Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 97 (2016) 144 – 147

Procedia Computer Science

CLOUD FORWARD: From Distributed to Complete Computing, CF2016, 18-20 October 2016, Madrid, Spain

Towards Data-driven Software-Defined Infrastructures

Pedro Garcia Lopez[a,*], Raul Gracia Tinedo[a], Alberto Montresor[b]

[a] *Universitat Rovira i Virgili, Departament d'Enginyeria Informàtica i Matemàtiques, Tarragona, Spain*
[b] *Università di Trento, Dipartimento di Ingegneria e Scienza dell'Informazione, Trento, Italy*

Abstract

The abundance of computing technologies and devices imply that we will live in a data-driven society in the next years. But this data-driven society requires radically new technologies in the data center to deal with data manipulation, transformation, access control, sharing and placement, among others.

We advocate in this paper for a new generation of Software Defined Data Management Infrastructures covering the entire life-cycle of data. On the one hand, this will require new extensible programming abstractions and services for data-management in the data center. On the other hand, this also implies opening up the control plane to data owners outside the data center to manage the data life cycle. We present in this article the open challenges existing in data-driven software defined infrastructures and a use case based on Software Defined Protection of data.

Peer-review under responsibility of organizing committee of the international conference on cloud forward: From Distributed to Complete Computing

Keywords: Data-driven services; software-defined protection; Software-Defined Storage; Data engineering; Middleware

1. Motivation

The data revolution will generate profound changes in our digital societies in the next years. The convergence of technologies like cloud computing, data-driven science and the Internet of Things will boost the creation of a data-driven economy. In this line the European Commission has defined five priority areas to boost digital innovation: 5G, cloud computing, internet of things, data technologies and cybersecurity.

In cloud computing settings, software-defined technologies have attracted a lot of attention in the last years because they offer increased flexibility, programmability and simplified provisioning of virtualized resources.

We advocate for a new generation of Software Defined Data-driven Infrastructures covering the entire life-cycle of data. Examples of data-driven services we are targeting in this proposal are the following:

* Pedro Garcia Lopez. Tel.: +34-977558510 ; fax: +34-977559710.
E-mail address: pedro.garcia@urv.cat

Peer-review under responsibility of organizing committee of the international conference on cloud forward: From Distributed to Complete Computing
doi:10.1016/j.procs.2016.08.293

- Data protection: access control, data sharing, content-based filters, encryption and privacy enhancing technologies.
- Data management: data reduction techniques like compression, deduplication or caching, redundancy mechanisms like coding or replication.
- Data manipulation: data analytics, event streaming, complex event processing, data warehouse and business intelligence, data workflows.
- Data transformation and data layers: code to manage and convert data to different formats, visualization technologies, layers on top of data like annotation, comments, location, or versions.
- Data placement: control where data is located, federated repositories, distributed stores, distributed replication and synchronization.

The importance of the software-defined approach is that the aforementioned services can be provided in a transparent way using control plane (micro-controllers, data policies) and data plane abstractions (active data). Inspired by Aspect Oriented Programming [1], we propose that all the aforementioned cross-cutting concerns to data must be provided in an orthogonal way by extensible services in the control plane.

Unfortunately, Software Defined Platforms are still in their infancy and do not provide higher level data management mechanisms. Software Defined Networking is the most advanced setting, but it is mainly focused on information flows and routing. SDN has also inspired Software Defined Storage architectures like IOFLow[2] and sRoute[3], which are focused on intercepting storage flows in distributed file systems like Samba. IOFLow's major services to the date are bandwidth differentiation and data caching. The flow-centric design of IOFlow and its stateless approach prevents the system to leverage the semantic view of files and folders, which is capital for enabling advanced data centric mechanisms.

A more data-centric approach is the previous works on Active Storage. The early concept of *active disk*[4] —i.e., hard drives with computational capacity— has been borrowed by distributed file system designers in HPC environments (i.e., *active storage*) for reducing the amount of data movement between storage and compute nodes. Concretely, Piernas et al.[5] presented an active storage implementation integrated in the Lustre file system that provides flexible execution of code near to data in the user space.

In this line, IOStack[6] leverages Active Storage technologies to intercept and modify object storage requests. IOStack's control plane is extensible at the data plane (object filters) and at the control plane (policies). IOStack architecture follows a more data-centric model (object-centric) than IOFlow, but it is still providing low level programming abstractions (object-level) and it is mainly targeting Cloud storage administrators.

Another recent work [7] presents Active Data as a new programming model allowing users to react to data life cycle progression. They present a formal model and different examples like a storage cache to Amazon-S3 or a cooperative sensor network. Finally, we want to outline a recent approach called Vertigo[8] advocating for flexible, user-centric and programmable micro-controllers for object storage. This approach presents an architecture in which the control logic of policies is directly *embedded into data objects*. This enables efficient execution of data-centric policies allowing for flexible manipulation of the data life cycle.

As stated before, we advocate for a next generation of cloud technologies covering the entire data life-cycle. This requires novel programming abstractions and services that boost the programmability and flexibility of cloud data infrastructures. IOSTACK [6] and Vertigo[8] are steps in the right direction because they provide transparent and extensible programming abstractions in the control plane that can be the seed for more advanced data abstractions.

But let us now outline the major challenges for the next generation of data-driven software-defined infrastructures.

- **Data centric approach**: The first requirement is to raise the level of abstraction: instead of manipulating blocks, objects or even files, the platforms should manipulate self-contained data entities. Such data-entities may be accessed by different protocols, and include meta-information and even code fragments enabling the annotation and layering of data services.
- **User centric approach**: Another important requirement is that data owners should retake control of their information in heterogeneous and distributed Cloud infrastructures. This means that owners should be able to manage their own control planes enabling the remote manipulation of data planes located in different providers.

- **Programmability and extensibility:** An important challenge is to simplify data management with novel programming abstractions. Examples like Amazon Lambda functions or Vertigo micro-controllers can inspire new data programming models because of their flexibility to intercept and react to data flows. Extensibility also implies that platforms must offer plugin-like models that can be extended by third parties very easily.
- **Interoperability:** Finally, it is important to construct on top of open data standards that ease the connection of disparate cloud providers. A good analogy is to achieve IFTTT (If This Then That) like mechanisms for cloud data infrastructures. The popular IFTTT service can now connect disparate Web services in a very simple and extensible way. Simple and extensible protocols could simplify data management across heterogeneous providers.

2. Use case: Software-Defined Protection of Data Repositories

Data protection is a priority for the European Union and it is defined as a fundamental right defined in Article 8 of the EU Charter of Fundamental Rights. Historically, the EU has played a crucial role in driving the development of national data protection laws in the member states.

The adoption of new data protection technologies can be however complex due to the heterogeneity of data infrastructures in the different service providers, companies and public administrations. To cope with this problem, several security projects aimed to provide different degrees of transparency using proxy architectures that intercept the flows to the data stores. But ad-hoc proxy-like interception involves non-standard layers that many providers are reluctant to deploy in their data infrastructures.

Software-defined protection is particularly needed: very often, software-defined architectures are overlooking how security services are provisioned and managed, leading to a scenario in which the datacenter infrastructure is flexible and virtualized, but the security infrastructure is hard-wired.

From a security and trust viewpoint, this separation from the physical layer could translate into a series of benefits to customers. First, "virtualizing" data protection eliminates the dependence on hardware that is expensive to maintain and manage. Because this is software-only, data protection is elastic, which means that data protection can be consumed (and payed for) in as "as you go" model. Second, programmatic management facilitates the automatic protection of data without the intervention of security administrators. Once policies are put in place, new content is automatically covered and controlled under the specified policies. Last but not least, the most important benefit: the possibility for customers to retain control over their data through programming abstractions and APIs which permit, among other things, to bring up, shut down and monitor data protection appliances on demand.

Currently, the "software-defined" paradigm falls short to fulfill the above desiderata. For this reason, we propose to extend this paradigm to the domain of data protection, by providing a collection of abstractions to empower customers with programmatic control over data, providing effective security, privacy, and trust management.

Let us show how we can fulfill the four requirements established in the previous section:

The first requirement (data-centric approach) is to raise the level of abstraction to self-contained data entities. In this line, we propose a novel programming abstraction for data manipulation: the datalet. A "datalet" is a programming abstraction that enables customers to control the entire life cycle of their data. A datalet collects together the computation, the communication protocol and the necessary security algorithms to enforce data protection. And very importantly, it acts as the "hook" to integrate data protection mechanisms into cloud computing environments in a seamless manner.

By leveraging existing knowledge and emerging solutions on cloud security, providing support for self-enforcing security solutions, datalets will contribute to the creation of novel data protection services for companies and public institutions.

The second requirement (user-centric approach) is that data owners retake control of information. A key feature of datalets is that their control plane may be managed remotely by the data owner. This property will enable the customers to take an active role on their data in the cloud, empowering them with full control and ensuring that they have the final word on all the decisions to be taken on their data. By moving not only the computation, but also the communication towards the data, datalets will need to be able to coordinate themselves across multiple protection domains (hybrid clouds) and across multiple datacenters (federated clouds).

The third requirement (programmability and extensibility) is also ensured by datalets. Third parties will be able to create new datalets offering data protection, sharing, transformation or even intelligent distributed placement. In distributed, multi-tenant systems this will also enable the sharing of information among tenants, regulated by service-level agreements quantifying not only the quality of service, but also the type and amount of information that could be potentially shared and collected.

The final requirement is interoperability. Datalets should also be integrated into existing data-intensive frameworks, enabling multiple tenants and multiple (public or private) cloud providers to share data and resources in a controlled, efficient and secure way. Multiple datalets distributed among different service providers may spark distributed computations querying the whole collection of data while ensuring proper protection of the individual datasets as demanded by their owners. These mechanisms can protect data against unwanted queries and also enforce controlled sharing of data entities to data-intensive frameworks.

3. Conclusions

We are living a data revolution that will require a new generation of data-driven cloud technologies. Existing Software-Defined Architectures are still in their infancy, providing low level programming abstractions and a focus on system automation. In this position paper, we expose four open challenges for data-driven software-defined infrastructures: data-centric approach, user-centric approach, programmability and extensibility, and interoperability. We also present a sample use case on Software-Defined Protection of Data Repositories that targets the aforementioned challenges. We believe that the convergence of Cloud technologies, data-driven science, and the Internet of Things requires novel data-driven software-defined models like the proposed in this position paper.

Acknowledgements

This work has been partly funded by the EU project H2020 “IOStack: Software-Defined Storage for Big Data” (644182).

References

1. Kiczales, G.. Aspect-oriented programming. *ACM Comput Surv* 1996;**28**(4es).
2. Thereska, E., Ballani, H., O’Shea, G., Karagiannis, T., Rowstron, A., Talpey, T., et al. Ioflow: a software-defined storage architecture. In: *ACM SOSP’13*. 2013, p. 182–196.
3. Stefanovici, I., Schroeder, B., O’Shea, G., Thereska, E.. sRoute: treating the storage stack like a network. In: *USENIX FAST’16*. 2016, p. 197–212.
4. Riedel, E., Gibson, G., Faloutsos, C.. Active storage for large-scale data mining and multimedia applications. In: *VLDB’98*. 1998, p. 62–73.
5. Piernas, J., Nieplocha, J., Felix, E.J.. Evaluation of active storage strategies for the lustre parallel file system. In: *ACM/IEEE Supercomputing’07*. 2007, p. 28.
6. Gracia-Tinedo, R., García-López, P., Sanchez-Artigas, M., Sampé, J., Moatti, Y., Rom, E., et al. IOStack: Software-defined object storage. *IEEE Internet Computing* 2016;.
7. Anthony Simoneta Gilles Fedaka, M.R.. Active data: A programming model to manage data life cycle across heterogeneous systems and infrastructures. *Future Gener Comput Syst* 2015;**53**:25–42.
8. Sampe, J., Sanchez-Artigas, M., Garcia-Lopez, P.. Vertigo: Programmable micro-controllers for software-defined object storage. In: *IEEE CLOUD’16*. 2016, .