Real-time business activity monitoring and analysis of process performance on big-data domains

Alejandro Vera-Baquero

Universidad Carlos III de Madrid, Spain

averabaq@gmail.com

Ricardo Colomo-Palacios

Østfold University College

ricardo.colomo-palacios@hiof.no

Owen Molloy

National University of Ireland, Galway

owen.molloy@nuigalway.ie

Abstract

Real-time access to business performance information is critical for corporations to run a competitive business and respond to a continuously changing business environment with ever-higher levels of competition. The timely analysis and monitoring of business processes are essential to identify non-compliant situations and react immediately to those inconsistencies in order to respond quickly to competitors. In this regard, the integration of Business Intelligence (BI) systems with Process Aware Information Systems (PAIS) can become a key tool for business users in decision making. However, current BI systems are not suitable for optimizing and improving end-to-end processes since these are normally business domain specific and are not sufficiently process-aware to support the needs of process improvement type activities. In addition, highly transactional business environments may produce vast amounts of event data that cannot be efficiently managed by the use of traditional storage systems which are not designed to manage vast amounts of event data. We introduce a cloud-based architecture that leverages big-data technology to support performance analysis on any business domain, in a timely manner and regardless of the underlying concerns of the operational systems. Likewise, we demonstrate the ability of the solution to provide real-time business activity monitoring on big-data environments with low hardware costs.

Keywords

Big Data, Cloud Computing, Business Activity Monitoring, Business Process Improvement

1. Introduction

ICT-based tools in general and real-time measurement and data analysis of the performance of operational activities is essential for companies to remain competitive [1]. The monitoring of

business process execution allows business users to detect error rates and non-compliant business situations, such as supply chain issues. This action must be performed on-time in order to react quickly to those situations. In a well running process it is expected that arrival (demand) and throughput rates should be in balance. Processes or activities which do not have the capacity to work to this arrival rate will cause delays and bottlenecks, thereby starving proceeding activities of input. This may result in increase delays and a loss of profit due to a waste of valuable resources that are underutilized, and consequently, a loss of customer satisfaction and loyalty.

A successful analysis of business processes is essential for organizations to gain competitiveness [2]. Moreover, process improvement based on analysis is seen as a way to lead organizations to effectiveness [3], [4]. Process models are often inadequately understood and optimized within organizations, and consequently processes under-performed. This leads to long response times, unbalanced utilization of resources, low service levels, and so on, thereby causing high costs and dramatic loss of profits to corporations [5]. In this regard, the use of advanced analytical techniques would help analysts to continuously improve their processes, thus meeting their business goals.

The combination of BI and Business Activity Monitoring (BAM) technologies may provide mechanisms to infer knowledge about business performance, but these are not sufficient for answering most of the demanding questions of today's business users. There currently exists an increasing demand for more advanced analytics such as root cause analysis of performance issues, predictive analysis and the ability to perform "what-if" type simulations. These features are powerful assets for analysts, expanding their knowledge beyond the limits of what current platforms typically offer. Furthermore, these platforms are normally business domain specific and have not been sufficiently process-aware to support the needs of process improvement type activities, especially on large and complex supply chains, where it entails integrating, monitoring and analysing a vast amount of dispersed, unstructured event logs produced on a variety of heterogeneous environments, in a timely manner.

In general, the monitoring and analysis of operational data aims to be fact-based and therefore empirically evaluated with real data which leads to trustworthy analysis results. However, this is complex to achieve as there exists a noticeable disconnect between idealized business processes and their actual event-data. Current BI platforms by their own do not fill this gap as they are focused on local decision making rather than end-to-end processes. As a consequence, their outputs tend to be unreliable since they are based on idealized models of reality rather than on observed facts [6]. In order to allow business users to gain visibility on their business processes, the execution outcomes must be gathered from operational sources, unified and correlated across organizational boundaries [7].

The latest advances in technology have made it possible for organizations to co-operate with each other, necessitating the integration of diverse business information systems across large and complex supply chains in several domains [8], [9]. In these scenarios, isolated optimisation within individual organizations is insufficient to optimize and improve end-to-end processes. This leads to the management of complex operational processes, where web services technology and cloud computing have become widely used, producing cross-functional event logs that are beyond company (and increasingly software) boundaries. This has promoted an incredible growth in corporate event data that needs to be merged for analysis [7]. Moreover, enterprises' business data are usually handled by heterogeneous systems which run on different

technological platforms, and even use incompatible standards. In addition, the continuous execution of distributed business processes (BP) may produce vast amounts of event data that cannot be efficiently managed by the use of traditional storage systems which are not adequate to manage event data of the order of hundreds of millions of linked records [10]. Therefore, innovative methods and techniques are needed to put real expert systems technology in the hands of business users. Nowadays, there exist emerging technologies such as big-data and cloud-computing that can be leveraged to drive the generation of business process analytics (BPA) solutions with the capabilities to produce outcomes on a timely basis. Notwithstanding, the successful implementation of a fully distributed BPA solution involves significant challenges that are not easy to address:

1. First, BI-like platforms must be re-engineered to support business process analytics, and these are typically business domain-specific.
2. Processes and enterprise events are intrinsically related to each other but these need to be correlated across organizational boundaries, and this is challenging.
3. Measuring and improving overall business performance is especially hard to achieve on highly distributed environments whose business processes are part of complex supply chains. In turn, these processes are typically executed under a variety of heterogeneous systems, which makes them even harder to measure.
4. Continuous execution of distributed business processes may produce a vast amount of event data that cannot be efficiently managed by means of traditional storage systems, which are not adequate to manage event data in the order of hundreds of millions of linked records.
5. Existing centralized approaches such as described in [11], cannot provide real-time analytics on complex business cases that produce large amounts of event data. These systems are not suitable to deal with such volumes of information since they neither include sharding mechanisms nor provide big-data support. Furthermore, these approaches may entail a significant latency from the time the event occurs on source to the time the event is recorded in central repositories. This pitfall is intensified on very large and complex supply chains which normally involve a high number of business units and a greater number of operational systems.
6. Dealing with highly distributed supply chains demands some collaborative analysis where individual stakeholders are geographically separate and need a platform to perform BPM in a collaborative fashion, rather than depending on a single centralized process owner to monitor and manage performance at individual supply chain nodes. This is especially complex to accomplish using centralized approaches.

This paper proposes a cloud-based solution aimed at addressing the aforementioned challenges. The overall solution has been previously introduced in [10][12], but this paper is focused on the internal details of the BAM part of the system instead, which achieves monitoring in real-time of processes whose execution outcomes are produced in big-data contexts. This leads us to a big-data analytics scenario, where supporting systems require very large scale processing to achieve timely results. Cloud computing is a key enabler for big data analytics since the huge volume of data to manage requires very large scale processing, especially in those cases where the scale of data exceeds a single computing node [13]. In this regard, the cloud-based infrastructure proposed is ideal for meeting the computational and storage needs of BPA applications over big-data [14]. This is especially interesting in business scenarios that involve very complex and highly distributed processes like supply chain management.

The next sections roll out the fundamentals of the solution proposed and give an overview of the system internals and overall architecture. The reminder of the paper is structured such as follows. First, we review the evolution of the state-of-the-art in the field of business activity monitoring and then we introduce the analytical framework with a special emphasis on the model and the proposed event correlation algorithm. The aims of the event-based model and correlation algorithm are twofold: 1) keeping the system agnostic to any business domain, and 2) allowing the system to integrate event data regardless of the underlying concerns of the operational systems. Following the foundations of the event correlation, we introduce the generation of metrics based on the sequence of correlated events. Finally, we evaluate the performance of the architectural solution proposed and we conclude with a brief summary, conclusions and outline of future works.

2. Evolution of Business Activity Monitoring Systems

The general concepts around process improvement have been well known for some time, and methodologies such as Lean and Six Sigma are widely applied as vehicles for understanding, measuring and analyzing process performance. Underlying principles are based around improving visibility as to what is actually happening as well as providing hard data to confirm hypotheses as to what is happening and where the problems lie. The actual metrics which we use to measure business processes are often grouped under different categories, such as Quality, Time, Flexibility and Cost [15]. Using appropriate BAM tools, we should be able to generate all of the required metrics relating to Time, such as process cycle time, defect (rework) rates, throughput rates, lead times, etc..

When it comes to measuring the overall process performance over a sustained period, Six Sigma calculations are typically used in conjunction with process measurements. Of key importance in Six Sigma are measurements of the distribution and variability of metrics, so that we can follow the changes in performance over time.

At the core of BAM is the monitoring and processing of business events. The capture of sufficient business process events (e.g. task start, task end) and their timely processing allows us to generate statistically valid process metrics, and to respond in a time-critical manner. Provided that we have a means of capturing and correlating business process events, we can produce metrics and visualizations of current process performance. Since the advent of service-oriented and event-driven architectures, there have been a number of frameworks proposed for the collection, processing and analysis of business process events. The company webMethods reflected the general consensus of the time, when they called BAM “The New Face of BPM” in a white paper from 2006 [16]. Their system was designed to be capable of collecting and correlating process event data from heterogeneous systems, but has since been acquired by SoftwareAG, and is used to provide real-time monitoring and metrics across SAP systems. A similar framework to provide process modeling, monitoring and analysis across heterogeneous systems was successfully developed by [17][18]. In this system, the event correlation is dependent on the use of context (or payload) data such as an order associated with a *create* order event, to achieve correlation of multiple events across the life of a process instance. At this time, the focus was very much on the service-oriented architecture, data warehousing and business intelligence aspects of the BAM frameworks, whereas the technologies of cloud computing and big data were not yet available. Nevertheless, it was widely recognized that application of BI, especially real-time BI to distributed business processes such as supply

chains, would drive improvements in operational efficiency [19]. While not central to the solution of the analytics problems, the emergence of the semantic web and ontologies also contributed to the definition and calculation of process metrics, in the absence of a standard for business process analytics data and its exchange [20][21]. Another technique which was immediately considered a good fit for BAM is that of Complex Event Processing (CEP), and a number of frameworks have demonstrated its usefulness in terms of dealing with event correlation and latency [22]. However, the problem of correlation of process events to specific process instances remains a problem to be solved, with a number of solutions proposed, such as [23] and [24].

With the advent of Big Data and Cloud Computing, it was clear that there were opportunities to use these technologies to gather and process vast amounts of information from across supply chains, not just for the purposes of monitoring of distributed business processes, but also for driving post-execution data analysis. This is perhaps a logical conclusion as businesses actually migrate their business processes to the Cloud. Nevertheless, this has a negative impact on the BI strategy of corporations as it increases their need for more complex solutions that can meet the data analysis demands in terms of latency and data volume. This is especially challenging on BAM contexts whose reports must be available in nearly real-time, and where the continued growth of event data during the business execution lifetime makes it difficult for those systems to deal with such volumes of data. This drives BAM solutions to adopt complex underlying architectures aimed to process data streams at speeds beyond the processing capabilities of a single large computer, thereby making the scalability a must on those systems at nowadays. Hence, next generation of BAM systems must be big-data ready for enabling elastic-scalable data analysis.

The application of Big Data to process analytics has the potential to add real scale to the data collection as well as the analysis itself. Companies such as IBM and Splunk are just two of many who are actively promoting solutions which claim to gather and synthesize data from customers, sensor, mobile devices and other sources. This opens up huge possibilities in terms of data mining, and the generation of new product ideas for example. However, as yet, there are no commercial products which harness Big Data and Cloud Computing to deliver a system-agnostic, process-aware BAM framework, such as the one proposed in the following sections.

3. The analytical framework

The proposed solution is aimed at providing integrated cloud services for monitoring and analysing business process performance over highly distributed environments. A set of BASU (business analytics service unit) nodes, along with a global master GBAS (Global Business Analytics Service) component (see Fig. 1), have been devised with the purpose of monitoring and analysing operational activities within both, local and global contexts. In a local context, the processes reside in an organization where they can be analysed and optimized by the means of BASU units. These units are attached to individual corporations for performing the managerial activities of their internal processes. An example of local business process analysis is the customer journey. In this case, the business process flows through different departments and systems that are in place such as product marketing, purchasing, servicing, billing, and support. An isolated BASU unit is capable of performing a local analysis and measuring the performance of these processes across different departments or business units. This is usually referred to as inter-departmental performance analysis, where processes are executed across departments or

different business units and where multiple heterogeneous data sources and applications are part of the value chain. In the non-local context, cross-organizational business process analysis can be attained on the GBAS module. This component supports the monitoring, analysis and optimization of large complex supply-chain processes like manufacturing and retail distribution. This global analysis supports these types of scenarios by allowing the integration and cooperation between organizations for optimizing distributed processes that cross both software and corporate boundaries. The overall solution is based on the provision of a set of local business analytical service units consolidated through a global business analytical service. In this way, multiple complex processes can be aggregated at local and global levels to provide process insights at the micro level (detailed customer transaction or interaction level) and at the macro level (the supply chain's overall health).

The overall architecture is depicted in Fig. 1 and aims to provide cloud computing services at very low latency response rates. These services can contribute to the continuous improvement of business processes through the provision of a rich informative environment that supports BPA and offers clear insights into the efficiency and effectiveness of organizational processes. These are exposed to third-parties as a number of APIs that can be leveraged by a wide range of analytical applications such as real-time monitoring, simulation, prediction and visualization, etc.

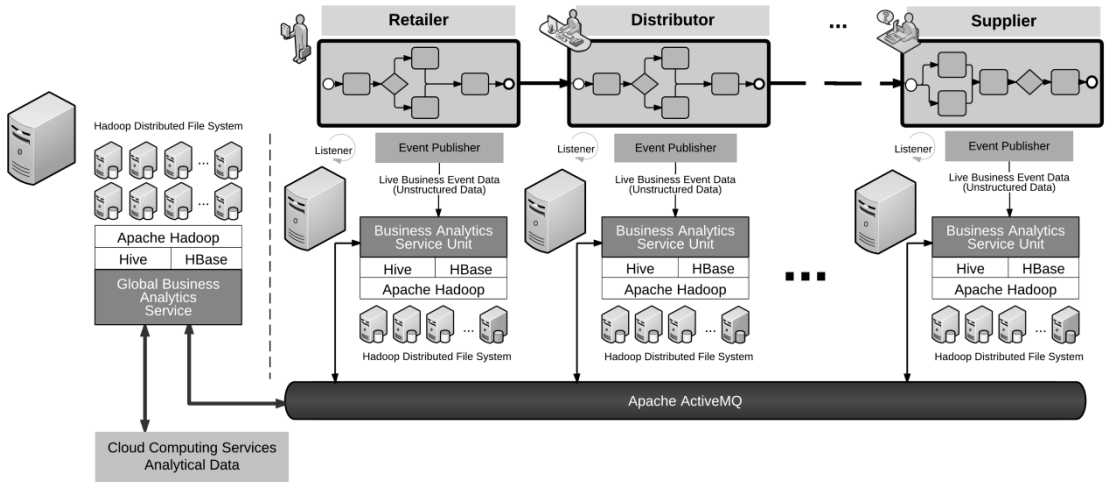

Fig. 1 - Architecture Overview

The devised architecture enables analysts to measure the performance of cross-functional business processes that are extended beyond the boundaries of organizations. The cloud-service components (BASU & GBAS) have the capabilities for collecting data originating from distributed heterogeneous enterprises systems, storing large amount of enterprise data and inferring knowledge from the gathered information. The successful integration of those components through enterprise service bus adapters completes the high-level architecture of the big data based solution proposed.

Each BASU component is attached to every operational business system along with their own local event repository. The event repository is a cloud data store that is built upon big-data technology. The big-data repository is designed as a cloud-based solution in order to allow the system to scale out easily on readily available hardware, which is essential for dealing with the data-intensive processing demands of the correlation process. Consequently, it allows us to

determine the trade-off between volume of data, KPI latency and hardware investments, and thereby we can easily adapt the analytical framework to multiple business domains with very specific business demands.

In this reagrd, the selection of the underlying technology is critical for addressing the real-time requirements of the BAM solution. As it will be covered later, the correlation process does need to have available the entire set of data stored in the event repository. On high-transactional, and consequently on big-data environments, this can be a real bottleneck for relational database management systems (RDBMS), where the distribution of the processing workload across multiple servers is mandatory for handling very large data volumes. It has been widely acknowledged by both the industry and academia, that RDBMS are hard to scale and tend to experience poor availability [25]. Additionally, they present serious challenges for handling big-data [26]. Even though modern RDBMS features sharding mechanisms with horizontal scaling capabilities [27] [26], they are not suitable for key partitioning [28] as they still rely on ACID properties, and the use of distributed systems techniques make them falling on the well-known CAP theorem [29] by sacrifycing either availability or partition-tolerance in detriment of consistency. In general, data stores that provide ACID guarantees tend to have poor availability [25], thus the use of a NoSQL approach is more adequate technological solution for fulfilling the aims of this work in terms of high-performance and horizontal scaling.

Herein, the event repository is implemented using the HBase product as big-data storage. HBase is a NoSQL, versioned, column-oriented data storage system that provides random real-time read/write access to big data tables and runs on top of the Hadoop Distributed Filesystem. HBase features powerful scaling capabilities. HBase clusters expand on commodity of HRegionServers, thus linearly increasing the storage and processing capacity. The technical documentation of this product reveals extraordinary clustering capabilities for providing data-intensive processing on large data tables. The distributed event repository is implemented as big-data tables over HBase, thereby exploiting its outstanding features for providing timely access to key data.

One of the main challenges of this approach relies on the integration of event data from operational systems whose business processes flow through a diverse set of heterogeneous systems such as business process execution language (BPEL) engines, ERP systems, CRM, workflows, etc. (see Fig. 2) as well as storing very large volumes of data in a global and distributed business process execution repository through the use of big-data technology. The monitoring and measurement of distributed information in real-time is the big challenge to be addressed in this paper. The monitoring of cross-organizational business processes is achieved by listening to state changes and business events from operational systems. This is achieved by capturing, collecting, unifying and storing the execution data outcomes across a collaborative network, where each node represents a participant organization within the global distributed business process. Hence, the proper instance identification is essential for measuring process performance as these must be correlated before any analysis is to be attempted.

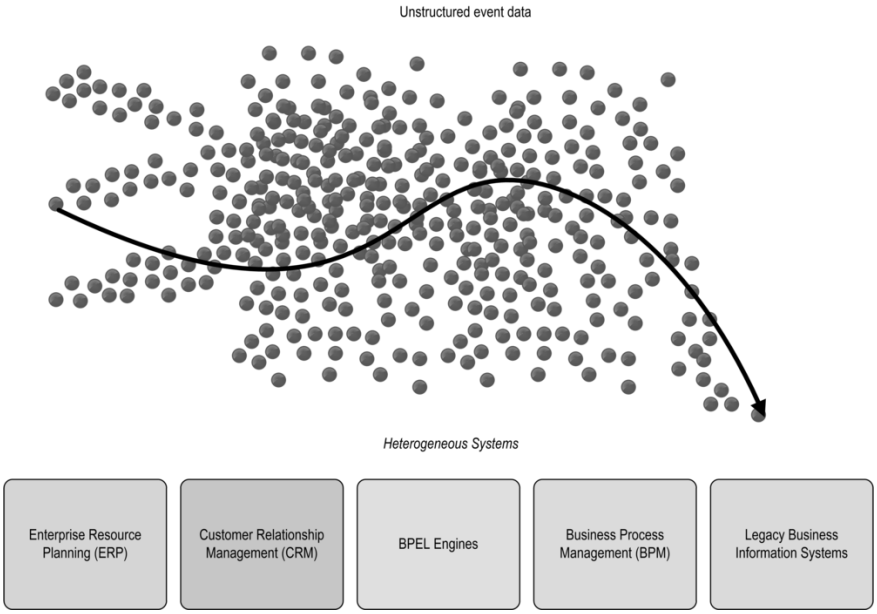

Fig. 2 - Distributed event correlation

3.1 Event Correlation Algorithm

Event correlation refers to the determination of the sequence of events produced by the execution of inter-related and consecutive process instances or activities. Event correlation is an essential part of the proposed framework for achieving the correct identification of process execution sequences. Without the ability to correlate events it would not be possible to generate metrics per process instance or activity [30], and thus business analysts are unable to identify exceptional situations or discover potential business opportunities. Furthermore, this sequence of event must be identified instantly in order to generate metrics in real-time.

This work proposes an event correlation mechanism based on the data shared between business processes during their execution. It is based on the conjunctive correlation method discussed in [31] and incorporates a generic construct that allows the framework to be agnostic to any business domain. This construct consists of two formalisms, an event-based model, which will be covered in the next section, and the foundations of the event correlation algorithm.

In an event-driven approach, such shared data usually makes reference to the message payload. This information can be used to identify the start and end event data for a particular process instance or activity. The main drawback of this approach is to determine which part of the event payload is used to identify and link the consecutive events.

The listening software is responsible for specifying which part of the message payload will be used to correlate the events of instances associated to a specific model. The following figure illustrates a sample of how three consecutive events in time (Event A, Event B and Event C) are correlated relating to the order number ‘A525’ contained in the message payload of the involved process.

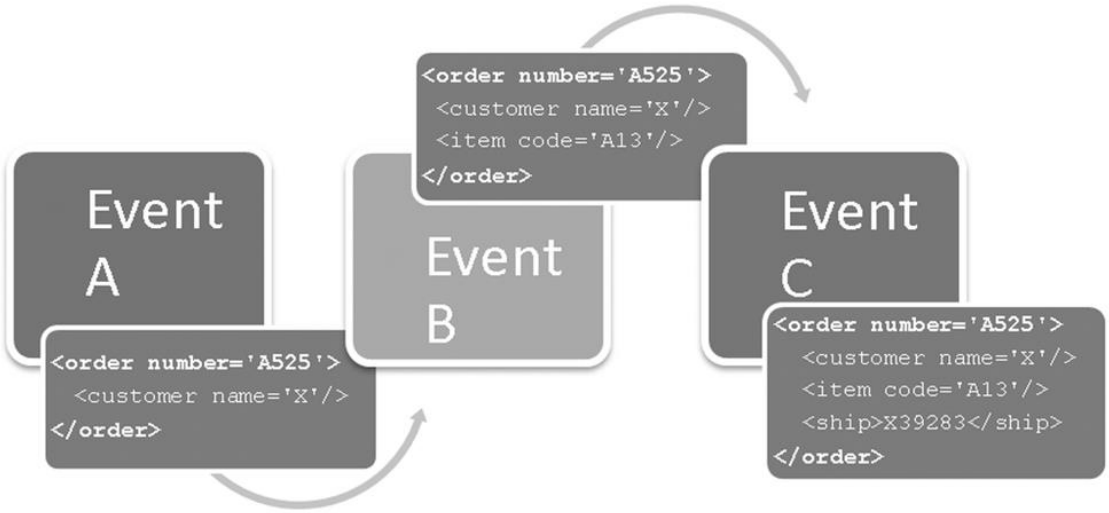

Fig. 3 - Event correlation sample based on some shared data

At destination, part of the message payload associated to the event is used to uniquely identify the associated process instance or activity by querying the cloud event store. The identification of the correct instance is attained by retrieving the exact instance associated to a determined process model, fired from a determined source and executed with a given correlation data. This triplet allows the proposed framework to determine which process instance or activity is the owner of a determined event. This mechanism is formalised in the next section.

Foundation of the Event Correlation Algorithm

The event correlation algorithm is the base of the system for identifying and linking sequences of inter-related events. This is essential for measuring processes and generating metrics per process instance or activity.

Domain of Discourse

The t-uple {S, D, P, E} composes the domain of discourse for the event correlation foundation. We define the domain of disclosure such as follows:

$S \equiv \{s_1,s_2,s_3,\ldots,s_n\}$ Defines the set of existing sources.
$D \equiv \{d_1,d_2,d_3,\ldots,d_n\}$ Defines the set of existing process definitions (models).
$P \equiv \{p_1,p_2,p_3,\ldots,p_n\}$ Defines the set of existing process instances.
$E \equiv \{e_1,e_2,e_3,\ldots,e_n\}$ Defines the set of existing event instances.

Definitions. Every discrete value of an event $e \in E$ belongs to an unique instance $p \in P$, which is defined by a process model $d \in D$, and whose instances are executed at a particular source $s \in S$. Every individual value on any of these sets is denoted such as follows:

$s \in S \equiv$ Defines the source *s*
$d \in D \equiv$ Defines the process definition (model) *d*
$p \in P \equiv$ Defines the process instance *p*
$e \in E \equiv$ Defines the event *e*

All discrete values associated to the set of events (E), process instances (P), models (D) and sources (S) are related to each other by definition, i.e. an event (E) implies a process instance (P) which must comply a process model (D) defined at a specific source (S). In this relationship (E->P->D->S), no discrete value can exists without the other, and this related data is denoted such as follow:

$e^p \in P \equiv$ Process instance of event *e*

$p^d \in D \equiv$ Process definition of instance *p*

$d^s \in S \equiv$ Source of the process definition *d*

The data-relationship constraints described above are expressed in the form of predicates which are defined below:

Predicate 1. For every process definition there exists an associated source.

$\forall x \in D(\exists y \in S : x^s = y)$

Predicate 2. For every process instance there exists a process definition that represents such process.

$\forall x \in P(\exists y \in D : x^d = y)$

Predicate 3. For every event there exists a process instance that is owner of such event:

$\forall x \in E(\exists y \in P : x^p = y)$

Before introducing the next predicate, let's define $e^d \in D$ such as the process definition that represents the event *e*.

Predicate 4. For every event there exists a process definition associated with the event which must be equal to the definition of the process instance that it refers to. This is denoted by the following formulation:

$$\forall x \in E\Big(\exists y \in P(\exists z \in D : y^d = z \ \wedge x^p = y \implies x^d = z)\Big)$$

Before introducing the next predicate, let's define $e^s \in S$ such as the source that originated the event *e*.

Predicate 5. For every event there exists a source from where the event originated which must be equal to the source of the process definition that it refers to. This is denoted by the following formulation:

$$\forall x \in E\left(\exists y \in P\Big(\exists z \in D(\exists \alpha \in S : z^s = \alpha \wedge y^d = z \ \wedge x^p = y \implies x^s = \alpha)\Big)\right)$$

Predicate 6. Let's define the correlation data of an event *e* as a set of key-value pairs $\{\{k_1,v_1\}, \{k_2,v_2\}, \{k_3,v_3\}, \ldots, \{k_n,v_n\}\}$ that is contained in the payload message of an event. This is denoted by e^{cor}. In turn, $e^k \in e^{cor}$ and $e^v \in e^{cor}$represent respectively a particular key and value pair of the correlation set for a particular event *e*.

Two correlation sets of two different events are equal if and only if, both sets have the same size and all key-value pairs from the first set are contained in the second set and vice versa.

$$e_i^{cor} = e_j^{cor} \Leftrightarrow \left(\forall x \in e_i^{cor}\left[\exists y \in e_j^{cor} : x^k = y^k \wedge x^v = y^v\right]\right) \wedge \left(\forall y \in e_j^{cor}[\exists x \in e_i^{cor} : y^k = x^k \wedge y^v = x^v]\right)$$

Before introducing the next predicate, let's define $p^e \in E$ such as the set of events of the process instance *p*.

Predicate 7. The event correlation set must be unique across process or activity instances, so that the inter-relation of events for a specific instance can be univocally identified. Hence, the following formulation must be compiled.

$$\forall x \in P\left(\forall \alpha \in x^e\left(\nexists y \in P : x \neq y \wedge (\forall \beta \in y^e : \alpha^{cor} = \beta^{cor})\right)\right)$$

Objective Function. Following the predicates stated above, we can define the objective function as the correlation function $C(e) = e^p$ that finds the process instance p of the event e in the domain of discourse for all existing events contained in E.

$$C(e) = \begin{cases} \forall x \in E[\exists y \in E : x^{cor} = y^{cor} \wedge x^d = y^d \wedge x^s = y^s] \Rightarrow y^p \\ otherwise, assign\ a\ new\ instance \end{cases}$$

The event correlation algorithm proposed in this paper is an implementation of the objective function formulated and based on the predicates defined above.

Table 1 - Event Correlation Algorithm

Algorithm. Event Correlation

```
1: function correlation (event in E) : return (p in P)
2: begin
3:   for every e in E
4:   do
5:      if (e^d == event^d) ∧ (e^s == event^s) ∧ (size(e^cor) = size(event^cor))
6:      then
7:         found := true
8:        for every c in e^cor
9:        do
10:          if ¬((c^k, c^v) in event^cor)
11:         then
12:             found := false
13:             break
14:         end if
15:       end do
16:
17:       if (found) then
18:          return e^p
16:     end if
20:   end do
21:  return {new identifier}
22: end
```

The algorithm complexity is $O(N^2)$ as it iterates over the correlation set of every event identified in the system. Since it is not possible to know, either estimate, beforehand the execution time of business processes, the event correlation mechanism must have available all historical data in order to identify previous instances. Even though the algorithm complexity is manageable, the input size can be extremely large in big-data contexts, thus making the algorithm inefficient and unable to correlate instances in real-time. In this case, clustering capabilities must be used to mitigate this handicap by distributing the processing load across different servers; however this might potentially span to hundreds or thousands of servers depending on the business case and the volume of data. In order to keep hardware investments at minimum, we have leveraged secondary indexes in HBase in order to achieve the algorithm to run in real-time. The big-data tables store the entire set of events $e \in E$ using an event model that is based on the business process analytics format (BPAF) which is ideal to meet the purposes of this research work.

3.2 Event-based model and repository

In order to enable analysts to infer knowledge about business performance we need to define an event model to provide the framework of a concrete understanding and representation of what needs to be monitored, measured and analysed [30]. The proposed event model is based on the BPAF standard, and it provides the information required to enable the global system to perform analytical processes over them, as well as representing any derived measurement produced during the execution of any business process flow. BPAF supports the analysis of audit data across heterogeneous PAIS systems [32], and it enables the delivery of basic frequency and timing information to decision makers, such as the cycle times of processes, wait time, etc. This format has been extended for meeting the requirements of the correlation algorithm.

As previously mentioned, the enterprise events are correlated as they arrive by querying the event repository for previous instances. This is achieved by fetching the existence of a process instance associated with the correlation data provided. If no data is returned, it means that a new process has been created at the source system; thereby a new process instance is generated at destination. In such a case, a new identifier to the process instance is assigned that will later be used to correlate the subsequent events as they arrive.

According to [10], the read operations over the row key in the event table are performed in the order of milliseconds on very small clusters that handle hundreds of millions of event records. Since read operations present very-low latency based on the row keys, we have leveraged this feature to enable the correlation algorithm to run at very low latency rates. This is achieved by using secondary indexes in HBase over the event correlation table. The correlation table is basically a register of correlation data associated to an event instance that follows the BPAF format. The idea behind this approach is to create an alternate HBase table that will be used as a secondary index for the event correlation table by using the triplet filter as a row key with the aim of speeding up the HBase scans. The row key is established as a byte stream sequence of strictly ordered key-value pairs for every event such as follows:

RowKey: Key1Value1Key2Value2KeyN…ValueNSourceModel cf: "event_correlation" {eventId}

This enables the event correlation mechanism to have immediate access to events that meet the set of key-value pair conditions for a specific source and model. At this stage, the event repository storage may grow to very high volumes of event information due to the continuous execution of processes over the business lifetime, and thus identification of consecutive

instances along the mass set of data becomes cumbersome. Depending on the business case, the volume of the event storage can rise to the order of TBs, PBs or even EBs of information.

The use of secondary indices requires data duplication but it provides an extraordinary response time on read operations, albeit to the detriment of writes. Achieving low rates on reads is essential for the correlation algorithm in order to rapidly identify process or activity instances, and is key for having metrics available on time.

Once the instances are correlated in real-time, we already have the event streams ready for analysis. Both event-data and process models together are essential to infer knowledge about process improvement. Process models by themselves represent the structural aspects of process instances, but a purely structural representation is not enough to construct a solid understanding of what needs to be monitored and improved. It is also required to capture and represent the behavioral state of these processes. Therefore the measurement of process performance is equally a critical factor, and thus the construction of metrics and KPIs must be accomplished at minimum latency.

3.4 Metrics

One key challenge in decision making is having access to all relevant information in order to undertake a performance and compliance assessment. In order to provide Business Activity Monitoring functionality in real time, the construction of metrics and KPIs must be performed at minimum latency. At the most basic level, operational systems deliver timing information on the event occurrence. The timestamp of these events can be used to generate metrics per instance or activity by analysing the state transitions on the event stream, thereby providing business analysts with an understanding of the behavioural aspects of business processes [33]. With the aim of keeping the latency of this process at minimum, the framework incorporates an intermediate in-memory cache solution with event data eviction. As the event correlation identifies sequence of events in time as they arrive, those events are temporarily stored in a distributed cache, so that a large number of events co-exist during a variable period of time, in both permanent storage and cache. The event cache has been implemented using Infinispan and configured as a distributed cache with replication (owner nodes). Every time an event is correlated in the stream chain, that event is cached and associated to the event stream of its process instance. In addition, there is an observer object which is continuously listening to events that transition towards a COMPLETED state. At that precise moment, the entire event stream is read from cache for that particular instance in place and forwarded to the data warehouse module for processing. Upon the notification of a new incoming instance that has finished its execution, the metrics processor analyses the event stream sequence of the instance thereof and produces the metrics according to the state transition changes based on zur Muehlen & Shapiro's model. These metrics are described in the previous sections and they are discussed in [33].

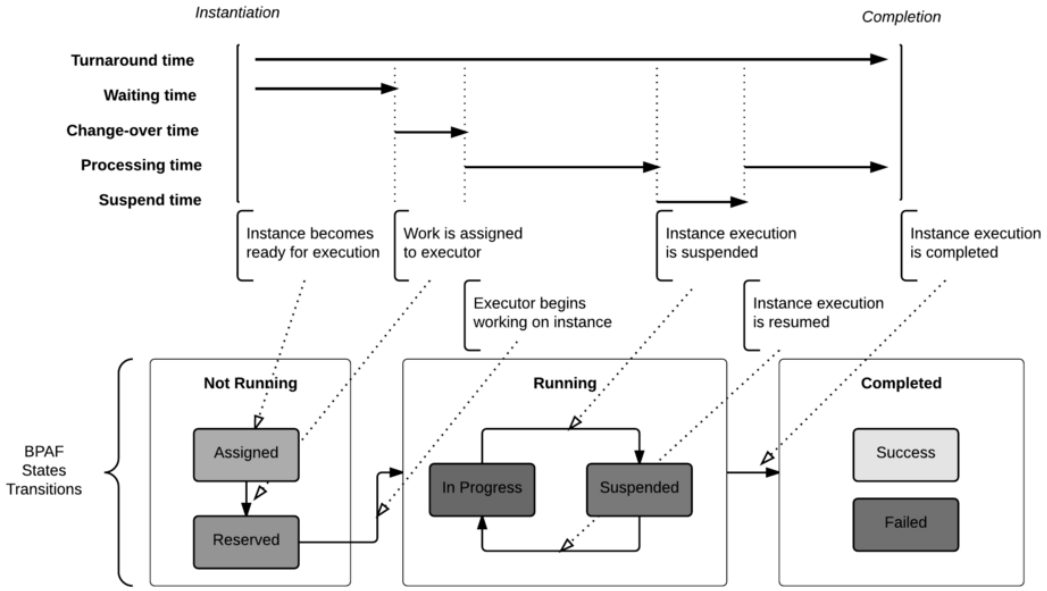

Fig. 4 - Performance metrics generation (adapted from [33])

Once the metrics are generated, these are stored back in cache and persisted in HBase. The reasons why metrics are stored in cache for a fixed period of time with data eviction backed by HBase, is twofold: 1) enabling low-cost time access to metrics of the latest instances with the aim of supporting real-time BAM capabilities, and 2) giving the framework real-time support for retrieving metrics of newly processed instances through API calls.

3.5 Deployment

The deployment of the cloud-based infrastructure previously discussed in an operational environment is challenging and it requires powerful computational resources to be able to provide timely analytics over big-data. Firstly, a BASU node must be provisioned per organization involved in the supply chain. Each BASU unit needs at least one server that will run the BASU instance, plus a big-data storage, which in turn, requires a set of clustered data nodes (1 master and a number of slaves). Depending on the nature of the business process that we aim to monitor and improve, we may have to deploy a considerable number of nodes, thereby the grade of complexity and the number of computational resources will grow significantly.

Fig. 5 illustrates the deployment of the IT solution in AWS (Amazon Web Services) platform. Therein we can notice that there is an availability zone per BASU node that is deployed over a virtual private cloud (VPC). Every node has a Cloudera CDH 4.7.1 installation with a specific number of HBase nodes (at least one master and multiple slaves), which in turns rely on HDFS for storage. All components are settled in a private network for data protection. Local analytical applications can access to their internal data within the organization through the BASU API, so that the analytical data is secured and inaccessible from outside the VPC. Global performance data is shared and published onto the GBAS component, thereby being visible and accessible to third-party applications through the global GBAS API. Consequently, business users are able to access those cloud services from anywhere at any time. The GBAS component has a similar deployment to the BASU nodes. The process performance data in GBAS is partially duplicated, so the number of data nodes should be greater than those deployed in a single BASU node. The

provision of these components in the cloud provides a powerful set of services that allow software engineers to build powerful ad-hoc analytical applications for doing timely performance analytics such as real-time monitoring, simulation, prediction and visualization. In addition, this solution fosters the collaboration between business users and across organizations by sharing cross-organizational performance information. This system provides a core infrastructure for the next generation of business intelligence systems to support business process intelligence.

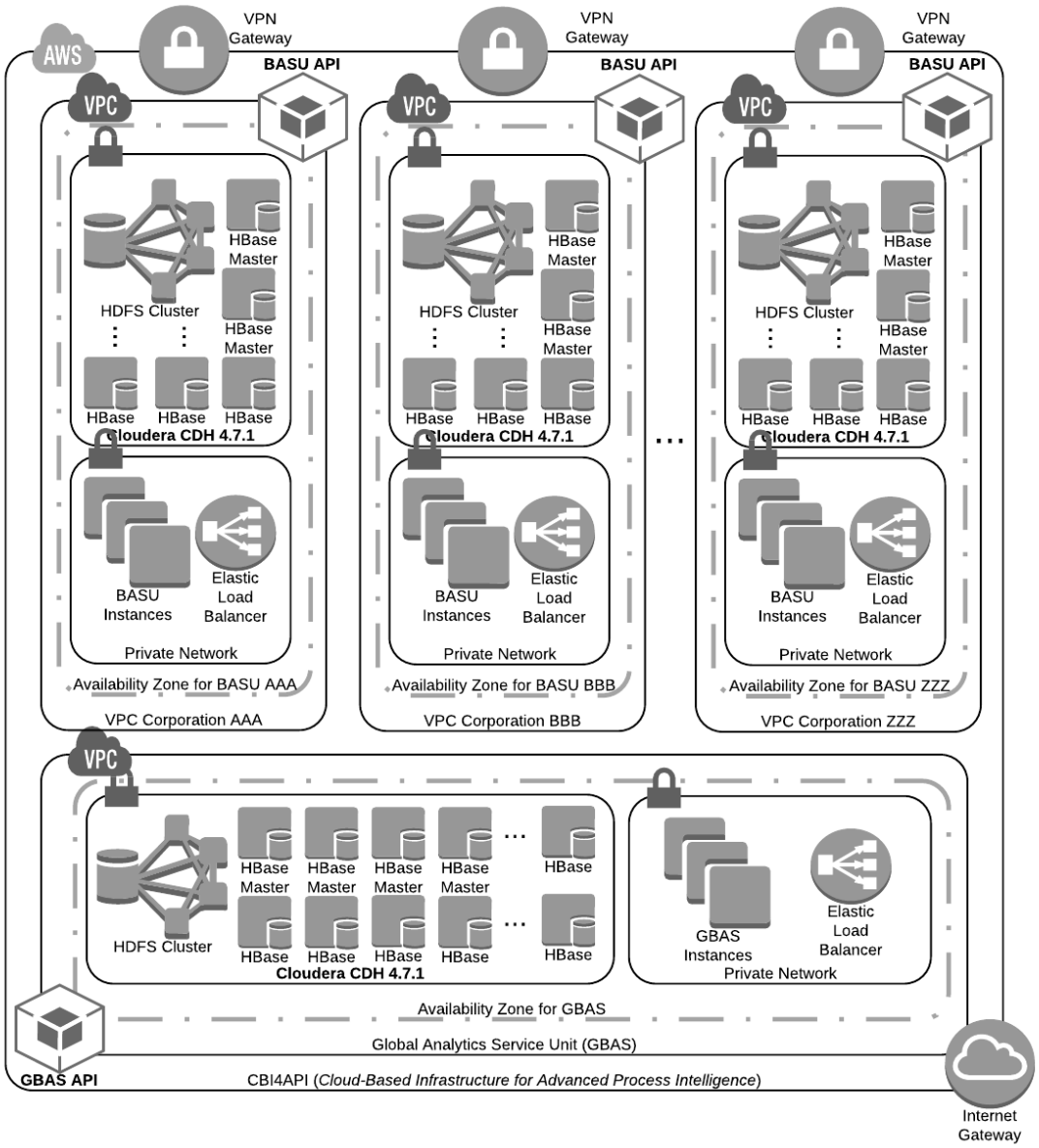

Fig. 5 - Infrastructure deployment on Amazon EC2 Services

4. Evaluation

A case study conducted in the area of smart cities has been leveraged to undertake the performance analysis of the solution. This case study aims to monitor and analyse the processes involved on smart services offered by the city of Chicago. The city of Chicago adopted in 2012 a common standard for 311 reporting known as "Open311" [34]. This open standard is being

adopted worldwide in multiple urban areas, and brings governments the ability to build uniform interoperable systems that allow citizens to interact with their cities in the form of a broad range of information and services.

The case study followed the methodology presented in [12] that integrates with the IT solution previously introduced with the aim of putting real BPI (Business Process Improvement) technology in business users' hands. Such methodology is beyond the scope of this paper, but its fundamentals have been applied to the real use cases of the city of Chicago to determine and represent the nature of the business domain such as process models, instance correlation essentials and specification of operational systems' interfaces, among others.

During the definition and configuration phase, we modelled the smart services as processes whose operational data is accessible through Open311 interfaces, thereby allowing the system to collect online data straight from the Open311 systems through the use of bespoke event capturing software (see Fig. 6). In this context, the listener is responsible for specifying which part of the message payload will be used for correlation of instances associated to a specific model (service). It captures the events and publishes them to the network throughout ActiveMQ message brokers. The listener emits the event information to different endpoints depending on the message format provided. Currently, the platform supports a variety of widely adopted formats for representing event logs such as XES, MXML and BPAF. Consequently, a different set of plugins are available per supported event format, and in turn, each plugin incorporates specific ETL (Extract, Transform & Load) functions to convert source event streams into BPAF.

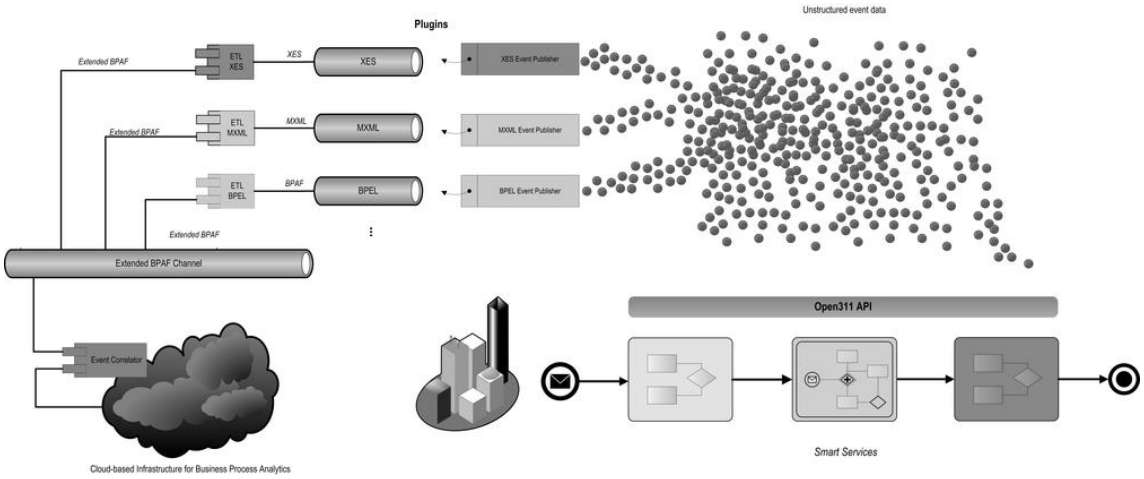

Fig. 6 - Contextual diagram of event capturing and correlation

For the listener implementation we devised an Open311 API client with ETL support that was continuously invoking the Open311 services to retrieve up-to-date status information of all requests available. Therein, the Open311 API endpoint acts as data source (extract phase), the incoming data in JSON format is analysed and converted into events in BPAF format (transformation phase), and those events are then forwarded to a specific channel for processing (load phase). The event correlation module was subscribed to this channel listening continuously for new incoming events. Thereby, the enterprise events are correlated as they arrive by querying the event repository for previous instances as the objective function stands.

The evaluation performance of the framework was carried out over single BASU unit deployed on an environment using a 4-nodes cluster for the big-data solution plus a number of nodes for

the functional modules. The infrastructure of the analytical framework was deployed on a variety of servers, and a vast amount of data was collected from the operational systems that rose to nearly 500 hundred millions of records. Outstanding results on the correlation algorithm were achieved, which manages to link consecutive instances in between 0 and 3 milliseconds for such volumes of data. Whereas the cluster size was very small, the correlation process performed at very low latency rates, thus exceeding the author's expectations. As expected, the performance of the metrics generation process was equally excellent as the entire operation is done in-memory thanks to the built-in cache system.

The experiment ran continuously during two months generating nearly 500 hundred million event data records, including event payload and derived information. This corresponds to approximately 50 million of structured events after processing, which entails a volume of 100GB of raw data assuming an estimated size of 2KB of raw data per event gathered at source. It is important to point out that this is a prospective study and ongoing and future effort is aimed to progressively increase the volume of data to the levels of TBs of information. Nonetheless we present the excellent preliminary results obtained in here.

The first significant finding on the outcomes is that the read operations over the HBase secondary index performed in the order of few milliseconds (see Fig. 7). This implies the event correlation algorithm can run at minimum latency, thus linking events as they occur without undergoing any delay that may impact the generation of metrics in real-time. The second finding is that the read execution time remained stable over time and did not increase as the number of events grew. This is especially important where the cluster size is very small. This shows that the IT solution framework is able to produce timely metrics for such high volumes of data with minimum hardware investments and using the event correlation algorithm proposed.

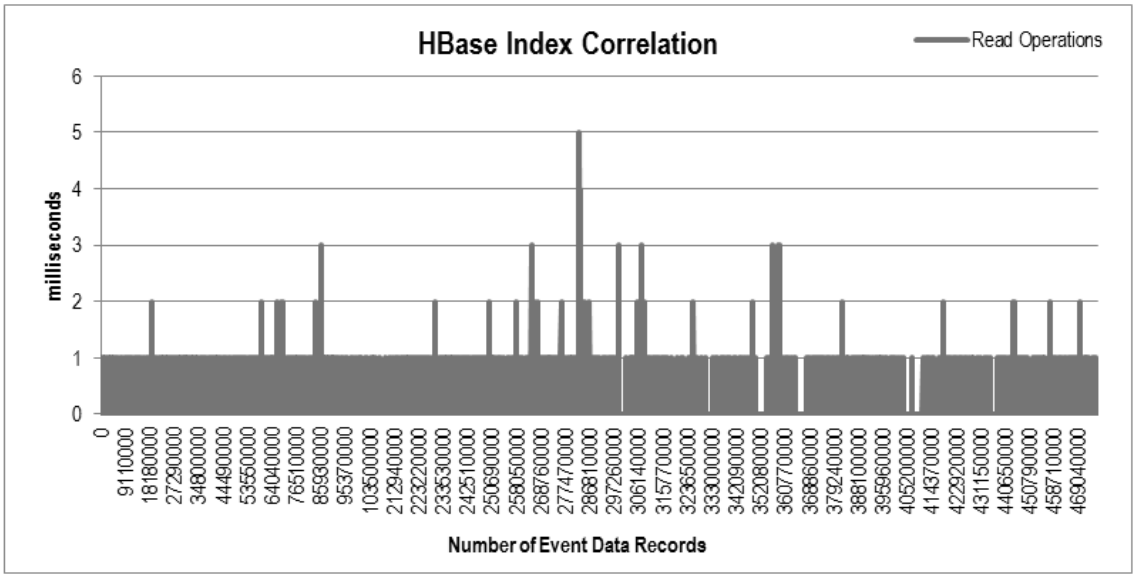

Fig. 7 - HBase read operation performance

With respect to the write operations, the response time obtained was slightly worse than the reads, but equally good for providing real-time outcomes (see Fig. 8). This was expected to occur, as it is known that HBase features outstanding performance on reads to the detriment of writes. Hence, this experiment has reinforced that fact.

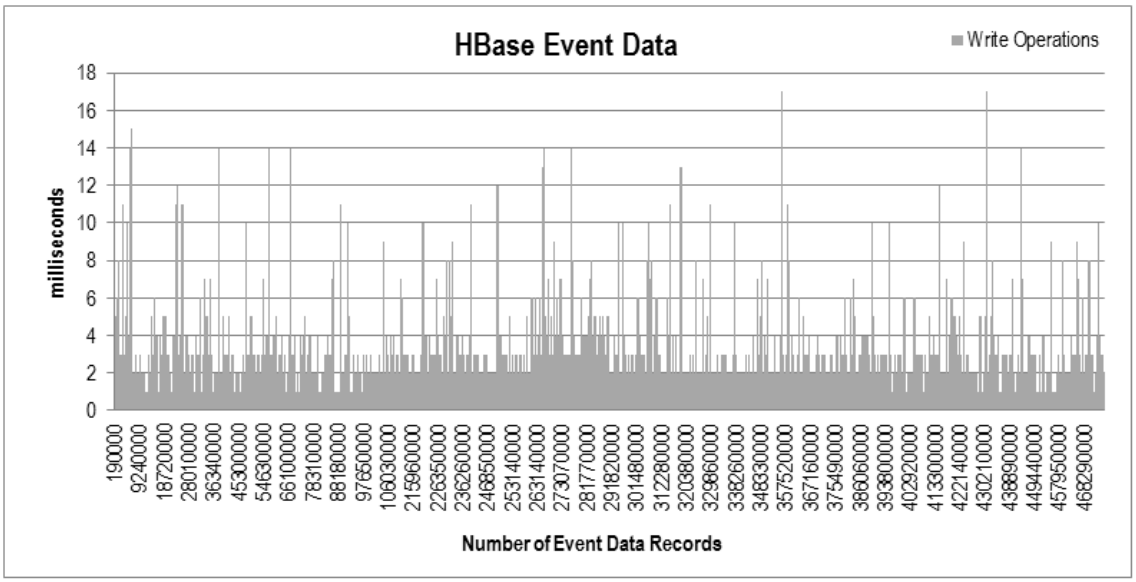

Fig. 8 - HBase write operation performance

The next graphic shows both I/O measures over time while the volume of event data grows (see Fig. 10). It can be clearly seen that rowkey scans (reads) performed much better than writes. Nevertheless, the writes experienced an excellent performance as it never rose above 18 milliseconds, thus running in between 2 and 4 milliseconds most of the time.

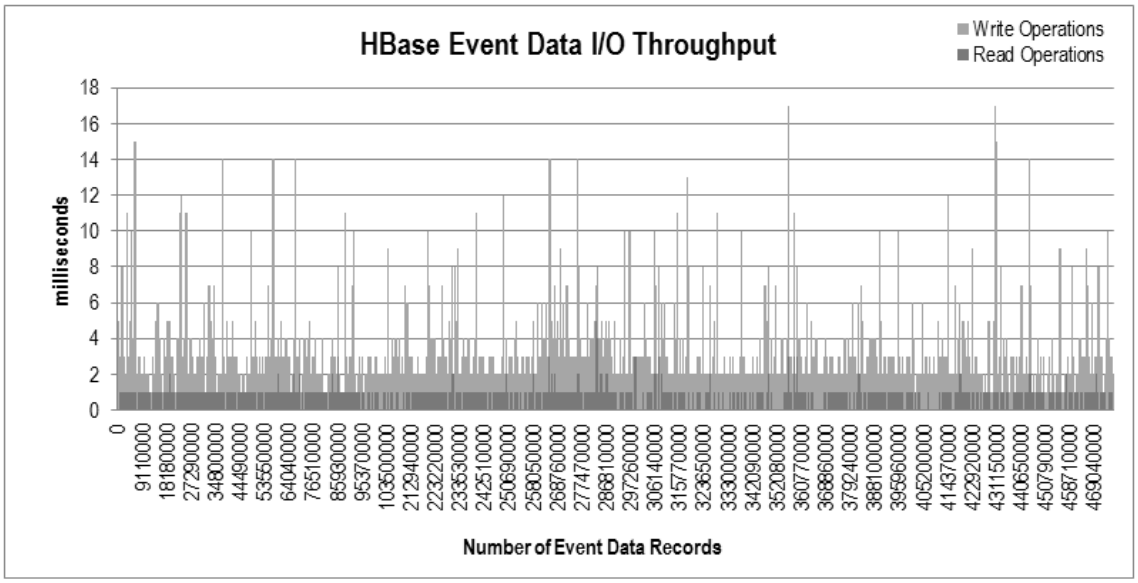

Fig. 9 - HBase read/write operation performance

From a purely event-correlation algorithm perspective, not only the read operations are crucial, but also the writes as the events are correlated by finding their predecessors in the event repository. Every event is stored immediately in the repository after being correlated in order that it can be found by its successors. Any significant delay in the events write may impact on the overall performance of the correlation algorithm. The next graphic overlaps both measures and provides an insight into the overall HBase I/O throughput in relation to the event correlation algorithm. The total amount of time taken by both operations during the algorithm execution is illustrated. It also specifies a real-time threshold set to 50 milliseconds, which indicates the margin of time for the system to produce real-time metrics after getting the instances correlated

in a big-data environment. This entails a margin of more than 30 milliseconds to analyse an event stream in memory and generate its corresponding metrics. This leads us to assert that the system proposed, along with its built-in correlation mechanism, can perform on real-time with only a 4-nodes cluster size for such volume of data.

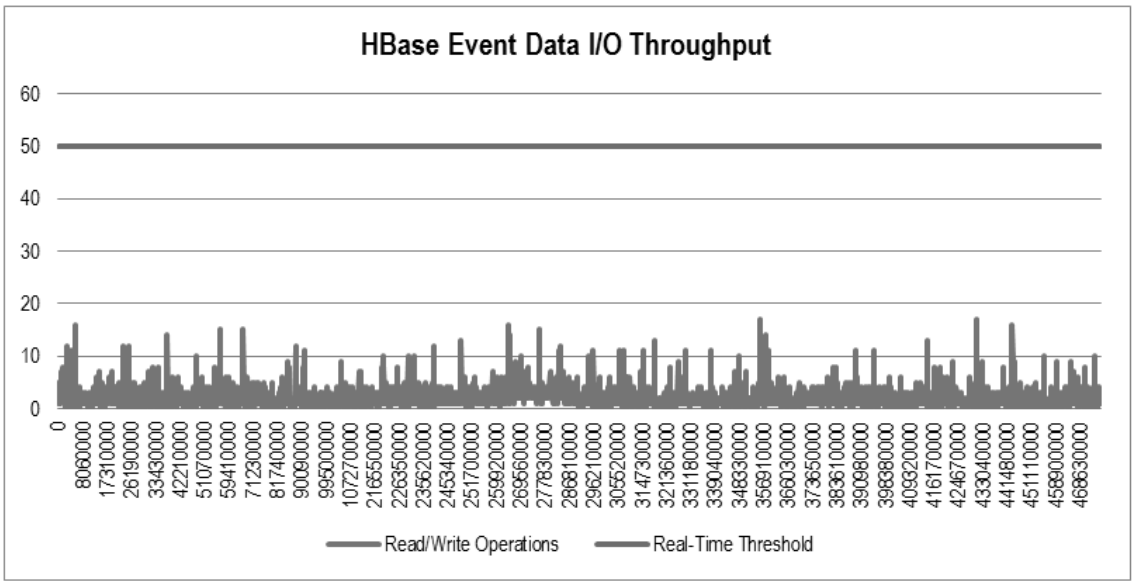

Fig. 10 - HBase overall I/O performance with real-time threshold

5. Conclusions and Future work

This paper has introduced a cloud-based infrastructure that supports business process analysis in the context of business performance improvement. Three main aims have been pursued in relation to the architectural solution devised: 1) to keep the system functionally agnostic to any business domain, 2) to provide analytical performance information in a timely manner, and 3) to integrate distributed event data regardless of the internal concerns of the enterprises' business systems. The BPAF format has been leveraged and extended to construct a generic model that represents the execution outcomes of any business process, thereby making the system non-domain dependent. Likewise, a correlation mechanism has been devised for linking and ordering vast amount of event instances per process or activity. Consequently, big-data technology has been used to effectively manage large volumes of event data, thus providing analytic metrics in real-time. Lastly, the system has been built upon an event-driven architecture that conducts data integration of end-to-end processes, thereby achieving the collection and unification of data regardless of the underlying technology of external sources.

Further efforts are still needed in order to complement the current work. In an ideal scenario, business process analytical techniques will be performed over a very large amount of data. This is produced by the continuous execution of processes during the business lifetime. Commonly, analysts need to know how the business behaved during a certain period of time, learn from the errors experienced in the past or see the evolution of business operations over time. Thus, historical analysis over the entire amount of data becomes key to analysts. In this regard, the scalability of the IT solution gains an enormous significance in the system evaluation. The elasticity features of the cloud-based solution are essential for increasing the computational power in order to meet the performance demands of the queries workload. The use of HBase clustering capabilities and in-memory cache distribution becomes essential for addressing

potential performance issues on event-correlation due to two main factors: 1) the high dependency of the event correlation mechanism on the data access, and 2) the high event-arrival rates on highly distributed environments. Future endeavours will be focused on gradually increasing the rate of the incoming events per second in order to analyse the impact of data overflow on cache. This could be easily solved by scaling out the distributed cache by adding new nodes, but this might affect the performance of the overall solution in highly transactional environments as the data have to be replicated across the cluster. Alternative technologies like kafka or big-data stream solutions such as Storm or Spark, must be evaluated to deal with huge input rates of event data that could easily overflow the cache event space. In order for this to happen, the input rate should be higher than the framework processing rate, and this is unlikely to happen since the event correlation algorithm runs in just a few milliseconds. This action would need the generation of millions of event data per second in order to cause the cache to overflow. Only extremely high transaction-rate business cases might be affected by this limitation. Even though the scalability features of the system can mitigate these edge-cases to some extent, future work will be focused on analysing the framework response based on this scenario.

Perhaps, the most interesting area to explore in the near future is the integration of the IT solution with process mining techniques. This could highly extend the system functionality by providing a fully integrated environment. The system has been shown to have great capabilities to provide monitoring activities in real-time as well as gathering distributed event logs regardless of operational system technology and location. This could close the loop between event generation and post-execution analysis by contributing with the provision of real-time monitoring activity services. In this way, the system may complement the myriad of existing tools and serve as event collector for supporting process mining functionality in real-time. Moreover, the framework has built-in support for converting BPAF event data into XES, which is the standard format used on process mining. This could be leveraged to extend the functionality of the framework beyond the monitoring and performance analysis features. XES data could be used in real-time for applying process mining techniques such as process model discovering, conformance checking, and so forth.

Other potential further research includes the gradual incorporation of services for supporting advanced functionality that can be supported by emerging technologies and optimization techniques. The provision of simulation techniques would highly empower the cloud-based functionality since structured data may serve as an input to simulation engines. This will enable business users to anticipate actions by reproducing what-if scenarios, as well as performing predictive analysis over augmented data that constitutes a base of hypothetical information. Likewise, this would enable analysts to reproduce live process instances and re-run event streams in simulation mode for diagnostic and root cause analysis purposes. Again, process mining tools could be leveraged in this regard. Collaborative business analytics is another potential research area to explore. The cooperation and data sharing between different companies or organizations using big-data would significantly improve not only the visualization of inter-related business analytical information in real-time, but also help to identify and collaboratively perform diagnostics and root-cause analysis on non-compliant situations along large and complex distributed business processes.

References

[1] J. Keller and H. A. von der Gracht, “The influence of information and communication technology (ICT) on future foresight processes — Results from a Delphi survey,” *Technological Forecasting and Social Change*, vol. 85, pp. 81–92, Jun. 2014.

[2] A. Delgado, B. Weber, F. Ruiz, I. Garcia-Rodríguez de Guzmán, and M. Piattini, “An integrated approach based on execution measures for the continuous improvement of business processes realized by services,” *Information and Software Technology*, vol. 56, no. 2, pp. 134–162, Feb. 2014.

[3] E. Herranz, R. Colomo-Palacios, A. de Amescua Seco, and M. Yilmaz, “Gamification as a Disruptive Factor in Software Process Improvement Initiatives,” *j-jucs*, vol. 20, no. 6, pp. 885–906, Jun. 2014.

[4] I. Ruiz-Rube, J. M. Dodero, and R. Colomo-Palacios, “A framework for software process deployment and evaluation,” *Information and Software Technology*, vol. 59, pp. 205–221, Mar. 2015.

[5] W. M. P. van der Aalst, J. Nakatumba, A. Rozinat, and N. Russell, “Business Process Simulation,” in *Handbook on Business Process Management 1*, P. D. J. vom Brocke and P. D. M. Rosemann, Eds. Springer Berlin Heidelberg, 2010, pp. 313–338.

[6] W. M. P. van der Aalst, “A Decade of Business Process Management Conferences: Personal Reflections on a Developing Discipline,” in *Business Process Management*, A. Barros, A. Gal, and E. Kindler, Eds. Springer Berlin Heidelberg, 2012, pp. 1–16.

[7] W. van der Aalst, “Process Mining: Making Knowledge Discovery Process Centric,” *SIGKDD Explor. Newsl.*, vol. 13, no. 2, pp. 45–49, May 2012.

[8] V. Stantchev, R. Colomo-Palacios, and M. Niedermayer, “Cloud Computing Based Systems for Healthcare,” *The Scientific World Journal*, vol. 2014, p. e692619, Apr. 2014.

[9] R. Colomo-Palacios, V. Stantchev, and A. Rodríguez-González, “Special issue on exploiting semantic technologies with particularization on linked data over grid and cloud architectures,” *Future Generation Computer Systems*, vol. 32, pp. 260–262, Mar. 2014.

[10] A. Vera-Baquero, R. Colomo-Palacios, and O. Molloy, “Business Process Analytics Using a Big Data Approach,” *IT Professional*, vol. 15, no. 6, pp. 29–35, Nov. 2013.

[11] A. Vera-Baquero and O. Molloy, “A Framework to Support Business Process Analytics,” 2012, pp. 321–332.

[12] A. Vera-Baquero, R. Colomo-Palacios, and O. Molloy, “Towards a Process to Guide Big Data Based Decision Support Systems for Business Processes,” *Procedia Technology*, vol. 16, pp. 11–21, 2014.

[13] A. O’Driscoll, J. Daugelaite, and R. D. Sleator, “‘Big data’, Hadoop and cloud computing in genomics,” *Journal of Biomedical Informatics*, vol. 46, no. 5, pp. 774–781, Oct. 2013.

[14] D. Talia, “Clouds for Scalable Big Data Analytics,” *Computer*, vol. 46, no. 5, pp. 98–101, May 2013.

[15] Andy Neely, Mike Gregory, and Ken Platts, “Performance measurement system design,” *Int Jrnl of Op & Prod Mnagemnt*, vol. 15, no. 4, pp. 80–116, Apr. 1995.

[16] J. Crump, “Business Activity Monitoring (BAM): The New Face of BPM,” Software AG, Business White Paper, 2006.

[17] C. Costello and O. Molloy, “Building a Process Performance Model for Business Activity Monitoring,” in *Information Systems Development*, W. Wojtkowski, G. Wojtkowski, M. Lang, K. Conboy, and C. Barry, Eds. Springer US, 2009, pp. 237–248.

[18] C. Costello, W. Fleming, O. Molloy, G. Lyons, and J. Duggan, “iWise: A Framework for Providing Distributed Process Visibility Using an Event-Based Process Modeling Approach,” in *8th International Conference on Enterprise Information Systems (ICEIS 2006)*, 2006.

[19] B.S. Sahay and Jayanthi Ranjan, “Real time business intelligence in supply chain analytics,” *Info Mngmnt & Comp Security*, vol. 16, no. 1, pp. 28–48, Mar. 2008.

[20] C. Costello and O. Molloy, “Towards a Semantic Framework for Business Activity Monitoring and Management,” in *AAAI Spring Symposium: AI Meets Business Rules and Process Management*, 2008, pp. 17–21.

[21] D. Kang, S. Lee, K. Kim, and J. Y. Lee, “An OWL-based semantic business process monitoring framework,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 7576–7580, May 2009.
[22] Christian Janiesch, Martin Matzner, and Oliver Müller, “Beyond process monitoring: a proof-of-concept of event-driven business activity management,” *Business Process Mgmt Journal*, vol. 18, no. 4, pp. 625–643, Jul. 2012.
[23] N. Herzberg, A. Meyer, and M. Weske, “An Event Processing Platform for Business Process Management,” in *Enterprise Distributed Object Computing Conference (EDOC), 2013 17th IEEE International*, 2013, pp. 107–116.
[24] G. Srdic and M. B. Juric, “Model for integrated monitoring of bpel business processes,” *Int. J. Coop. Info. Syst.*, vol. 22, no. 02, p. 1350008, Jun. 2013.
[25] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, “Dynamo: Amazon’s Highly Available Key-value Store,” in *Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles*, New York, NY, USA, 2007, pp. 205–220.
[26] K. Grolinger, W. A. Higashino, A. Tiwari, and M. A. Capretz, “Data management in cloud environments: NoSQL and NewSQL data stores,” *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 2, no. 1, p. 22, Dec. 2013.
[27] R. Cattell, “Scalable SQL and NoSQL Data Stores,” *SIGMOD Rec.*, vol. 39, no. 4, pp. 12–27, May 2011.
[28] V. Padhye and A. Tripathi, “Scalable Transaction Management with Snapshot Isolation for NoSQL Data Storage Systems,” *IEEE Transactions on Services Computing*, vol. 8, no. 1, pp. 121–135, Jan. 2015.
[29] S. Gilbert and N. Lynch, “Brewer’s Conjecture and the Feasibility of Consistent, Available, Partition-tolerant Web Services,” *SIGACT News*, vol. 33, no. 2, pp. 51–59, Jun. 2002.
[30] O. Molloy and C. Sheridan, “A Framework for the use of Business Activity Monitoring in Process Improvement,” in *E-Strategies for Resource Management Systems: Planning and Implementation*, E. Alkhalifa, Ed. IGI Global, 2010, pp. 21–46.
[31] H. R. Motahari-Nezhad, R. Saint-Paul, F. Casati, and B. Benatallah, “Event correlation for process discovery from web service interaction logs,” *The VLDB Journal*, vol. 20, no. 3, pp. 417–444, Sep. 2010.
[32] WfMC, “Business Process Analytics Format Specification,” *Workflow Management Coalition*, 2012. [Online]. Available: http://www.wfmc.org/Download-document/Business-Process-Analytics-Format-R1.html. [Accessed: 25-Feb-2012].
[33] M. zur Mühlen and R. Shapiro, “Business Process Analytics,” in *Handbook on Business Process Management 2*, J. vom Brocke and M. Rosemann, Eds. Springer Berlin Heidelberg, 2010, pp. 137–157.
[34] “Code for America Innovation Team Arrives in Chicago to Develop New Open 311 System,” *US Fed News Service, Including US State News*, Washington, D.C., India, 2012.