Task Oriented Data Exploration with Human-in-the-Loop. A Data Center Migration Use Case.

Alfredo Alba, Chad DeLuca, Anna Lisa Gentile, Daniel Gruhl,
Linda Kato, Chris Kau, Petar Ristoski, Steve Welch
aalba@us.ibm.com,delucac@us.ibm.com,annalisa.gentile@ibm.com,dgruhl@us.ibm.com,
kato@us.ibm.com,ckau@us.ibm.com,petar.ristoski@ibm.com,welchs@us.ibm.com
IBM Research Almaden, CA, US

ABSTRACT

Data exploration is a task that inherently requires high human interaction. The subject matter expert looks at the data to identify a hypothesis, potential questions, and where to look for answers in the data. Virtually all data exploration scenarios can benefit from a tight human-in-the-loop paradigm, where data can be visualized and reshaped, but also augmented with missing semantic information - that the subject matter expert can supplement *in itinere*. In this demo we show a novel graph-based data exploration model where the subject matter expert can annotate and maneuver the data to answer specific questions. This demo specifically focuses on the task of migrating data centers, logically and/or physically, where the subject matter expert needs to identify the function of each node - a server, a virtual machine, a printer, etc - in the data center, which is not necessarily directly available in the data and to be able to plan a safe switch-off and relocation of a cluster of nodes. We show how the novel human-in-the-loop data exploration and enrichment paradigm helps designing the data center migration plan.

ACM Reference Format:
Alfredo Alba, Chad DeLuca, Anna Lisa Gentile, Daniel Gruhl, Linda Kato, Chris Kau, Petar Ristoski, Steve Welch. 2019. Task Oriented Data Exploration with Human-in-the-Loop. A Data Center Migration Use Case.. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW '19 Companion), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3308560.3317306

1 INTRODUCTION

Understanding the functions implemented within a data center is an extremely challenging problem, due to countless machine re-configuration, software updates, changing software installation, failures, malevolent external attacks, etc. Being able to quickly and precisely characterize the nature, role, connections - which are often not explicitly declared - of the multitude of nodes in a data center becomes paramount when planning data center migrations.

Migrating data centers - either physically or moving applications to the cloud - is a time and resource intensive task. Preparing a migration plan involves intensive data analysis, often based on the logs and the network activities of each node in the data center. Discovering and understanding connections and dependencies can be very laborious and missing any of them can result in unplanned failures during the migration. Traditional data analysis tools offer little support during the plan-making phase, which can take many man hours.

In this work we propose a data exploration solution that allows the subject matter expert to interactively augment the data with structured knowledge and semantic information which is not initially present in the data. We combine traditional Information Extraction techniques together with human-in-the-loop learning to construct a semantic representation of the functions provided by the data center.

The contribution of this work is twofold. First, we propose a novel technique to extract semantic knowledge about nodes or clusters of nodes in the data center. While available structured knowledge about data center nodes - processes numbers, port numbers, IP addresses... - is readily available, semantic knowledge about each node is not formally encoded. We propose the use of logs from nodes in the data center to extract semantic information about the running processes. The knowledge extraction is performed with a human-in-the-loop model: we identify repeating patterns in the logs and ask a subject matter expert to label them (e.g. a certain log might indicate that the node hosts a "database"), we then generate regular expressions to label similar processes (i.e. similar logs) accordingly. The operation is repeated iteratively until the subject matter expert is satisfied with the label coverage. Second we leverage the added semantic knowledge, together with all other already available information, within a visual discovery framework to support data center migration planning.

In this demo we will show how to explore the functions of a data center using our novel exploration tool. We will demonstrate the value provided by the semantic knowledge added interactively. During the live demo we will showcase the use of the tool to answer questions such as “Find all nodes running a database application" followed by “Find all webservers connecting to those databases".

The main advantage of our solution is that it enables subject matter experts to quickly explore, characterize and augment complex data. Specifically in this case information about data centers, where we help the subject matter experts to quickly combine (i) information in the nodes' logs together with (ii) iteratively added knowledge that is not available from the logs to create a cartography of the data center.

In the following we will give an account of available research on data exploration, specifically focusing on data center migrations (Section 2) and then describe the specific use case that will be

WWW '19 Companion, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6675-5/19/05.
https://doi.org/10.1145/3308560.3317306

demonstrated during the live demo (Section 3). In Section 4 we depict conclusions of this work and our planned future work.

2 STATE OF THE ART

The Knowledge Discovery pipeline proposed by Fayyad et al. [7] has been widely accepted and most of the existing data analysis systems comply to its principles. Knowledge discovery is the process of identifying unsuspected relationships and summarizing the data in novel ways that are both understandable and useful to the data owner [9]. The knowledge discovery process usually comprises five steps, i.e. data selection, data processing, data transformation, data analysis, and evaluation and interpretation of the discoveries.

In the case of knowledge discovery for data center exploration, the most abundant type of available data is in the form of machine logs. Therefore, a great deal of work has been devoted to adapt the steps of the pipeline to this specific format. The common ground for knowledge discovery from log-like data is the usage of rules - in the form of regular expressions, filters, etc. Lemoudden and El Ouahidi [11] use tokenization, regular expressions, dictionaries and timestamp filter. Similarly, we use regular expressions and dictionaries, but rather than using a pre-constructed set of filters, we iteratively construct those during data exploration with a human-in-the-loop model. The timestamp is a feature that is also used in many other works analyzing logs [2, 10]. Nonetheless, these works are focused on the identification of user sessions, e.g., in Web based search [10] or within e-commerce frameworks [2]. While the temporal component is paramount for certain analysis tasks (e.g. discovering cyber-attacks in data centers), it is not a key component to depict a cartography of the data center, i.e., for understanding the functionalities provided by each node (or clusters of nodes) - and in this scenario we find that helping the user to semantically tag the nodes leads to better insights. In other words, understanding the nature and context of a particular event is far more valuable to our use case than understanding exactly when that event occurred.

In terms of generating a semantic view of data centers, there are several works in the state of the art, many of which are ontology-based solutions where the logs are aligned to a specific available knowledge representation of the data center [4, 6, 12, 13, 15]. While in this work we also create a semantic representation of the data, we do not assume the existence of any target knowledge, but we let the semantic representation emerge from the human interaction: each time the subject matter expert adds semantic tags to characterize the logs, we collect and organize them in a growing taxonomy. The work by Mavlyutov et al. [14] is similar in this sense of not using any pre-existing target knowledge, but letting the semantic representation emerge from the data. The proposed Dependency-Driven Analytics (DDA) infers a compact dependency graph from the logs, which constitutes a high level view of the data that facilitates the exploration. Similarly, we generate high level, semantic views of the data center, but we do so by adding semantic tags to logs, with a human-in-the-loop paradigm rather than solely relying on parsing.

Last but not least, logs are not the only source of information available in data centers. A plethora of sources of data are available, including network data, configuration management, databases, data from monitoring devices and appliances, etc., all of which can be leveraged to generate a model of the data center [3, 5, 8]. While we use all this data, the major novelty of our work is that we combine the views derived from the data with the semantic tags iteratively added by the human exploring the data.

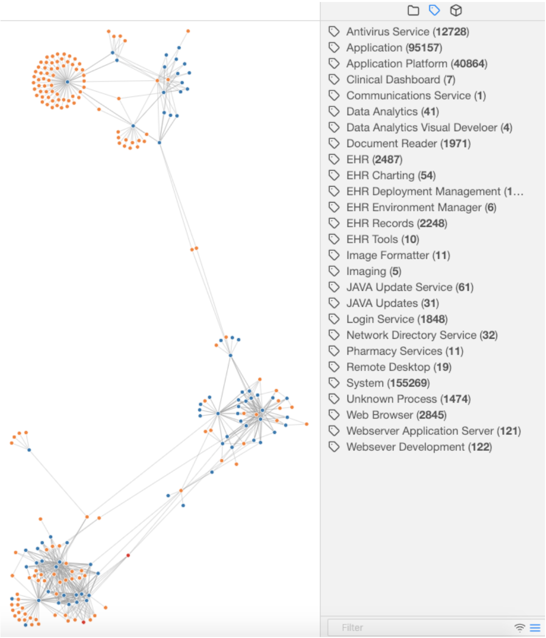

Figure 1: shows all nodes in the data center running a specific software (i.e. EHR) and all dependencies involving these nodes classified as being of type "EHR"

3 USE CASE: DATA CENTER EXPLORATION

During the demo we will showcase how we use the system to transition from raw data, to semantically enriched data, to full graph visualization and exploration of the data center.

3.1 Creating a Data Center Cartography

As a starting point we use all network information collected from the data center and build a color coded representation of all the nodes, as well as their incoming and outgoing connections. The created graph is (i) too big to be effectively visualized and (ii) does not contain crucial information about which processes are running on each node, to be able to design a migration plan. Therefore we collect information on running processes from all the nodes to try and characterize the nature of dependencies between each machine. The task is to effectively extract entities from each process log, where the entities of interest are the processes running on the machine. There are numerous hurdles to perform this task. First, the nature of the logs is single lines with a command which has been run on the machine and potentially a number of parameters. The format of the string is highly dependent on the operating system and on the specific command, but it is not a regular natural language sentence - therefore state of the art entity extraction tools fail. Regular expressions are helpful with identifying the meaningful parts of the process logs and matching them - when possible - to a list of candidate processes or applications. Nonetheless, not all processes are known a priori, therefore leaving us with numerous entities

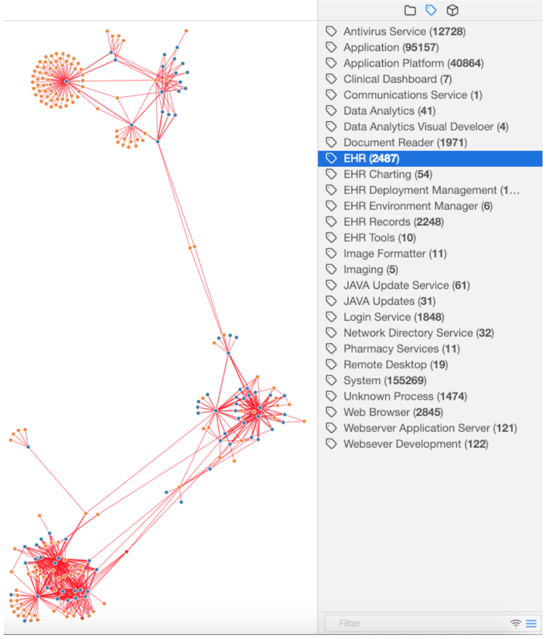

Figure 2: Example of selecting the tag "EHR" to highlight all dependencies that have been classified as being of type "EHR". All dependencies are highlighted in red.

which are nearly impossible to automatically resolve (the entities extracted from the logs can have obscure names, e.g. “xyz.exe”). We adapt our human-in-the-loop approach for corpus analysis [1] and perform the following approach: (i) we group logs that share the same entities and identify helpful clues in the parameters or in the directory paths (ii) we show these to the subject matter expert who can add a label if she recognizes which application is running. We then create a rule exploiting this knowledge and apply it to the whole graph. As the user keeps exploring and annotating the data, we dynamically add all the new tags as exploring dimensions, which can be immediately used to query and visualize the data. During the live demo we will show how we generate the graph from raw data and the effect of adding additional semantic tags - via the human-in-the-loop data enrichment approach - to the graph.

3.2 Exploring tagged dependencies

After the subject matter expert has interacted with the system, creating and enriching the cartography of the data center, the next step is to perform in-depth analysis. Specifically, when considering the task of planning the data center migration, one of the paramount questions to answer is to identify nodes that support the same (or dependent) processes, or as we refer to them in the following, “affinity groups". Finding a set of nodes that form an affinity group, followed by understanding the nature of the dependencies between nodes in the group is both valuable and challenging. An example of an affinity group may be a group of servers involved in providing and maintaining Electronic Health Records (EHR) from an EHR application. Figure 1 shows the initial step of surfacing an affinity group related to EHR. Each node is serving and/or consuming a service identified as EHR. Where nodes are connected by lines, we can say the nature of the dependencies between those nodes includes EHR (and likely many more classification as well). To emphasize this, Figure 2 shows the same cluster when the user selects EHR from the list on the right side of the screen, highlighting all dependencies involving EHR. This shows all tags involved in the current visualization, but given that the original query asked for a EHR affinity group, it follows that every dependency would be highlighted in red. To drill down further, a user can interactively select the other tags represented in the visualization to surface the dependencies where those tags are also involved. As Figure 3 shows, selecting "Network Directory Service" highlights a subset of the visualized dependencies. By visualizing the dependencies, tightly coupled groups are easily identified by the human eye. In the case of "Network Directory Service", one can immediately see there are 2 main nuclei, each with a set of non-overlapping dependent nodes. This sort of visualization also exposes insights that may not have been directly queried. The vast majority of business critical application groups will include a resiliency layer to provide redundancy, data backups, etc. However, the list of categories included in the EHR cluster does not include "Resiliency"or "Backup". To a user familiar with dependency grouping, this anomaly is striking. In fact, we learned on a call with the client that this particular cluster suffered an outage the week before, without any disaster recovery in place, effectively taking the whole business application offline.

4 CONCLUSIONS AND FUTURE WORK

Data exploration is a task that inherently requires a tight human interaction. The more complex the data and the scenarios, the more the need of a carefully designed methodology to support a human-in-the-loop paradigm. In this work we explored a particularly complex data exploration task, in the context of data centers, with the specific goal of supporting the subject matter expert to design a migration plan. Migrating a data center is a difficult and very critical task faced by many enterprises looking to increase flexibility, reduce costs, and enhance resiliency. The process requires a deep understanding of the dependencies between nodes, as well as the relations between larger clusters of nodes, either to each other and/or with other shared resources (eg. data stores, APIs, etc). Typically designing a migration plan can take 3-6 months, followed by 12-18 months for execution. Additionally, inevitable “missteps" along the way can cost substantial amounts of time and effort to resolve, along with the larger risk of business application outages. Our approach enables subject matter experts to obtain the necessary understanding of a data center in a faster and more accurate fashion. The tool has been used internally to help the subject matter experts in the design of migration plans, which has resulted in a positive reduction of design time and a significant reduction of mistakes/misunderstandings. The subject matter expert is provided with a clearer, more precise picture of the world they are working in, such that the job of planning and executing a smooth data center migration is far more realizable.

Future work for us focuses not only on reducing this “time to understanding”, but also assisting the practitioner in considering more complex parallel moves of functionality to the cloud, potentially reducing the migration time ever further. Given the nature

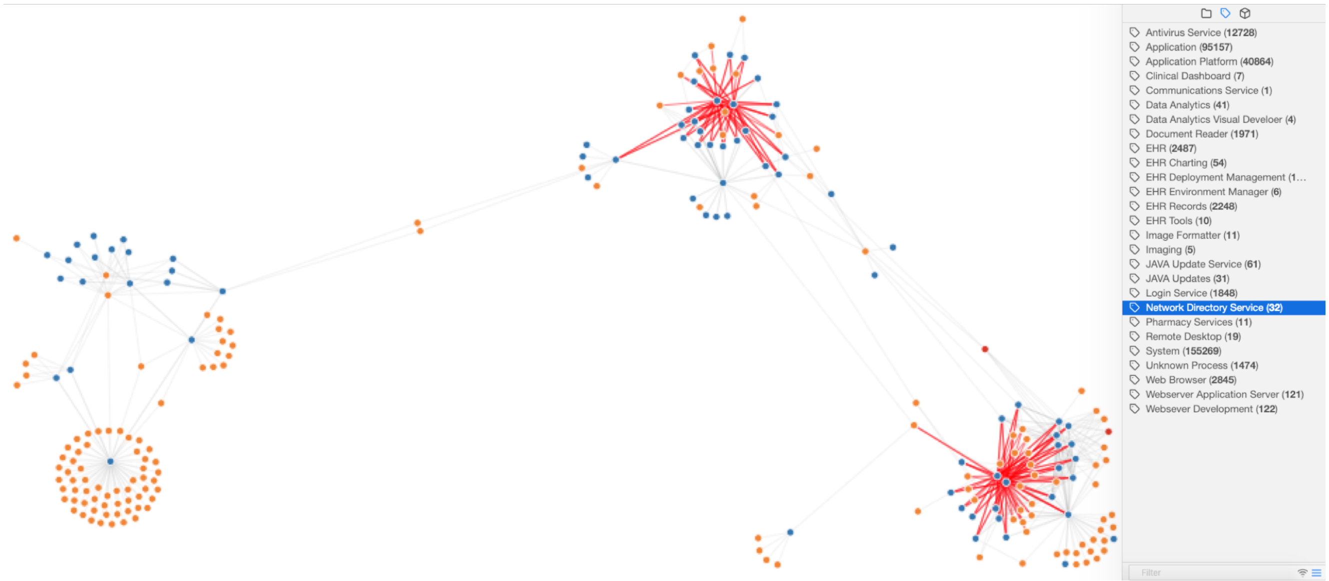

Figure 3: Example of selecting the tag "Network Directory Service" to highlight all dependencies that have been classified as being of type "Network Directory Service", a subset of dependencies are highlighted in red. The list of tags on the right shows only tags used in the visualization (i.e. the query results).

of our data model, there are opportunities for more advanced techniques (graph analysis and machine learning) to assist the human in annotating data. Automatic cluster suggestions using graph analysis is an area of active exploration. With the dependencies and annotations (tags) already in place, representing the data as a graph is a natural fit. Further analysis has the possibility of deepening our understanding of the relationships between nodes in such a way that rough identification of node clusters may become viable. This would involve starting with the nuclei and expanding outwards, only following paths of importance above a human-defined threshold until the analysis identifies a likely stopping point, thus firming the outer boundary of the cluster. While graph analysis looks at the nature of dependencies and relies heavily on tags, there are likely additional machine learning techniques that can be employed to enhance the tagging process, while keeping the human in the loop. This is an active area of exploration and will potentially speed up the tagging process significantly, but more importantly, broaden the set of entities with tags applied, thus providing a richer dataset to analyze.

REFERENCES

[1] Alfredo Alba, Anni Coden, Anna Lisa Gentile, Daniel Gruhl, Petar Ristoski, and Steve Welch. 2017. Multi-lingual Concept Extraction with Linked Data and Human-in-the-Loop. In *Proceedings of the Knowledge Capture Conference, K-CAP 2017, Austin, TX, USA, December 4-6, 2017*, Óscar Corcho, Krzysztof Janowicz, Giuseppe Rizzo, Ilaria Tiddi, and Daniel Garijo (Eds.). ACM, 24:1–24:8. https://doi.org/10.1145/3148011.3148021

[2] Mahmoud Awad and Daniel A Menasc. 2015. Automatic Workload Characterization Using System Log Analysis. In *Computer Measurement Group Conf.*

[3] Zakaria Benzadri, Faiza Belala, and Chafia Bouanaka. 2013. Towards a formal model for cloud computing. In *International Conference on Service-Oriented Computing*. Springer, 381–393.

[4] David Bernstein, Santa Clara, Nipoma Court, and David Bernstein. 2010. Using Semantic Web Ontology for Intercloud Directories and Exchanges 2330 Central Expressway Using Semantic Web Ontology for Intercloud Directories and Exchanges 2330 Central Expressway. (2010).

[5] Robert H Bourdeau and Betty HC Cheng. 1995. A formal semantics for object model diagrams. *IEEE Transactions on Software Engineering* 21, 10 (1995), 799–821.

[6] Yu Deng, Ronnie Sarkar, Harigovind Ramasamy, Rafah Hosn, and Ruchi Mahindru. 2013. An Ontology-Based Framework for Model-Driven Analysis of Situations in Data Centers. (2013). https://doi.org/10.1109/SCC.2013.98

[7] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. 1996. From data mining to knowledge discovery in databases. *AI magazine* 17, 3 (1996), 37.

[8] Tyrone Grandison, E Michael Maximilien, Sean Thorpe, and Alfredo Alba. 2010. Towards a formal definition of a computing cloud. In *Services (services-1), 2010 6th World Congress on.* IEEE, 191–192.

[9] David J Hand. 2007. Principles of data mining. *Drug safety* 30, 7 (2007), 621–622.

[10] Yongyao Jiang, Yun Li, Chaowei Yang, Edward M Armstrong, Thomas Huang, and David Moroni. 2016. Reconstructing Sessions from Data Discovery and Access Logs to Build a Semantic Knowledge Base for Improving Data Discovery. *ISPRS Int. J. Geo-Inf* 5, 5 (2016). https://doi.org/10.3390/ijgi5050054

[11] Mouad Lemoudden and Bouabid El Ouahidi. 2015. Managing Cloud-generated Logs Using Big Data Technologies. In *International Conference on Wireless Networks and Mobile Communications (WINCOM).*

[12] Li Liao, Yuzhong Qu, and Hareton K N Leung. 2005. A Software Process Ontology and Its Application. In *ISWC2005 Workshop on Semantic Web Enabled Software Engineering*. 1–10.

[13] K Magoutis, M Devarakonda, N Joukov, and N G Vogl. 2008. Galapagos : Model-driven discovery of end-to-end application âĂŞ storage relationships in distributed systems. *IBM J. REs & DEV* 52, 4 (2008), 367–377.

[14] Ruslan Mavlyutov, Carlo Curino, Boris Asipov, and Philippe Cudre-mauroux. 2017. Dependency-Driven Analytics : a Compass for Uncharted Data Oceans. In *8th Biennial Conference on Innovative Data Systems Research (CIDR âĂŸ17).*

[15] Lamia Youseff, Maria Butrico, and Dilma Da Silva. 2008. Toward a Unified Ontology of Cloud Computing. In *Grid Computing Environments Workshop, 2008. GCE '08.*