult and may also cause unnecessary delays as well as inefficiencies. This work is also based on the Simple Log file Clustering Tool (SLCT) (Vaarandi, 2003) tool and a visualization tool has been further developed that can be used to view log files based on the clusters produced by the SLCT tool. Authors claim that results their solution further help in easing the summarization of a large amounts of data contained in the log files from network devices. The approach further helps in expediting analyses of events to detect any possible errors, faults or exceptions in networks. Drawbacks of this approach are the same as in previous approach, i.e., it is also based on using unstructured and almost not expressive data. This limits the approach in detection of different possible events (i.e., faults).

In (Beeferman, 2000), clustering is applied on log of queries for a search engine. Clustering is used to mine a collection of different and multiple user transactions over the search engine to discover clusters of similar queries as well as similar URLs. Identifying different queries from logs and then using clustering for different queries from the log, the authors claim that it enhances the process of web search. Clustering of different queries into different clusters in a meaningful manner helps in computing results faster for new queries that are similar to the queries that have already been recorded and categorized in clustering. This approach helps in enhancing the process of search but it is however limited to unstructured and raw log data (which is also sometimes referred to as click-through data). That limits the approach for detection and correlation of different events in terms of efficiency, accuracy and effectiveness.

In addition to the above-mentioned solutions, there are several other approaches that attempt to model data using semantics for the purposes of automating the process of Web Service discovery, composition and execution. Ontology Web Language for Services (OWL-S) (Paolucci, 2003), extended from DAML (Fensel, 2002), is considered as pioneer approach for semantically modelling web service description. It is based on OWL ontologies to describe different aspects of a web service to be known as Semantic Web Service (SWS) (SWSF, 2005). Web Service Modeling Framework (WSMF) (Fensel, 2002) is another similar and well-known approach proposed as a comprehensive framework to model different aspects of service consumers and service providers, known as Semantic Web Services (Roman, 2006). This approach is based on the principles of maximizing de-coupling between service consumers and service providers by providing mediation (Mocan, 2006), (Cimpian, 2005). The WSMF is realized by modelling ontology WSMO (Roman, 2006), description language WSML (de Bruijn, 2005), and execution environment WSMX (Recuerda, 2005), (Moran, 2004). Semantic Web Services Framework (SWSF) is another approach, having conceptual model as Semantic Web Service Ontology (SWSO) and language Semantic Web Service Language (SWSL). SWSO is based on three ontologies, i.e. service profile, model and grounding. It enables formal service descriptions and reasoning (Sirin, 2007) on Web Services. WSDL-S (Akkiraju, 2005) proposes a mechanism to enhance existing Web Services Description languages with semantics, in particular focusing on the services' functional descriptions. All these approaches attempted to formally describe Web Services descriptions or other relevant aspects, but none of these approaches attempt to formally represent or describe execution logs during execution of Web Services.

There are also several tools that attempt to process logs regardless of structures of such logs. Some of the tools are Adiscon LogAnalyzer (Adiscon, 2011), WebLog Expert (WebLog, 2016), GitHub Log-analyzer (Github, 2014), Retrospective Log Viewer Software (Retrospective, 2016) and XpoLog Log Analysis Platform (XpoLog, 2016). These tools were found to be applicable for currently available logging solutions. However, these tools were not found to be able to employ one or more analytical solutions to perform analysis in collective as well as meaningful manner.

To summarize the related work, most of the clustering solutions that have been reviewed so far either attempt to cluster logs that are not formalized and structures, or approaches like Semantic Web Service focused only on formalizing descriptions of web services and user requests. Such approaches do not specify issues related to processing of logs and especially having more than one analytical solutions analysing data in a collective and meaningful manner. This paper proposes to use semantic logs and stream clustering to allow different analytical techniques to analyse events in logs in integrated as well as meaningful manner.

3 PROPOSED SOLUTION

This section presents the proposed solution. The proposed solution is two-fold. First, employs stream clustering for processing of log events. Stream clustering was chosen because events are executed in applications in a stream like manner where logs are produced as event execution progresses in applications. However, employing stream clustering based solution was not straightforward. In case of large-scale applications, logs being produced are also large in scale. That means, incoming log events, especially from large-scale applications, can be large in number (volume), large in speed with which the log events are generated (velocity) and may also have different variety of log events. This fulfils the definition of big data. Therefore, the proposed solution should be able to handle log events, not only in streaming manner, but also in large-scale. For this purpose, BIRCH based stream clustering solution has been proposed.

3.1 BIRCH based Stream Clustering for Log Events

Logs are produced as events that occur while an application is being executed. The events are produced in a continuous and streaming manner. Therefore, it is important to be able to process such logs in a streaming manner. BIRCH (Zhang, 1997) based approach has been utilized to cluster log events, streaming during execution of an application, into different clusters. Events are categorized into different clusters using stream clustering. The categorization could be based on a particular category, status, component, functional, non-functional properties or any other application specific features. Clustering of logs based on data stream of events from logs is carried out by BIRCH approach as described in Table 1. BIRCH uses clustering feature (CF) which is based on number of data points (N), linear sum (LS) and squared sum (SS). Therefore, CF = {N, LS, SS}.

Table 1: Stream Clustering Algorithm for Log Events.

```
Inputs:
1. A set of n Log Events from
   Semantic Logs (LE1, LE2, LE3, …
   LEn).
2. An integer k for number of
   clusters to be formed.

Algorithm:
1. For n Log Events LE1 to LEn,
   compute clustering feature CF.
2. Build CF-Tree with a branching
   factor B and Threshold T using
   (Zhang, 1997).
3. Perform initial clustering using
   hierarchical clustering as in
   (Zhang, 1997).
4. Perform cluster refining by doing
   additional pass-overs over the
   data points and re-assigning
   points to closest centroids

Output:
1. K clusters with each cluster
   containing a set of Log Events
   belonging to that cluster as {C1,
   C2, C3 … Ci … Ck}.
2. Ci = {LE1, LE2 … LEx}
```

3.2 Clustering of Log Events from Semantic Logs

This section presents extended semantic logging model that is customized specifically for clustering of log events. The proposed extension of the semantic logging model based on the previous works of the author (Shafiq, 2014b) is shown in table 2. The semantic model encapsulates important and relevant information like global clustering solution, intermediate refined clustering solutions, centroids for different clusters and so on. Rest of the semantic logging model contains elements like different types of annotations including semantic and syntactic or simple annotations.

Table 2: Extended Semantic Logging Model for Stream Clustering.

Class GlobalClustering
hasCluster **type** Cluster
multiplicity = multi-valued

Class RefinementClustering
hasCluster **type** Cluster
multiplicity = multi-valued

Class Cluster
hasLogEvent **type** LogEvent
multiplicity = multi-valued
hasCentroid **type** LogEvent

Class SimpleAnnotation
attribute(s) as defined in (Shafiq, 2016).

Class SemanticAnnotation
attribute(s) as defined in (Shafiq, 2016).

Class Application
attribute(s) as defined in (Shafiq, 2014b).

Class Component
attribute(s) as defined in (Shafiq, 2014b).

Class Property
attribute(s) as defined in (Shafiq, 2014b).

Class Input
attribute(s) as defined in (Shafiq, 2014b).

Class Output
attribute(s) as defined in (Shafiq, 2014b).

Class LogEvent
attribute(s) as defined in (Shafiq, 2014b).

Class Context
attribute(s) as defined in (Shafiq, 2014b).

Class KeyValuePair
attribute(s) as defined in (Shafiq, 2014b).

Application element may contain one or more components. Components may have one or more properties which could be functional or non-functional. Components may also have inputs and outputs. Log Events may contain context and application specific data as key-value pairs. This model is inspired from W3C (W3C, 2001), Web Service Modeling Ontology (WSMO) (Roman, 2006) and then using Meta-Object Facility (MOF) (MOF, 2002) to be able to model data from logs and clustering in a standardized way.

Figure 1 depicts an overview on how stream clustering and semantic logging based solution can allow processing of log events into different clusters in a streaming manner, followed by performing further analytical solutions like association rule mining, classification and social network analysis (Shafiq, 2014a), (Shafiq, 2014b), (Shafiq, 2015).

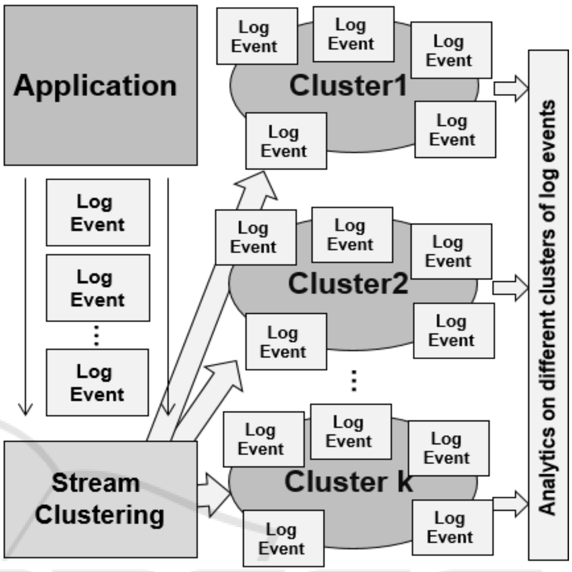

Figure 1: Overall architecture of integrated analytics based on stream clustering and semantic logs for application monitoring and management.

4 EVALUATION AND RESULTS

Stream clustering allows formation of clusters of incoming log events as a stream in a continuous manner. Once clusters area formed, other analytical solutions can be applied on data within each cluster to carry out further analysis. In this way, analysis can be carried out on data that is more relevant to a particular feature or situation based on which clustering is performed. For example, if the clustering is based on difference between a given event related to a particular user or a group of users, the further analytics could be performed on the cluster that contains data that is specific to that user of group of users. On the other hand, if clustering is based on state of an event being successful or fault in execution, in that case, further analytics can be performed on data within each cluster that may contain data specific or related to a particular state of event (i.e., successful or fault). This may boost analytics by providing more relevant data that is systematically categorized into different clusters.

Different experiments were performed on a use-case application that was used in building previously mentioned analytical approaches (Shafiq, 2014a), (Shafiq, 2014b), (Shafiq, 2015). The experiments were performed machine with Intel Core CPU 2.50 GHz, with 6 GB of RAM, and on Microsoft Windows 7, 64-bit operating system. The experiment was based on two key steps. First step was about performing classification on all the available log events at a given point in time. Second step was about performing classification on dataset that was categorized into different clusters by the stream clustering based approach for clustering log events into different categories. Two different types of measures were used to perform clustering. First measure was about clustering different log events based on users performing a transaction. Second measure was about clustering different log events based on event status (i.e., successful or failure).

Bayesian classifier was trained using a training dataset to classify problem types based on different features of log events. Such problem types could be including external communication issue, internal communication issue, database connection issue, external service connectivity issue, login failure, connection timeout issue, network down issue.

Experiment was initiated with comparing performance of Bayesian classifier trained on all the given data of log events to the Bayesian classifier that was trained on data categorized into different clusters based on the log events originating from different users. There were several advantages and disadvantages that were noticed about performance of the Bayesian classifier that was trained on clustered data. In-cases where significant number of log events were found to be categorized for a particular user, the classifier performed well. However, there were some cases for some users did not have significant amount of log events. In such cases, the Bayesian classifier was limited with data to be trained which had negative impact on performance of the classifier.

Table 3a presents a comparison of results from classifiers from data with or without using clustering based on log events originating from different users. For each of the cases, one-third of the data was used in training while rest of the data was used in experimentation and evaluation. An overall 2% increase in rounded Mean Average Precision was observed for classifier that was used for dataset without and with clustering having clusters based on different users. Note that in case of stream clustering for data categorized for different users, average of precision was calculated for results of classifier for data from all different users. Although an overall improvement was achieved in precision of classifier but it was noted that for some of the cases, precision was rather decreased. After examining the results, it was found out that for such cases, clustering of log events based on users results in having no or significantly less number of log events to be categorized that could be used by classifier for training purposes and hence resulted in a decrease in performance. Another lesson learned from this experiment was that it is important to decide the criteria for clustering. It is important that whichever criteria is chosen, should have reasonable distribution of log events into different clusters.

Table 3a: Comparison of Precision of Classifiers with or without Stream Clustering (based on different users).

Classified Problem Types	Precision without stream clustering	Average Precision with stream clustering
External Communication	0.81	0.80
Internal Communication	0.91	0.95
Database Connection	0.82	0.86
External Service	0.92	0.94
Login failure	0.72	0.87
Transaction Timeout	0.84	0.71
Network down	0.64	0.69

Table 3b: Comparison of Precision of Classifiers with or without Stream Clustering (based on different event statuses).

Classified Problem Types	Precision without stream clustering	Precision with stream clustering
External Communication	0.81	0.84
Internal Communication	0.91	0.96
Database Connection	0.82	0.89
External Service	0.92	0.87
Login failure	0.72	0.78
Transaction Timeout	0.84	0.84
Network down	0.64	0.71

Table 3b presents a comparison of results from classifiers from data with or without using clustering based on log events with different statuses. For each of the cases, one-third of the data was used in training while rest of the data was used in experimentation and evaluation. An overall 4% increase in rounded Mean Average Precision was observed for classifier that was used for dataset without and with clustering having clusters based on log events with different statuses (i.e., any type of faults and failures). In this experiment, an overall improvement was noticed in precision of classifier but it was further noted that for some of the problem types, precision was rather decreased where data was pre-processed using clustering. It was found that for such cases, clustering of log events based on statuses of log events resulted in some of the clusters to have no or significantly less number of log events to be categorized that could be used by classifier for training purposes. This caused a decrease in performance by classifier. Therefore, it is important that whichever criteria for clustering is chosen, the distribution of data into different clusters should be even as much as possible.

Comparing to the rest of the related work, most of the approaches were found to be either focusing only on formalizing or structuring data while other approaches focused only on using data analytics based techniques for mining unstructured data which could be used to monitoring and management of software applications. In addition to it, there has been very less emphasis on scientific work on building systematic ways on how different analytics techniques can be integrated together so that analytics could be performed in a combined and meaningful manner. Author believes that, just like Semantics have played a key role in data integration, in the similar manner, it can play an important and key role in integrated analytics. This paper demonstrates that once data (i.e., logs) are well structured and formalized using Semantic logging and analytics model, it makes it feasible to have multiple analytics based techniques to work together as integrated analytics. If the logs had not been structured and formalized, it would have almost not feasible or at least have been extremely difficult to integrate logs based on different features. Almost no approaches have been found to that could address the issues of monitoring and management of software applications using integrated analytics.

In addition to using semantics, unsupervised clustering of data stream is another important aspect of integrated analytics. First, it has the ability to process streaming data in a continuous manner. Second, BIRCH (Zhang, 1997) which has been utilized in this paper to perform stream clustering, is an unsupervised data mining approach which does not require prior labelling about data. Third, the steps of performing global clustering as an initial stream and then performing refinements in clustering in continuous manner makes this approach a suitable candidate to survive in big data platforms. All these properties will allow stream clustering to be able to continuous processing incoming data as stream, use multiple computation and data nodes to maintain and refine a global solution while performing local computations on different physical nodes, and hence cope up with large-scale data, i.e., big data.

5 CONCLUSIONS

This paper proposes an integrated analytics approach of using unsupervised stream clustering and semantics for enhanced and automated monitoring and management of applications. Semantics are used to formalize and structure logs from application execution which are then utilized by analytical approaches to process and identify different types of possible patterns and correlations. BIRCH algorithm (Zhang, 1997) was utilized for carrying out clustering of streaming log events in an unsupervised and continuous manner. It was found to significantly help in improving the process of application monitoring and management by letting different analytics techniques work together in an integrated manner. During the experimentation, there were a few key findings that were learned. First is criteria of clustering. It is crucial to decide the criteria for clustering so that events could be categorized into different clusters as even as possible. If a clustering criteria is chosen that causes uneven distribution of data into different clusters, it will also cause varying performance of analytics that will be performed on data within different clusters. This is because in some of the clusters there could be more data that help the analytical approaches to learn and identify interesting patterns while other clusters with limited or lesser data may cause analytical approaches to be limited in learning and identifying data within the clusters. Another important factor is use of semantics for logs and analytics that makes data to be structured and formalized. The structured and formalized data makes it feasible to have multiple analytics based techniques to work together as integrated analytics. Last but not least, unsupervised stream clustering in itself is another very important factor in enabling integrated analytics. It is because it can process streaming log data in a continuous manner, it is an unsupervised approach that does not require any previous knowledge or labelling of data and its ability to maintain a global and master clustering solution with refinements from different other intermediate solutions makes it feasible to be used for data that is large in scale, i.e., big data. Experimental evaluation shows how the combination of semantics and stream clustering made it even better to have efficient and effective application monitoring and management. Next steps are to use and adapt more data mining techniques to use semantically formalized data to further enhance application monitoring and management.

REFERENCES

W3C Semantic Web activity, 2001. W3C Recommendations on RDF and OWL. Available at http://www.w3.org/2001/sw

The Object Management Group, 2002. Meta-Object Facility, version 1.4, 2002. Available at http://www.omg.org/technology/documents/formal/mof.htm

Semantic Web Service Framework, 2005. SWSF version 1.0. SWSF Available from http://www.daml.org/services/swsf/1.0/, 2005.

de Bruijn, J., 2005. D16 WSML specification. WSMO Deliverable available at http://www.wsmo.org/TR/d16/, February 2005.

Mocan, A., Moran, M., Cimpian, E., Zaremba. M., 2006. Filling the gap – extending service oriented architectures with semantics. In IEEE International Conference on e-Business Engineering 2006 (ICEBE 2006), pages 594–601, Oct 2006.

Friedman, N., Geiger, D., Goldszmidt, M., 1997. Bayesian Network Classifiers, *Journal of Machine Learning*, vol. 29, pages 131-163, November 1997.

Vaarandi, R., 2003. A Data Clustering Algorithm for Mining Patterns From Event Logs, *2003 IEEE Workshop on IP Operations and Management (IPOM 2003)*, 1-3 October 2003, Kansas City, Missouri, USA.

Hand, D., Mannila, H., Smyth, P., 2001. Principles of Data Mining, *The MIT Press*, 2001.

Berkhin, P., "Survey of Clustering Data Mining Techniques", (see http://citeseer.nj.nec.com/berkhin02survey.html), 2002.

Makanju, A., Brooks, S., Nur Zincir-Heywood, A., Milios, E., 2008. LogView: Visualizing Event Log Clusters, *Conference on Privacy, Security and Trust (PST 2008)*, Fredericton, NB, Canada.

Vaarandi, R., 2003. A Data Clustering Algorithm for Mining Patterns From Event Logs, *IEEE Workshop on IP Operations and Management (IPOM 2003)*, 1-3 Oct 2003, Kansas City, Missouri, USA.

Beeferman, D., Berger, A., 2000. Agglomerative clustering of a search engine query log, *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 407-416, 20-23 August 2000, Boston, MA, USA.

Paolucci, M., Srinivasan, N., Sycara, K., Nishimura, T., 2003. Towards a Semantic Choreography of Web Services: from WSDL to DAML-S, in the proceedings of *International Conference on Web Services (ICWS 2003)*, June 2003.

Fensel, D., Bussler, C., 2002. The web service modeling framework WSMF. *Electronic Commerce Research and Applications*, pages 209–231, 2002.

Roman, D., Lausen, H. Keller, U., 2006. D2v1.3. Web Service Modelling Ontology (WSMO). Deliverable, http://www.wsmo.org/TR/d2/v1.3/, Oct 2006.

Cimpian E., Mocan, A., Scharffe, F., Scicluna, J., Stollberg, M., 2005. D29v0.1 WSMO Mediators, WSMO Final Draft, December 2005, Available at: http://www.wsmo.org/TR/d29/v0.1/

Martin-Recuerda, F., Sapkota, B., (eds.), 2005. WSMX Triple-Space Computing. Deliverable D21, 2005; available at: http://www.wsmo.org/TR/d21

Moran, M., Polleres, A., Kopecký, J., WSMX Grounding, WSMX Working draft D26v0.1, December 2004. Available at http://www.wsmo.org/2004/d26/v0.1

Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., Katz, Y., 2007. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, June 2007.

Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M., Sheth, A., Verma, K., 2005. Web Service Semantics – WSDL-S. Technical note Available from http://lsdis.cs.uga.edu/library/download/WSDL-S-V1.html, April 2005.

Adiscon LogAnalyzer, available at http://wiki.rsyslog.com/index.php/PhpLogCon

WebLog Expert Log Analyzer, available at http://www.weblogexpert.com

GitHub LogAnalyzer, available at https://github.com/wvanbergen/request-log-analyzer

Retrospective Log Viewer Software, available at www.retrospective.centeractive.com

XpoLog Log Analysis Platform, available at http://www.loganalysis.com

Zhang, T., Ramakrishnan, R., Livny, M., 1997. BIRCH: A New Data Clustering Algorithm and Its Applications, *Springer Journal on Data Mining and Knowledge Discovery*, Vol 1, Issue 1, pp 141-182, June 1997.

Shafiq, O., Alhajj, R., Rokne, J., 2014. Handling incomplete data using Semantic Logging based Social Network Analysis Hexagon for Effective Application Monitoring and Management, *International Conference on Advances in Social Networks Analysis and Mining (IEEE/ACM ASONAM 2014)*, August 2014, Beijing, China.

Shafiq, O., Alhajj, R., Rokne, J., Reducing Problem Space using Bayesian Classification on Semantic Logs for Enhanced Application Monitoring and Management, *Intl Conf on Cognitive Informatics and Cognitive Computing (IEEE ICCI-CC 2014)*, August 2014, London, UK.

Shafiq, O., Alhajj, R., Rokne, J., 2015. Reducing Search Space for Web Service Ranking using Semantic Logs and Semantic FP-Tree based Association Rule Mining, *9th IEEE Intl Conf on Semantic Computing (IEEE ICSC 2015)*, Feb 2015, Anaheim, CA, USA.

Shafiq, O. M., 2016. Event Segmentation using MapReduce based Big Data Clustering, *2016 IEEE International Conference on Big Data (IEEE BigData 2016)*, December 2016, Washington, DC, USA.