# Measuring data-centre workflows complexity through process mining: the Google cluster case

**Damián Fernández-Cerero[1] · Ángel Jesús Varela-Vaca[1] · Alejandro Fernández-Montes[1] · María Teresa Gómez-López[1] · José Antonio Alvárez-Bermejo[2]**

## Abstract

Data centres have become the backbone of large Cloud services and applica-tions, providing virtually unlimited elastic and scalable computational and storage resources. The search for the efficiency and optimisation of resources is one of the current key aspects for large Cloud Service Providers and is becoming more and more challenging, since new computing paradigms such as Internet of Things, Cyber-Physical Systems and Edge Computing are spreading. One of the key aspects to achieve efficiency in data centres consists of the discovery and proper analysis of the data-centre behaviour. In this paper, we present a model to automatically retrieve execution workflows of existing data-centre logs by employing process mining tech-niques. The discovered processes are characterised and analysed according to the understandability and complexity in terms of execution efficiency of data-centre jobs. We finally validate and demonstrate the usability of the proposal by applying the model in a real scenario, that is, the Google Cluster traces.

**Keywords** Cloud computing · Business process management · Scheduling · Process mining · Process discovery · High performance computing

✉ Damián Fernández-Cerero
damiancerero@us.es

Ángel Jesús Varela-Vaca
ajvarela@us.es

Alejandro Fernández-Montes
afdez@us.es

María Teresa Gómez-López
maytegomez@us.es

José Antonio Alvárez-Bermejo
joseantonio@ual.es

[1] Department of Computer Languages and Systems, University of Seville, 41012 Seville, Seville, Spain

[2] Department of Computer Science, University of Almería, 04120 Almería, Spain

## 1 Introduction

Data centres are the backbone of large Cloud services and applications, providing virtually unlimited elastic and scalable computational and storage resources. Such environments are constantly evolving, from traditional monolithic systems tightly coupled to the MapReduce workload under execution [10], to highly flexible infrastructures which employ virtualisation and containerisation [13] to improve resource efficiency in Cloud-Computing scenarios.

Modern resource-managing and workload-scheduling models enable data-centre operators to share computing resources among heterogeneous applications and frameworks, such as MapReduce applications, frameworks for real-time analytics and on-demand Virtual Machines for Platform as a Service (PaaS) business models.

This search for the efficiency and optimisation of resources is one of the current key aspects for large Cloud Service Providers. Several strategies have been proposed to achieve this goal, including: (i)resource orchestration and managing models [21, 24, 26, 37, 48, 53]; (ii) scheduling algorithms and consolidation models [5, 15, 16, 18, 29]; and (iii) workload migration models [41, 58, 59].

The improvement in resource efficiency in data centres is becoming more and more challenging because centralised data centres present numerous limitations when new computing paradigms, such as Internet of Things [23], Edge Computing [50], cyber-physical systems [3] and Fog Computing [4], are under consideration.

The aforementioned models have something in common: to optimise their execution, it is necessary a deep knowledge about the system behaviour, actors and workloads involved. Such knowledge has been traditionally gathered through the human experience of data-centre operators and detailed log extraction and analysis [43], such as those presented in Fig. 1. Human-based analysis may lack objectivity and be error-prone.

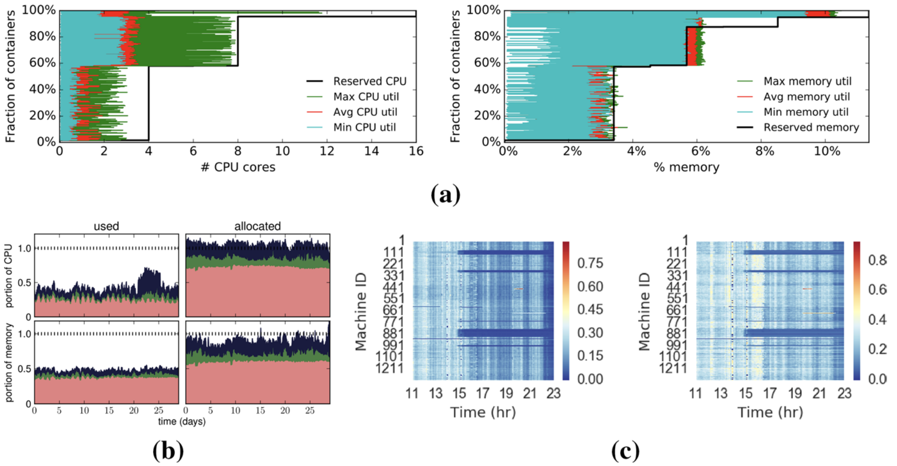


**Fig. 1** Detailed log analysis examples [7, 43]. It can be noticed that these analyses are complex and hard to apply directly to a decision-making system

In our approach, we propose to take a set of existing data-centre logs and to automatically retrieve execution workflows in order to infer how the data-centre operation evolves. The adoption of the proposed model based on a real-time process mining engine employing new process metrics which may unveil useful information about the complexity and behaviour of the workload could be directly applied to develop new scheduling, workload consolidation, container migration and resource and energy efficiency algorithms. Compared to raw log analysis, process mining models provide richer information which may be though to discover through human analysis, enabling more accurate decisions. In addition, such decisions require low computational resources, allowing fast operations in critical infrastructures, such as data centres. A data-centre log represents a set of executions (i.e. jobs and tasks) performed in the past sorted by time. Process mining [55] is a well-recognised discipline which utilises various techniques to infer processes from execution traces [2]. The application of process mining in data-centre contexts facilitates the extraction of the evolution of the data-centre operation and resource planning. Nonetheless, the results provided by process mining techniques (i.e. workflows) are not always easy to understand, and even to interpret depending on the case. Discovered processes might be unleashed into "spaghetti-like" models or "lasagna-like" processes [54, 57] derived from the diversity of the possible order execution of the tasks and its variety.

Spaghetti-like processes, which are related to unstructured processes, tend to appear in contexts where the tasks executed in each moment can vary according to the complex decision behind, as occurs in a data-centre scenarios. Lasagna processes represent more structured processes. Figure 2 shows the result of the direct application of process mining techniques to a piece of the Google data-centre log [45]. We can observe certain well-structured zones of the process (cf., right hand); nonetheless, the vast majority of processes present a spaghetti-like scheme. Therefore, the understandability and complexity of the process make the analysis almost unbearable for human beings.

Usually, processes are influenced by certain factors, including: *i*) time events and *ii*) social events, such as the Super Bowl. It is crucial to understand and identify these factors for an optimal interpretation of the results of the mined processes. Therefore, depending on the perspective, the characterisation and analysis of the mined process can help to determine improvements such as detecting bottlenecks and misused resources. Nevertheless, the simplification of lasagna and spaghetti processes is an open problem in the process mining domain [8, 51].

In this first approach, we pursue the transformation of data-centre execution logs into event log traces to characterise the process and to determine the understandability and complexity of data-centre jobs in terms of execution efficiency. Derived from this context and the problems to be addressed, the aims of the paper are:

**Fig. 2** Spaghetti process. This kind of process provides little usefulness to decision making

- **OBJ1** Define a framework which may establish the cornerstone in the application of process mining in the field of data centres and Cloud and Edge Computing.
- **OBJ2** Study the usefulness and the applicability of process mining in the enhancement of data-centre performance by analysing the processes generated from data-centre execution logs.
- **OBJ3** Analyse the behaviour and homogeneity of the extracted processes by means of statistical tests.
- **OBJ4** Validate the usability of the proposal by applying our approach to a real scenario, that is, the Google Cluster traces presented in [45].

The rest of the paper is organised as follows: The related work is discussed in Sect. 2. Section 3 briefly presents the proposed framework. In Sect. 4, we discuss the raw input data utilised for the use cases. The Log Extractor and XES Generation cases and tools are presented in Sect. 5. In Sect. 6, we briefly illustrate the process mining and discovery models and how they are employed in this work. The results obtained in our evaluation are presented and analysed in Sect. 7. Finally, the paper is summarised, and the conclusions and future work are presented in Sect. 8.

## 2 Related work

Data-centre logs have been the source of information to gather knowledge on many aspects of data-centre operation, such as the behaviour of the scheduling and resource orchestration frameworks, workload classification and characterisation, detection of vulnerabilities, performance issues and anomalies.

Many authors [11, 12, 28, 35, 49] have focused on the analysis of Google data-centre traces in order to properly characterise the workloads present in Cloud-Computing environments. In [28], Liu et al. study: (a) the management of machines, performing a deep analysis on the frequency and pattern of machine maintenance and (b) the cluster resource utilisation by performing an analysis on the life cycle of jobs and tasks. In [49], Sebastio et al. study the machine dynamic life cycle distributions to propose a data-driven model to enable the estimation of the expected number of available machines at any instant of time. Di et al. [11] analyse the resource utilisation per application and classify applications via a K-means clustering algorithm to show the underlying Pareto principle. Mishra et al. [35] propose a fine-grained algorithm to classify the workload present in the traces and its application to the Google Cloud Backend. In [12], Di et al. perform a detailed statistical analysis on job submission, resource request and machine utilisation patterns in order to compare Cloud-Computing workloads to those present in grid computing.

In addition, other authors [1, 14, 44] analyse Google traces with different objectives. In [14], El-Sayed et al. analyse traces to precisely predict patterns of job termination and failures and provide strategies to mitigate such failures through task-cloning policies. Abdul-Raman et al. [1] analyse the topology and resource request patterns of the workload to model the user behaviour through sessions. In [44], Reiss et al. analyse the traces to show that heterogeneity is present at all levels, which imposes challenges in the development of effective cloud-based resource schedulers. Moreover, authors have been working with data-centre traces from various providers in order to apply scheduling optimisation strategies to improve performance and energy efficiency [17].

Process mining has been applied in several fields to properly discover the processes followed by users or systems, by analysing event logs [25]. Depending on the field, different perspectives could be used to discover the processes, such as the activities executed, people involved, resources used and location where the actions occur. The versatility of process mining techniques has brought about their application to several scenarios [9], being health care [30, 40, 46] and IT [31, 38, 47] the most active areas. To the best of our knowledge, process mining has not been used before in data-centre contexts, whose characteristics make necessary certain types of specific adaptations.

Process discovery groups a set of algorithms that facilitate the creation of a workflow process model that represents the traces observed [32]. Typically, the process discovery has been based on the tasks analysis, but during last years, other resources as the persons who execute them or even its location have been included [42].

Process mining is a relevant topic very well received by the enterprises, bringing about the evolution of the research solution tools (e.g. ProM [56]) towards commercial solutions (e.g. Disco™and Celonis™); however, how to create the logs to be used in the tools is still a challenge.

## 3 Evaluation approach

Our approach aims to characterise the workflows of data centres from their raw execution logs. We propose the framework depicted in Fig. 3 to achieve this objective. The definition of the framework facilitates the applications of a methodology that includes each step to discover business processes from data-centre behaviour, including the actors and the workloads involved. Derived from the complexity of these systems, it is necessary to provide a mechanism to adjust the process discovery for a better analysis.

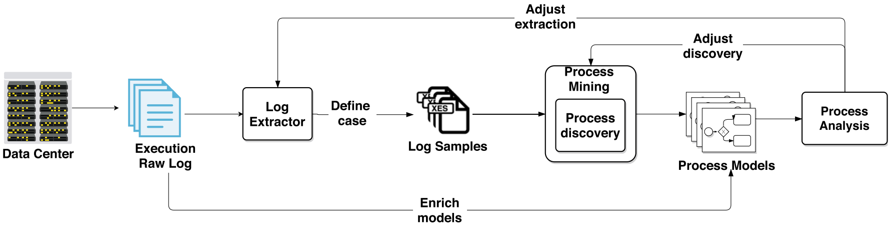


**Fig. 3** Framework overview. This framework extracts and adjusts information from raw data-centre logs to build rich process models, which indicators and metrics may be used to make data-centre operation decisions

First of all, the framework begins from a data-centre raw execution log (cf., *Execution Raw Log* in Fig. 3). We defined the most relevant terms used in process mining area to define *Log*, *Event Log*, *Event* and *Trace*.

**Definition 1** (*Execution Raw Log*) *Let RL be a raw log which consists of a multiset of records*, $\{r_1, r_2, \dots, r_m\}$, *in which* $r_i$ *is a tuple of attributes* $\langle a_1 : t_1, a_2 : t_2, \ldots,$ $a_n : t_n\rangle$ *where an attribute is identified by a name* $(a_i)$ *and described by a data type* $(t_i)$.

An event log for a process mining purposes is formed of multiset of traces:

**Definition 2** (*Event Log*) *Let L be an event log* $L = [\tau_1, \dots, \tau_m]$ *as a multiset of traces* $\tau_i$.

A trace is composed of a tuple with an identifier and a sequence of events that occurred at a particular time *t*:

**Definition 3** (*Trace*) *Let* $\tau$ *be a trace* $\tau = \langle case\_id, \mathcal{E}\rangle$ *which consists of a case_id that identifies the case, and a sequence of events* $\mathcal{E} = \{\varepsilon_1, \dots, \varepsilon_n\}$, $\varepsilon_i$ *occurring at a time index i relative to the other events in* $\mathcal{E}$.

An event occurrence is a triple with an identifier of an activity that occurred at a particular time stamp which may have additional information:

**Definition 4** (*Event occurrence*) *Let* $\varepsilon$ *be an event occurrence* $\varepsilon = \langle activity\_id, time\ stamps, others\rangle$ *which is specified by the identity of an activity which produces it and the related time stamps.* $\varepsilon$ *may store more information, such as states, labels and resources.*

Data-centre logs are usually huge in size, which makes them hard to process in many cases. Due to this, in this work, we perform two analyses which vary in the size of the logs: the first one takes the whole log file as input, whilst in the second analysis, we perform a random sampling to reduce the complexity of the log files. In addition, this second analysis enables us to have enough samples to perform partitionability and statistical tests to check the homogeneity of the internal processes over time.

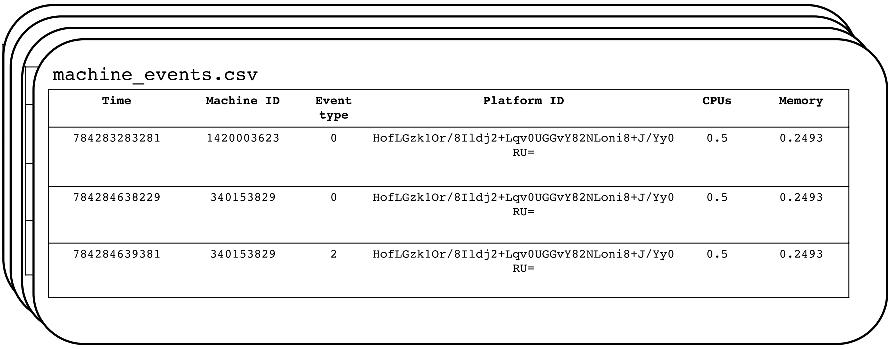
Therefore, the definition of transformations (cf., *Log Extractor* in Fig. 3) aims to extract an event log from data-centre raw execution logs. Depending on the purpose of the analysis, the extraction should define which pieces of data in the raw log should be considered as *case_id*, *activity_id*, and *event*, respectively. The log extraction consists of the specification of a sequence of sentences, that are, recipes, on how to extract certain information from a raw log and how to align it to the definitions of *case_id*, *activity_id* and *event occurrences*.

Once the event logs have been obtained, these logs are introduced in a process discovery tool (cf., *Process Mining–Process Discovery* in Fig. 3) which retrieves the workflows. Subsequently, these workflows are analysed (cf., *Process Analysis* in Fig. 3), and as the result of such analysis, the process discovery model and/or log extraction may be adjusted (cf., *Adjust discovery* and *Adjust extraction* in Fig. 3). On one hand, the aforementioned adjustments may include deletion of irrelevant cases or activities. On the other hand, the discovered workflows can be enriched (cf., *Enrich models* in Fig. 3) with extra information to carry out other type of analysis.

## 4 Experiment data: *execution raw log*

In this work, we used the data-centre raw execution log presented in [45], in order to illustrate our approach in a real scenario. These raw logs represent 29 days of operation time in a Google cluster composed of approximately 12,500 servers. The trace includes hundreds of thousands of jobs, which are submitted by final users. In this context, a final user may be a data-centre operator, a final client or even an auto-scaling software. Each job is composed of one to tens of thousands of tasks, which are the minimum workload unit to be deployed on an available server. These tasks are not gang scheduled [36], but are often processed in parallel. The information is provided in the form of timed events which may be grouped in the following categories:

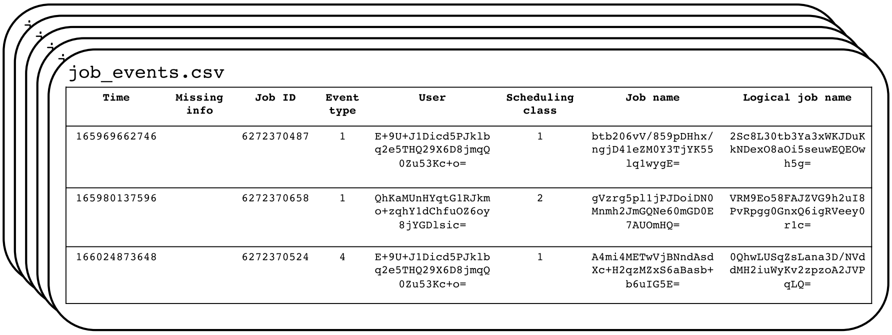
- **Machine** events, such as a fail or addition of machines, and the attributes related to those machines. Figure 4 shows an example of machine events.
- **Job** events, which represent the workload processing in the cluster. A job, which is composed of tasks, may be submitted, scheduled, evicted, or killed, for instance. Figure 5 shows an example of job events.
- **Task** events, which represent the minimum atomic workload to be executed by a server and their requirements, such as CPU, RAM and architecture. Both jobs and tasks have the same life cycle. Figure 6 shows an example of tasks events.



machine_events.csv

| Time | Machine ID | Event type | Platform ID | CPUs | Memory |
|---|---|---|---|---|---|
| 784283283281 | 1420003623 | 0 | HofLGzk1Or/8Ildj2+Lqv0UGGvY82NLoni8+J/Yy0RU= | 0.5 | 0.2493 |
| 784284638229 | 340153829 | 0 | HofLGzk1Or/8Ildj2+Lqv0UGGvY82NLoni8+J/Yy0RU= | 0.5 | 0.2493 |
| 784284639381 | 340153829 | 2 | HofLGzk1Or/8Ildj2+Lqv0UGGvY82NLoni8+J/Yy0RU= | 0.5 | 0.2493 |

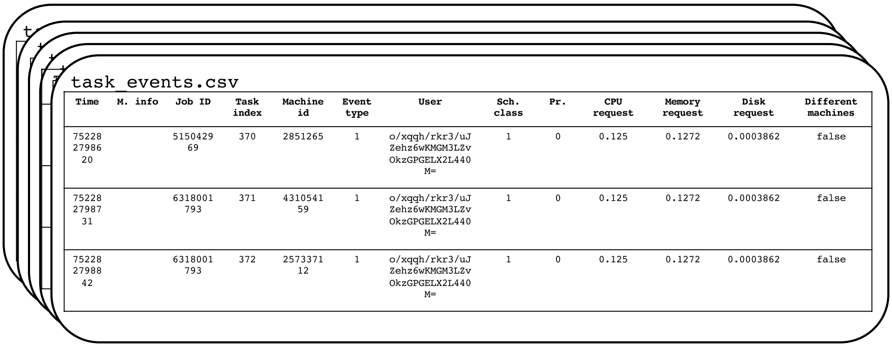
**Fig. 4** Machine events log raw data example. This log consists on the following parameters: time, machine ID, event type, platform ID, machine CPU and memory



job_events.csv

| Time | Missing info | Job ID | Event type | User | Scheduling class | Job name | Logical job name |
|---|---|---|---|---|---|---|---|
| 165969662746 |  | 6272370487 | 1 | E+9U+J1Dicd5PJklbq2e5THQ29X6D8jmqQ0Zu53Kc+o= | 1 | btb206vV/859pDHhx/ngjD41eZM0Y3TjYK55lq1wygE= | 2Sc8L30tb3Ya3xWKJDuKkNDexO8aOi5seuwEQEOwh5g= |
| 165980137596 |  | 6272370658 | 1 | QhKaMUnHYqtG1RJkmo+zqhY1dChfuOZ6oy8jYGDlsic= | 2 | gVzrg5pl1jPJDoiDN0Mnmh2JmGQNe60mGD0E7AUOmHQ= | VRM9Eo58FAJZVG9h2uI8PvRpgg0GnxQ6igRVeey0r1c= |
| 166024873648 |  | 6272370524 | 4 | E+9U+J1Dicd5PJklbq2e5THQ29X6D8jmqQ0Zu53Kc+o= | 1 | A4mi4METwVjBNndAsdXc+H2qzMZxS6aBasb+b6uIG5E= | 0QhwLUSqZsLana3D/NVddMH2iuWyKv2zpzoA2JVPqLQ= |

**Fig. 5** Job events log raw data example. This log contains the following information: time, missing information (if present), job ID, event type, user ID, scheduling class, job name and logical job name



task_events.csv

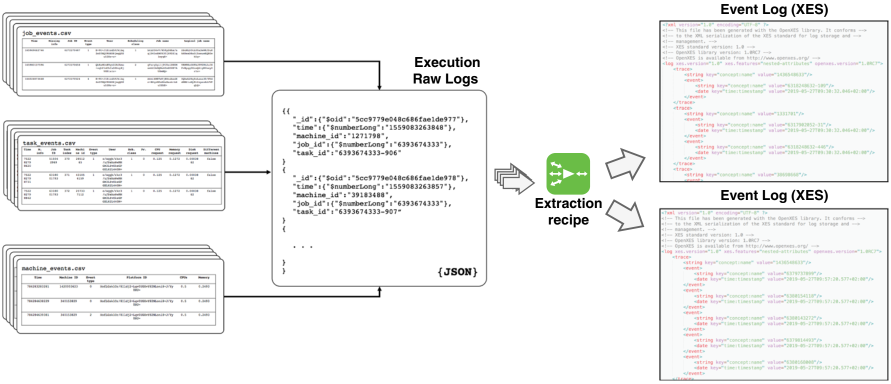
| Time | M. info | Job ID | Task index | Machine id | Event type | User | Sch. class | Pr. | CPU request | Memory request | Disk request | Different machines |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 752282798620 |  | 515042969 | 370 | 2851265 | 1 | o/xqqh/rkr3/uJZehz6wKMGM3LZvOkzGPGELX2L440M= | 1 | 0 | 0.125 | 0.1272 | 0.0003862 | false |
| 752282798731 |  | 6318001793 | 371 | 431054159 | 1 | o/xqqh/rkr3/uJZehz6wKMGM3LZvOkzGPGELX2L440M= | 1 | 0 | 0.125 | 0.1272 | 0.0003862 | false |
| 752282798842 |  | 6318001793 | 372 | 257337112 | 1 | o/xqqh/rkr3/uJZehz6wKMGM3LZvOkzGPGELX2L440M= | 1 | 0 | 0.125 | 0.1272 | 0.0003862 | false |

**Fig. 6** Task events log raw data example. This log is composed of the following attributes: time, missing information (if present), job ID, event type, task index, machine ID where the task event occurred, event type (in this work we only consider events representing task submission), user ID, scheduling class, priority, CPU, memory, and disk required, and whether the even happened in different machines

The raw 29-day log of 42GB is divided into hundreds of comma-separated value files, of 50 MB each. Task and job events trace files represent the vast majority of files. In this work, we consider five random files among those task events to study the behaviour and background processes of the data-centre operation and scheduling. Furthermore, we gather four random samples of each log file to analyse the heterogeneity for several general, quality and partitionability metrics among samples.

## 5 Log extractor: XES generator

In general, we consider a data-centre raw execution log as input, and as output, a set of traces such as those shown in Fig. 7. The raw log files described in the previous section may be presented in a unstructured (NoSQL) or structured (SQL) fashion. Most of the current process mining tools accept XES [22] and MXML



```
{{
   "_id":{"$oid":"5cc9779e048c686fae1de977"},
   "time":{"$numberLong":"1559083263848"},
   "machine_id":"1271798",
   "job_id":{"$numberLong":"6393674333"},
   "task_id":"6393674333-906"
}
{
   "_id":{"$oid":"5cc9779e048c686fae1de978"},
   "time":{"$numberLong":"1559083263857"},
   "machine_id":"39183488",
   "job_id":{"$numberLong":"6393674333"},
   "task_id":"6393674333-907"
}
{

   . . .

}
}
```

**Fig. 7** Log extraction overview. The XES files are constructed from the execution raw logs following the recipes that are described in the following sections

formats. We are able to define a set of recipes by using the ELE transformation language [52], to convert raw execution logs onto XES formatted files. In Fig. 7, two different extractions are depicted as the result of the obtention of various event logs (i.e., XES) for two different analysis purposes.

In Example 5.1, we provide a piece of code to illustrate the resulting log extraction, containing the information relative to *machine_id*, *job_id* and *start_date* in XES format.

**Example 5.1: Generated event log in XES format**

```
<log xes.version="1.0"
  xes.features="nested-attributes"
  openxes.version="1.0RC7">
  <trace>
    <string key="concept:name" value="1436548633"/>
    <event>
      <string key="concept:name" value="6318248632"/>
      <date key="time:timestamp"
        value="2019-05-27T09:56:52.186+02:000"/>
    </event>
    [...]
  </trace>
  [...]
</log>
```

All the resources, thus, raw logs (i.e., json format), the XES files and the processes discovered that are employed in this work are freely available at http://www.idea.us.es/thejournalofsupercomputing/.

### 5.1 Cases of log extraction

In this work, we propose three cases of log extraction to illustrate our approach:

- **Case** *Simple*, which analyses the processes of the users according to their execution of jobs. This case is an illustrative case used as a minimum-complexity comparison for the rest of cases.
- **Case** ***A***, which studies the processes related to the machines in terms of their execution of jobs.
- **Case** ***B***, which evaluates the processes according to the deployment of tasks onto machines.

For the complex cases (cases *A* and *B*), we performed a two-step analysis:

- **Analysis without file sampling**, where we process the entire log files. As a result, we compare five individuals (i.e., identified by 150, 276, 311, 402 and 478) composed of a high number of events, as shown in Tables 2 and 3.
- **Analysis with file sampling**, where we gather four random samples of ∼ 5000 events from each log file. As a result, we compare five groups (i.e., identified by 150, 276, 311, 402 and 478) composed of four samples each, which are, in turn, composed of a lower number of events, as shown in Tables 2 and 3 (e.g. id. 150 sample 1,2,3,4). This deep analysis enables us to perform deep sequentiality and homogeneity tests between samples.

In the Case *Simple*, we show the second step alone (*Analysis with file sampling*, as shown in Table 1), since the results obtained with the first analysis are too simple and self-contained in the second analysis.

#### 5.1.1 Case *simple*: process of the users according to their execution of jobs

The workload executed in a data centre is composed of jobs, which are submitted by users. These users can be real data-centre operators, final users and even automated processes, like auto-scaling frameworks. The event log employed in this case is structured as follows:

- ***case_id***: *user_id.*
- ***activity_id***: *job_id.*
- ***timestamp***: *start_date.*

The XES logs extracted following this case for the samples of experiment are described in Table 1.

**Table 1** Sample details for the *Analysis with file sampling* of the Case *Simple*

| Id | Sample | Traces | Events | Size (MB) |
|---|---|---|---|---|
| 150 | | | | |
| | 1 | 69 | 300 | 1.3 |
| | 2 | 98 | 291 | 1.3 |
| | 3 | 94 | 422 | 1.3 |
| | 4 | 81 | 364 | 1.3 |
| 276 | | | | |
| | 1 | 79 | 219 | 1.3 |
| | 2 | 40 | 121 | 1.3 |
| | 3 | 69 | 202 | 1.3 |
| | 4 | 69 | 233 | 1.3 |
| 311 | | | | |
| | 1 | 61 | 123 | 1.3 |
| | 2 | 75 | 159 | 1.3 |
| | 3 | 60 | 140 | 1.3 |
| | 4 | 59 | 279 | 1.3 |
| 402 | | | | |
| | 1 | 34 | 68 | 1.3 |
| | 2 | 30 | 60 | 1.3 |
| | 3 | 27 | 54 | 1.3 |
| | 4 | 53 | 86 | 1.3 |
| 478 | | | | |
| | 1 | 37 | 110 | 1.3 |
| | 2 | 30 | 71 | 1.3 |
| | 3 | 58 | 138 | 1.3 |
| | 4 | 50 | 186 | 1.3 |

It should be borne in mind that in this case, we only perform the deep analysis with log-file sampling

### 5.1.2 Case *A*: process of machines according to their execution of jobs

During the data-centre operation, the incoming jobs must be distributed among several machines. In this test case, the processes can help to understand how machines are used according to the jobs deployed on them. Hence, the resulting event log has the next form:

- ***case_id***: *machine_id*.
- ***activity_id***: *job_id*.
- ***timestamp***: *start_date*.

The start date of a job can be determined by the first task submitted within that particular job. To do this, the tasks are grouped by job_id. These grouped tasks are then ordered by start date, and finally, the first one is chosen. The XES logs extracted

**Table 2** Id and sample details for the analysis of the Case *A*

| Id | Sample | Traces | Events | Size (MB) |
|---|---|---|---|---|
| **150** | | **8437** | **21510** | **3.6** |
| | 1 | 3526 | 4513 | 1.3 |
| | 2 | 3533 | 4542 | 1.3 |
| | 3 | 3166 | 4291 | 1.3 |
| | 4 | 3276 | 4399 | 1.3 |
| **276** | | **11678** | **74498** | **11.4** |
| | 1 | 3813 | 4778 | 1.3 |
| | 2 | 3880 | 4855 | 1.3 |
| | 3 | 3956 | 4823 | 1.3 |
| | 4 | 4156 | 4933 | 1.3 |
| **311** | | **11126** | **65326** | **10.0** |
| | 1 | 3948 | 4886 | 1.3 |
| | 2 | 3286 | 4202 | 1.3 |
| | 3 | 3827 | 4761 | 1.3 |
| | 4 | 3761 | 4825 | 1.3 |
| **402** | | **11708** | **79850** | **12.1** |
| | 1 | 2253 | 3749 | 1.3 |
| | 2 | 2325 | 3627 | 1.3 |
| | 3 | 1929 | 3265 | 1.3 |
| | 4 | 3579 | 4502 | 1.3 |
| **478** | | **11660** | **59646** | **9.3** |
| | 1 | 4119 | 4785 | 1.3 |
| | 2 | 3665 | 4216 | 1.3 |
| | 3 | 3819 | 4781 | 1.3 |
| | 4 | 3938 | 4885 | 1.3 |

It can be noticed that the values for the whole files used in the first analysis without file log sampling (bold) are higher than those provided by the samples employed in the second analysis with file log sampling

in this case for the samples of experiment data given in Sect. 4 are described in Table 2.

### 5.1.3 Case *B*: process of the deployment of tasks onto machines

During the data-centre operation time, a set of tasks, which belong to jobs, are deployed. As a result, the tasks are distributed by the machines in the data centre according to the scheduling policies. In this test case, the processes may help to understand how tasks are executed depending on the chosen machines. Hence, the resulting event log has the next form:

- ***case_id***: *task_id*.

**Table 3** File and sample details for the analysis of Case *B*

| File | Sample | Traces | Events | Size (MB) |
|---|---|---|---|---|
| 150 | | 18,929 | 25,034 | 4.9 |
| | 1 | 4133 | 4960 | 1.3 |
| | 2 | 4031 | 4970 | 1.3 |
| | 3 | 3375 | 4928 | 1.3 |
| | 4 | 3711 | 4926 | 1.3 |
| 276 | | 70,106 | 80,265 | 16.4 |
| | 1 | 4519 | 4990 | 1.3 |
| | 2 | 4608 | 4990 | 1.3 |
| | 3 | 4275 | 4991 | 1.3 |
| | 4 | 4506 | 4999 | 1.3 |
| 311 | | 62,184 | 71,070 | 14.5 |
| | 1 | 4887 | 4982 | 1.3 |
| | 2 | 4850 | 4979 | 1.3 |
| | 3 | 4696 | 4971 | 1.3 |
| | 4 | 4338 | 4992 | 1.3 |
| 402 | | 69,217 | 126,677 | 22.9 |
| | 1 | 2257 | 4991 | 1.3 |
| | 2 | 2453 | 4992 | 1.3 |
| | 3 | 1917 | 4987 | 1.3 |
| | 4 | 4185 | 4998 | 1.3 |
| 478 | | 61,072 | 68,324 | 14.1 |
| | 1 | 4616 | 4998 | 1.3 |
| | 2 | 4829 | 4998 | 1.3 |
| | 3 | 4491 | 4994 | 1.3 |
| | 4 | 4415 | 4995 | 1.3 |

It can be noticed that the high values of the parameters may indicate the complexity of this case compared to the rest of cases

- ***activity_id***: *machine_id*.
- ***timestamp***: *start_date*.

In this case, the start date of a *task_id* is well determined in the raw execution log. Hence, the log extraction has only to align this date with the time stamp. The resulting XES logs, which represent the output of the application of the recipes described in this case taking as input the samples of experiment data presented in Sect. 4, are described in Table 3.

**Table 4** Tiny sample of event log

| Case id (Traces) | Activity Id (Events) | Time stamp |
|---|---|---|
| 1436548633 | 6318248632 | $2019-05-27T09:56:51.186+02:00$ |
| 1331701 | 6317902052 | $2019-05-27T09:56:52.186+02:00$ |
| 1331701 | 6318248632 | $2019-05-27T09:56:54.196+02:00$ |
| 38698668 | 6318447049 | $2019-05-27T09:56:52.186+02:00$ |

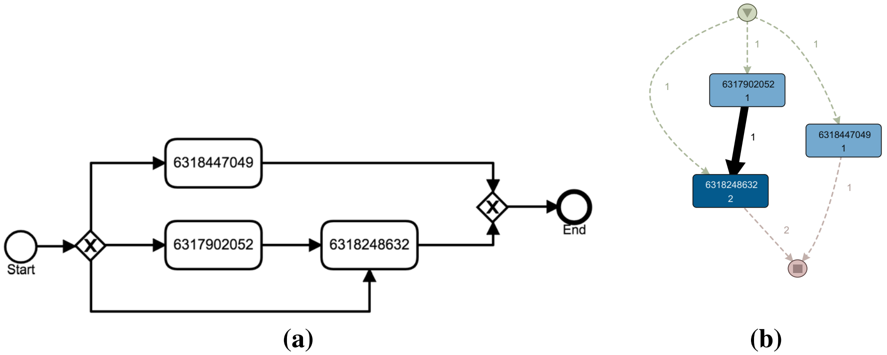


**Fig. 8** Process discovered by ProM and Disco tool. All the processes are fully contained. In Disco, the colour intensity represents the number of event occurrences

## 6 Process mining: process discovery

Process mining has emerged as a new research area within business process management. Process mining provides a family of solutions that includes process discovery, conformance checking, and process enhancement [55].

Process discovery provides multiple mechanisms of analysis of processes that are unknown or non-intrinsically defined by the organisation. Process discovery enables multi-perspective analysis of processes depending on the required needs. Even simulation may be a suitable tool to carry out performance analysis, which may help to develop various analysis, such as time analysis, resource analysis, and case analysis.

Nowadays, process discovery is a mature discipline which has been well received by the industry to discover complex processes [33]. Proof of this is the number of algorithms (e.g. Alpha algorithm, Inductive mining, Heuristic miner) and tools (e.g. ProM [56], Disco™and Celonis™) that have emerged during last years.

As aforementioned, process discovery aims to obtain a model that covers all the possible traces. In order to illustrate the process discovery, a tiny sample of XES event log is given in Table 4. In the example, *Case id* may represent a machine, *Activity id* may represent a task executed by the machine, and finally, *Timestamp* may denote the start date of the task. Figure 8 shows two possible outputs of the processes mined by ProM (cf., BPMN process in the left) and Disco tool suite (cf., process map in the right), which covers every trace in the table.

The best fitness values for activities have been employed to obtain both models. The fitness is the capacity of the discovered model to cover the traces of an event log.

Relevant information can be extracted from the process models. For instance, three tasks (cf., 6318248632, 6317902052 or 6318447049) are firstly deployed in all the cases, as well as a pattern consisting on the execution of the task 6318248632 just after 6317902052 is recognised. In the figure, extra information might be shown, as the number of traces that are represented by each transition and the number of repetitions for each activity. Moreover, this extra information may help to infer other relevant operation-related events. For instance, in two of the cases, the last activity is 6318248632, but in the third case, the last activity is 6318447049. Another important aspect to remark in the use of process mining techniques is the number of events maybe not the same as the number activities represented in the resulting process as shown in this example. The log has for events, and the process only shows three activities as the result of grouping the repetition of events (i.e., 6317902052). This is one of the advantages of using this type of techniques against the analysis of raw logs with a huge number of events making it difficult to correlate the behaviour of an event that occurs many times.

In our approach, *ProM* tool with *inductive miner* is employed and the *fitness* of activities value ranged from 0.00 to 0.20 to discover process models. ProM has been used to be a free tool with a wide set of algorithms available; moreover, it provides an easy way to be connected with other software components. Although the best fitness is reached when the value is 0.00, this value does not provide results in several of our samples. For this reason, we increase the range of fitness to 0.20. A deeper analysis on the mined process models is described in the following sections.

## 7 Analysis of results

Typically, an analysis of soundness and correctness of process models should be carried out, but inductive process discovery techniques [19] ensure the soundness and correctness of the process models obtained [27]. Thus, the processes discovered are always complete, have a proper completion and have no dead transitions.

Since our approach aims is to characterise the data-centre workflows in terms of the understandability and complexity, we propose the usage of the following main set of metrics in order to correctly analyse the process:

- **General metrics** that enable the analysis in terms activities, gateways and transitions of the model.
- **Quality metrics** that allow the analysis of models in terms of the complexity of the models.
- **Skewness metrics** which enable the analysis of models in terms of similarity of the models.
- **Partitionability metrics** enable the analysis of the process model in terms of the relationships between sub-components in the whole process model.

Regarding the quality metrics, in the literature, several metrics are used to measure how "good" a design of a business process model is [6, 34, 39]. The next set of metrics is adapted to measure the understandability and the complexity, which, in turn, allows the objective comparison of the discovered processes:

- **Density** (***Δ***): the ratio of transitions divided by the maximum number of possible transitions. The lower the value of density, the higher the understandability and the lower the complexity. All the metrics are computed based on the control-flow graph $G = (V, E)$, where $V$ and $E$ represent the vertices (activities) and arcs (transitions), respectively. Therefore, we can denote the density formally as follows:

$$\Delta(G) = \frac{|E|}{|V| \cdot (|V| - 1)} \tag{1}$$

- **Cyclomatic number** (CC): the number of paths needed to visit all activities. The cyclomatic can be seen as a complexity metric; thus, the lower the value of CC, the lower the level of complexity. CC can be defined as:

$$\mathrm{CC}(G) = |E| - |V| + 1 \tag{2}$$

- **Coefficient of connectivity** (CNC): the ratio of transitions to activities. The greater the value of CNC, the greater the complexity of configuration workflows. Notwithstanding, models with the same CNC value might differ in complexity regarding this parameter [34]. CNC can be written as:

$$\mathrm{CNC}(G) = \frac{|E|}{|V|} \tag{3}$$

.

- **Control-flow complexity** (CFC) enables the measurement of the complexity in terms of the potential transitions after a split, depending on its type. The higher the value of the CFC, the higher the overall structural complexity of a workflow. CFC may be denoted as follows:

$$\mathrm{CFC}(G) = \sum_{c \in S_{and}} 1 + \sum_{c \in S_{xor}} |c_{xor} \bullet| + \sum_{c \in S_{or}} 2^{|c_{or} \bullet|} - 1 \tag{4}$$

  where $|c_{xor} \bullet|$ and $|c_{or} \bullet|$ represent the output transitions of a XOR gateway and OR gateway, respectively.
- **Sequentiality** ($\Xi$) denotes the ratio computed by dividing the number of arcs between nodes without connectors by the total number of arcs. The lower the value of sequentiality, the higher the structural complexity of a workflow is expected. It may be formally described as:

$$\Xi(G) = \frac{|E \cap (V \times V)|}{|E|} \tag{5}$$

  where $V \times V$ represent the set of transitions of two activities.

We present the results for tiny sample in Table 5, as shown in Fig. 8, in order to illustrate the calculation of the quality and general metrics.

The skewness and partitionability analysis may help to provide extra data on the behaviour in terms of variability and similarity. The following statistical parameters are proposed to evaluate the understandability and complexity of the models for the *general* and *quality* metrics presented in both cases:

- Mean ($\mu$), Median ($\widetilde{\mu}$), and Standard Deviation $\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}$.
- Fisher skewness coefficient ($\gamma_1$), which measures the skewness of the distribution of a variable according to its mean: $\gamma_1 = \frac{\mu_3}{\mu_2^{3/2}} = \frac{\mu_3}{\sigma^3}$.
- $F$, $F_{critic}$ and $p$ value related to the one-way ANOVA tests [20]; we performed to test whether the samples are statistically homogeneous over time or not.

In the next subsections, the resulting metrics for each case are analysed and compared.

### 7.1 Results for the case *simple*

In general, this type of process may help to understand which are the most representative users in terms of the number of jobs submitted. A random sample is shown in Fig. 9 for illustration purposes. This process is obviously simpler and more understandable when compared with the spaghetti-like ones (cf., Fig. 2). Regarding the control-flow structure, the process is composed of two gateways (i.e., XOR-split and XOR-join). Multiple paths, each one representing the sequence of jobs executed by a user, can be observed. Although the process is composed of multiple paths, just one can be executed each time due to the XOR gateway. Looking at in depth, we can highlight that there are very significant users with a large number of jobs, whilst others only submitted one or two jobs. All the remaining discovered processes of the case have a similar structure.

Regarding the metrics, the results shown in Table 6 represent the *general*, *quality*, *skewness* and *partitionability* metrics obtained for the analysis of the *Case Simple* with log-file sampling. It should be borne in mind that one metric is homogeneous among samples when the $F$ ratio is lesser than $F_{critic}$. Therefore, the results obtained for all metrics are statistically not homogeneous, which implies that the behaviour of inherent processes varies significantly over time. It can be noticed that the values for sequentiality $\Xi$ are 1 and 2 orders of magnitude

**Table 5** General and quality metrics

| Sample | Traces | Events | Act. | Trans. | Gate. | $\Delta$ | CC | CNC | CFC | $\Xi$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Tiny | 4 | 4 | 3 | 8 | 2 | 1.33 | 6 | 2.67 | 3 | 0.125 |

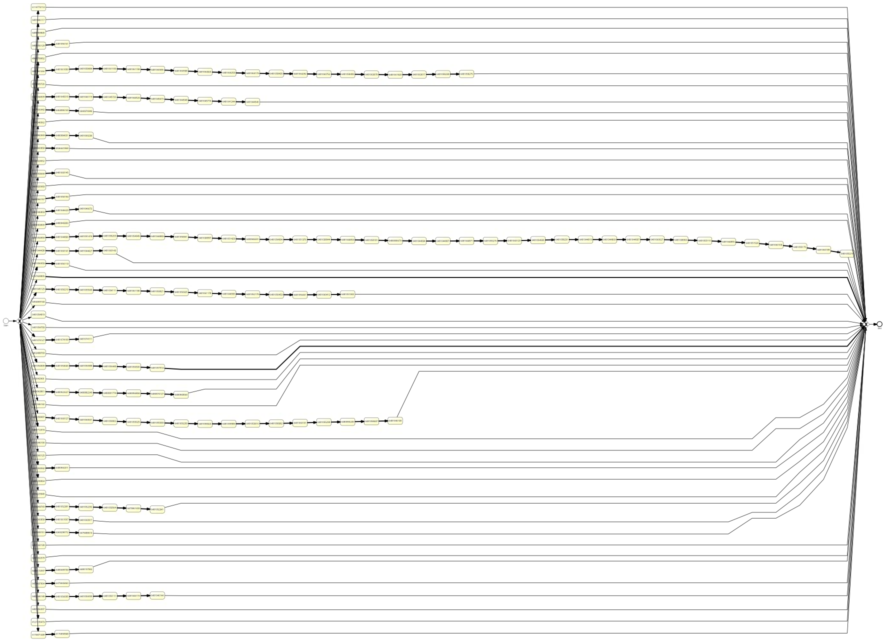


**Fig. 9** Process discovered for sample 487-4. This process clearly presents the simplicity and the sequentiality of the Case *Simple*

superior to those of the Case $A$ and $B$, respectively. Notwithstanding, the density levels ($\Delta$) are comparable to those of the Case $A$.

On one hand, both the number of events and the number of traces between samples present huge differences: the values range from 54 to 422 and from 27 to 98, respectively. Nonetheless, the standard variation of the number of events and transactions represents the $\sim 57\%$ of the mean, and the standard variation of the number of traces reduces to $\sim 43\%$ of its mean.

On the other hand, the number of gateways keeps stable due to the simplicity of the case of study. The low number of transitions and gateways highlights the extremely low complexity of the models due to the multiple options of execution paths introduced. As a general conclusion, we can confirm the low complexity and high understandability of the processes discovered in all the samples due to the low number of activities, transitions and gateways.

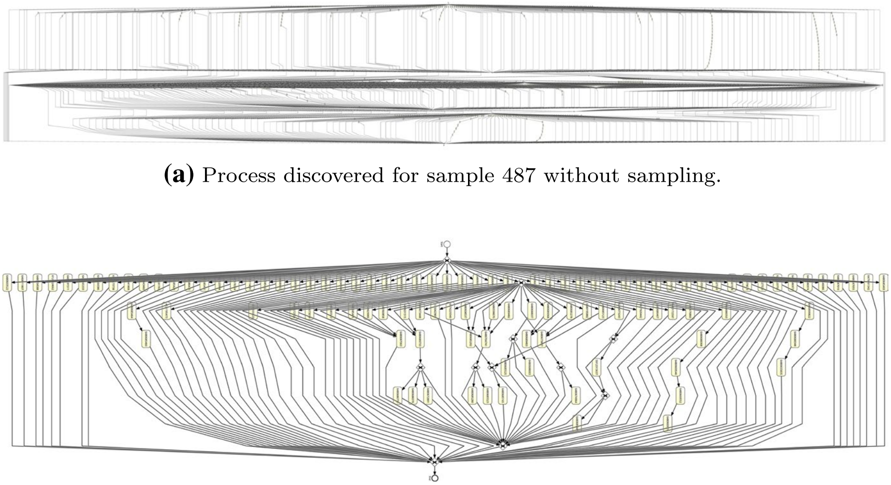
The ANOVA tests performed showed that the behaviour of the processes in terms of almost all metrics is heterogeneous between log files, which may indicate that the complexity of the process is time dependent. This conclusion may help us to build a set of time-related metrics that could support decision making, especially in scheduling, migration, consolidation and efficiency tasks.

**Table 6** General, quality, skewness, partitionability and variance analysis results obtained for the Case *Simple*

| Id-Sample | Traces | Events | Act. | Trans. | Gate. | $\Delta(10^{-3})$ | CC | CNC | CFC | $\Xi(10^{-1})$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 150-1 | 69 | 300 | 300 | 371 | 2 | 4.14 | 72 | 1.24 | 70 | 6.23 |
| 150-2 | 98 | 291 | 291 | 391 | 2 | 4.63 | 101 | 1.34 | 99 | 4.94 |
| 150-3 | 94 | 422 | 422 | 518 | 2 | 2.92 | 97 | 1.23 | 95 | 6.33 |
| 150-4 | 81 | 364 | 364 | 447 | 2 | 3.38 | 84 | 1.23 | 82 | 6.33 |
| 276-1 | 79 | 219 | 219 | 300 | 2 | 6.28 | 82 | 1.37 | 80 | 4.67 |
| 276-2 | 40 | 121 | 121 | 163 | 2 | 11.23 | 43 | 1.35 | 41 | 4.97 |
| 276-3 | 69 | 202 | 202 | 273 | 2 | 6.72 | 72 | 1.35 | 70 | 4.87 |
| 276-4 | 69 | 233 | 233 | 304 | 2 | 5.62 | 72 | 1.30 | 70 | 5.39 |
| 311-1 | 61 | 123 | 123 | 186 | 2 | 12.40 | 64 | 1.51 | 62 | 3.33 |
| 311-2 | 75 | 159 | 159 | 236 | 2 | 9.39 | 78 | 1.48 | 76 | 3.56 |
| 311-3 | 60 | 140 | 140 | 202 | 2 | 10.38 | 63 | 1.44 | 61 | 3.96 |
| 311-4 | 59 | 279 | 279 | 340 | 2 | 4.38 | 62 | 1.22 | 60 | 6.47 |
| 401-1 | 34 | 68 | 68 | 104 | 2 | 22.83 | 37 | 1.53 | 35 | 3.27 |
| 401-2 | 30 | 60 | 60 | 92 | 2 | 25.99 | 33 | 1.53 | 31 | 3.26 |
| 401-3 | 27 | 54 | 54 | 83 | 2 | 29.00 | 30 | 1.54 | 28 | 3.25 |
| 401-4 | 53 | 86 | 86 | 141 | 2 | 19.29 | 56 | 1.64 | 54 | 2.34 |
| 487-1 | 37 | 110 | 110 | 149 | 2 | 12.43 | 40 | 1.35 | 38 | 4.90 |
| 487-2 | 30 | 71 | 71 | 103 | 2 | 20.72 | 33 | 1.45 | 31 | 3.98 |
| 487-3 | 58 | 138 | 138 | 198 | 2 | 10.47 | 61 | 1.43 | 59 | 4.04 |
| 487-4 | 50 | 186 | 186 | 238 | 2 | 6.92 | 53 | 1.28 | 51 | 5.71 |
| $\mu$ | 62.40 | 176.08 | 176.08 | 240.44 | 2.00 | 11.59 | 65.36 | 1.42 | 63.36 | 4.34 |
| $\widetilde{\mu}$ | 60 | 157 | 157 | 225 | 2 | 9.51 | 63 | 1.43 | 61 | 4.04 |
| $\sigma$ | 27.11 | 100.56 | 100.56 | 121.81 | 0.00 | 7.47 | 26.97 | 0.12 | 26.97 | 1.51 |
| $\gamma_1$ | 1.44 | 0.84 | 0.84 | 0.70 | N/A | 0.91 | 1.41 | − 0.13 | 1.41 | 0.37 |
| $F$ | 9.34 | 15.53 | 15.53 | 17.05 | N/A | 17.49 | 9.34 | 7.91 | 9.34 | 6.37 |
| $F_{critic}$ | 3.06 | 3.06 | 3.06 | 3.06 | N/A | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 |
| $p$ | 0.00 | 0.00 | 0.00 | 0.00 | N/A | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

## 7.2 Results for Case *A*

As aforementioned, this case can help to understand how machines are used according to the jobs deployed on them. Figure 10 shows the discovered process for the case from two perspectives: (a) without sampling and (b) with sampling. On one hand, the process obtained without sampling (cf., sub-figure a) is huge, very complex and non-understandable since we cannot easily analyse the structure. On the other hand, the sampled process (cf., sub-figure b) is less complex and much more understandable. Some conclusions can be drawn from it: some paths are composed of only one activity and others have a more complex path structure. The path with one activity may be the consequence of that particular



**(a)** Process discovered for sample 487 without sampling.

**(b)** Process discovered for the first sample of 487.

**Fig. 10** Processes discovered by Case *A*. It can be noticed that the sampling reduced the complexity of the process, which enables a richer and more useful processing

machine executing only one job. We can conclude that the analysis with sampling is much more useful than that without sampling.

Tables 7 and 8 present the *general*, *quality*, *skewness* and *partitionability* results provided by the analysis without and with log-file sampling for Case *A*, respectively.

Regarding general metrics in Table 7, although there is a big difference in the number of events between samples ranging from 59,646 and 21,510 events, the number of mapped activities keeps more homogeneous, thus, 1820 at maximum in the worst case and half of that in the best case. All the samples present the similar behaviour with regard to transitions and gateways. However, transitions represent the paths between activities within the model and gateways represent

**Table 7** General, quality, skewness and partitionability results obtained for the Case *A* without log-file sampling

| Id | Traces | Events | Act. | Trans. | Gate. | $\Delta(10^{-3})$ | CC | CNC | CFC | $\Xi(10^{-2})$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 150 | 8437 | 21510 | 1428 | 2271 | 93 | 1.11 | 844 | 1.59 | 1342 | 7.29 |
| 276 | 11678 | 74498 | 1601 | 3159 | 45 | 1.23 | 1559 | 1.97 | 1557 | 3.01 |
| 311 | 11126 | 65326 | 1597 | 3132 | 45 | 1.23 | 1536 | 1.96 | 1534 | 4.00 |
| 402 | 11708 | 79850 | 1820 | 3592 | 38 | 1.09 | 1773 | 1.97 | 1771 | 2.70 |
| 478 | 11660 | 59646 | 958 | 1807 | 40 | 1.97 | 850 | 1.89 | 848 | 8.47 |
| $\mu$ | 10921.80 | 60166 | 1480 | 2792 | 52.2 | 1.33 | 1312 | 1.88 | 1410.40 | 5.05 |
| $\widetilde{\mu}$ | 11660 | 62746 | 1597 | 3132 | 45 | 1.23 | 1536 | 1.96 | 1534 | 3.80 |
| $\sigma$ | 1409.86 | 22989.39 | 323.70 | 730.06 | 23.01 | 3.66 | 434.79 | 0.16 | 349.20 | 2.64 |
| $\gamma_1$ | − 2.08 | − 1.61 | − 1.24 | − 0.53 | 2.14 | 2053 | − 0.39 | − 1.99 | − 1.24 | 0.61 |

**Table 8** General, quality, skewness, partitionability and variance analysis results obtained for the Case *A*

| Id | Traces | Events | Act. | Trans. | Gate. | $\Delta(10^{-3})$ | CC | CNC | CFC | $\Xi(10^{-2})$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 150-1 | 3526 | 4513 | 300 | 621 | 40 | 6.92 | 322 | 2.07 | 320 | 4.99 |
| 150-2 | 3533 | 4542 | 291 | 593 | 28 | 7.03 | 303 | 2.04 | 301 | 6.07 |
| 150-3 | 3166 | 4291 | 422 | 832 | 45 | 4.68 | 411 | 1.97 | 409 | 10.46 |
| 150-4 | 3276 | 4399 | 364 | 723 | 36 | 5.47 | 360 | 1.99 | 358 | 8.99 |
| 276-1 | 3813 | 4778 | 219 | 470 | 35 | 9.84 | 252 | 2.15 | 250 | 5.74 |
| 276-2 | 3880 | 4855 | 121 | 255 | 16 | 17.56 | 135 | 2.11 | 133 | 3.92 |
| 276-3 | 3956 | 4823 | 202 | 406 | 17 | 10.00 | 205 | 2.01 | 203 | 5.42 |
| 276-4 | 4156 | 4933 | 233 | 469 | 24 | 8.68 | 237 | 2.01 | 235 | 8.74 |
| 311-1 | 3948 | 4886 | 123 | 267 | 21 | 17.79 | 145 | 2.17 | 143 | 3.37 |
| 311-2 | 3286 | 4202 | 159 | 337 | 24 | 13.41 | 179 | 2.12 | 177 | 3.86 |
| 311-3 | 3827 | 4761 | 140 | 300 | 22 | 15.42 | 161 | 2.14 | 159 | 3.00 |
| 311-4 | 3761 | 4825 | 279 | 580 | 46 | 7.48 | 302 | 2.08 | 300 | 10.00 |
| 402-1 | 2253 | 3749 | 68 | 153 | 14 | 33.58 | 86 | 2.25 | 84 | 1.31 |
| 402-2 | 2325 | 3627 | 60 | 136 | 14 | 38.42 | 77 | 2.27 | 75 | 2.94 |
| 402-3 | 1929 | 3265 | 54 | 118 | 10 | 41.23 | 65 | 2.19 | 63 | 1.69 |
| 402-4 | 3579 | 4502 | 86 | 184 | 11 | 25.17 | 99 | 2.14 | 97 | 1.63 |
| 487-1 | 4119 | 4785 | 110 | 221 | 11 | 18.43 | 112 | 2.01 | 110 | 10.86 |
| 487-2 | 3665 | 4216 | 71 | 152 | 10 | 30.58 | 82 | 2.14 | 80 | 2.63 |
| 487-3 | 3819 | 4781 | 138 | 290 | 19 | 15.34 | 153 | 2.10 | 151 | 5.17 |
| 487-4 | 3938 | 4885 | 186 | 367 | 18 | 10.67 | 182 | 1.97 | 180 | 7.90 |
| $\mu$ | 3487.75 | 4480.90 | 181.30 | 373.70 | 23.05 | 16.91 | 193.40 | 2.10 | 191.40 | 5.08 |
| $\widetilde{\mu}$ | 3665 | 4542 | 157 | 324 | 18 | 13.41 | 193.40 | 2.10 | 191.40 | 4.99 |
| $\sigma$ | 631.48 | 469.92 | 105.81 | 207.95 | 11.61 | 111.31 | 102.39 | 0.09 | 102.39 | 8.61 |
| $\gamma_1$ | − 1.45 | − 1.35 | 0.78 | 0.71 | 0.80 | 1.03 | 0.64 | 0.26 | 0.64 | 0.46 |
| $F$ | 9.71 | 7.18 | 15.53 | 16.17 | 7.07 | 14.93 | 16.59 | 6.37 | 16.59 | 1.40 |
| $F_{critic}$ | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 2.77 |
| $p$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.27 |

It can be noticed that all metrics except sequentiality ($\Xi$) are statistically heterogeneous

control-flow elements that enable the convergence and divergence of paths. A high number of transitions and gateways may increase considerably the complexity of the models due to the multiple options of execution paths introduced. In summary, we can confirm the high complexity and low understandability of the processes discovered in all the log files without log-file sampling due to the high number of activities, transitions and gateways.

The best log file by considering activities, transitions and gateways is 478. However, observing the mean ($\mu$) and median ($\widetilde{\mu}$) according to activities, the more representative log files are 150 and 311, whilst 150 and 311 are the most representative in terms of transitions, and with regard to gateways, 276, and 311. Therefore, we can conclude that the models of the log file 150 and 311 can be taken as representative for the *Case A* since they can be considered as the most understandable and less complex in terms of general metrics.

By analysing $\gamma_1$, the coefficient shows a negative skew with regard to activities and transitions but positive regarding gateways. Thus, the models present an asymmetrical distribution when negative or positive coefficient are reached. Although the number of transitions is negative, it is relatively close to zero; thus, it almost achieves a symmetrical distribution. This coefficient has helped us to identify that the complexity and understandability of the models present non-homogeneous trends in terms of symmetry distribution according to the general metrics.

Regarding the quality metrics, the reference models are 402, 150 and 478. The log file 402 is the best case for density, whilst 150 excels for CC and CNC, and 478 for CFC. Thus, these are the models that obtained the best score in terms of complexity and understandability regarding quality metrics. Comparing the results with general metrics, the log file 311 has the second highest rate of density and CFC, and the worst CNC. However, the log file 150 has been also selected as the model reference. Although other samples may be considered as relevant, the log file 311 represents better the values for general metrics, and 402 and 478 represent the log files according to quality metrics.

We may conclude that the log file 150 is the most representative in terms of general and quality metrics. Hence, it can be considered the most understandable and less complex model, taking into account the difficulty of treating processes like these.

Comparing the above results with those provided by the analysis with log-file sampling (cf., Table 8), the number of traces, events, activities, gateways and transitions is drastically reduced since the sample covers only a part of the whole log file. Nevertheless, the reduction in the number of elements may help to reduce the complexity and to improve the understandability of the case as remarked at the beginning of the section. We can see how the density in all the samples is higher due to the ratio of activities versus transitions is lower than that provided by the analysis without sampling. It should be noticed that even though a higher density usually implies a worst complexity, in this case, these trends are not present. In terms of CC, CNC, CFC and $\Xi$, we can conclude that in some cases, the log-file sampling increases the complexity. For instance, the values of CNC resulting of the analysis with sampling are higher than those provided by the analysis without sampling. Nevertheless, the lower values of CC, CFC and the higher value of $\Xi$ indicate that the process models obtained with the analysis with sampling are less complex and more understandable than those provided by the analysis without sampling.

As a summary of the Case *A*, we may conclude that the analysis of the datacentre workflows employing log-file sampling present a simpler behaviour than that without sampling.

### 7.3 Results for Case *B*

As previously mentioned, this case may help to understand how tasks are executed depending on the chosen machines. We want to remark that the discovered process without sampling is impossible to depict due to the limitations of the modellers when importing, representing, and exporting these models. Nonetheless, Fig. 11 shows the discovered process obtained with the analysis with log-file sampling. In the process, the paths represent the machines where the tasks are executed. On one hand, in this particular case, there are paths with only one activity, which means that those particular tasks are only executed in one machine. On the other hand, there are paths where multiple machines are used to execute the tasks.

The results shown in Tables 9 and 10 represent the *general*, *quality*, *skewness* and *partitionability* metrics obtained with the analysis without and with log-file sampling, respectively, in Case $B$. Due to the complexity of the analysis without sampling, for log file 402, the $\Xi$ metric is not provided, since we were unable to compute it due to limitations of the tools employed. Such non-present values are denoted by a dash symbol in Table 9.

Regarding general metrics, there exist a huge number of traces and events with differences, multiplying by five the number of events in the worst cases. Notwithstanding, the number of mapped activities keeps reasonably stable between 8433 and 11, 702. In this case, it is necessary to emphasise the huge number of transitions and gateways. Such a trend demonstrates the enormous complexity related to the number of optional paths in the model. Similarly to the previous case, we confirm the high complexity and low understandability of the processes discovered in all the log files due to the high number of activities, transitions and gateways.

The best results are provided by log files 150 and 276 for activities; 276 for transitions; and 402 for gateways. The mean ($\mu$) and median ($\widetilde{\mu}$) indicate that the most representative log files are 311, 402 and 276, according to activities, transitions and gateways, respectively. Therefore, we can conclude that the models of the log files 276 and 402 can be taken as representative for the Case $B$, since they can be considered as the most understandable and less complex in terms of general metrics.

The models discovered in Case $A$ seem less complex than those of Case $B$. However, the ratio between activities and transitions is similar in both cases. Furthermore, in Case $B$, the number of gateways in the models is enormous compared with that of Case $A$. In the same way, the selection of reference models in both cases according to general metrics is the result of different analyses. For this reason, the resulting reference models differ from those obtained for Case $A$.

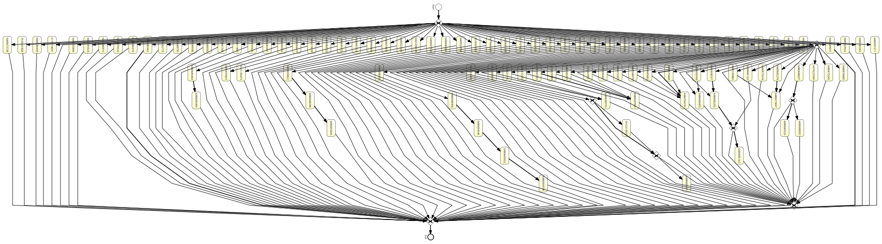


**Fig. 11** Process discovered for the first sample of 487 in Case $B$. It can be noticed the higher complexity of this case

**Table 9** General, quality, skewness and partitionability results obtained for the Case *B* without log-file sampling

| File | Traces | Events | Act. | Trans. | Gate. | $\Delta(10^{-4})$ | CC | CNC | CFC | $\Xi(10^{-3})$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 150 | 18929 | 25034 | 8433 | 10225 | 1410 | 1.44 | 1793 | 1.21 | 8587 | 2.74 |
| 276 | 70106 | 80265 | 8433 | 10199 | 1433 | 1.43 | 1767 | 1.21 | 8586 | 0.33 |
| 311 | 62184 | 71070 | 11116 | 13683 | 2072 | 1.11 | 2568 | 1.23 | 11541 | 1.08 |
| 402 | 69217 | 126677 | 11702 | 13251 | 1207 | 0.97 | 1550 | 1.13 | 11947 | – |
| 478 | 61072 | 68324 | 11659 | 14487 | 2282 | 1.07 | 2829 | 1.24 | 12143 | 0.74 |
| $\mu$ | 56301.60 | 74274 | 10269 | 12369 | 1680.80 | 1.20 | 2101.40 | 1.21 | 10561 | 0.97 |
| $\widetilde{\mu}$ | 62184 | 72672 | 11116 | 13251 | 1433 | 1.11 | 1793 | 1.21 | 11541 | 0.91 |
| $\sigma$ | 21280.51 | 36236.07 | 1691.50 | 2018.42 | 467.36 | 0.22 | 560.82 | 0.04 | 1815.31 | 1.06 |
| $\gamma_1$ | − 2.04 | 0.21 | − 0.53 | − 0.39 | 0.55 | 3594 | 0.60 | − 1.69 | − 0.55 | 1501.32 |

**Table 10** General, quality, skewness, partitionability and variance analysis results obtained for the Case *B*

| Id-Sample | Traces | Events | Act. | Trans. | Gate. | $\Delta(10^{-4})$ | CC | CNC | CFC | $\Xi(10^{-3})$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 150-1 | 4133 | 4960 | 3526 | 7094 | 67 | 5.71 | 3569 | 2.01 | 3576 | 5.07 |
| 150-2 | 4031 | 4970 | 3533 | 7198 | 122 | 5.77 | 3666 | 2.04 | 3664 | 3.61 |
| 150-3 | 3375 | 4928 | 3166 | 6427 | 106 | 6.41 | 3262 | 2.03 | 3260 | 5.76 |
| 150-4 | 3711 | 4926 | 3276 | 6643 | 89 | 6.19 | 3368 | 2.03 | 3366 | 1.96 |
| 276-1 | 4519 | 4990 | 3813 | 7685 | 82 | 5.29 | 3873 | 2.02 | 3871 | 5.07 |
| 276-2 | 4608 | 4990 | 3880 | 7787 | 43 | 5.17 | 3908 | 2.01 | 3906 | 3.08 |
| 276-3 | 4275 | 4991 | 3956 | 8018 | 136 | 5.12 | 4063 | 2.03 | 4061 | 10.23 |
| 276-4 | 4506 | 4999 | 4156 | 8413 | 151 | 4.87 | 4258 | 2.02 | 4256 | 10.10 |
| 311-1 | 4887 | 4982 | 3948 | 7961 | 43 | 5.11 | 4014 | 2.02 | 4012 | 3.01 |
| 311-2 | 4850 | 4979 | 3286 | 6653 | 54 | 6.16 | 3368 | 2.02 | 3366 | 1.95 |
| 311-3 | 4696 | 4971 | 3827 | 7675 | 19 | 5.24 | 3849 | 2.01 | 3847 | 0.52 |
| 311-4 | 4338 | 4992 | 3761 | 7541 | 54 | 5.33 | 3781 | 2.01 | 3779 | 6.63 |
| 402-1 | 2257 | 4991 | 2253 | 4553 | 14 | 8.97 | 2301 | 2.02 | 2299 | 0.66 |
| 402-2 | 2453 | 4992 | 2325 | 4670 | 21 | 8.64 | 2346 | 2.01 | 2344 | 1.71 |
| 402-3 | 1917 | 4987 | 1929 | 3868 | 10 | 10.40 | 1940 | 2.01 | 1938 | 0.78 |
| 402-4 | 4185 | 4998 | 3579 | 7252 | 78 | 5.66 | 3674 | 2.03 | 3672 | 0.83 |
| 487-1 | 4616 | 4998 | 4119 | 8431 | 182 | 4.97 | 4313 | 2.05 | 4311 | 4.27 |
| 487-2 | 4829 | 4998 | 3665 | 7391 | 63 | 5.50 | 3727 | 2.02 | 3725 | 3.11 |
| 487-3 | 4491 | 4994 | 3819 | 7803 | 162 | 5.35 | 3985 | 2.04 | 3983 | 3.33 |
| 487-4 | 4415 | 4995 | 3938 | 8046 | 171 | 5.19 | 4109 | 2.04 | 4107 | 4.23 |
| $\mu$ | 4002.88 | 4980.32 | 3438.04 | 6944.56 | 70.60 | 6.15 | 3507.52 | 2.02 | 3505.88 | 3.22 |
| $\widetilde{\mu}$ | 4338 | 4990 | 3665 | 7391 | 54 | 5.50 | 3727 | 2.02 | 3725 | 3.01 |
| $\sigma$ | 917.73 | 21.60 | 640.80 | 1306.97 | 55.39 | 1.54 | 667.00 | 0.01 | 667.04 | 2.79 |
| $\gamma_1$ | – 1.19 | – 1.65 | – 1.29 | – 1.27 | 0.72 | 1.78 | – 1.24 | 0.67 | −1.24 | 1.21 |
| $F$ | 10.95 | 13.83 | 9.71 | 9.60 | 5.94 | 8.52 | 9.47 | 3.47 | 9.46 | 4.21 |
| $F_{critic}$ | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 | 3.06 |
| $p$ | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | 0.00 | 0.02 |

It can be noticed that in this case, both the results in terms of density ($\Delta$) and sequentiality ($\Xi$) are one order of magnitude inferior to those of the Case *A*. In this case, even the sequentiality is not homogeneous among samples

By analysing $\gamma_1$, the coefficient shows a negative skew with regard to activities and transitions but positive regarding gateways. Although the skewness values are negative and positive in all cases, the values are moderately close to zero in five of eight metrics. Such values highlight a trend to a symmetrical distribution.

Moreover, the samples belonging to Case *B* have a more symmetrical distribution than those of Case *A*, since the skewness coefficient is closer to zero in most of the metrics.

Regarding the quality metrics, the log files 402 and 276 may be taken as the reference models. The log file 402 provides the best results for the density ($\Delta$) parameter, CC and CNC, whilst the log file 276 provides the best results for CFC.

Consequently, these are the models that obtained the best score in terms of complexity and understandability regarding quality metrics.

Comparing the results with general metrics, the quality metrics match with the models selected by general metrics; precisely, 276 and 402 have been also selected as representative. In conclusion, the log files 276 and 402 are chosen as representative models for the Case *B*.

As in Case *A*, similar conclusions can be obtained by comparing the aforementioned metric with those obtained with the analysis with log-file sampling (cf., Table 9) and with (cf., Table 10). The metrics $\Delta$, CC and CNC are higher for the analysis with log-file sampling. Thus, the process models obtained without sampling are less complex than those obtained with sampling. Nevertheless, CFC and $\Xi$ determine that the models obtained with sampling are less complex and more understandable than those obtained by the analysis without sampling.

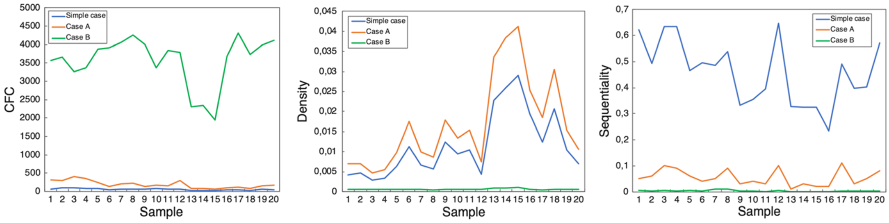
Case *A* seems to have less quality in comparison with Case *B* according to $\Delta$, CC and CNC. In contrast, Case *B* has worst quality than Case *A* in terms of CFC. In both cases, the chosen reference models differ, as shown in Fig. 12.

As summary, in the light of the visual and metrics analysis of the Case *A* and *B*, we can confirm the hypothesis that the analysis of the data-centre workflows with sampling is easier due to the lower complexity and higher understandably of the models obtained with log-file sampling.

## 8 Conclusions and future work

In this paper, we have coped with the problem of extracting the actual workflows present in data centres by performing several analyses of raw data-centre execution logs. We applied process mining strategies to discover the internal workflows. We presented a model that enables the obtention of the process models which cover all the traces of a raw event log.

In addition, we validated the usability of the proposal by applying our model in a real scenario, that is, the Google Cluster traces presented in [45]. The case study demonstrates the complexity of the workflows employed in data centres and the



**(a)** $CFC$ **(b)** $Density(\Delta)$ **(c)** $Sequentiality(\Xi)$

**Fig. 12** Comparison in terms of CFC, *Density*($\Delta$), *Sequentiality*($\Xi$) between cases and log-file samples. In these figures, the high complexity of case *B* is clearly depicted, due to the high CFC, and the low *Density*($\Delta$) and *Sequentiality*($\Xi$)

ability of process mining to unleash information that could help to improve the data-centre resource efficiency.

We learned the following important lessons:

1. **The data-centre log complexity** The complexity of the data-centre logs presents a real challenge, since it has a non-unified form which requires a pre-transformation or a querying process previous to XES transformation.
2. **The selection of case analysis** Several cases have been explored before to select the cases shown in the paper. The selection of wrong cases may lead to untreatable event logs that cannot be handled by process mining tools.
3. **The limitation of process mining tools** One of the challenges, under constant improvement, is the analysis of huge event logs, such as data-centre execution logs used in the paper. It is especially apparent in non-commercial tools, although it can also be found in commercial solutions. Consequently, such logs need to be divided and preprocessed before the application of any process discovery technique
4. **Insufficient metrics to represent complexity** The metrics used in this work may help in the understanding of the model complexity in terms of certain general characteristics. However, more suited metrics should be developed to optimally understand data-centre related processes.

As future work, this is the first step towards the development of a new research line based on the application of process mining modes to Cloud-Computing and Edge-Computing environments. The next steps include:

- Development of novel metrics for the measurement of model quality to easily represent the complexity of data-centre workloads and scheduling operations.
- A stronger statistical analysis on the presented scenarios;
- An analysis on the typical time window where process models keep homogeneous to support decision making;
- Application of new transformations to unveil hidden inefficiencies.
- Application of process mining to online scheduling processes as a tool to help decision making.
- Application of process mining to those processes is carried out by data-centre operators and any other human-related operations.
- Application of the presented models on other large-scale data-centre scenarios and comparison between the obtained results.
- Perform efficiency analysis based on the results of process mining models to improve resource and energy efficiency in data centres.

**Acknowledgements** Ministry of Science and Technology of Spain, ECLIPSE (RTI2018-094283-B-C33) and REACT (RTI2018-098062-A-I00) projects; University of Seville, VI Plan Propio de Investigación y Transferencia – US 2018, Proyecto Precompetitivo 2018/00000520; Junta de Andalucía, PIRADMIDE and METAMORFOSIS projects; European Regional Development Fund (FEDER). The authors would like to thank the Cátedra de Telefónica "Inteligencia en la Red" of the Universidad de Sevilla for its support.

## References

1. Abdul-Rahman OA, Aida K (2014) Towards understanding the usage behavior of google cloud users: the mice and elephants phenomenon. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science. pp 272–277. https://doi.org/10.1109/CloudCom.2014.75
2. Augusto A, Conforti R, Dumas M, Rosa ML, Maggi FM, Marrella A, Mecella M, Soo A (2019) Automated discovery of process models from event logs: review and benchmark. IEEE Trans Knowl Data Eng 31(4):686–705. https://doi.org/10.1109/TKDE.2018.2841877
3. Bhuiyan MZA, Kuo S, Lyons D, Shao Z (2019) Dependability in cyber-physical systems and applications. TCPS 3(1):1:1–1:4
4. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing. ACM, pp 13–16
5. Buyya R, Beloglazov A, Abawajy J (2010) Energy-efficient management of data center resources for cloud computing: a vision. architectural elements, and open challenges. arXiv preprint arXiv:1006.0308
6. Cardoso J (2005) Control-flow complexity measurement of processes and weyuker's properties. In: 6th International Enformatika Conference, vol 8. pp 213–218
7. Cheng Y, Anwar A, Duan X (2018) Analyzing alibaba's co-located datacenter workloads. In: 2018 IEEE International Conference on Big Data (Big Data). pp 292–297 https://doi.org/10.1109/BigData.2018.8622518
8. Conforti R, Rosa ML, ter Hofstede AHM (2017) Filtering out infrequent behavior from business process event logs. IEEE Trans Knowl Data Eng 29(2):300–314
9. Dakic D, Stefanovic D, Cosic I, Lolic T, Medojevic M (2018) Business application: a literature review. In: 29TH DAAAM International symposium on intelligent manufacturing and automation. https://doi.org/10.2507/29th.daaam.proceedings.125
10. Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51(1):107–113
11. Di S, Kondo D, Cappello F (2013) Characterizing cloud applications on a google data center. In: 2013 42nd International Conference on Parallel Processing. pp 468–473 https://doi.org/10.1109/ICPP.2013.56
12. Di S, Kondo D, Cirne W (2012) Characterization and comparison of cloud versus grid workloads. In: 2012 IEEE International Conference on Cluster Computing. pp 230–238. https://doi.org/10.1109/CLUSTER.2012.35
13. Dua R, Raja A.R, Kakadia D (2014) Virtualization vs containerization to support PaaS. In: 2014 IEEE International Conference on Cloud Engineering. IEEE, pp 610–614
14. El-Sayed N, Zhu H, Schroeder B (2017) Learning from failure across multiple clusters: a trace-driven approach to understanding, predicting, and mitigating job terminations. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). pp 1333–1344. https://doi.org/10.1109/ICDCS.2017.317
15. Fernández-Cerero D, Fernández-Montes A, Jakobik A, Kolodziej J (2018) Stackelberg game-based models in energy-aware cloud scheduling. In: ECMS. pp 460–467
16. Fernández-Cerero D, Fernández-Montes A, Kolodziej J, Lefèvre L (2018) Quality of cloud services determined by the dynamic management of scheduling models for complex heterogeneous workloads. In: 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC). IEEE, pp 210–219
17. Fernández-Cerero D, Fernández-Montes A, Ortega JA (2018) Energy policies for data-center monolithic schedulers. Expert Syst Appl 110:170–181
18. Fernández-Cerero D, Jakóbik A, Grzonka D, Kołodziej J, Fernández-Montes A (2018) Security supportive energy-aware scheduling and energy policies for cloud environments. J Parallel Distrib Comput 119:191–202. https://doi.org/10.1016/j.jpdc.2018.04.015
19. Ghawi R (2016) Process discovery using inductive miner and decomposition. CoRR. arXiv:1610.07989
20. Girden ER (1992) ANOVA: repeated measures. 84. Sage, Thousand Oaks
21. Gog I, Schwarzkopf M, Gleave A, Watson R.N, Hand S (2016) Firmament: fast, centralized cluster scheduling at scale. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). pp 99–115

22. Group XW, et al. (2016) IEEE standard for eXtensible event stream (XES) for achieving interoperability in event logs and event streams. IEEE Std 1849–2016. pp 1–50. https://doi.org/10.1109/IEEESTD.2016.7740858
23. Gubbi J, Buyya R, Marusic S, Palaniswami M (2013) Internet of things (IoT): a vision, architectural elements, and future directions. Future Gener Comput Syst 29(7):1645–1660
24. Hindman B, Konwinski A, Zaharia M, Ghodsi A, Joseph AD, Katz R, Shenker S, Stoica I (2011) Mesos: a platform for fine-grained resource sharing in the data center. In: Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, NSDI'11, Boston, MA. USENIX Association, Berkeley, CA, USA, pp 295–308
25. Varela-Vaca AJ, Galindo JA, Ramos-Gutiérrez B, Gómez-López MT, Benavides D (2019) Process mining to unleash variability management: discovering configuration workflows using logs. In: Proceeedings of the 23nd International Systems and Software Product Line Conference- Volume 1, SPLC 2019, Paris, France, September 10–14, 2018. p 298
26. Karanasos K, Rao S, Curino C, Douglas C, Chaliparambil K, Fumarola GM, Heddaya S, Ramakrishnan R, Sakalanaga S (2015) Mercury: hybrid centralized and distributed scheduling in large shared clusters. In: USENIX Annual Technical Conference. pp 485–497
27. Leemans SJJ, Fahland D, van der Aalst WMP (2015) Scalable process discovery with guarantees. In: Gaaloul K, Schmidt R, Nurcan S, Guerreiro S, Ma Q (eds) Enterprise, business-process and information systems modeling. Springer, Cham, pp 85–101
28. Liu Z, Cho S (2012) Characterizing machines and workloads on a Google cluster. In: 2012 41st International Conference on Parallel Processing Workshops. pp 397–403. https://doi.org/10.1109/ICPPW.2012.57
29. Lo D, Cheng L, Govindaraju R, Ranganathan P, Kozyrakis C (2016) Improving resource efficiency at scale with heracles. ACM Trans Comput Syst (TOCS) 34:6:1–6:33
30. Mans RS, Schonenberg MH, Song M, van der Aalst WMP, Bakker PJM (2009) Application of process mining in healthcare—a case study in a Dutch hospital. In: Fred A, Filipe J, Gamboa H (eds) Biomedical engineering systems and technologies. Springer, Berlin, pp 425–438
31. Măruşter L, van Beest NRTP (2009) Redesigning business processes: a methodology based on simulation and techniques. Knowl Inf Syst 21(3):267
32. Maruster L, Weijters AJMM, van der Aalst WMP, van den Bosch A (2002) Discovering direct successors in process logs. In: Discovery Science, 5th International Conference, DS 2002, Lübeck, Germany, November 24–26, 2002, Proceedings. pp 364–373. https://doi.org/10.1007/3-540-36182-0_37
33. Maruster L, Weijters AJMM, van der Aalst WMP, van den Bosch A (2002) Process mining: discovering direct successors in process logs. In: Discovery Science, 5th International Conference, DS 2002, Lübeck, Germany, November 24–26, 2002, Proceedings. pp 364–373. https://doi.org/10.1007/3-540-36182-0_37
34. Mendling J (2008) Metrics for business process models. Springer, Berlin, pp 103–133
35. Mishra AK, Hellerstein JL, Cirne W, Das CR (2010) Towards characterizing cloud backend workloads: insights from Google compute clusters. SIGMETRICS Perform Eval Rev 37:34–41
36. Moschakis IA, Karatza HD (2011) Performance and cost evaluation of gang scheduling in a cloud computing system with job migrations and starvation handling. In: 2011 IEEE symposium on computers and communications (ISCC). IEEE, pp 418–423
37. Ousterhout K, Wendell P, Zaharia M, Stoica I (2013) Sparrow: distributed, low latency scheduling. In: Proceedings of the twenty-fourth ACM symposium on operating systems principles. ACM, pp 69–84
38. Pérez-Álvarez JM, Maté A, López MTG, Trujillo J (2018) Tactical business-process-decision support based on KPIs monitoring and validation. Comput Ind 102:23–39
39. Pérez-Castillo R, Fernéndez-Ropero M, Piattini M (2019) Business process model refactoring applying ibuprofen. An industrial evaluation. J Syst Softw 147:86–103. https://doi.org/10.1016/j.jss.2018.10.012
40. Perimal-Lewis L, Teubner D, Hakendorf P, Horwood C (2016) Application of process mining to assess the data quality of routinely collected time-based performance data sourced from electronic health records by validating process conformance. Health Inform J 22(4):1017–1029
41. Piao JT, Yan J (2010) A network-aware virtual machine placement and migration approach in cloud computing. In: 2010 Ninth International Conference on Grid and Cloud Computing. IEEE, pp 87–92

42. Pika A, Wynn MT, Fidge CJ, ter Hofstede AHM, Leyer M, van der Aalst WMP (2014) An extensible framework for analysing resource behaviour using event logs. In: Advanced Information Systems Engineering—26th International Conference, CAiSE 2014, Thessaloniki, Greece, June 16–20, 2014. Proceedings. pp 564–579
43. Reiss C, Tumanov A, Ganger GR, Katz RH, Kozuch MA (2012) Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In: Proceedings of the third ACM symposium on cloud computing. ACM, p 7
44. Reiss C, Tumanov A, Ganger GR, Katz RH, Kozuch MA (2012) Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In: Proceedings of the third ACM symposium on cloud computing, SoCC '12. ACM, New York, , pp 7:1–7:13. https://doi.org/10.1145/2391229.2391236
45. Reiss C, Wilkes J, Hellerstein JL (2011) Google cluster-usage traces: format+ schema. Google Inc., White Paper, Mountain View, pp 1–14
46. Rozinat A, de Jong ISM, Günther CW, van der Aalst WMP (2009) Process mining applied to the test process of wafer scanners in ASML. IEEE Trans Syst Man Cybern Part C 39(4):474–479
47. Sahlabadi M, Muniyandi R, Shukur Z (2014) Detecting abnormal behavior in social network websites by using a process mining technique. J Comput Sci 10(3):393–402. https://doi.org/10.3844/jcssp.2014.393.402
48. Schwarzkopf M, Konwinski A, Abd-El-Malek M, Wilkes J (2013) Omega: flexible, scalable schedulers for large compute clusters. In: Proceedings of the 8th ACM European Conference on Computer Systems. ACM, pp 351–364
49. Sebastio S, Trivedi KS, Alonso J (2018) Characterizing machines lifecycle in google data centers. Perform Eval 126:39–63. https://doi.org/10.1016/j.peva.2018.08.001
50. Shi W, Cao J, Zhang Q, Li Y, Xu L (2016) Edge computing: vision and challenges. IEEE Internet Things J 3(5):637–646
51. Tax N, Sidorova N, van der Aalst WMP (2019) Discovering more precise process models from event logs by filtering out chaotic activities. J Intell Inf Syst 52(1):107–139. https://doi.org/10.1007/s10844-018-0507-6
52. Valencia-Parra A, Ramos-Gutiérrez B, Varela-Vaca AJ, Gómez-López MT (2019) https://github.com/IDEA-Research-Group/ELE
53. Verma A, Pedrosa L, Korupolu M, Oppenheimer D, Tune E, Wilkes J (2015) Large-scale cluster management at google with borg. In: Proceedings of the Tenth European Conference on Computer Systems. ACM, p 18
54. van der Aalst W (2016) Analyzing "lasagna processes". Springer, Berlin, pp 387–409. https://doi.org/10.1007/978-3-662-49851-4_13
55. van der Aalst WMP (2016) Process mining—data science in action, 2nd edn. Springer, Berlin
56. van Dongen BF, de Medeiros AKA, Verbeek HMW, Weijters AJMM, van der Aalst WMP (2005) The prom framework: a new era in process mining tool support. In: Applications and Theory of Petri Nets 2005, 26th International Conference, ICATPN 2005, Miami, USA, June 20–25, 2005, Proceedings. pp 444–454. https://doi.org/10.1007/11494744_25
57. vander Aalst WMP (2011) Analyzing "spaghetti processes". Springer, Berlin. https://doi.org/10.1007/978-3-642-19345-3_12
58. Xiao Z, Song W, Chen Q (2013) Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Trans Parallel Distrib Syst 24(6):1107–1117
59. Ye K, Jiang X, Huang D, Chen J, Wang B (2011) Live migration of multiple virtual machines with resource reservation in cloud computing environments. In: 2011 IEEE 4th International Conference on Cloud Computing. IEEE, pp 267–274