2014 IEEE International Conference on Cloud Engineering

Principles of Software-defined Elastic Systems for Big Data Analytics

Hong-Linh Truong, Schahram Dustdar
Distributed Systems Group, Vienna University of Technology
E-mail: {truong, dustdar}@dsg.tuwien.ac.at

Abstract

Techniques for big data analytics should support principles of elasticity that are inherent in types of data and data resources being analyzed, computational models and computing units used for analyzing data, and the quality of results expected from the consumer. In this paper, we analyze and present these principles and their consequences for software-defined environments to support data analytics. We will conceptualize software-defined elastic systems for data analytics and present a case study in smart city management, urban mobility and energy systems with our elasticity supports.

1. Introduction

The main characteristics of big data described through the "four V's of volume, variety, velocity and veracity" [1] have steered the discussion and the development of big data techniques into big computing infrastructure (e.g., high performance and data intensive computing/cloud systems), big data storage and scalable data structures (e.g., BigTable and Cassandra), scalable computation frameworks (e.g., Hadoop/MapReduce and S4), and scalable data mining algorithms [2], [3], [4], [5], [6]. However, few discussions have been focused on dynamic and flexible data analytics processes that rely on multi-dimensional elasticity perspectives from consumers and providers, while leveraging these existing powerful computing infrastructures, frameworks, and algorithms. We argue that elasticity principles, such as, resource, quality and cost elasticity [7], should be investigated as fundamental guidelines for developing new data analytics platforms to tackle issues in big data analytics, for example:

- data analytics can be carried out by computational models utilizing software algorithms as well as humans.
- consumers can have different cost/quality requirements in a single analytics which require different types of data taken into the analytics at different times of the analytics.
- data and computing resources are utilized differently to produce different outputs for the same type of analytics.

We argue that these exemplified issues reflect the multiple types of elasticity inherent in big data analytics that data analytics software should be supported. Furthermore, these types of elasticity should be programmed by means of software-defined Application Programming Interfaces (APIs) at runtime to enable dynamic changes. Therefore, we need to understand basic principles of elasticity in big data analytics and possible software-defined APIs for managing and control the elasticity in big data analytics.

In this paper, we describe complex dependencies on data analytics processes (Section 2). We contribute an analysis of elasticity principles for big data analytics (Section 3). Based on that, we conceptualize software-defined elastic systems for data analytics (Section 4), and present a case study of how we currently apply such principles in smart city, urban and mobility systems (Section 5).

2. Dependencies in Data Analytics Processes

Conceptually, given input data, in an analytics we utilize an *analytics (structured or unstructured) process* (e.g., a scientific workflow [8]) which consists of different *analytics tasks* (e.g., an activity in the workflow) to understand and process the data. Analytics processes and tasks rely on *computational models* which implement algorithmic steps to analyze data followed specific data models. A task can invoke a service/program which encapsulates a computational model or a set of tasks can implement a computational model.

Today's one of the popular forms of (big) data analytics is that we have voluminous (static and streaming) data aggregated from different sources and then analyzed at a center place, such as, cloud data centers. In this form, typically the number of analytics processes and the computational models are limited because data models of both input data and output data are known. To face with input data volume and velocity, the common solution is to provision more computing units (e.g., virtual machines). This form is shown in Figure 1(a) and well-supported using common technologies, such as MapReduce, Big Table, and scientific/data analytics workflows.

Having diverse types of data will increase the number of analytics processes and computational models substantially, as different types of data and their compositions require different analytics processes, tasks and computational models (as shown in Figure 1(b)). Thus, we need to go beyond the typical provisioning of more computing units by also provision more computational models and analytics processes at runtime.

Now let us consider the role of dynamic changes of quality of results of the analytics. A simple form of quality of results can consist of performance (e.g., deadline of the analytics), cost (e.g., monetary prices to be paid), quality of data (QoD) (e.g., data accuracy), and forms of output data (e.g., a comma-separated values data or a chart). Due to the four V's characteristics, the consumer expects to have different quality of results. The reason is that the consumer is always

978-1-4799-3766-0/14 $31.00 © 2014 IEEE
DOI 10.1109/IC2E.2014.67

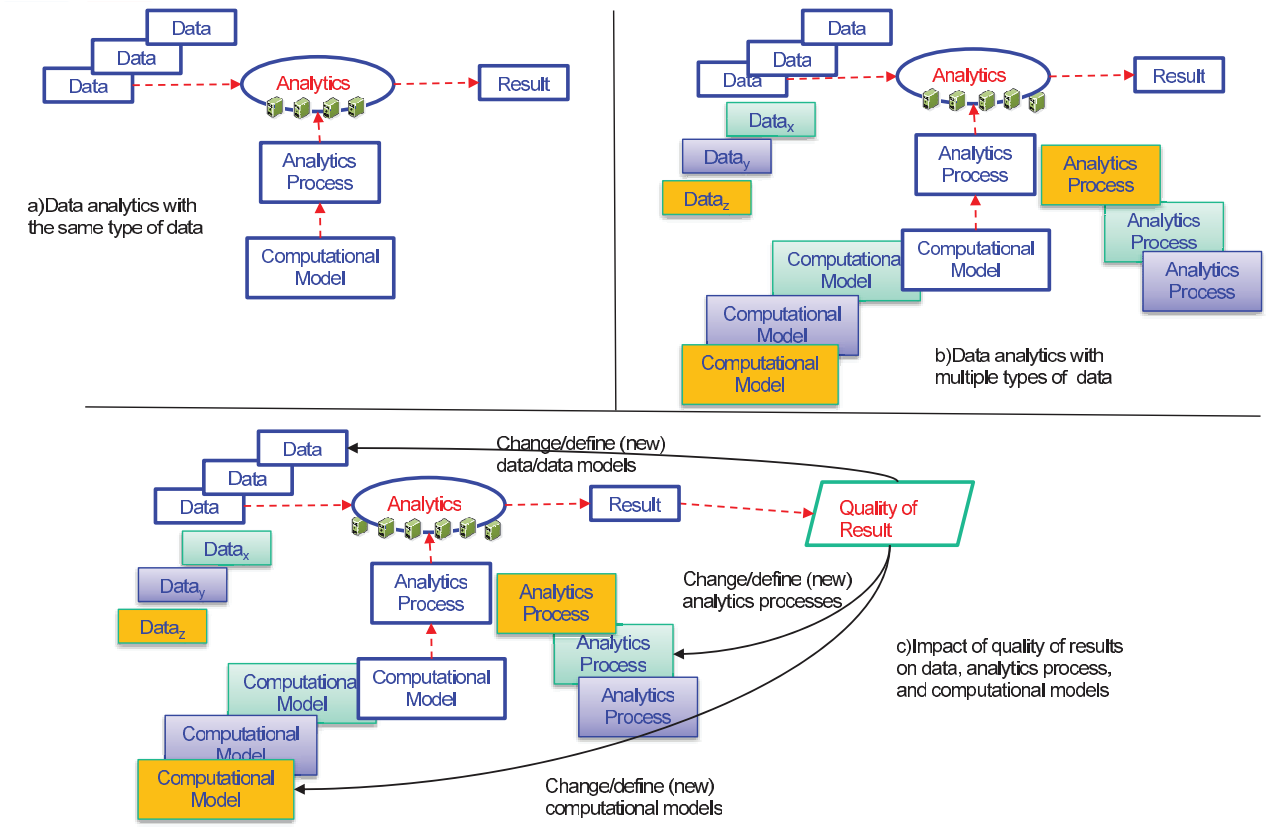

Fig. 1. Changes in data analytics and their complex consequences

constrained with time, quality and cost, as the consumer carries out analytics for different goals. Shown in Figure 1(c), when we know the expected form of data outputs from our analytics, we could focus on steering other parameters of the quality of results, such as QoD, performance and cost. This will influence on several aspects on the analytics, such as which types of data will be taken into the analytics, which analytics processes will be invoked and which computational models will be considered, besides the question of which and how many computing units will be used. In a much more complex situation when data analytics is used for "finding the right form of results", e.g., which is the right plan for putting a factory in a city, the form of output data might not be known in advanced or might be changed during the analytics. This triggers, at runtime, not only the need to change data sources, analytics processes, and computational models but also the need to define new data models, analytics tasks and computational models.

The above-mentioned scenarios show that we cannot just support big data of the same type where we focus on

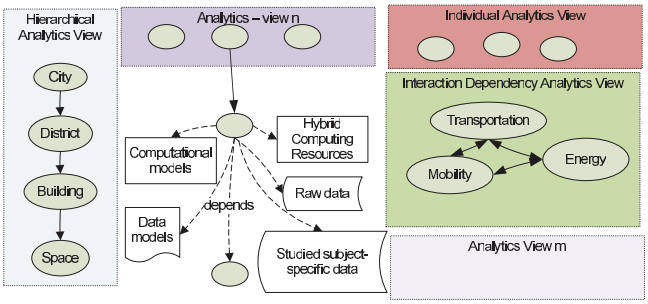

Fig. 2. Multi-view data analytics

provisioning big computational resources for a number of small processes and computational models. In addition, we cannot just support big data gathered in the same place in which we put most effort on, e.g., provisioning Hadoop plus big cloud computing platforms or centralized data mining algorithms. Instead, we must also be able to support quality of result elasticity for the consumer through different views. For example, Figure 2 depicts the analytics under different views,

which require different ways to compose different data and computational models, utilize different data sources, and meet different quality of results. For example, in principle, one analytics for smart cities could be based on a `Hierarchical Analytics View` in which each hierarchical layer, such as `City, District, Building` and `Space`, requires different computational models and data. Another analytics could be based on an `Interaction Dependency Analytics View` in which analytics subjects, such as `Transportation, Energy` and `Mobility` simulations, have complex dependencies that require cause-effect analysis based on their interactions.

3. Elasticity Principles for Data Analytics

3.1. Elasticity of Data and Computational Models

The big data we face does not come from a single type of objects (e.g., big data of twitter messages [9]) but from multiple types of objects from different sources. Therefore, data and computational models for analytics subjects, their dependencies, and their relevant information, quality, policy and processes are complex and diverse. In many cases these models are designed for specific types of objects for specific types of analytics. One challenge is that we probably do not have a clear picture of how many analytics subjects and which kinds of analytics goals for these subjects will be evolving during the analytics. Therefore, our analytics techniques should allow us to decide/select computational models for analytics subjects and to allow the definition/composition of (new) data models based on existing data models and computational services during the analytics. Essentially, this calls for the *management and modeling* of the elasticity of data and computational models during the analytics.

3.2. Data Resource Elasticity

Data in big data analytics in our view will be provided, managed and shared by different providers but based on different data concerns (e.g, different quality of data, privacy, data retention, licensing, and data contract) [10]. The data concerns are crucial that must be designed together with data collection, summarization, exchange and analytics. Currently, service computing techniques have been employed to provision data under different models, such as, the data-as-a-service (DaaS) and data marketplaces. Furthermore, the Internet of Things have also enabled the provisioning of opportunistic data. While it is true that big data analytics require us to put computation on data close to the data, it is also true that in big data analytics we cannot assume data being stored in a single (big) source. Here, the principle of data resource elasticity – data resources can be taken into account in the analytics at runtime when we need them (e.g., due to the expected quality of results) – will need to be supported. This can be achieved through the enabling of the access data from distributed places, e.g., for different customers and analysis processes, based on elasticity controls [11]. Being analytics atop of DaaS, data elasticity controls must also be address other challenges, e.g., pricing mechanism, data privacy assurance, and data contract.

3.3. Elasticity of Human- and Software-based Computing Units

While big data analytics discuss the use of elastic computing resources or crowds separately, we believe that big data analytics should consider the elasticity of hybrid types of computing units: in addition to computing resources, human-based computing units (also called teams and crowds) should be supported to be part of the analytics to solve complex problems that algorithms could not figure out. The role and function of human-based computing units would differ from software components atop big computing infrastructures but nevertheless they all together establish different hybrid compute units for big data analytics. Beyond traditional collaborative working environments in which humans can communicate and share information via common portals or human-based workflows based on crowds [12], we will need to focus on proactive hybrid computing unit formation and life-cycle management in specific data analytics and analytics phases. This requires elasticity techniques to utilize knowledge about experts (such as, skills, domains, availability, and cost) to automatically match and suggest right human-based computing units for solving particular issues during the data analytics (e.g., evaluating the quality of the output of a computational model).

3.4. Elasticity of Quality of Result in Data Analytics

Big data analytics means multi-scale data analytics utilizing diverse types of computational models and computing units (e.g., clusters, Grid, clouds, and crowds). The goal is not just to be able to find insights from vast amount of data based on the consumer's expected quality of result in data analytics. The quality of result is formed, e.g., based on quality of data, cost, and time. It includes complex trade-offs among different quality aspects and has a profound impact/dependency on other criteria, such as, data/computational model elasticity, data resource elasticity, and hybrid compute unit elasticity. This would require us to provide flexible mechanisms to solve several issues related to data storage overload, quality control, and performance to produce meaningful data analytics results. This is much more than just imposing quality data control for data analytics processes.

4. Software-defined Elastic Systems for Data Analytics

4.1. Elastic Objects and Software-defined APIs

Our goal is to provide software-defined elastic systems for data analytics (SES-DA) that support the above-mentioned

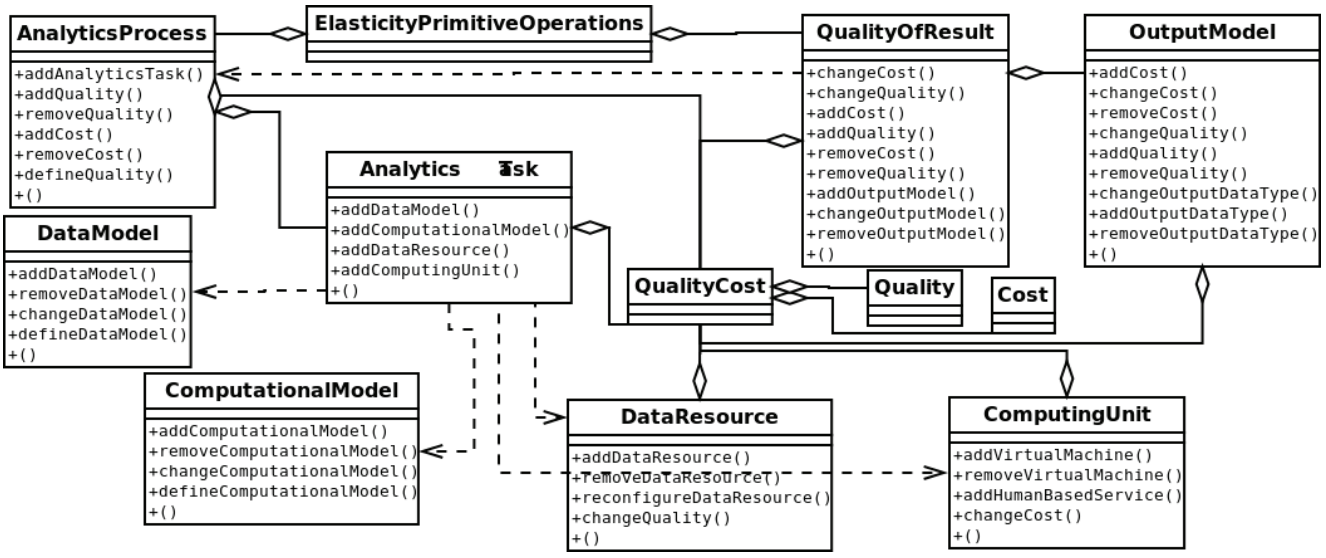

Fig. 3. Examples of elasticity capabilities for software-defined elastic systems for data analytics

elasticity principles atop cloud computing environments. In our view, SES-DA will be constructed from different types of elastic service units, which support the management, creation and execution of data models, computational models, analytics processes, and computing units within the data analytics. Based on the principles mentioned in Section 3, we define different types of *elastic objects*, including *QualityOfResult, AnalyticsProcess, AnalyticsTask, DataModel, ComputationalModel, DataResource, Quality, Cost, ComputingUnit* and *OutputModel*, shown in Figure 3. Elastic objects are associated with elasticity primitive operations. Basically, these operations allow us to add, remove, configure and define (new) data and computational models, computing units, data resources, analytics tasks and analytics processes at runtime. These types of elastic objects are instantiated and executed by SES-DA core elastic service units which do not only provide suitable functionality to enable data analytics (e.g., providing data, analytics algorithms, and computation) but also provide software-defined APIs for managing them. When executing these types of elastic objects, the SES-DA core elastic service units will perform the binding and invoke suitable cloud service units offered by cloud providers. For example, an elastic object represented a data resource can be mapped to data stored in a MongoDB or DynamoDB DaaS.

4.2. Conceptualizing SES-DA

Figure 4 outlines our conceptual SES-DA for supporting elasticity principles in big data analytics. First, we must be able to capture, represent and manage different types of relevant information and relationships (e.g, data, analytics processes, data sources, and dependent analytics subjects). For the data associated with/relevant to analytics subjects, we support two types of DaaS: *Raw Data* and *Subject-specified* DaaS. Data in the first type of DaaS are gathered from different existing techniques, such as, sensors, instruments, and crowds, but not bound to any specific analytics models. For example, we can have urban mobility and energy data from companies/government organizations; they are relevant to different analytics subjects but they do not characterize/represent the status of a specific analytics subject. The second type of DaaS is for managing analytics results of specific analytics subjects. This type includes data associated with analytics subjects which are obtained from data analytic processes that analyze and extract data from the first type of DaaS.

The elasticity of data models and computational models is supported through both design and execution activities. Core, known models are available at design time but their binding to the concrete execution is the subject of elasticity control. This involves elasticity and variability modeling techniques. On-the-fly data models (e.g., using runtime Extract, Transform and Load (ETL) techniques) and computational models (e.g., Python and Matlab scripts) are defined during runtime to deal with complex situations. To execute analytics tasks both software and people can be utilized to carry out the analytics as computing units (supported by *Data Analytics Workflow Platform*, *Hadoop/MapReduce Platform*, or *Human-based Service Platform*). These computing units are modeled as software-based services (SBS) or human-based services (HBS). SES-DA will include *Elasticity Control* to steer the elasticity of different types of computing units to invoke suitable SBS/HBS for suitable tasks and *Programming and Execution Platform* to execute elastic analytics processes. Both consumers and analytics processes can trigger the definition and changes of data and computational models. We need to manage such evolution and utilize different services, such as, provenance management and quality of information evaluation, to ensure that quality is guaranteed and change can be recorded.

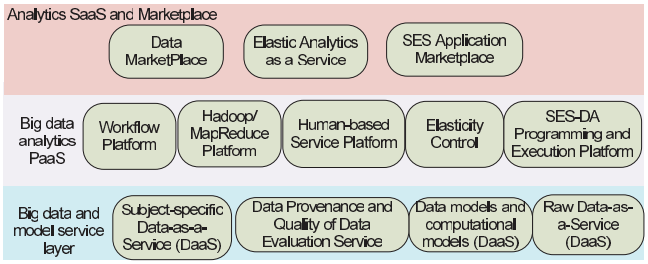

Fig. 4. Conceptual service units of SES-DA

5. Case Study – Data Analytics in Smart Cities

Let us outline a scenario of big data analytics in smart cities as a case study through which explains the above-mentioned principles. In this scenario, we are interested in supporting data analytics for studying and simulating environments, urban and mobility problems in smart cities that we are currently developing in the context of the Pacific Control Cloud Computing Lab (PC3L)[1] and Urban Energy and Mobility (URBEM)[2]. In this scenario, shown in Figure 5, we focus on key research issues related to cloud-based services and processes for big data analytics, such as, modeling and analysis techniques for data and computational model elasticity, data resource elasticity, and elasticity of software and humans as computing units.

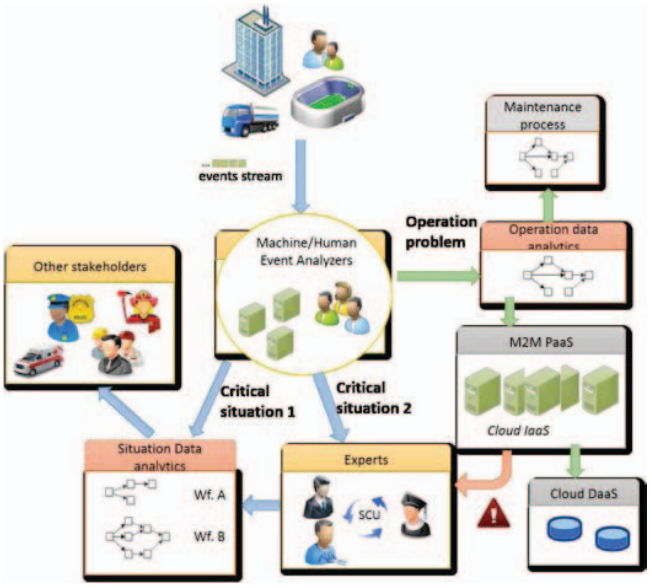

Fig. 5. Example of data analytics elasticity for smart cities

Elastic Data Resources: In our framework, near-realtime sensors can send vast data into an Machine-to-Machine (M2M) cloud system whether sensory data will be stored into NoSQL DaaS where other types of data describing analytics subjects and their dependencies (e.g., buildings and transport networks) are stored into graph-based and relational based data storage. Furthermore, big sources of historical data about transportation, energy consumption and other city-wide data are also provided via DaaS. Due to several constraints, e.g., regulation and economical issues, not all data can be stored in the same places. Therefore, during the analytics, data from different DaaS can be accessed and utilized differently.

Elastic Analytics Process: the big data analytics in our framework starts with a combination of software and humans to analyze near-realtime data – *Machine/Human Event Analyzers*. Most of near-realtime data will be handled by complex software processing complex events. Hence such complex software relies on computational resources elasticity to deal with the volume of the data (e.g., the number of virtual machines is based on the the load of events); this is, e.g., performed by our SYBL service [11]. However, an important issue is how to scale the problem solving when the complex software detects some critical situations? We approach this question by using human-based workflows [13] and using high-level elasticity control language [11] to invoke human-based services, when needed, e.g., when the quality of data is low. The following list shows an example of how to invoke human-based services within an analytics service:

```
#for a service unit analyzing chiller status
#SYBL.ServiceUnitLevel
Mon1 MONITORING accuracy = Quality.Accuracy
Cons1 CONSTRAINT accuracy < 0.7
Str1 STRATEGY CASE Violated(Cons1):
Notify(Incident.DEFAULT, ServiceUnitType.HBS)
```

In this case, both software and humans are involved and human-based services (indicated by `ServiceUnitType.HBS`) are scaled out to examine the situations. This requires us to not only able to scale human-based resources together with machine-based resources but also to able to model the variability of processes dealing with complex events.

Data Resource Elasticity: Using elasticity techniques for data and computational models that are specified both in analytics processes and supported by elasticity controls, we could automatically decide if, given a situation, the next steps will be executing `Operation Data Analytics` to examine if there are some operation problems within the cities (e.g., chiller and air quality problems in large-scale buildings) that would lead to other analytics and actions for maintaining the sustainability of objects with detected problems. Hence these operation data analytics need data resource elasticity support because, due to the situation, requirements for quality of result are very different. Some situations, we do accept a low quality of data output but require a very fast response. For other cases, we do require highly accurate results that should be based on several types of data from DaaS. This leads to the question of how we model the elasticity and variability of the analytics process and how the analytics execution platform can take into account elasticity requirements and behavior. Our approach is to extend analytics process specification with data,

1. http://pc3l.infosys.tuwien.ac.at
2. http://urbem.tuwien.ac.at

things, and human elasticity requirements and variability. At the runtime infrastructure, we use high-level languages, like SYBL, to control the elasticity.

Elastic Hybrid Compute Units: Given the `Operation Data Analytics` process, depending on the data and the expected quality of result, we use SYBL to control machine-based computing units (e.g., virtual machines and networks). The next situation in our case study is that the result from `Operation Data Analytics` might need to be examined by people or the result from `Machine/Human Event Analyzers` signaling different critical situations (`Critical Situations 1` & `2`). Hence we invoke elasticity control to form units of human-based services and configure them in the right structure [14]. The question is how to provision such units and control the quality of their work. Our approach aims at supporting diverse forms of human-based compute units, which can be social compute units (SCU), crowds or hybrid compute units [14]. Furthermore, there is a need to coordinate the activities within computing units and among them. For this, we are devising new techniques to manage hybrid compute units life-cycle and elasticity capability primitives as well as working on cross hybrid compute units coordination protocols.

6. Related Work

Elasticity has been discussed w.r.t. resource elasticity and database elasticity [15] but to our best knowledge, principles of elasticity for data analytics processes have not been thoroughly discussed. Currently, software-defined environments for big data are mainly designed for hardware/software resource management networking [16], storage [17] and machine-based computing units [18]. Our work suggests a multi-dimensional elasticity perspective, leading to the design of software-defined capabilities for not only resources but also quality and costs associated with data and computational models, analytics tasks and processes, and hybrid computing units. Tools for big data analytics have been intensively discussed [5]. We design big data analytics processes from another perspective, where we need to deal with the elasticity of results and data by utilizing elastic hybrid computing units and analytics processes.

7. Conclusions and Future Work

In this paper, we presented main principles of elasticity for big data analytics and we described conceptual software-defined elastic systems for achieving elasticity in big data analytics. Elasticity principles should be investigated deeply in big data analytics techniques to ensure that we can deal with not only the volume, the diversity, and the distribution of data but also expected quality of results. At the end, it is the quality of result which will drive how we take into account of which data sources, algorithm, machines, or people into analytics processes. We cannot assume that we will be able to centralize big data analytics with a huge amount of computational resources for single types of data.

Currently, we are focusing on software-defined elastic systems that support the presented elasticity principles by developing fundamental elastic service units for managing, creating and executing models, computing units and processes. We also work on multi-perspective analytics process variability to enable the modeling of different types of elasticity.

Acknowledgment: This paper is partially supported by the Pacific Control Cloud Computing Lab, and by the European Commission in terms of the CELAR FP7 project (FP7-ICT-2011-8 #317790) and of the FP7 EU project SmartSociety.

References

[1] IBM, P. Zikopoulos, and C. Eaton, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*, 1st ed. McGraw-Hill Osborne Media, 2011.

[2] A. Menon, "Big data @ facebook," in *Proceedings of the 2012 Workshop on Management of Big Data Systems*, ser. MBDS '12. New York, NY, USA: ACM, 2012, pp. 31–32.

[3] A. Jacobs, "The pathologies of big data," *Commun. ACM*, vol. 52, no. 8, pp. 36–44, Aug. 2009.

[4] V. Borkar, M. J. Carey, and C. Li, "Inside "big data management": Ogres, onions, or parfaits?" in *Proceedings of the 15th International Conference on Extending Database Technology*, ser. EDBT '12. New York, NY, USA: ACM, 2012, pp. 3–14.

[5] A. Kumar, F. Niu, and C. Ré, "Hazy: Making it easier to build and maintain big-data analytics," *Commun. ACM*, vol. 56, no. 3, pp. 40–49.

[6] W. Fan and A. Bifet, "Mining big data: Current status, and forecast to the future," *SIGKDD Explor. Newsl.*, vol. 14, no. 2, pp. 1–5.

[7] S. Dustdar, Y. Guo, B. Satzger, and H. L. Truong, "Principles of elastic processes," *IEEE Internet Computing*, vol. 15, no. 5, pp. 66–71, 2011.

[8] E. Deelman, D. Gannon, M. Shields, and I. Taylor, "Workflows and e-science: An overview of workflow system features and capabilities," *Future Gener. Comput. Syst.*, vol. 25, no. 5, pp. 528–540, May 2009.

[9] J. Lin and D. V. Ryaboy, "Scaling big data mining infrastructure: the twitter experience," *SIGKDD Explorations*, vol. 14, no. 2, pp. 6–19, 2012.

[10] H. L. Truong, M. Comerio, F. D. Paoli, G. R. Gangadharan, and S. Dustdar, "Data contracts for cloud-based data marketplaces," *IJCSE*, vol. 7, no. 4, pp. 280–295, 2012.

[11] G. Copil, D. Moldovan, H. L. Truong, and S. Dustdar, "Sybl: An extensible language for controlling elasticity in cloud applications," in *CCGRID*. IEEE Computer Society, 2013, pp. 112–119.

[12] A. Marcus, E. Wu, D. Karger, S. Madden, and R. Miller, "Human-powered sorts and joins," *Proc. VLDB Endow.*, vol. 5, no. 1, pp. 13–24.

[13] M. Reiter, U. Breitenbücher, S. Dustdar, D. Karastoyanova, F. Leymann, and H. L. Truong, "A novel framework for monitoring and analyzing quality of data in simulation workflows," in *eScience*. IEEE Computer Society, 2011, pp. 105–112.

[14] H.-L. Truong, S. Dustdar, and K. Bhattacharya, ""conceptualizing and programming hybrid services in the cloud"," *International Journal of Cooperative Information Systems,*, 2013.

[15] D. Agrawal, A. El Abbadi, S. Das, and A. J. Elmore, "Database scalability, elasticity, and autonomy in the cloud," in *Proceedings of the 16th International Conference on Database Systems for Advanced Applications - Volume Part I*, ser. DASFAA'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 2–15.

[16] G. Wang, T. E. Ng, and A. Shaikh, "Programming your network at run-time for big data applications," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: ACM, 2012, pp. 103–108.

[17] E. Thereska, H. Ballani, G. O'Shea, T. Karagiannis, A. Rowstron, T. Talpey, R. Black, and T. Zhu, "Ioflow: A software-defined storage architecture," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, ser. SOSP '13. New York, NY, USA: ACM, 2013, pp. 182–196.

[18] R. S. Montero, R. Moreno-Vozmediano, and I. M. Llorente, "An elasticity model for high throughput computing clusters," *Journal of Parallel and Distributed Computing*, vol. 71, no. 6, pp. 750 – 757, 2011, special Issue on Cloud Computing.