

Challenges in Resource Provisioning for the Execution of Data Wrangling Workflows on the Cloud: A Case Study

Abdullah Khalid A Almasaud, Agresh Bharadwaj, Sandra Sampaio*, and Rizos Sakellariou

Department of Computer Science, The University of Manchester, UK

Abstract. Data Wrangling (DW) is an essential component of any big data analytics job, encompassing a large variety of complex operations to transform, integrate and clean sets of unrefined data. The inherent complexity and execution cost associated with DW workflows make the provisioning of resources from a cloud provider a sensible solution for executing these workflows in a reasonable amount of time. However, the lack of detailed profiles of the input data and the operations composing these workflows makes the selection of resources to run these workflows on the cloud a hard task due to the large search space to select appropriate resources, their interactions, dependencies, trade-offs and prices that need to be considered. In this paper, we investigate the complex problem of provisioning cloud resources to DW workflows, by carrying out a case study on a specific Traffic DW workflow from the Smart Cities domain. We carry out a number of simulations where we change resource provisioning, focusing on what may impact the execution of the DW workflow most. The insights obtained from our results suggest that fine-grained cloud resource provisioning based on workflow execution profile and input data properties has the potential to improve resource utilization and prevent significant over- and under-provisioning.

1 Introduction

Data Wrangling (DW) is the most widely used term to refer to the process of transforming data, from the format in which it was originally collected into a desired format suitable for analysis [4]. The reason for the transformation of “raw” data is the need to turn unrefined data into a valuable asset, from which intelligence is to be obtained and used to benefit business and science. A DW workflow has similarities with traditional Extraction-Transformation-Loading (ETL) processes commonly used in data warehousing [16]. The main similarity consists in that DW encompasses operations for preparing data to be integrated before disseminated, including operations for data cleaning, format transformation, summarization and integration [11]. However, DW and ETL differ in their level of reusability with regards to the number of applications whose requirements each

* corresponding author: s.sampaio@manchester.ac.uk

is able to fulfil; while ETL activities are carefully designed to fulfil the requirements of multiple use cases, DW workflows are tailored for the purposes of a single data analysis task, making their reusability a challenge.

The fact that each DW workflow is tailored to a specific task causes such workflows to significantly differ in their level of complexity, depending on a number of factors, e.g., number of inputs, format of inputs and need for complex transformations [11], availability of metadata [8], need for metadata reconciliation [15], data quality [3], size of inputs, to name a few. While some DW workflows are simple, involving a couple of nodes, other workflows may involve dozens of nodes encapsulating complex processing, such as the transformation of a JSON formatted file into a CSV one, aggregation of values in multiple columns, the joining of two files, etc. In addition, complex DW workflows are usually expensive in terms of the resources they consume and can incur long execution times, leading data wranglers to resort to cloud services for executing their workflows in a reasonable amount of time. As a result, research on resource allocation for complex DW workflows is timely, particularly in domains such as Smart Cities, where typical road traffic analysis workflows present a high-level of complexity, as they perform operations that go beyond simple data manipulations over a single dataset. Thus, the main challenge in this scenario is how to best provision resources to fulfil the performance requirements of this type of workflow, while seeking to find a balance between the conflicting interests of cloud service users, i.e., minimization of financial cost, against those of cloud service providers, i.e., the maximization of resource utilization.

Cloud resource provisioning encompasses all activities that lead to the selection and use of all resources (e.g., CPU, storage and network) needed for the execution of a job submitted by a cloud service user, considering Quality of Service (QoS) requirements and Service Level Agreement (SLA) [14]. Provisioning of resources can be done ‘on-demand’, whereby resources are promptly provided to urgent jobs, or by long-term reservation, where resources are reserved for later use. While each approach presents advantages, on-demand provisioning often causes too many jobs to simultaneously use the same resource, leading to interference and performance degradation. On the other hand, long-term reservation often causes many resources to be in an idle state [1].

This paper considers the problem of cloud resource provisioning for complex and data intensive DW workflows, by providing an investigation into the impact of varying levels of cloud resource provisioning on the performance of these workflows. The main aim is to provide insights that can be used in the development of solutions that answer the following questions:

1. What is a ‘good’ amount of resources to choose for the execution of complex DW workflows, aiming at avoiding significant over- and under-provisioning (i.e., preventing more or less resources than the amount actually needed to be allocated [9])?
2. How can the execution profile of workflows, size of input datasets and intermediate results be used in the development of criteria for resource provisioning?

3. How can the information in (2) above be effectively and efficiently used in resource provisioning?

To obtain useful insights, we show results for a number of simulations exploring the performance behaviour of complex DW workflows under varying levels of resource provisioning, considering the resources that have most impact on these workflows. As a use case, we take a typical data analysis workflow in the Smart Cities domain. Our simulations are performed using a widely used Cloud Workflow Simulator, WorkflowSim [2], as well as real-world data. The main contributions of this paper can be summarized as follows:

- Identification of properties and profile information of complex DW workflows that can be used in cloud resource provisioning.
- Identification of the resources that most impact on the performance of these workflows and their level of impact.
- Design and rationale of a number of simulations for revealing the level of impact of cloud resources on the performance of DW workflows.
- Discussion of results, derived insights, and challenges to be addressed in future research.

The rest of the paper is organized as follows: Section 2 describes some related work. Section 3 provides background information on the workflows used in our investigation. Section 4 describes the rationale behind each performed simulation. Section 5 provides a description of the simulations, the obtained results and discussion, and Section 6 concludes and describes further work.

2 Related Work

Early work on cloud resource provisioning mostly focused on the development of general techniques for static and dynamic provisioning, as the survey by Guruprasad et al. [1] indicates, where approaches more susceptible to resource over- or under-provisioning are described. Greater concern about performance and other SLA requirements led to the development of QoS based techniques, an example being the work by Singh et al. [12]; more specifically, this work suggests that identification, analysis and classification of cloud workloads, taking into account QoS metrics, should be performed before scheduling, to avoid violation of SLA. A survey by the same authors, in [13], classifies various works in cloud resource provisioning according to different types of provisioning mechanisms, and focuses on typical cloud workload types, such as Web sites, online transaction processing, e-commerce, financial, and internet applications as well as mobile computing, which account for the bulk of cloud workloads.

Recent Software Engineering trends towards self-management, minimization of energy consumption as well as the impact of machine learning, complex data preparation and analysis on the success of both business and science have significantly influenced research in cloud resource provisioning. Examples of work addressing self-management include Gill et al. [6], which addresses limitations

in resource management by proposing an autonomic resource management technique focused on self-healing and self-configuration; and Gill and Buyya [5], which addresses self-management of cloud resources for execution of clustered workloads. An example of cloud resource provisioning work considering energy consumption is Gill et al. [7], which proposes a technique for resource scheduling that minimizes energy consumption considering a multitude of resources, in order to better balance the conflicting requirements of high reliability/availability and minimization of the number of active servers. Exploring different types of workloads, the work by Pietri et al. [10] proposes a cloud resource provisioning approach to handle large and complex scientific workflows, where an algorithm for efficiently exploring the search space of alternative CPU frequency configurations returns Pareto-efficient solutions for cost and execution trade-offs.

Similarly to the work by Pietri et al. [10], the work in this paper focuses on cloud resource provisioning for workloads resulting from the execution of complex workflows. Pietri et al. focus on scientific workflows, while we focus on data-intensive DW workflows, which share similarities with subsets of activities found in many scientific workflows, in that these also require Data Wrangling (DW). Traffic DW workflows, in particular, are highly complex because of the presence of functions that go beyond simple data manipulations over a single data unit. Rather, examples of such functions include spatio-temporal join operations using time, latitude and longitude proximities to integrate files, functions to iterate over a number of rows in a file to remove redundant data, etc. Considering that DW workflows are data-intensive and require complex analysis, identification of the resources that may mostly impact on the performance of this type of workload (e.g., via job profiling), and the level of impact of these resources (via experimentation or cloud simulation) can provide awareness of the challenges that need to be addressed.

3 A Traffic DW Workflow

The work in this paper makes use of a DW Workflow from the Smart Cities domain [11]. In particular, this workflow (illustrated in Figure 1) answers the following traffic-related question: *What is the typical Friday Journey Time (JT) for the fragment of Chester Road stretching from Poplar Road to the Hulme area between 17:00 and 18:00?* Note that input *File 1* and *File 2* are “raw” traffic data files, each of size 1GB, describing data from two collection sites on Chester Road (in the city of Manchester, UK). *File 3* holds information about distances between data collection sites across the city, with less than 100KB. Each of the main files is reduced and prepared for integration by having extraneous columns and rows that do not match the specified week day and day time removed, as well as some single columns split into two. *Files 1* and *2* are then merged vertically using the union operation before being horizontally merged with *File 3*. Note that, as *File 1* and *File 2* are significantly reduced at this point, the merge with *File 3* does not incur high execution costs as it generally would if the reduction of these files had not taken place before this merge. The information is

then grouped by ID of collection site before the data is summarised and journey time, calculated. In total, there are 13 operations preceded by an operation for uploading the files onto the environment used.

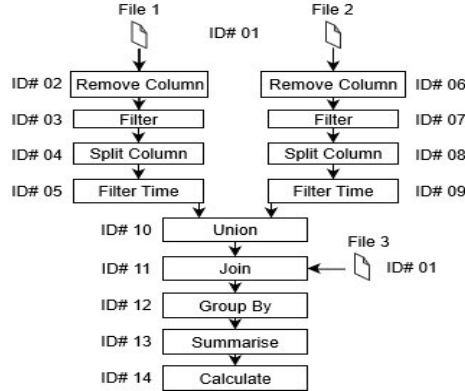


Fig. 1. An illustration of the Workflow used in the simulations.

4 Methodology

Profiling of the workflow described in Section 3¹ has revealed that traffic DW contains operations that are I/O or CPU intensive, or a combination of both, depending on functionality and input/output size. For example, the *ID1-Read* operation is I/O intensive, however, when consuming *File 3*, it incurs a much lower cost than *Files 1* and *2*, due to file size. On the other hand, operations *ID13-Summarise* and *ID14-Calculate* are mostly CPU intensive. To observe over- and under-provisioning, investigation into the levels of performance improvement or degradation, as variations on the amount of the most impacting resources are made, is required. To fulfil this purpose, three sets of simulations were performed using WorkflowSim [2], in which the execution of the workflow was simulated. The first set encompasses simulations where variations of CPU *Million Instructions Per Second* (MIPS) are made while all other simulation parameters remain fixed, to observe how variations in the availability of CPU resources in isolation impact on the execution time of the workflow. The second set encompasses simulations where parameters that define maximum available bandwidth are varied while other parameters remain fixed, including CPU MIPS, to observe how variations in bandwidth in isolation affect the execution time of the workflow. The third set encompasses simulations where the number of VMs are varied while all other parameters remain fixed, to observe how the different types of parallelism, inter- and intra-operator, can be explored and performance gains obtained, while

¹ Using *workflow-profiler* (<https://github.com/intel/workflow-profiler>).

mimicking a cloud environment where multiple nodes are available for the execution of a task. Note that WorkflowSim, by default, performs task clustering by allocating a single Virtual Machine (VM) per branch of a workflow. Also note that the choice of parameters used in the simulations was made by performing additional simulations (outside the main scope of this paper) with each parameter in WorkflowSim, and selecting the ones that had the most impact on execution time. We believe the three sets of simulations we describe serve to help identify a ‘good’ amount of resources for the execution of complex DW workflows, by revealing the number of and which operations are CPU or I/O bound, the extent to which specific resources should be increased or decreased to obtain performance gains and what correlations exist between parameters (such as input size, CPU/IO-bound classification and resource availability), potentially resulting in the development of models that can be used to avoid significant over- and under-provisioning. The results of the three sets of simulations are presented in the next section.

5 Simulation, Results and Discussion

Simulations were carried out by installing and running the WorkflowSim simulator on a Windows 10 computer with an Intel i7-7500U processor and 16GB memory using Java version 1.8.0.191. A number of runs were repeated up to five times to validate that the simulator gave consistent results every time. The results of the three sets of experiments are presented and analyzed next.

Simulation Set 1 (CPU MIPS): For this set of simulations, a single-CPU VM that operates a range of MIPS values equally distributed between a minimum and maximum, $M_{min} = 1000$ and $M_{max} = 5000$, with 1000 MIPS steps is assumed, with all other simulation variables remaining fixed. Figure 2 shows how the total execution time of the complete CPU differs as the value of provisioned MIPS increases. On the level of individual operations of the workflow, the performance improvement varies as shown in Figure 3; we observe that the reduction in execution time is more significant for operations that are CPU intensive. As expected, the main observation is a clear linear inverse relation between workflow execution time and MIPS in this set of simulations, as detailed in Figure 2. The increase in MIPS in this simulation proved beneficial in reducing the execution time of all DW operations in the workflow, although the degree of reduction of the execution time needs to be taken into account as most cloud service providers will charge higher fees for more powerful CPUs.

Simulation Set 2 (Bandwidth): I/O bound tasks are usually affected by the bandwidth and transfer rates available in the execution environment. To explore potential execution time reductions, a number of three different bandwidth values are used in this set of simulations. The values used are 1, 15 and 225 MB/s, with all other parameters remaining fixed. It can be observed that the impact of the variation in bandwidth on performance may not be as profound as it was

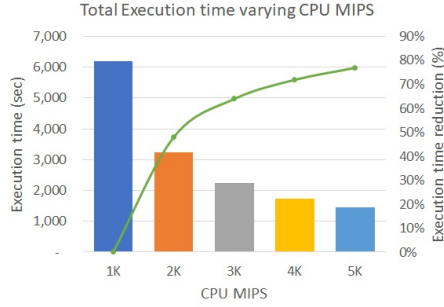


Fig. 2. Workflow execution time with varying CPU MIPS values.

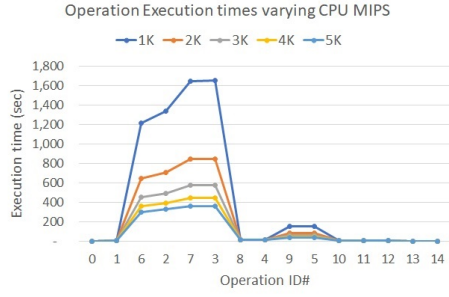


Fig. 3. Execution times of individual operations varying CPU MIPS.

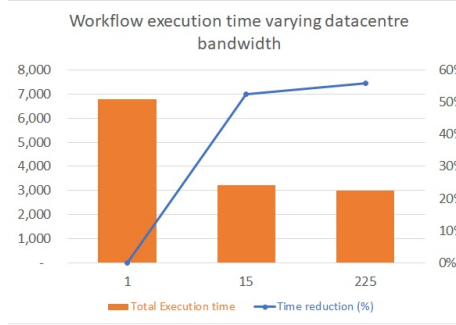


Fig. 4. Workflow execution time with varying max available bandwidth.

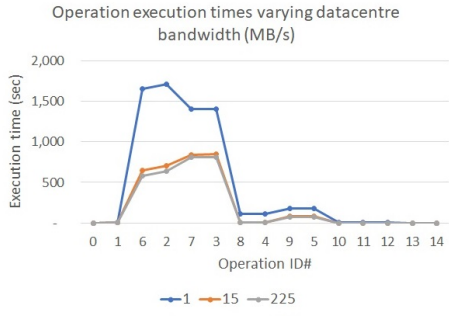


Fig. 5. Execution times of individual operations for each bandwidth.

in the case of CPU MIPS, as shown in Figure 4. Considering the execution time of individual operations in the workflow, it is also observed that execution time improvements are more significant for certain operations, as shown in Figure 5, specifically, those that were profiled as both I/O and CPU bound and that also process large inputs. Even though, execution time improvements are not equally significant for all operations, increase in bandwidth is still beneficial for reducing the execution time of the whole workflow. It is worth pointing out that, increasing bandwidth from 15 MB/s to 225 MB/s, leads to very small savings in execution time. This observation raises the question of whether this increase is worth paying for.

Simulation Set 3 (Number of Virtual Machines): In this set of simulations, the number of VMs used in the execution is varied while other parameters remain fixed. First, the variation is performed on the “original” two-branched workflow, to a maximum of the number of branches in the workflow, exploring *inter-operator parallelism*. Next, the input data files are partitioned so that the same workflow operations are performed on fragments of the original files, increasing

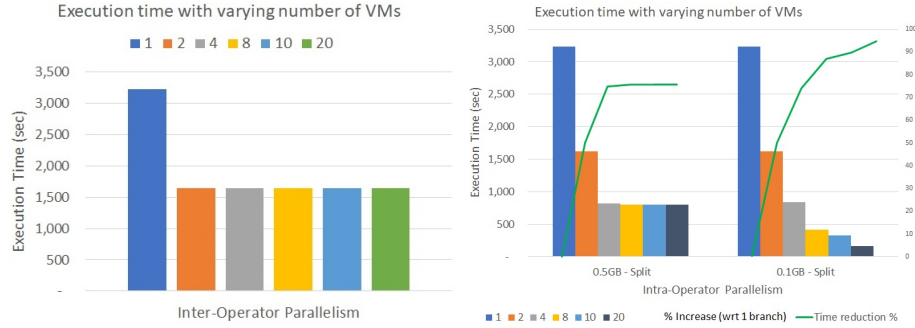


Fig. 6. Workflow execution time applying Inter-Operator Parallelism.

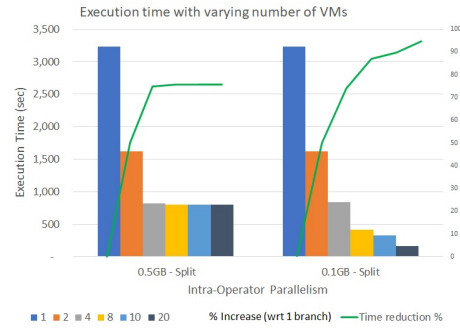


Fig. 7. Workflow execution time applying Intra-Operator Parallelism.

the level of parallelism by exploring *intra-operator parallelism*. When exploring **inter-operator parallelism**, it is observed that the increase in number of VMs does not have any impact on the timings of individual operations, but it reduces the total execution time of the complete workflow, by assigning operations located in different branches of the workflow to run on different VMs. Figure 6 shows how the increase in VMs is beneficial, up to the number of branches in the workflow. To explore **intra-operator parallelism**, the input data is partitioned by a factor of 2 and 10, combining both types of parallelism, as shown in Figure 7. It is observed that the total workflow execution time on a single VM is not significantly affected. However, data partitioning allows an increase in the degree of parallelism, by increasing the number of VMs used in the execution of the workflow. This results in a reduction of the total execution time of the whole workflow (as shown in Figure 7), as partitions of the same file are simultaneously input, processed and output on different VMs. Further increases in the number of VMs without further data partitioning can lead to no performance gains, as can be seen from the *0.1GB-split* case in Figure 7, where an increase to 20 VMs is ideal for obtaining performance gains, but if further increases are desired without further data partitioning, i.e., to more than 20 VMs, no further performance gains are obtained.

Discussion: Three main observations are derived from the results presented in the previous section, discussed in the following: **(1)** CPU MIPS is the parameter that mostly impacts execution time and one of the most costly cloud resources. However, a balanced combination of CPU MIPS provisioning with the provisioning of other impacting resources can result in financially viable parameter configurations, while still providing similar performance gains. **(2)** Bandwidth has a more modest impact on execution time, showing less significant performance gains with increases in availability, as the size of intermediate results gradually decreases due to the application of data reduction operations, rendering no more than 56% improvement at best. Presence of data reduction opera-

tions early in the workflow execution can potentially lessen the benefit of higher bandwidths, generating opportunities for resource release before the workflow execution is over. **(3)** Variations in the number of provisioned VMs show a substantial impact on execution time, particularly when both inter- and intra-operator parallelisms are combined to speed up execution. The extent to which performance gains are observed depends mainly upon the number of workflow branches, limiting exploitation of inter-operator parallelism, and intermediate data size, limiting exploitation of intra-operator parallelism. Clearly, not all operations in the workflow benefit from higher numbers of VMs, particularly those that input, process and output smaller data sizes. Finally, resource balancing involving multiple resources for obtaining execution time reductions incurs different cost and performance implications, and so an effective solution to the problem of finding a ‘good’ amount of resources to balance financial cost and performance benefits, avoiding under- and over-provisioning, probably involves combining not only the Pareto-efficient set of configurations that finds the best cost-benefit balance involving multiple resources for the execution of one job, but also for several jobs that may be waiting to be executed simultaneously.

6 Conclusions and Further Work

To help finding answers to the research questions that motivate this paper (Section 1), we performed a number of simulations using a representative DW workflow. The results have shown that, depending on the execution profile of the DW workflow, more than one resource can have a significant impact on execution performance; and that job execution profiles, if considered when provisioning cloud resources, have the potential to improve decision making and avoid over- and under-provisioning. While choices regarding which subset of resources to focus on and their provisioning levels have an impact on performance and financial costs, these decisions are, to a large extent, job-dependent. Therefore, we believe that models to find the configurations that return the best cost-performance trade-off for a job, such as the work by Pietri et al. [10], should be extended to consider multiple resources as well as multiple jobs, based on the execution profile of the individual jobs and their performance requirements. Numerous challenges in developing such a solution need to be faced, such as: (i) how to efficiently obtain profiles of the jobs, which may require a large number of experiments or simulations; (ii) how to accurately identify the most relevant profile metadata to be used for cloud resource provisioning; (iii) how to devise an efficient and effective mechanism for making use of the profile information at the time of cloud resource provisioning. We intend to investigate these challenges in the future, experimenting also with a variety of different DW workflows. One direction to address these challenges may be that job profiles are generated at run time and Machine Learning or related training techniques become important components of an effective solution.

Acknowledgement: Partial support from the H2020 I-BiDaaS project (grant agreement No. 780787) is gratefully acknowledged.

References

1. Bhavani, B.H., Guruprasad, H.S.: Resource provisioning techniques in cloud computing environment: A survey. *International Journal of Research in Computer and Communication Technology* **3**, 395–401 (2014)
2. Chen, W., Deelman, E.: Workflowsim: A toolkit for simulating scientific workflows in distributed environments. *IEEE 8th International Conference on E-Science* pp. 1–8 (2012)
3. de F. Mendes Sampaio, S., Dong, C., Sampaio, P.: DQ²S - A framework for data quality-aware information management. *Expert Syst. Appl.* **42**(21), 8304–8326 (2015)
4. Furche, T., Gottlob, G., Libkin, L., Orsi, G., Paton, N.W.: Data wrangling for big data: Challenges and opportunities. In: *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France.* pp. 473–478 (2016)
5. Gill, S.S., Buyya, R.: Resource provisioning based scheduling framework for execution of heterogeneous and clustered workloads in clouds: from fundamental to autonomic offering. *Journal of Grid Computing* **17**(3), 385–417 (2019)
6. Gill, S.S., Chana, I., Singh, M., Buyya, R.: RADAR: self-configuring and self-healing in resource management for enhancing quality of cloud services. *Concurrency and Computation: Practice and Experience* **31**(1) (2019)
7. Gill, S.S., Garraghan, P., Stankovski, V., Casale, G., Thulasiram, R.K., Ghosh, S.K., Ramamohanarao, K., Buyya, R.: Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge. *Journal of Systems and Software* **155**, 104–129 (2019)
8. Hellerstein, J.M., Sreekanti, V., Gonzalez, J.E., Dalton, J., Dey, A., Nag, S., Ramachandran, K., Arora, S., Bhattacharyya, A., Das, S., Donsky, M., Fierro, G., She, C., Steinbach, C., Subramanian, V., Sun, E.: Ground: A data context service. In: *CIDR 2017, 8th Biennial Conference on Innovative Data Systems Research, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings* (2017)
9. Nahrstedt, K.: To overprovision or to share via qos-aware resource management? In: *Proceedings of the Eighth International Symposium on High Performance Distributed Computing (Cat. No.99TH8469), Redondo Beach, CA, USA, 6-6 Aug.* pp. 205–212 (1999)
10. Pietri, I., Sakellariou, R.: A Pareto-based approach for CPU provisioning of scientific workflows on clouds. *Future Generation Computer Systems* **94**, 479–487 (2019)
11. Sampaio, S., Aljubairah, M., Permana, H.A., Sampaio, P.: A conceptual approach for supporting traffic data wrangling tasks. *Computer Journal* **62**(3), 461–480 (2019)
12. Singh, S., Chana, I.: Q-aware: Quality of service based cloud resource provisioning. *Computers & Electrical Engineering* **47**, 138–160 (2015)
13. Singh, S., Chana, I.: Cloud resource provisioning: survey, status and future research directions. *Knowledge and Information Systems* **49**(3), 1005–1069 (2016)
14. Singh, S., Chana, I.: A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of grid computing* **14**(2), 217–264 (2016)
15. Stonebraker, M., Ilyas, I.F.: Data integration: The current status and the way forward. *IEEE Data Eng. Bull.* **41**(2), 3–9 (2018)
16. Vassiliadis, P.: A survey of extract-transform-load technology. *International Journal of Data Warehousing and Mining* **5**, 1–27 (2009)