Generating Workload for ERP Applications through End-User Organization Categorization using High Level Business Operation Data

Gururaj Maddodi
Utrecht University
Utrecht, Netherlands
g.maddodi@uu.nl

Slinger Jansen
Utrecht University
Utrecht, Netherlands
slinger.jansen@uu.nl

Rolf de Jong
AFAS Software
Leusden, Netherlands
rolf.dejong@afas.nl

ABSTRACT

For software companies performance testing is an essential part of new application development. In this paper we present a performance engineering method that extracts the workload of an existing legacy ERP application with more than 1 million users and generates workload for a radically new version of the application. The workload is used to classify groups of end user organizations, i.e., enterprises whose customers are end users of the application, with unsupervised machine learning techniques. The method shows that (1) workload for new application testing and architecture validation can be generated from legacy application behavior, (2) end user organizations have significantly different usage patterns, and (3) for ERP applications, high-level operations, such as a salary calculations, provide a useful method for analyzing and generating workload, as opposed to for instance low level page views. The method is evaluated within a Dutch software company, where it is found to be accurate and effective for performance engineering.

KEYWORDS

Workload Generation; Software Usage Behavior; Unsupervised Learning; Software Performance Engineering

ACM Reference Format:

Gururaj Maddodi, Slinger Jansen, and Rolf de Jong. 2018. Generating Workload for ERP Applications through End-User Organization Categorization using High Level Business Operation Data. In *ICPE '18: ACM/SPEC International Conference on Performance Engineering, April 9–13, 2018, Berlin, Germany*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3184407.3184432

1 INTRODUCTION

Performance testing of software under development is essential for the software development life-cycle, as it helps to make choices for architecture to improve performance. For accurate results in performance testing, it is essential to use realistic workload; workload that the software system is expected to handle in production use. For software under development, a similar application in a similar domain or a predecessor can be used as a usage reference to estimate the kinds of load in production use. The usage data from an application in production can be used to detect usage patterns and to simulate realistic workloads, even if the new architecture and even some of the application's features are significantly different.

Several works have investigated workload modeling using production usage data. Many such research works use low-level application usage parameters, such as page accesses [11, 18, 22] or resource-level metrics [2, 4]. However, in a highly complicated software system such as Enterprise Resource Planning (ERP) software, using low-level concepts such as page access is too course grained, as for instance some pages might incur complex operations such as salary and pension calculations. A high-level abstraction helps to capture the features of ERP software and map them from the old application to a new application (also on different types of interfaces). We hypothesize and illustrate in this paper that with the behavior patterns and high level operations, we can create more accurate workloads for realistic usage simulation on new products.

In order to investigate the production workload of ERP systems, several aspects have to be considered. ERP applications are complicated and have several distinct domains that serve needs of different types of end-user organizations. Since these domains are distinct, different workload patterns exist and analysis needs to be performed separately for each. There is also a hierarchical structure in ERP software: for example, users receive an invoice for the order they place. The invoice contains details of one or more items that were ordered at a particular time. Processing more invoices require more resources as will more items within each invoice. These features are not captured by low-level metrics such as page access. For these reasons, high-level abstractions based on business operation metrics is proposed to get more realistic insight into usage of an ERP software and simulate workload for performance testing.

The contributions of this paper are: (1) Workload metrics are defined using High-level abstractions to identify categories of end-user organizations that help to translate workload patterns of existing application to new applications, (2) An exploratory study of the usage of an ERP application from a big software producing company is presented combining several machine learning and statistical methods, and (3) A mechanism is presented to simulate workload for an ERP application that is under development at the same company, based on patterns discovered from the older application.

The remainder of the paper is organized as follows: Section 2 describes how the work in this paper is related to and is different from

ICPE '18, April 9–13, 2018, Berlin, Germany

ACM ISBN 978-1-4503-5095-2/18/04…$15.00
https://doi.org/10.1145/3184407.3184432

similar works; with main similarity being that all the works employ some type of machine leaning technique to detect patterns in usage and use them to simulate workload while primary difference being that we use high-level abstractions for end-user organization categorization and workload simulation on new application. In Section 3, a description of the research motivation, case-study context, and research questions is presented. Section 4 describes the research method being used to address the research questions. A description of data attribute selection strategy and data collection for facilitating identification of categories of end-user organization is discussed, followed by an explanation of each step of the knowledge discovery process, and mechanism for simulating workload in new application is discussed in subsections. Section 5 shows the results from the statistical experiments applied on the application usage data. In particular, we present the identified clusters, and simulate workload on new version of the software taking examples from the discovered knowledge. Section 7 present discussion of results and the threats to validity of the study; and the section 8 concludes the paper, where we show that more accurate workload can be generated when taking behavioral aspects of customer groups into account.

2 RELATED WORKS

Workload categorization and simulation using existing usage data is studied extensively in literature. The application areas include e-commerce, cloud applications, big data applications, etc. The results of categorization has been used for performance testing, load prediction, and in some instances resource scaling. Not much research is done on ERP applications, although on-line shopping and e-commerce applications come close. In many of these works, the main theme is to categorize users of the software, generating concrete groups of users or tasks with definitive usage patterns, which helps to model the behavior accurately on a test setup. In this section, a few of the related works in the area of workload and performance testing is presented.

In Menasce et al. [15], workload characterization based on customer behavior graphs for e-commerce sites is investigated. State transition graphs called customer behavior model graphs (CBMG) are used to capture user navigation patterns. User behavior is expressed in terms of sessions, i.e. a sequence of requests users perform during on-line shopping such as browse, search, and add to cart. Metrics used are average session length, number of purchased items per customer, and visit-to-buy ratio. K-means clustering is used to categorize the customers. The CBMG associated with a specific cluster has certain characteristics in terms of session length, buy-to-visit ratio, add-to-cart-to-visit ratio, etc.

Moreno et al. [18] study Google cloud trace logs to identify patterns in user requests. The authors defined cloud workload in terms of "users" and "tasks", where user is a combination of submission rate, CPU, and memory requested while task is combination of session length, average CPU, and memory utilization. Users and tasks are clustered using k-means algorithm. From the resulting clusters workload is simulated and compared to the production load. In Elijorde et al. [4], dynamic resource allocation for virtual machines (VM) is proposed based on clustering of virtual machines according to workload patterns. The authors identify workload metrics which help to cluster workload, i,e, parameters such as CPU and memory requested and CPU and memory used are used to classify workloads. Here the users are identified on the basis of the amount of resources requested on the VMs. These works define high-level metrics in terms of customer/user of the application and session/task they produce. We were inspired by these works to use high level operations. However, our work is in a different domain and concerns workload generation for a new version of an ERP application, as opposed to simulating workload for the same application.

In Kulkarni et al. [13] a classification algorithm is proposed for categorizing cloud workloads. The classification is done to identify if the workloads are based on metrics such as their I/O, computation, or communication (network I/O) intensiveness on BigDataBench workloads on Cloudera Impala SQL query engine. The queries are classified as belonging to either of the three categories, and based on the type, infrastructure choice is made. In this work the authors know the categories of workloads that exist in the system, hence a supervised approach is sufficient while we need to discover the patterns which will help us in decision making in architecture selection.

In Mian et al. [16], workload prediction models for data intensive workloads are created. Although the authors acknowledge that the variation in tenant types (Online Transaction Processing, Online Analytical Processing, or a combination) could affect the resource, the models are built considering multi-tenant scenario. In Herbst et al. [7], proactive forecasting of resource requirement based on classification of workload in terms of intensity is proposed. The workload classes are based on intensity metrics such as burstiness and relative monotonicity. Workload classes are predetermined based on historical data and the framework dynamically forecasts workload resource requirement. This is not a problem when the types of workload is known, but without knowing the workload types building workload models is inaccurate.

In [17], an analysis of Google cloud back-end workload is presented for forecasting future workloads. The workload is defined in terms of tasks, and a task classification methodology is proposed such that ones with similar resource requirements are grouped together. The workload dimensions used for task classification are task duration, average core usage, and average memory usage. For categorization, the authors use K-means clustering algorithm. Then, the classes which are based on individual workload dimension is merged if the coefficient of variation between then are almost equal to get the final workload classes. Similarly, [19] study Google cloud workload by clustering tasks that are received based on resource required. This is used to predict resource requirement for future tasks that are predicted to belong to a specific cluster.

Aggarwal et al. [1] use clustering to find characteristics of MapReduce jobs on Apache Hadoop. For clustering, the workload metrics selected are map and reduce tasks, i.e. bytes read/written to the file system, format of the input/output files, and type of compression. In [20], Hadoop job workloads are categorized according to their data transformation patterns and running times. The parameters chosen were input size, shuffle size, output size, job duration, map task time, and reduced task time; it also used K-means clustering for workload types. Jia et al. [8] use clustering to categorize workload into different types of queries on big data workloads. The authors

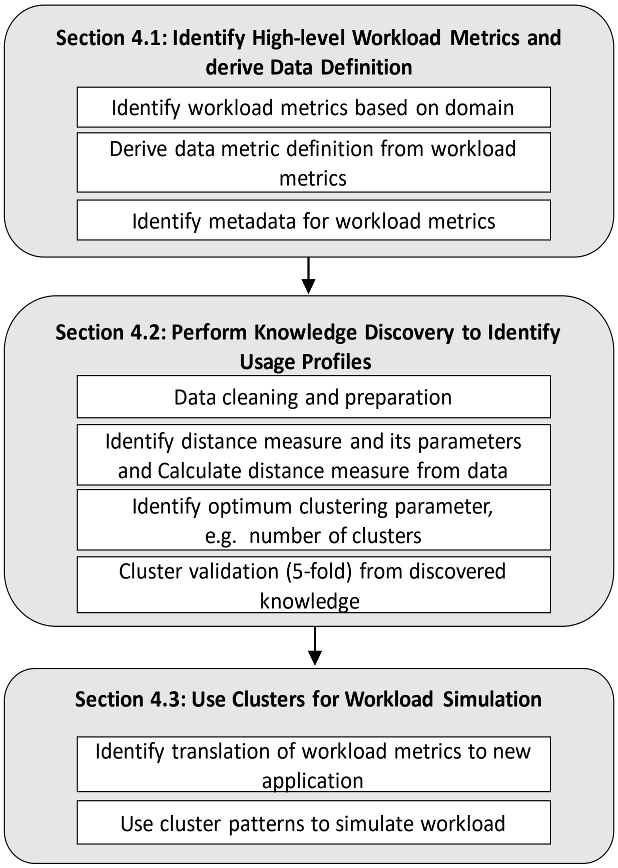

Figure 1: The research steps are modeled in a process diagram. The paper follows these steps and the Section headings are the same as the titles of the main process steps.

identified 32 different types of workloads such as sort, word count, and identified 45 metrics associated with each workload. In these works, the level of abstraction identified to categorize workloads is low and in terms of metrics on infrastructure side rather than from the user point of view.

We define high level abstraction for clustering, which makes it applicable to other enterprise applications. While in [15, 18] the authors do consider a somewhat higher level of abstraction, they are still tied to the application itself rather than to the underlying business model. In our previous work [14], we surveyed a number workload generation methods and performance testing methods available and we describe the workload and performance testing used at the case company in more detail.

3 RESEARCH CONTEXT

The research concerns a theory testing case study; our aim is to prove that high level metrics and end user organization clustering are supportive pillars for reliable workload generation in ERP.

3.1 Case-study Scenario

The research is conducted at a Dutch ERP software vendor called AFAS Software. The privately held company currently employs over 350 people, annually generates €100 million in revenue, and has been highly profitable since it was founded. The case-company currently delivers a fully integrated ERP suite called *Profit 2016*, which is used daily by more than 1.000.000 professional users from more than 10.000 End-User organizations.

From here on in this paper, we will refer to the case-company as ERPComp and their currently most successful product *Profit 2016* as ERPSoft. ERPSoft comprises of several distinct domain modules that cater to different needs of end-user organizations ranging from small businesses to huge retailers, and businesses in the domains of health care, educational institutions, retailers, accountancies, etc. The domains in ERPSoft that facilitate different types of business operations are: *Sales* which facilitates companies selling their products to buyers; *Purchasing* for companies buying raw materials from other suppliers; *human resource management (HRM)* which automates operations such as salary and leave registration.

ERPComp is developing a new version of their software, which we will call as ERPSoftNext. ERPSoftNext is a cloud-based application, developed using a model-driven development approach. Because ERPSoftNext will replace ERPSoft, it is expected to handle similar usage, even if the architectures of the products differ significantly.

The research questions that we investigate in this study are:

(1) What are the high-level metrics that can define the behavior of end-user organizations using an ERP application?
(2) what types of end-user organizations groups exist in an ERP application usage?
(3) How can usage sourced from high-level workload metrics be used for simulation in future versions of software products with different architectures?

The research follows an investigative approach into performance engineering using machine learning in the context of a case study. The research steps are modeled in Figure 1. In the next Section, we explain how the workload is extracted from the existing 'legacy' software application.

4 RESEARCH METHOD AND APPROACH

In this section, we describe the research methods of describing the data attributes, data collection strategy, and data analysis and presentation techniques.

4.1 Usage Data Extraction

4.1.1 Workload Metrics Identification and Data Attribute Derivation: In order to translate workload from ERPSoft running on different types of interfaces to ERPSoftNext, we have chosen to express the application usage in terms of high-level abstractions or workload metrics, i.e. interface and architecture independent parameters. Experts at the ERPComp were consulted to identify the metrics, which along with their definitions, enumerated below:

(1) **End-user organizations**: End-user Organizations or simply organizations purchase the license for the use of

the ERPSoft provided by ERPComp. The licenses vary based on requirements of the ORGANIZATION.

(2) **Roles**: ROLES indicate the personnel involved with or within an ORGANIZATION in performing business operations. Different types of ROLES have different access levels and functionality available in the application. ROLES can be distinguished into: Customer (who purchases an end product), Supplier (who supplies raw materials), and Employee.

(3) **Business-Events**: BUSINESS-EVENTS are the business operations performed by ROLES. BUSINESS-EVENTS can be classified into four types based on the domain: 1. Sales Related BUSINESS-EVENTS, are the ones where customer ROLE is involved along with one or more items, 2. Purchases Related BUSINESS-EVENTS, involve supplier and items, 3. HRM Related BUSINESS-EVENTS, involve the employees, and 4. Other BUSINESS-EVENTS that are not associated with any ROLES, such are mostly bookkeeping tasks executed by ORGANIZATIONS.

(4) **Business-Event Attributes**: BUSINESS-EVENT ATTRIBUTES are the details of items that are associated with a BUSINESS-EVENT. An example for BUSINESS-EVENT ATTRIBUTES is individual products purchased by a customer from an ORGANIZATION such as an electronic store under a sales order. The order might contain for example a monitor and mouse in which case both are BUSINESS-EVENT ATTRIBUTES under BUSINESS-EVENT sales order. Only sales and purchase domains have BUSINESS-EVENT ATTRIBUTES associated with **business-event**s as a single ROLE can have many items purchased/sold under a **business-event**, while HRM **business-event**s have only one ROLE with details such as salary slips and leave requests.

There is a hierarchical pattern that can depict the operation of an ORGANIZATION in an abstract way using the metrics described above, i.e. ORGANIZATIONS have instances of ROLES that perform BUSINESS-EVENTS which themselves have BUSINESS-EVENT ATTRIBUTES. This pattern is used in modeling on ERP application, please refer to [21] for more details.

Table 1 shows the ROLES, and associated BUSINESS-EVENTS, and if BUSINESS-EVENTS LINES are present in ERPSoft for that specific domain. We study only the BUSINESS-EVENTS associated with the *sales* domain because it is the most used part of the application.

From the chosen high-level workload metrics described above, data attributes were constructed. As in the work of [15], where the authors use ratios such as visit-to-buy ratio to represent user behavior, we use the ratio of BUSINESS-EVENTS to ROLE and BUSINESS-EVENT ATTRIBUTES to BUSINESS-EVENT to represent behavior of an ORGANIZATIONS. We call them *BE-factor* and are described in the equations 1 and 2.

$$BE - factor = \frac{No.\ of\ Business - Events}{Instance\ of\ Role} \tag{1}$$

$$BE - factor = \frac{No.\ of\ Business - Event\ Attributes}{Instance\ of\ Business - Event} \tag{2}$$

The *BE-factors* represent the number of BUSINESS-EVENTS that are generated by an instance of ROLE and BUSINESS-EVENT ATTRIBUTES for an instance of BUSINESS-EVENT in an ORGANIZATION.

Table 1: List of Domains in ERPSoft, and Associated ROLES and BUSINESS-EVENTS and BUSINESS-EVENT ATTRIBUTES

Domain/Roles	Business-Events	Business-Event Attributes
HRM/Employee	Salary Slip generation	No
	Employee data change	
	Time Sheet filling	
	Absence Registration	
	Leave Registration	
Sales/Customers	Sales quotation creation	Yes
	Sales order creation	
	Delivery note creation	
	Invoice (sales) generation	
	Invoice (projects) generation	
	Invoice (subscriptions) generation	
Purchases/ Suppliers	Purchase quotation creation	Yes
	Purchase order creation	
	Goods received creation	
	Invoices (purchases) generation	
General/ Automated	Financial entries addition	No
	Workflow mutations	
	Time sheet (invoiced) generation	
	Cost estimation	

BE-factors were chosen because of the assumption that it represents the behavior of the ROLES associated with the ORGANIZATIONS more accurately in terms of use of ERPSoft, and hence the behavior of the ORGANIZATION. This leads to identifying groups of similar ORGANIZATIONS that use the software in a similar manner. Using absolute BUSINESS-EVENT numbers will group the ORGANIZATIONS based on their sizes, as bigger ORGANIZATIONS will have more instances of ROLES (for e.g. customers), and hence more BUSINESS-EVENTS.

Since We choose to use the sales part of ERPSoft, the *BE-factors* in the data will be related to sales domain. Sales domain includes subtypes such as: sales, projects, and subscriptions that ORGANIZATIONS sell to the ROLE of type CUSTOMER, varying in the way in which the selling process is arranged. We only consider invoices to calculate for BE-factors. This is because orders and delivery notices are highly correlated with invoices, i.e. invoices follow orders and deliveries in the process.

The distribution of *BE-factor* data most of the times shows a positive skew. In such cases a single value does not describe the distribution very well. Taking quartiles that divide the distribution into four regions or at three points captures data in finer granularity and helps to reduce error. The points that divide distribution at 25% (1qtl), 50% (med), and 75% (3qtl) are taken. Maximum and minimum points are not considered as they do not add any information (mostly 0 or 1 for min and very high value for max). An example of an attribute is the *BE-factor* is sales invoices per customer denoted as $SIpC_{1qtl}$ with 25% value denoted using a subscript. Table 2 shows the *BE-factor* data attributes with the quartiles from rows 2 to 7. The remaining rows, represents the metadata about the ORGANIZATIONS. Rows from 8 to 10 describes the type of items or goods the ORGANIZATIONS sell represented as SItem, PItem, and SuItem for sales, projects, and subscription respectively. Rows 11 to 15 represent the license or sub-functionality of ERPSoft that ORGANIZATIONS use represented by LType. In the table 2, for Boolean variables "yes" value represents that the feature is present and "no" value represents that the feature is not present.

Data was extracted from production database servers at ERP-Comp running ERPSoft. SQL queries were written such that only the active ROLES within an ORGANIZATION, i.e. ROLES having at least one BUSINESS-EVENT over the period of one year, were selected. This was due to the assumption, along with advice from experts at the ERPComp, that the ROLE instances without any BUSINESS-EVENTS for over a year can be considered not to be associated anymore with the ORGANIZATION.

4.1.2 Data Cleaning and Preparation: Initial inspection of the data showed that there were several duplicates, as all the attributes of some record pairs had identical or almost identical values. This was due to clients replicating their production database instances and using the new instance for testing purposes, which resulted in duplicate record pairs. With help from experts at ERPComp, we used specific attributes of few selected tables, i.e. the tables used were Financial Entries, Projects, and Subscriptions to identify similar databases. Based on expert suggestion, the data was collected from the mentioned tables that have at least 10,000 or more records. We calculated hash of the resulting records and compared all the database pairs. The number 10,000 was chosen based on the expert suggestion because the probability of two clients having 10,000 similar records in those tables is highly unlikely and it could be said with nearly 100% confidence that the compared databases are duplicate pairs and one them is a test instance. It was not clear which of the identified duplicate pairs was the production database and which one was the test copy. Since the duplicate pairs have similar database names in that the ORGANIZATION license number is same, but the characters representing instances were different. One convention we used was that the database instance with higher characters for instance representation as test instances.

4.2 Clustering and End User Organization Classification

In order to categorize the ORGANIZATIONS, several steps need to be followed. First, a distance or dissimilarity measure should be identified to represent dissimilarity between the data points. Second, an algorithm and its input parameters should be determined to categorize the ORGANIZATIONS. Finally, the resulting categorization should be verified for stability. In the subsections below, we describe the steps.

4.2.1 Distance Measure. Typically, clustering algorithms do not take the raw data as it is. A pair-wise measure of distance or dissimilarity matrix between data points has to be provided to the algorithm. For datasets containing a mix of different types of data (i.e. continuous or numeric, nominal etc.), a distance measure called Gower Distance [6] is popularly used. Gower Distance calculates a pair-wise dissimilarity matrix by combining distance/dissimilarity measures of each attribute of data points.

The Gower distance algorithm needs to calculate a dissimilarity matrix. Based on the dataset we have, the following specification was used: As an ORGANIZATION cannot have negative value for BUSINESS-EVENTS, for instance there cannot be negative number of orders being created, the *BE-factor*s are a ratio-scaled continuous

Table 2: List of Identified Metrics for Clustering ORGANIZATIONS with their Relevance and Data Type

Attribute	Relevance	Type
Org_{id}	Identifier which uniquely identifies an ORGANIZATION	-
$SIpC_{1qtl}$ $SIpC_{med}$ $SIpC_{3qtl}$	Number of sales invoices per customer with quartiles	$\mathbb{R}_{\geq 0}$
$SILpSI_{1qtl}$ $SILpSI_{med}$ $SILpSI_{3qtl}$	Number of invoice lines per sales invoice with quartiles	$\mathbb{R}_{\geq 0}$
$PIpC_{1qtl}$ $PIpC_{med}$ $PIpC_{3qtl}$	Number of project invoices per customer with quartiles	$\mathbb{R}_{\geq 0}$
$PILpPI_{1qtl}$ $PILpPI_{med}$ $PILpPI_{3qtl}$	Number of invoice lines per project invoice with quartiles	$\mathbb{R}_{\geq 0}$
$SuIpC_{1qtl}$ $SuIpC_{med}$ $SuIpC_{3qtl}$	Number of subscription invoices per customer with quartiles	$\mathbb{R}_{\geq 0}$
$SuILpSuI_{1qtl}$ $SuILpSuI_{med}$ $SuILpSuI_{3qtl}$	Number of invoice lines per subscription invoice with quartiles	$\mathbb{R}_{\geq 0}$
$SItem_{article}$ $SItem_{ToW}$ $SItem_{course}$	Indicates if an ORGANIZATION uses physical goods, uses type of work (ToW, i.e. billing based on work type and time), and/or courses in sales invoices	{yes, no}
$PItem_{article}$ $PItem_{ToW}$ $PItem_{course}$	Indicates if an ORGANIZATION uses physical goods, uses type of work (ToW, i.e. billing based on work type and time), and/or courses in project invoices	{yes, no}
$SuItem_{article}$ $SuItem_{ToW}$ $SuItem_{course}$	Indicates if an ORGANIZATION uses physical goods, uses type of work (ToW, i.e. billing based on work type and time), and/or courses in subscription invoices	{yes, no}
$LType_{SB}$	Small Business (SB) license is a limited set of functionality available for ERP and Accountancy focusing on use in smaller ORGANIZATIONS, e.g. 1 to 5 employees	{yes, no}
$LType_{ERP}$	ERP license offers functionality to automate the secondary processes in a ORGANIZATION such as bookkeeping, HRM and payrolling, order management, project administration, subscription administration and general functionality like reporting, workflow management, and BI	{yes, no}
$LType_{HRM}$	HRM/Payroll license is a limited set of ERP focusing on HRM and payrolling mostly used by bigger ORGANIZATIONS having another solution for ERP	{yes, no}
$LType_{ACC}$	Accountancy (ACC) license is a specific set of ERP focusing on administration offices such as accountants, there is some limitation of the functionality they use and also some specific functionality for this kind of ORGANIZATIONS	{yes, no}
$LType_{ASB}$	Accountancy and Small Business (ASB) license is a combination of Accountancy and Small Business features	{yes, no}

variables [10]. For example $SIpC_{1qtl}$ (along with variables in row 2 to 7 in the table 2) are ratio-scaled continuous variables. Also, as there can be '0' values for the BE-factors, we calculate the rank scores and then treat them as interval-scaled [10]. The binary variables, such as $SItem_{article}$ (and variables from row 8 to 21 in the table 2), indicate a particular item type or license is present or not and there is no ambiguity for a 'NO' value. Hence they are treated as symmetric binary variables [10].

4.2.2 Clustering and Selection of Optimum Input Parameters. Since clustering is an exploratory approach, generally the best parameters to get the optimal clusters for a given algorithm are not known. There is a plethora of techniques to determine the input parameters to clustering algorithms and validate clustering quality. One technique commonly used is the Internal Cluster Validation techniques [12] which does not use any external information such as class memberships (as many times it is not available beforehand). One of the main parameters required for most clustering algorithms is number of clusters to be generated. There are several internal validation methods that could be used to estimate the optimal cluster number, we used a few of them as they suited the dataset that we have (for e.g. some measures only work on continuous numeric variables, and so doesn't suit the dataset):

(1) **Silhouette Width** is a measure of similarity of an data point to its own cluster compared to other clusters. The value ranges from -1 to 1. A higher value suggests better clusters.
(2) **Dunn Index** helps to identify clusters that are compact, with a small variance among the members of the clusters. A higher value indicates well separated clusters with high correlation among cluster members.
(3) **Connectivity** indicates the degree of connectedness of clusters determined by k-nearest neighbors. A lower value indicates lower correlation for cluster members to their nearest neighbors in the data space.
(4) **Davies-Bouldin (DB) Index** calculates for each cluster, ratio of within cluster distance to between cluster distance. Then over all the clusters an average value is calculated. A smaller value represents better clustering result.

We use Partitioning Around Medoid (PAM) clustering method, which is a partition-based method. PAM takes the distance or dissimilarity matrix and the number of clusters to be generated as input. It works by randomly selecting data points (equal to the specified number of clusters) as cluster centers, then iteratively assigns the remaining data points to the closest cluster centers (using median), meanwhile recalculating the cluster centers until the clusters are as far apart as possible.

In order to find out the best number of clusters, clustering should be run several number of times using the same Gower distance calculated from the dataset, but with the number of clusters to be generated varying, for e.g. 2 to 20. Then, for each experiment, the value of the internal validation metrics are calculated. By plotting the internal validation metrics, the best number of clusters can be deternmined. Then, the data set can be appended with a cluster membership vector (the cluster/class to which a data point belongs) obtained by the cluster model given by the optimum cluster number.

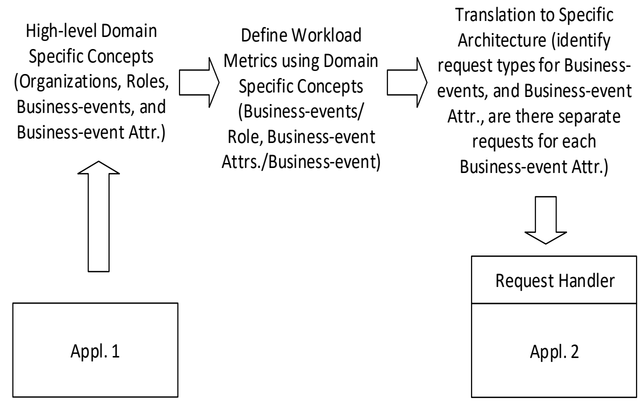

Figure 2: Translation of Workload Metrics to New Application.

4.2.3 Classification Model from Discovered Knowledge: Clusters can be validated by the knowledge discovered using supervised learning approach. This can be accomplished by building a classification model from the clusters. We built a classification model from the clusters using the cluster membership assigned to the data points as class variable. only binary variables as predictors for the classification model, as they represent the type of ORGANIZATIONS that are present in the clusters. The classification is a multi-class classification problem, since there are more than 2 clusters (or classes) in the dataset. To build a classification model, we used Randomforest classifier as it allows multiple classes. To test the hypothesis, an n-fold validation can be used by dividing the dataset into training and test sets. Then, by only using to training set to build clusters and building classification model from those clusters, the membership of data points in test set can be predicted, the stability of clusters can be calculated.

4.3 Workload Simulation

In this section, we demonstrate how we can translate the patterns from the existing application onto a new application with different architecture and workflow pattern. Figure 2 shows the translation of workload metrics to new application.

In order to translate the workload to the new system, the high-level workload metrics have to be expressed in terms of request types that the new application architecture can handle. We illustrate it here in the case-study using architecture of ERPSoftNext. ERPSoftNext runs on Command-Query Responsibility Segregation (CQRS) [9, 23] back-end architecture and Event Sourcing [5]. CQRS architecture separates actions that change the state of the system, i.e. commands, and the queries which just reads the current state of the system. The changes made by commands are propagated to the query side through events that register the changes and project them to the query side.

The workload metrics, i.e. BUSINESS-EVENTS and BUSINESS-EVENT ATTRIBUTES creation occur on the command side as they create new state of the system. Each type of BUSINESS-EVENT and BUSINESS-EVENT ATTRIBUTE translate to a particular type of command in ERPSoftNext, for example a sales invoice will translate to a sales

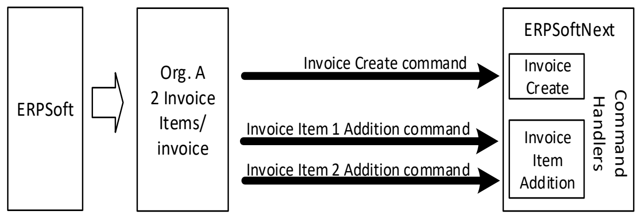

Figure 3: Illustration of Translation of Workload Metrics from ERPSoft to ERPSoftNext.

invoice creation command and adding invoice items to an invoice leads to sales invoice item addition command.

The Figure 3 shows the translation of workload metrics from ERPSoft to ERPSoftNext. Let us say that one of the cluster is ORGANIZATION with 2 invoice items per invoice, then the workload on ERPSoftNext for one invoice creation for a ROLE will have 3 commands: one for invoice creation, and 2 for addition of items to the created invoice.

The research method used is described in the algorithm 1. For the sake of simplicity, method shown applies only to sales *BE-factor* and not to project and subscription.

5 RESULTS

In this section, the tests to determine the optimum cluster number and validation of determined cluster model is presented. In order to evaluate the accuracy and stability of clustering , we used n-fold validation where we sampled 5% of the dataset as test set and the remaining 95% was used to build clusters and classification model. In order to calculate the accuracy of prediction for n-fold tests, we built a clustering and classification model on the full dataset to know the cluster memberships for the data points in 5% test dataset beforehand.

5.1 Clustering of ORGANIZATIONS for Knowledge Discovery

First step in clustering is to determine the number of clusters. We ran tests using PAM on several cluster numbers ranging from 5 to 28. We calculated the internal cluster validation measures mentioned in the section 4, i.e. silhouette width, Dunn index, connectivity index, and DB index. The results of the test are shown in the Figure 4.

Overall, it can be observed that as the cluster number increases, internal validation measures approach optimum values. It can also be observed from the plot that internal validation measures reach optimum values at cluster number 22, i.e. silhouette width is high, Dunn index is low and does not change much after 22, connectivity is low, and DB index is minimized. The internal validation measures only help determine cluster numbers which give optimum clusters, but the choice selection is based on expert knowledge. With expert opinion, we chose to 22 as the optimum cluster number to use.

Figure 5 shows the clusters obtained from the data. The clusters are plotted by extracting features using t-Distributed Stochastic Neighborhood (t-SNE) algorithm. t-SNE is a non-linear dimensionality reduction technique which extracts a small number of features that account for most of the variance in the data. t-SNE

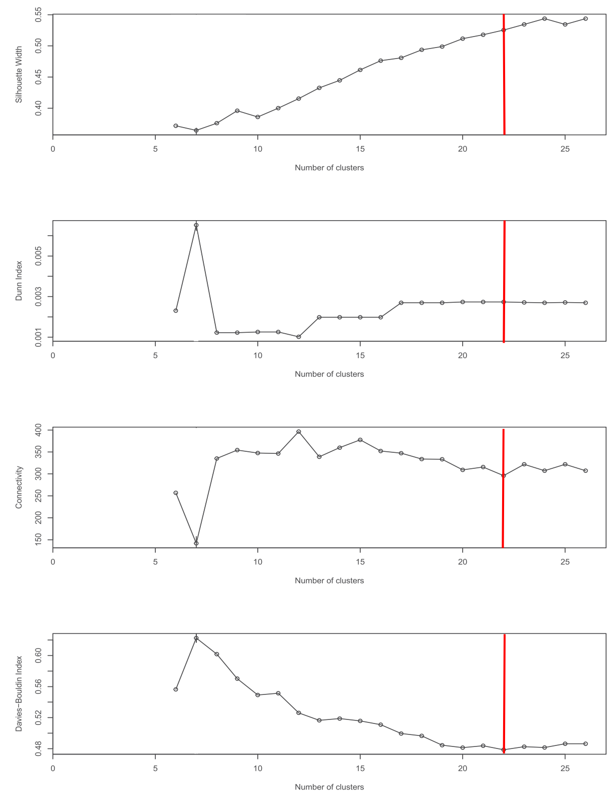

Figure 4: Plot of Internal Validation Metrics Using Calculated for Clustering Models with Clusters Numbers from from 6 to 28.

algorithm can be used to visualize localized similarities in the multi-dimensional data by reducing it to two or three features. The plot shows distinct clusters of ORGANIZATIONS.

5.2 Classification Validation

For each fold of the 5-fold validation, the sub dataset containing train or model building set was used to build the clustering model. Using the discovered cluster labels, we built a classification model. Then, the test set was used to predict the cluster membership from the classification model. Using the already known membership for the data points in the test set from the cluster model of full dataset, we calculated the accuracy of prediction of membership to clusters for test set data points.

The table 3 shows the prediction accuracy for 5-fold validation. An overall prediction accuracy over 5-fold validation of 93.2% was obtained from the tests. The values for folds 2 and 4 were affected because the sampling chose too many data points from a specific cluster as could be the case when there are a lot of clusters in the data and relatively small number of data points. This validation indicates that the clusters are stable.

Algorithm 1: Categorization of ORGANIZATIONS and Validation

$Data \leftarrow (Org_{id},\ SIpC_{1qtl},\ ..,\ SILpSI_{1qtl},\ ..,\ PIpC_{1qtl},\ ..,\ PILpPI_{1qtl},\ ..,\ SuIpC_{1qtl},\ ..,\ SuILpSuI_{1qtl},\ ..,\ SItem_{article},\ SItem_{ToW},\ SItem_{course},\ PItem_{article},\ PItem_{ToW},\ PItem_{course},\ SuItem_{article}, SuItem_{ToW},\ SuItem_{course},\ LType_{SB},\ LType_{ERP},\ LType_{HRM},\ LType_{ACC},\ LType_{ASB})$

for $i \leftarrow 1$ ***to*** $NumOfColumns(Data)$ **do**

  if $data(,i)$ *is numeric* **then**

    $data(,i) \leftarrow rank(data(,i))$;

$Diss_{Gower} \leftarrow GowerDistance(Data)$;

for $j \leftarrow 2$ ***to*** m **do**

  $Clusters_j \leftarrow PAM(Diss_{Gower}, j)$;

  $SilWidth_j \leftarrow SilhouetteWidth(Clusters_j)$; $Dunn_j \leftarrow DunnIndex(Clusters_j)$; $Conn_j \leftarrow Connectivity(Clusters_j)$; $DB_j \leftarrow DBIndex(Clusters_j)$;

$Clusters_{optimum} \leftarrow optimum(SilWidth_j, Dunn_j, Conn_j, DB_j)$ *where* $j = 2$ *to* m;

$Data_{mem} \leftarrow Data + Clusters_{optimum}[membership]$;

for $k \leftarrow 1$ ***to*** n **do**

  $TrainSet \leftarrow sample(Data_{mem})$;

  $TestSet \leftarrow Data_{mem} - trainSet$;

  $classificationModel \leftarrow Randonforest(TrainSet)$;

  $Prediction \leftarrow Predict(TestSet)$;

  $Accuracy \leftarrow TruePositives(Prediction)/Size(TestSet)$

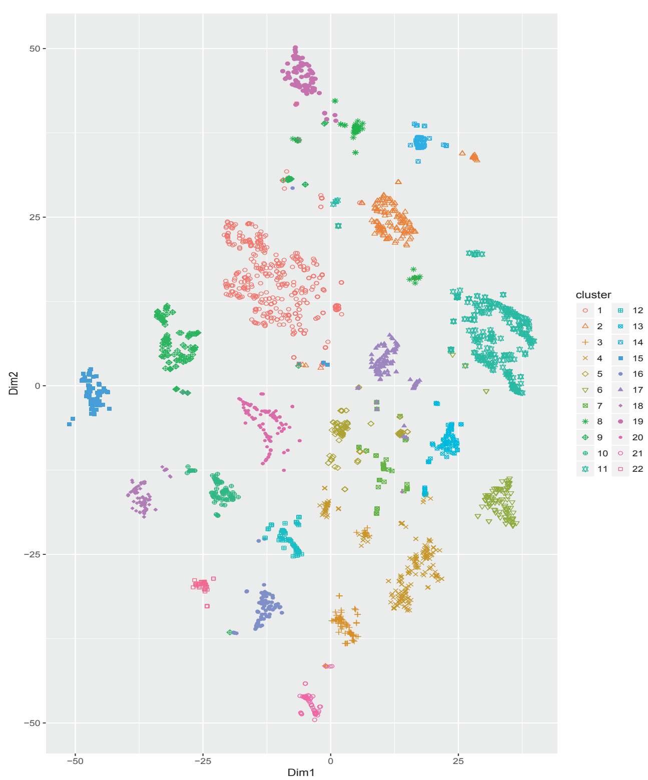

Figure 5: Plot of t-sne Reduced Dimensions Showing Clusters of organizations.

Table 3: Classification Accuracy for 5-fold Validation for Cluster Membership Prediction

Fold	Testset size	True Positives	Accuracy
1	154	148	96.1%
2	154	134	87%
3	154	147	95.45%
4	154	139	90.02%
5	154	150	97.4%

5.3 Analysis of the ORGANIZATION Clusters

In the table 4 the statistics and information about the clusters is presented. It can be seen from the table 4 that, every cluster has a specific LType or a combination and combined with type/s of items they sell categorize the ORGANIZATIONS into clusters which shows up as variations in BE-factors.

Even if the ORGANIZATIONS have similar license types between clusters, the type of invoice make a difference. For example in the table 4, clusters cl4 and cl9 both have high number of ORGANIZATIONS (more than 98%) ERP license; the difference comes from the types of invoice they have. Cluster cl4 has SItem and SuItem of type article as majority while cl9 has PItem of type article in majority.

Furthermore, not just type of invoice (sales, project, or subscription), but also the type of item makes a difference. For e.g. in table 4 clusters cl3 and cl19 are similar in terms of license type. Both have project invoices, even though cl19 also has subscription invoices. But if we see the project invoices, the item types are different. Cluster cl3 has only PItem type article, cl19 has PItem type article and ToW. This suggests that, by knowing the information about the ORGANIZATIONS, the expected workload depends on the number of ROLE instances present in the ORGANIZATION.

Table 4. Cluster Statistics Showing the Number of BE-Factor for Sales, Projects, and Subscriptions Along with Percentage of organizations in the Cluster having a Specific Item and License Type

No.	BE-Factor SIpC/SILpSI	BE-Factor PIpC/PILpPI	BE-Factor SuIpC/ SuILpSuI	SItem %			PItem %			SuItem %			LType %				
				article	ToW	course	article	ToW	course	article	ToW	course	SB	ERP	HRM	ACC	ASB
cl1	(1,2,3)/(1,2,3)	(0,0,0)/(0,0,0)	(0,0,0)/(0,0,0)	100	0	4	2	0.7	0	0	0.1	0	1	98	1.5	3	0
cl2	(1,2,3)/(1,2,3)	(0,0,0)/(0,0,0)	(0,0,0)/(0,0,0)	85	100	5	0	0.5	0	0	1.5	0	1.5	98	0.5	2	0
cl3	(0,0,0)/(0,0,0)	(1,1,2)/(1,1,2)	(1,2,3)/(1,1,2)	13	0	4	100	0	4	100	5	1	1	99	2	7	0
cl4	(1,2,3)/(1,2,3)	(0,0,0)/(0,0,0)	(2,3,5)/(1,1,2)	100	19.5	6.5	6.5	2.5	0	100	4	1	0.4	100	2	0.8	0
cl5	**(1,1,2)/(1,2,3)**	**(1,2,4)/(1,2,4)**	**(1,2,3)/(1,1,2)**	**97**	**80**	**16**	**88**	**94**	**1.5**	**100**	**12**	**1**	**0.6**	**100**	**1**	**0**	**0**
cl6	(0,0,0)/(0,0,0)	(1,2,4)/(1,2,3)	(0,0,0)/(0,0,0)	0	1	0	0	100	1	0.8	1	0	0.8	1	0.7	97	0
cl7	(2,3,5)/(1,1,2)	(3,6,8)/(1,2,3)	(2,5,6)/(1,1,2)	94	4	94	83	87.5	12.5	100	4	0	0	96	21	3	0
cl8	(1,1,2)/(1,2,3)	(1,2,2.5)/(1,1,3)	(0,0,0)/(0,0,0)	96.5	78	8	76	97.5	3.5	0	0	0	1	100	0	1	0
cl9	(0,0,0)/(0,0,0)	(1,1,2)/(1,1,2)	(0,0,0)/(0,0,0)	12.5	0	3.5	100	0	0.7	0	0	0	1	98	3	1.5	0
cl10	(0,0,0)/(0,0,0)	(1,2,4)/(1,3,5)	(4,6,11)/(1,1,1)	5	2	1	100	100	1	100	19	1	2	0	0	100	0
cl11	(0,0,0)/(0,0,0)	(1,2,4)/(1,1,2)	(0,0,0)/(0,0,0)	2	5	3	0	100	0.9	0	0	0	0	99.5	3	0	0
cl12	(0,0,0)/(0,0,0)	(0,0,0)/(0,0,0)	(3,6,6)/(1,1,1)	10	5	1	0	0	0	95	9	1	0	0	0	100	0
cl13	(0,0,0)/(0,0,0)	(1,2,4)/(1,1.5,3)	(2,3,5)/(1,1,1)	9	3	5	0	100	0	100	12.5	2	0	100	4	0	0
cl14	(0,0,0)/(0,0,0)	(0,0,0)/(0,0,0)	(0,0,0)/(0,0,0)	0	0	14	0	0	1	0	0	0	31.5	100	0	1.5	0
cl15	(1,2,4)/(1,1,1)	(0,0,0)/(0,0,0)	(0,0,0)/(0,0,0)	94.5	12.5	0.0	0	0	0	0	0	0	3.5	0	0	3.5	95.5
cl16	(0,0,0)/(0,0,0)	(1,2,4)/(1,1,2)	(0,0,0)/(0,0,0)	5.5	1.8	1	97	0	0	0	1	0	13	3.5	0.8	100	0
cl17	(0,0,0)/(0,0,0)	(1,2,5)/(1,2,3)	(2,4,6)/(1,1,2)	7	1.5	6	100	100	2.2	100	18.5	3	0	100	0	0.5	0
cl18	**(0,0,0)/(0,0,0)**	**(1,2,4)/ (2,4,8.5)**	**(4,6.5,11)/ (1,1,1)**	**1**	**3**	**2**	**0**	**100**	**0**	**100**	**18**	**0.0**	**2**	**2**	**2**	**100**	**0**
cl19	(0,0,0)/(0,0,0)	(1,2,5)/(1,2,4)	(0,0,0)/(0,0,0)	1	1.8	6	100	100	2.5	0	0	0	0	100	2	2	0
cl20	(0,0,0)/(0,0,0)	(0,0,0)/(0,0,0)	(1,2,3)/(1,2,2)	0	2.5	3.5	0	0	0	100	3.5	0.7	0.5	100	3.5	2.5	0
cl21	(0,0,0)/(0,0,0)	(1,1,2)/(1,1,2)	(3,4,7)/(1,1,1)	6	0	0	100	0	0	100	0	0	6	8	3	100	0
cl22	(0,0,0)/(0,0,0)	(1,2,4)/(1,2,4.5)	(0,0,0)/(0,0,0)	6	0	0	100	100	0	0	0	0	14	2	0	100	0

6 WORKLOAD SIMULATION TESTS

In this section we present simulation of workload on ERPSoftNext using the clusters discovered from the simulations the affect of different organization behavior patterns on resource consumption can also be analyzed. Since the temporal aspect of the workload in production is not taken into account in this work, we make few assumptions on the distribution of the workload. We assume that the workload is evenly distributed. This means that total business-events obtained by multiplying *BE-factor* by the number of roles (which is for the period of a year), is distributed over each day of the year evenly. In order to demonstrate the effect of variation in business-event attributes on resource consumption, we take two clusters, cl5 and cl18 from the table 4, that have similar *PIpC BE-factor* but different *PILpPI BE-factor*. With the different usage pattern of the two organization classes, we can check the impact of difference in behavior of organizations in terms of business-events to resource consumption.

We simulate the workload on a test server running ERPSoftNext using a custom-built workload generator. Figure 6 shows the setup for workload simulation from existing system ERPSoft to the system under development ERPSoftNext. As discussed in section 4, the business=events and business=event attributes translate to commands in the CQRS architecture of ERPSoftNext. For the sake of simplicity of implementation, business=event attributes are implemented as separate commands. Hence, the total number of business=events corresponds to the total number of invoice create commands and business=event attributes in each business=event to the total invoice attribute create commands. By using the command template for the business-event and business-event attributes, corresponding workload can be generated.

Let us assume that the number of instances of the role customer in the organizations belonging to clusters cl5 and cl18 is 10,000. This leads to 22,500 project invoices for both organizations in clusters cl5 and cl18. Distributing the total number of invoices over a year, makes around 62 invoices per day. The total number of attributes in invoices will then be 139 for an organization in cl5 taking into account the quartiles. Similarly, for the organization in cl18, we have 287 invoice lines. We also assume that the invoices are generated at the same time, as it helps us to record resource utilization in a convenient way. We used a custom-built workload generator for simulations. The workload generator tool contains JSON template files for specific commands that will be invoked when generating a specific type of command. The actual workload generator fills in templates with dummy data and generates JSON messages as command for the application. In the setup, we use ELK stack [3] to capture and display the resource statistics.

Figure 7[a] shows the simulation of workload for organization in clusters cl5 and Figure 7[b] for cl18. From Figures 7[a] and [b], the first plots show the number of commands processed. The difference in the first plots is that the total number of commands is dissimilar. This is because in ERPSoftNext the CQRS back-end system receives more commands for cl18 than cl5, as more invoice attributes are present, even though the number of invoices that is created is the

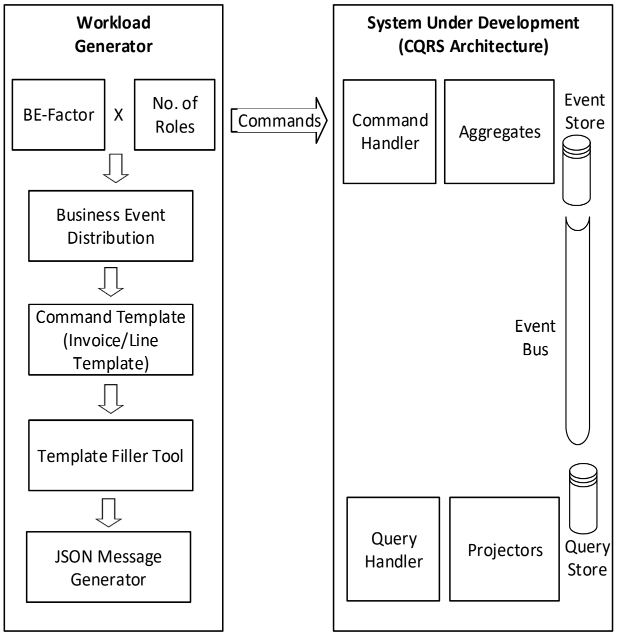

Figure 6. Workload Simulation on System Under Development by Extracting Patterns from Existing System.

same. The average duration for processing commands is higher on cl18, as more requests are being processed simultaneously. Also, for a similar number of invoices for the two clusters, the average CPU consumption for cl5 is around 15% while for cl18 it is around 25%. This is because of a higher number of requests processed simultaneously, as cl18 generates more invoice attributes, and hence more requests. Disk transfer is also higher in cl18 compared to cl5.

If low level metrics, such as page accesses, were used in categorization of organizations, the details of the advanced business processes in ERPSoft would not have been captured in a transportable format to ERPSoftNext. In other words, only through organization categorization of high-level operations, we can create more accurate workloads for realistic usage simulation on new products. Furthermore, the significant workload difference between the clusters proves that clustering -user organizations is useful for performance testing.

7 DISCUSSION AND THREATS

The concepts presented in the paper are applicable to other ERP software applications, as business actions in the ERP domain are similar in other ERP software applications. Therefore, the high-level of Organizations, Roles, Business-Events, and Business-event Attributes are generalizable to most, if not all cases.

In an ERP software case, identifying representative metrics for simulating workload is essential, especially in the case if an existing application is used with a different architecture. With the chosen high-level metrics defined in terms of business-events and roles in an organization, we show that the organizations can be categorized into groups based on the usage of an application. It was shown that the items organizations sell and the license for ERP software have a strong correlation to application usage. We choose three abstract concepts to represent items sold and license types in relation to the case-study, but depending on the scenario different metrics might need to be determined.

There are validity threats in this work. One of the main threats is that the study is conducted at single case-company. This is a limitation since getting production data from ERP application is difficult to the reasons of sensitive and confidentiality. This is also why not many studies are available in literature on enterprise applications.

Additionally, simulating workload exactly similar to the load on the production server requires temporal data of ERP workload generated by the roles. Most organizations choose to generate business-events at a specific period of year or month or is based on behavior of roles. Without studying the temporal relation of the generation of business-events, simulations will be inaccurate. We plan to study temporal aspect of ERP workload in future work. Also load generated by modify and view actions needs to be analyzed separately.

8 CONCLUSION

In this paper an extensive study at a large software company to simulate workload on an application under development is presented. We defined high-level abstractions or workload metrics in terms of business operations, business-events, business-event attributes, and roles, which are translated to a new application with a different architecture. From the high-level abstractions, data metrics were derived that characterize the behavior of roles in end-user organizations. Metrics were extracted from an existing application in use by the end-user organizations and used to derive the patterns using an unsupervised learning approach. Next, a workload simulation mechanism is described by translating the high-level workload to actual workload on a an application under development using a different architecture, i.e., the CQRS framework. From the simulations we showed that for organizations with similar number of business-events but varying business-event attributes show significant difference in resource consumption, which prompts that more usage based performance testing is required when redesigning system.

In future work, we plan to investigate temporal behavior of organizations, for instance to analyze peak load on the servers. In addition other parts of the enterprise application could be investigated to gain overall understanding of usage. We also wish to investigate reactive architectures that self-adapt to the behavior of the organizations during production usage.

ACKNOWLEDGMENTS

This is an AMUSE Paper. This research was supported by the NWO AMUSE project (628.006.001): a collaboration between Vrije Universiteit Amsterdam, Utrecht University, and AFAS Software in the Netherlands. The NEXT Platform is developed and maintained by AFAS Software. Please see amuse-project.org for more information.

REFERENCES

[1] Sonali Aggarwal, Shashank Phadke, and Milind Bhandarkar. 2010. Characterization of hadoop jobs using unsupervised learning. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 748–753.

Figure 7. Kibana Dashboard for Workload Simulation for an organization in [a] cl5 and [b] cl18. We in particular note the longer ramp-up and steady drop in CPU and disk transfer at the end of the test of the organization in cl5.

[2] Claudia Canali and Riccardo Lancellotti. 2017. AGATE: Adaptive Gray Area-based TEchnique to Cluster Virtual Machines with Similar Behavior. *IEEE Transactions on Cloud Computing* (2017).

[3] Saurabh Chhajed. 2015. *Learning ELK Stack*. Packt Publishing Ltd.

[4] Frank Elijorde and Jaewan Lee. 2015. Attaining Reliability and Energy Efficiency in Cloud Data Centers Through Workload Profiling and SLA-Aware VM Assignment. *International Journal of Advances in Soft Computing & Its Applications* 7, 1 (2015).

[5] Martin Fowler. 2005. Event sourcing. *Online, Dec* (2005), 18.

[6] John C Gower. 1966. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika* (1966), 325–338.

[7] Nikolas Roman Herbst, Nikolaus Huber, Samuel Kounev, and Erich Amrehn. 2014. Self-adaptive workload classification and forecasting for proactive resource provisioning. *Concurrency and computation: practice and experience* 26, 12 (2014), 2053–2078.

[8] Zhen Jia, Jianfeng Zhan, Lei Wang, Rui Han, Sally A McKee, Qiang Yang, Chunjie Luo, and Jingwei Li. 2014. Characterizing and subsetting big data workloads. In *Workload Characterization (IISWC), 2014 IEEE International Symposium on*. IEEE, 191–201.

[9] Jaap Kabbedijk, Slinger Jansen, and Sjaak Brinkkemper. 2012. A case study of the variability consequences of the CQRS pattern in online business software. In *Proc. of the 17^{th} European Conference on Pattern Languages of Programs*. 2:1–2:10.

[10] Leonard Kaufman and Peter J Rousseeuw. 2009. *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons.

[11] Dao-Lei Liang and Hai-Bo Chen. 2017. An Online Mall CRM Model Based on Data Mining. In *Quantitative Logic and Soft Computing 2016*. Springer, 599–606.

[12] Yanchi Liu, Zhongmou Li, Hui Xiong, Xuedong Gao, and Junjie Wu. 2010. Understanding of internal clustering validation measures. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 911–916.

[13] Kunal Kulkarni Xiaoyi Lu and Dhabaleswar K DK Panda. 2016. Characterizing Cloudera Impala Workloads with BigDataBench on InfiniBand Clusters. (2016).

[14] Gururaj Maddodi, Slinger Jansen, Jan Pieter Guelen, and Rolf de Jong. 2016. The daily crash: a reflection on continuous performance testing. In *11th International Conference on Software Engineering Advances*. IARIA, 171–180.

[15] Daniel A Menascé, Virgilio AF Almeida, Rodrigo Fonseca, and Marco A Mendes. 1999. A methodology for workload characterization of e-commerce sites. In *Proceedings of the 1st ACM conference on Electronic commerce*. ACM, 119–128.

[16] Rizwan Mian, Patrick Martin, Farhana Zulkernine, and Jose Luis Vazquez-Poletti. 2013. Towards building performance models for data-intensive workloads in public clouds. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*. ACM, 259–270.

[17] Asit K Mishra, Joseph L Hellerstein, Walfredo Cirne, and Chita R Das. 2010. Towards characterizing cloud backend workloads: insights from Google compute clusters. *ACM SIGMETRICS Performance Evaluation Review* 37, 4 (2010), 34–41.

[18] Ismael Solis Moreno, Peter Garraghan, Paul Townend, and Jie Xu. 2014. Analysis, modeling and simulation of workload patterns in a large-scale utility cloud. *IEEE Transactions on Cloud Computing* 2, 2 (2014), 208–221.

[19] Jemishkumar Patel, Vasu Jindal, I-Ling Yen, Farokh Bastani, Jie Xu, and Peter Garraghan. 2015. Workload estimation for improving resource management decisions in the cloud. In *Autonomous Decentralized Systems (ISADS), 2015 IEEE Twelfth International Symposium on*. IEEE, 25–32.

[20] Kai Ren, Garth Gibson, YongChul Kwon, Magdalena Balazinska, and Bill Howe. 2012. Hadoop's Adolescence; A Comparative Workloads Analysis from Three Research Clusters.. In *Proceedings of 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, SCC 2012, (Washington, DC, USA)*. 1452.

[21] Henk van der Schuur, Erik van de Ven, Rolf de Jong, Dennis M M Schunselaar, Hajo A Reijers, Michiel Overeem, Machiel de Graaf, Slinger Jansen, and Sjaak Brinkkemper. 2017. NEXT: Generating Tailored ERP Applications from Ontological Enterprise Models. In *10th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (POEM)*.

[22] André van Hoorn, Christian Vögele, Eike Schulz, Wilhelm Hasselbring, and Helmut Krcmar. 2014. Automatic extraction of probabilistic workload specifications for load testing session-based application systems. In *Proceedings of the 8th International Conference on Performance Evaluation Methodologies and Tools*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 139–146.

[23] Greg Young. [n. d.]. CQRS and Event Sourcing. Feb. 2010. *URl: http://codebetter.com/gregy oung/2010/02/13/cqrs-and-event-sourcing* ([n. d.]).