Usage Analytics: A Process to Extract and Analyse Usage Data to Understand User Behaviour in Cloud

Manoj Kesavulu(✉), Duc-Tien Dang-Nguyen(✉), Marija Bezbradica(✉), and Markus Helfert

Lero, School of Computing, Dublin City University, Dublin, Ireland
manoj.kesavulu2@mail.dcu.ie,
{duc-tien.dang-nguyen,marija.bezbradica,markus.helfert}@dcu.ie

Abstract. Usage in the software field deals with knowledge about how end-users use the application and how the application responds to the users' action. Understanding usage data can help developers optimise the application development process by prioritising the resources such as time, cost and man power on features of the application which are critical for the user. However, in a complex cloud computing environment, the process of extracting and analysing usage data is difficult since the usage data is spread across various front-end interfaces and back-end underlying infrastructural components of the cloud that host the application and are of different types and formats. In this paper, we propose usage analytics, a process to extract and analyse usage to understand the behavioural usage patterns of the user with the aim to identify features critical to user. We demonstrate how to identify the features in a cloud based application, how to extract and analyse the usage data to understand the user behaviour.

Keywords: Usage data · Data extraction · Analytics · Application · Features · Cloud · User behaviour and usage pattern

1 Introduction

Cloud computing is an emerging paradigm, bringing many advantages to both users and services providers while making dynamic platforms, cost-effective, flexible, on-demand resource provisioning, and many others. Consequently, there are significantly increasing developments for cloud computing infrastructures and platforms, from large high-tech enterprises such as Microsoft (with Microsoft Azure), Amazon (with Amazon EC2), and IBM (with IBM Cloud), to name a few. According to Forbes, by the end of 2018, spending on IT-as-a-Service for data centers, software and services will be $547B[1]. The number of Cloud-based services has increased rapidly and strongly, offering various advantages

[1] https://www.forbes.com/sites/louiscolumbus/2017/04/29/roundup-of-cloud-computing-forecasts-2017/#7155e14e31e8.

A. Holzinger et al. (Eds.): CHIRA 2017, CCIS 654, pp. 109–124, 2019.
https://doi.org/10.1007/978-3-030-32965-5_6

over traditional software including reducing time to benefit, scalability, accessing through various interfaces and so on. Unfortunately, difficulties in understanding the way different users using the applications and the provided resources for the cloud services still exist. Understanding usage data of an application has various uses such as to personalise the application according to the end-user's preferences [14], profiling users for security [1], improvement in marketing of software products [3] and to analyse the performance of the application in the deployed environment for maintenance purposes [2,12]. Normally, cloud providers and users use several monitoring and analytics tools to ensure the cloud services and the applications deployed on them execute as intended. Traditionally, the cloud provider (vendor) provides application performance management tools to monitor the status of the deployed applications. These tools provide mainly a vast amount of usage data of the resources used which can be turn into some knowledge for resource provisioning or error diagnostics in the cloud systems at the infrastructure and application level [8]. However, it is non-trivial to obtain user-related information, for example, user's behaviour, user's usage pattern, which features of the application are critical for a user, from the data extracted using existing APM tools. In order to analyse these information, advanced data analytics on extracted usage data is required [6].

In this study, we present a novel usage analytics process, explain and demonstrate how to extract and analyse usage in cloud services and applications that can help reveal behavioural indicators. Extraction of usage data of the features provided by the cloud applications could help software developers and architects to make an informed decision for the development/improvement of functionalities of the system according to end-user usage patterns. Typically, usage data consists of different types and formats of data extracted from multiple sources. Analytical solutions refer to the use of various analysis techniques and methods such as data mining, machine learning, reasoning, and other methods to extract useful knowledge and insights from large data set. For example, a company can use analysis techniques to understand customers' behaviour and predict how they are engaged or which customers are least likely to quit. These insights can be discovered via customers' profiles, memberships they subscribe to, or their generated content (comments, clicks, and other interactions). Developers can understand if some functions do not work properly via the usage data generated by the actions performed by the user with the application. User interests can be modelled by extracting browsing behaviour when accessing web application [7]. Such analytical solutions are considered as increasingly critical tools for modern enterprise to get an informational advantage, and have evolved from a matter of choice to a fundamental requirement in the present competitive business environments. Applying these solutions, thus, is a key to discover insights from the applications' usage. Every user has their own pattern when using an application or a service. Understanding these patterns could help to improve the service or discover the trends in advance. These patterns, can be discovered from the usage data.

In our previous work, the criteria for the usage data are defined and analyse the existing usage data extraction techniques according to the defined criteria and propose a usage data extraction framework [9]. We proposed **Usage Analytics**, a set of potential novel solutions that could help tackle various challenges in the cloud domain. We provided an overview of usage analytics in the cloud environment and proposed how to discover insights using these analytics solutions [6]. In this paper, we present the usage analytics process and provide the techniques for the key activities of the process and demonstrate using an experiment conducted.

Consequently, the aims of this paper are:

– To discuss the challenges in understanding the user behaviour in a cloud environment.
– To provide an overview of what is usage analytics in the cloud environment and demonstrate the key activities of the usage analytics process;
– To provide techniques that can be used to identify, extract and analyse the usage data.

The remainder of the paper is structured as follows: in section Sect. 2 we discuss the challenges in understanding user behaviour in cloud and discuss the concept of usage data with the focus on cloud applications. In Sect. 3, We present the Usage Analytics Process. In Sect. 4, we implement the usage analytics process explaining the key activities of the process with the aim to understand the user behaviour. In Sect. 5, we discuss the conclusions drawn and directions for the future work.

2 Related Work

User behaviour in the context of this research is defined as "the set of aggregated actions performed by user with the cloud application". Several studies exist on monitoring and analysing user behaviour for different purposes in the information systems domain. In [11], the authors provide a tool-kit that exploits the hardware sensors and software capabilities of contemporary mobile devices like PDAs and smartphones to capture objective data about human behaviour and social context together with objective data about application usage and highly subjective data about user experience. In [4], the authors show the importance of users' attitude towards the website and attitude towards the internet in explaining attitude towards the brand and consumer behaviour. Web mining techniques help in understanding the user access patterns in websites with the aim to recommend relevant topics and their placement on the website pages [13]. Understanding the changes in user behaviour can also help to make Application programming logic adaptive [5], the authors present and discuss a logic based approach for automatically learning and updating models for users from their observed behaviour. It is worth noting that these usage data potentially can be exploited in situation-emotional analytics [10], which aims at recognizing the emotions and changes of software situations in order to improve the quality and user experience levels. These emotional information are now extracted via external biometric recording devices, e.g., recording devices that record the eye and gaze-tracking signal. We firmly believe that, usage information at the application levels, will be very useful for this type of learning and potentially can replace eye and gaze-tracking information.

User behaviour can be understood by analysing usage data and can various data sources in cloud, and the major challenge is they can be in any form and format, which brings many challenges for analysis. The main questions for usage data extraction is what usage data should be extracted and how to map the raw usage data with the right applications or services. Considering the multi-tenant architecture of the cloud, different applications share the same physical and virtual resources. This raises challenge as in how to separate and extract the logs that represent each application from the instance (VM) co-hosting the applications. Another important challenge is handling with different contextual information. A system usually has a lot of branches, and thus the systems behaviors may be quite different under different input data or environmental conditions. Knowing the execution behaviour under different inputs or configurations can greatly help system operators to understand system behaviors. However, there may be a large number of different combinations of inputs or parameters under different system behaviors. Such complexity poses difficulties for analyzing contextual information related to the state of interest. In order to understand user behavior, descriptive statistics, e.g., mean, total, standard variation, most frequent value, etc., are typically used to obtain meaningful insights such as the basic behaviors of the users. These information can be also used to classify the user based on the correlation and demographic similarities among them. In order to understand the patterns from user behavior, we propose to exploit all of the usage data from multiple layers of the cloud environment, usage data of a cloud-based application is spread across front-end interfaces (web-browser, smart phone app/client and command-line interface) and the back-end (server instance and database instance) in a cloud environment [9] and formulate as the transition states of a graph. This type of graph can be used to mine execution patterns and to model relationships among different user behavior patterns. This kind of approach can also be used to discover some problems under some specific context. To discover these contextual factors, we propose to use the decision trees to learn the conditions, which allows us to determine any possible connections between the contexts and change in behavior of the user. In this study, we focus to exploit the data in application logs and refer to the work in [6] as shown in Fig. 1, where the authors have classified the usage data sources at the back-end of cloud into three groups, coming from three main sources, as follows: the system logs from the cloud services from the back-end of the cloud system, the application logs, and the logs from the virtual machines (VMs). As discussed earlier, the usage data could be extracted from various sources in a cloud environment and these data are of different type and formats. Hence, usage data has to be classified according to categories. Table 1 provides a information on how usage data could be classified (refer [9] for a deeper discussion of classification).

Table 1. Usage data classification (Source [9]).

Who is using the application	(a) User ID (b) IP address
Where the application is being hosted	(a) Web server (b) Database
What the end user does	(a) Application (b) Page (c) Method (d) Function (e) Button that is accessed (f) Action that is performed
When the user performs the operation	(a) Date and time (b) Session ID
How long it takes to complete the operation	(a) Duration (b) Query duration
Operation details	(a) Errors (b) Background tasks (c) Number of records loaded
What application features are used by the user and how	(a) Clickstream (b) View (c) Focus (d) API calls

Usage data in a cloud environment can be mainly divided into three categories: 1. System logs contain a wealth of information to help manage systems. Most systems print out logs during their executions to record system runtime actions and states that can directly reflect system runtime behaviours. System developers and architects usually use these logs to track a system to detect and diagnose system anomalies.

2. The second type of this data is the user-level usage data generated as a result of user interaction with a cloud-based application. Some examples of usage data are application logs, for example the assessment data (wiki, forum, message), the activity data (clicks, time spent), server logs, and so on. They can be extracted by the applications themselves or via Web cookies (from web browser). Such data in the cloud is spread across various interfaces such as Web browser, mobile applications and command line interfaces on the front-end and server and database on the back-end.

3. The last type of usage data is the VM logs, typically generated from the VMs running the applications or services. This type of logs contains the usage of the CPU, memories, as well as running tasks, time of starting and stopping

and others. Figure 1 shows a summary of the three main sources of usage data in the back-end of a cloud environment.

In this paper, we consider application logs at the back-end as the usage data source. The application logs reveal information such as:

– IP address of the user
– HTTP request and responses between the browser and the application server running in the VM
– Timestamp
– Application events triggered by user
– Errors and warnings
– Application resource access
– Diagnostics data
– Network configurations
– Database requests and responses.

In the next section, we present the usage analytics process and explain how the usage data are identified, tools and techniques used for extraction, how the extracted usage data could be analysed to identify the behavioural usage patterns and discuss how to evaluate the usage data and the usage analytics process.

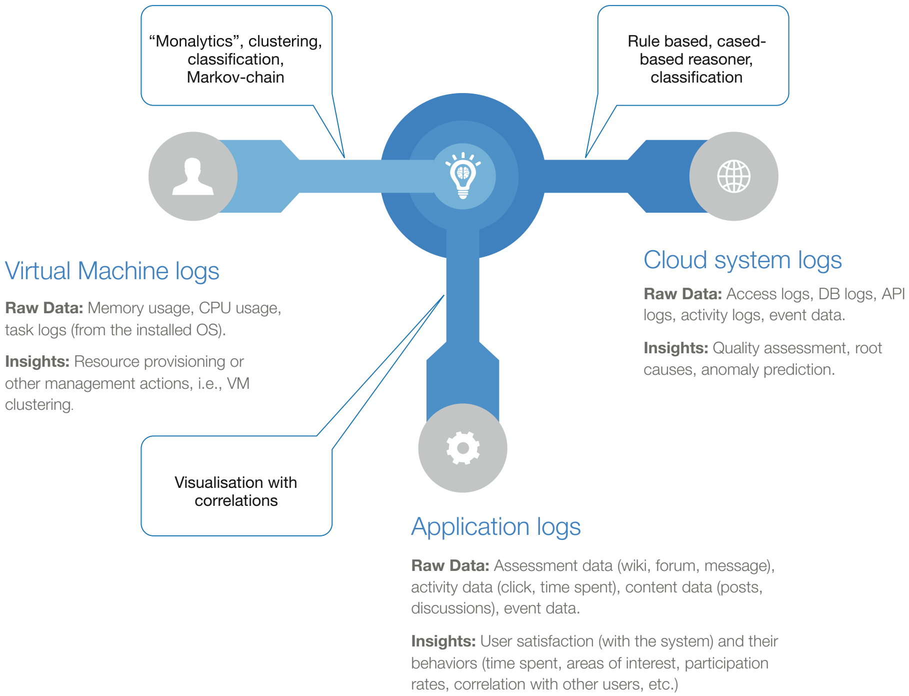

Fig. 1. Three main sources of usage data in cloud-based environment (Source: [6]).

3 Usage Analytics Process

Usage analytics process consists of steps explaining how to identify, extract and analyse the usage data with the aim to understand how users of an application use the features provided by the application. A high-level diagram of the usage analytics process is represented in Fig. 2.

Fig. 2. Experimental setup for Usage Analytics.

The first step is to classify the usage data which helps to know who is the user, what the user does, when does the user perform an action and so on. Refer to Table 1 in Sect. 2 for the usage data classification. The second step is to identify the features provided by the application. In order to identify the feature, it is essential to investigate the application architecture, identify the functionalities of the application, actions a users can perform, investigate the application catalogue provided by the developers and test the application for any features missing in the catalogue and then finally build a list of the features identified. The third step is to identify the usage data sources, it includes exploring the available usage data sources as shown in Fig. 1 in Sect. 2 while comparing the usage data classification as discussed in the first step. The fourth step includes carefully exploring the usage data source and identifying the format and type of the usage data to identify the user, action performed, time when the action performed and so on. The final step is to analyse the data, it includes using statistical analysis techniques to understand which feature was most used by which user, if the user consistent with accessing a particular feature, what is the frequency of the access of a feature and so on. This helps identifying distinguishing feature usage patterns of a user. Based on the different feature usage patterns exhibited by an user, we can identify which features are critical for the user.

4 Implementation

In this section, we present the experimental setup for the usage analytics process with the open-source application “Odoo Notes app”[2] chosen as the use-case. Odoo is a set of open-source range of easy to use business applications that form a complete suite of tools to accompany any business need. It is used by over 3.7 million users and around 950 business partners. Odoo notes is an online collaboration task management and to-do application. Users of this application are provided with a kanban-style dashboard to create, organise and edit notes. A complete set of functionalities provided this application is discussed in Sect. 4.2. The focus of this research is to extract and analyse usage data generated when user interact with the application through a browser.

[2] https://apps.odoo.com/apps/online/note/.

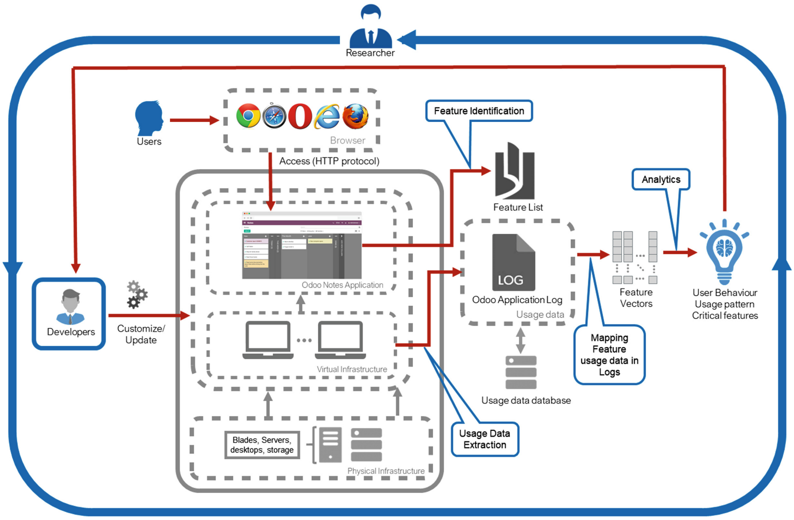

Fig. 3. Experimental setup for Usage Analytics.

4.1 Experimental Setup

The Odoo Notes application (community edition version 10) is deployed as a service in a virtual machine running Ubuntu 16.10 LTS operating system to simulate the nature of the cloud environment. Users access the application through as browser over the internet. Application logs are generated and stored in the VM hosting the application. Odoo Notes logs contain information such as Timestamp, HTTP requests and responses, Database requests and responses, IP address, Event name, Object ID and so on. Figure 3 shows the overall picture of Usage Analytics process. The first step in the process is to identify the list of features provided by the application.

4.2 Application Features Identification

Odoo notes is an online collaborative task management and to-do application. Users can create, edit and organise “notes” (similar to a sticky-note) in a kanban style dashboard. Users also have an option to share the notes with other users and invite other users to work together in a collaborative manner. Users can add comments to the notes, send private and broadcast messages to other users. A complete list of features provided by Notes application is available in the Odoo website. For the purpose of this study, we have carefully tested each feature and chosen only those features that a user explicitly can interact. This filtered set of features is shown in Table 2. The features F1–F17 are defined by the developer, features f18–F21 are manually identified by testing the application execution. We are aware of the possibility that the application may contain more features and many other features may be developed and implemented in the newer versions. But, in this paper, for the demonstration purpose, only the features listed in Table 2 are considered.

The focus of this research is to understand the user behaviour by analysing user interactions with the application. Feature F17 refers to a notification to the user about other users who are active. Since users do not have an option to interact with this feature of the application, it is neglected in this study. An interesting point to note is that the features listed here are initially defined by the developers of the application. However, we are aware of the fact that there could be other features being identified and categorised in the later versions of the application. Usage Analytics process is designed in a way to accommodate additional features being added to the list. Once the list of features to focus on is prepared, the next step is to extract the usage data.

Table 2. List of features of Odoo Notes Application.

#	Feature	Description
F1	Create stages (Columns)	Break down your to-do list into stages which will be converted to columns into your dashboard
F2	Create notes	Add notes to your stages. Each note correspond to a mini-project that you will move from one stage to another as your project moves forward
F3	Kanban view	Drag and drop notes easily from one stage to another in the kanban view
F4	Text layout	Insert text styles like headers bold, italic, lists and fonts with a simple WYSIWYG editor
F5	File attachments	Attach text files image files document files to your notes
F6	Tags	Add tags to your notes for a clear organization
F7	Filters and groups	Search notes easily with smart filters
F8	Colors	Group your notes by color as a way to categorize your tasks. There are 9 colors to choose from and a colorless option
F9	Import	Upload any text file or document to your notes
F10	Export	Export notes as HTML plain text or DocuWiki text documents
F11	Invite people	Add coworkers to your notes so they can follow the discussions and receive notifications
F12	Authorship color	Every author typing some text in a note has a different background color to show who wrote what. You can link a name to a color

Table 2. (*continued*)

#	Feature	Description
F13	Timeline slider	See the history of changes made to a note through a timeline from first to last sentence
F14	Share	Easily share your notes with your colleagues by sending them as link or embed URL
F15	Access Settings	Choose what others can do with your notes by granting viewing or editing access
F16	Chat	Enable chat for real time discussion with the people following your notes
F17	Show Connected Users*	See who is connected to your notes right now
F18	Delete a Note	Deletes a note, including the comments, attachments and tags attached
F19	Edit Note	Open the note in edit mode. Change the title, text and add comments
F20	Duplicate Note	Create a duplicate copy of the note in the same stage (column)
F21	Post Comment	Add comment to a note, all collaborators can view the comments

4.3 Usage Data Extraction

The Usage data extraction step includes identification of the usage data sources, identification of the usage data and the corresponding data extraction process. In a cloud environment, some of the typical usage data sources are application logs, application cache, server logs, cloud system logs, Application Programming Interface (API) event data, network logs on the back-end and browser cache, browser cookies, desktop client log, mobile app logs, mobile system logs and so on. Depending on the type of usage data and the purpose of analysis, appropriate usage data source can be considered. In this paper, we focus on application logs as the usage data source, type 2, as discussed in Sect. 2. Odoo notes application log entries contain the following information, explained using a small subset of the log data as shown in Log 1.1:

Log 1.1. Odoo Notes sample application log entries.

```
2018-03-28 12:24:10,966 8970 INFO ? werkzeug:
   ↪ 136.206.48.84 - - [28/Mar/2018 12:24:10] "GET /
   ↪ web_editor/static/src/js/transcoder.js HTTP/1.1"
   ↪ 200 -
2018-03-28 12:24:11,256 8970 DEBUG Odoo_Database odoo.
   ↪ api: call ir.ui.view().read_template(u'web_editor
   ↪ .colorpicker')
2018-03-28 12:24:11,260 8970 INFO Odoo_Database
   ↪ werkzeug: 136.206.48.84 - - [28/Mar/2018
   ↪ 12:24:11] "POST /web/dataset/call HTTP/1.1" 200 -
2018-03-28 12:24:11,309 8970 INFO Odoo_Database
   ↪ werkzeug: 136.206.48.84 - - [28/Mar/2018
   ↪ 12:24:11] "POST /web/webclient/translations HTTP
   ↪ /1.1" 200 -
2018-03-28 12:24:11,324 8970 INFO ? werkzeug:
   ↪ 136.206.48.84 - - [28/Mar/2018 12:24:11] "GET /
   ↪ web_editor/static/src/xml/editor.xml?debug
   ↪ =1522239851310 HTTP/1.1" 200 -
2018-03-28 12:33:46,695 8970 DEBUG Odoo_Database odoo.
   ↪ api: call note.tag(4,).read([u'name', u'color'])
```

The first and second fields in all the log entries represent the time stamp. The keywords *INFO, DEBUG* represent the level of logging, other possible options are *CRITICAL, ERROR and WARN*. The fourth field represents the target data store, a "?" symbol represents requests to load web-pages and web scripts. The name of the database in this sample is "Odoo_Database". The keyword "werkzeug" is the name of the logger tool. The next field shows the IP address of the user (or the person who made the request, for example, administrator of the application). Next field of importance is the HTTP request and response between the browser and the application server along with the response code. One interesting fact to notice here is that Odoo application logs can configured to capture the API calls which reveal a vast amount of information about features such as object ID of the elements (for example, ID of the tag is 4), operation performed (for example, the last line of the log entry represents read operation with the variables - the user defined name of the tag and the colour assigned by user to the note).

User actions can be identified by analysing the logs which helps reveal individual user's pattern of interactions with the application. In the next section we discuss how the logs are analysed to reveal usage patterns of the users.

4.4 Analysis

The first step in analysing the usage data is to map the feature data to the log entries. This step involves identifying the keywords in the log entries that refer to the actions performed by the user to the keywords in the log entries. A mapping file is created to aid this purpose as shown in 1.2.

Log 1.2. Feature mapping.

```
F3: Reorder / rearrange note
2018-03-27 12:58:22,875 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 12:58:22] "POST /web/dataset/resequence HTTP/1.1"
    ↪  200 -

F18: Delete a note
2018-03-27 13:48:15,166 20602 INFO Odoo_Database odoo.
    ↪ models.unlink: User #1 deleted note.note records
    ↪ with IDs: [33]

F19: Create a new note
2018-03-27 13:49:23,804 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:49:23] "POST /web/dataset/call_kw/note.note/
    ↪ create HTTP/1.1" 200 -

F20: Duplicate a note
2018-03-27 13:51:30,902 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:51:30] "POST /web/dataset/call_kw/note.note/
    ↪ copy HTTP/1.1" 200 -

F18: Edit note
2018-03-27 13:52:44,783 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:52:44] "POST /web/dataset/call_kw/note.note/
    ↪ read HTTP/1.1" 200 -
2018-03-27 13:52:44,806 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:52:44] "POST /mail/read_followers HTTP/1.1"
    ↪ 200 -
2018-03-27 13:52:44,831 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:52:44] "POST /web/dataset/search_read HTTP
    ↪ /1.1" 200 -
```

```
2018-03-27 13:52:55,493 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:52:55] "POST /web/dataset/call_kw/note.note/
    ↪ write HTTP/1.1" 200 -

F21: Open commenting in note - loads users list
2018-03-27 13:54:57,434 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:54:57] "POST /web/dataset/call_kw/note.note/
    ↪ message_get_suggested_recipients HTTP/1.1" 200 -

F21: Post comment in note
2018-03-27 13:56:18,848 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:56:18] "POST /web/dataset/call_kw/note.note/
    ↪ message_post HTTP/1.1" 200 -

F6: Create new tag in note
2018-03-27 13:59:29,608 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:59:29] "POST /web/dataset/call_kw/note.tag/
    ↪ name_create HTTP/1.1" 200 -
2018-03-27 13:59:32,082 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 13:59:32] "POST /web/dataset/call_kw/note.tag/
    ↪ create HTTP/1.1" 200 -

F6: Add new tag
2018-03-27 14:10:14,925 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 14:10:14] "POST /web/dataset/call_kw/note.tag/
    ↪ read HTTP/1.1" 200 -
2018-03-27 14:10:14,929 20602 INFO Odoo_Database
    ↪ werkzeug: 136.206.48.84 - - [27/Mar/2018
    ↪ 14:10:14] "POST /web/menu/load_needaction HTTP
    ↪ /1.1" 200 -
```

The mapping file will help identify the keywords to search in the logs file for entries that reveal the actions performed by users. Google's colab tool[3] is used for identifying the moments at which a specific user performed a specific task. Colab tool is built on Jupyter notebook[4], which is an open-source web application used for data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and so on. Keywords identified

[3] https://colab.research.google.com.
[4] http://jupyter.org.

from the mapping file are used to search the log. The keywords in the odoo notes application logs are generally found in in the HTTP POST request and API call fields. *Feature F3: Reorder/rearrange note* is represented in the log entry by the keyword "resequence". *Feature F18: Delete a note* can be identified using the keyword "delete note.note" and the log entry also provides the ID of the note deleted. In this sample log, the ID of the note deleted is "33". Similarly, *Feature F19: Create a note* can be identified by the keyword "note.note/create". While these are keywords are simple to recognise by their names, some keywords are difficult to identify. For example, the log entry for the *Feature F21: Open comment box in note* do not use keywords such as comment or open in the HTTP request. However, the keywords "note.note/message_get_suggested_recipients" reveal that a comment box was opened by the user as the application loaded the message recipient suggestions. Hence, a mapping file should be carefully created by anticipating the actions performed and understanding the programming logic of the application.

Statistical analysis could be used to identify the frequency by which a user accessed a feature, amount of time spent using a feature, consistency in accessing the feature and so on. Each feature can be ranked for each user or a group of users' usage pattern, which is yet to be implemented and tested, one of the aims of our future work. Snapshots of the user browser window are captured with the frequency of four snapshots per minute. The behavioural patterns of a users resulted from the analysis could be validated using the snapshots captured. Machine learning techniques can be used to cluster users who exhibit similar behavioural patterns, correlation between user clusters can be understood to help improve the ranking of the features. Developers can use this knowledge about the criticality of the features from the user's perspective to optimise the development of the application.

5 Conclusion and Future Work

We presented the usage analytics process, discussed and demonstrated the key activities: Feature Identification, Usage Data Extraction, Analysis and discussed how the usage data and the process of usage data extraction and analysis can be evaluated. The usage analytics process demonstrated with an experiment in this paper shows how to successfully understand behavioural usage patterns of the user in a cloud environment. The behavioural indicators could help the developers of the applications in cloud to optimise the software development process by prioritising the development resources to the features critical to the user.

For our future work, we aim to (1) test this experiment with users and implement the evaluation plan discussed in Sect. 5; (2) use machine learning algorithms to identify more behavioural indicators by analysing the correlation between the group of user exhibiting similar usage patterns; (3) include usage data from other relevant usage data sources (for example, browser sessions, cookies etc.) to improve the proposed extraction and analysis process.

Acknowledgements. This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre (www.lero.ie).

References

1. Al-Bayati, B., Clarke, N., Dowland, P.: Adaptive behavioral profiling for identity verification in cloud computing: a model and preliminary analysis. GSTF J. Comput. (JoC) **5**(1), 21 (2016)
2. Bezemer, C.P., Zaidman, A., Platzbeecker, B., Hurkmans, T., Hart, A.: Enabling multi-tenancy: an industrial experience report. In: IEEE International Conference on Software Maintenance, pp. 1–8, September 2010. https://doi.org/10.1109/ICSM.2010.5609735
3. Bucklin, R.E., Sismeiro, C.: Click here for internet insight: advances in clickstream data analysis in marketing. J. Interact. Mark. **23**(1), 35–48 (2009)
4. Castañeda, J.A., Rodríguez, M.A., Luque, T.: Attitudes' hierarchy of effects in online user behaviour. Online Inf. Rev. **33**(1), 7–21 (2009). https://doi.org/10.1108/14684520910944364. http://www.emeraldinsight.com/doi/10.1108/14684520910944364
5. Corapi, D., Ray, O., Russo, A., Bandara, A.K., Lupu, E.C.: Learning rules from user behaviour. In: Iliadis, Tsoumakasis, Vlahavas, Bramer (eds.) Artificial Intelligence Applications and Innovations III, vol. 296, pp. 459–468. Springer, Boston (2009). https://doi.org/10.1007/978-1-4419-0221-4_54
6. Dang-Nguyen, D.T., Kesavulu, M., Helfert, M.: Usage analytics: research directions to discover insights from cloud-based applications. In: International Conference on Smart Cities and Green ICT Systems (SMARTGREENS) (2018, accepted)
7. Gasparetti, F.: Modeling user interests from web browsing activities. Data Min. Knowl. Disc. **31**(2), 1–46 (2016). https://doi.org/10.1007/s10618-016-0482-x
8. Kesavulu, M., Dang-Nguyen, D.T., Helfert, M., Bezbradica, M.: An overview of user-level usage monitoring in cloud environment. In: The UK Academy for Information Systems (UKAIS) (2018)
9. Kesavulu, M., Helfert, M., Bezbradica, M.: A usage-based data extraction framework for cloud-based application - an human-computer interaction approach. In: International Conference on Computer-Human Interaction Research and Applications (CHIRA), Madeira, Portugal (2017)
10. Märtin, C., Herdin, C., Engel, J.: Model-based user-interface adaptation by exploiting situations, emotions and software patterns. In: International Conference on Computer-Human Interaction Research and Applications (2017)
11. Mulder, I., Ter Hofte, G.H., Kort, J.: SocioXensor: Measuring user behaviour and user eXperience in conteXt with mobile devices. In: Proceedings of Measuring Behavior, pp. 355–358, January 2005
12. Petruch, K., Tamm, G., Stantchev, V.: Deriving in-depth knowledge from IT-performance data simulations. Int. J. Knowl. Soc. Res. **3**(2), 13–29 (2012). https://doi.org/10.4018/jksr.2012040102. http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/jksr.2012040102

13. Xu, G., Zhang, Y., Yi, X.: Modelling user behaviour for Web recommendation using LDA model. In: Proceedings - 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT Workshops 2008, pp. 529–532 (2008). https://doi.org/10.1109/WIIAT.2008.313
14. Yang, J., et al.: Multimedia recommendation and transmission system based on cloud platform. Future Gener. Comput. Syst. **70**, 94–103 (2017)