May 2016

Scientific advisor

Francoise Baude (PR), Scale team CNRS I3S
Mail: Francoise.baude@unice.fr
Web page:http://www-sop.inria.fr/members/francoise.baude/

Predictive Big Data Analytics: safe dynamically adaptable data streaming analytics

Context:
As the popularity of Big Data explodes, more and more use cases are implemented using this kind of technologies. But there are some use cases that are not properly tackled by classic Big Data models and platforms like Apache Hadoop MapReduce because of these models intrinsic batch nature. These cases are those where online processing of new data is required as soon as they enter the system, in order to aggregate to the current analysis results the newest information extracted from these incoming data. Such on-line and continuous processing pertains to what is known as continuous query and trigger in the more focused context of databases [1][2], or also as complex event processing in publish-subscribe systems [3]. More generally, processing the incoming data is known as Data Stream processing, and in the big data area is known as real-time data analytics.

Moreover, some situations have the need of what could be named *anticipatory* analytics: given gathered data originating from various sources and combined to get meaningful information out of them, the goal is to **adapt** the current analytics in such a way that it can match to the anticipated coming situation, somehow ahead of time. For instance, doing short-term weather forecast for local places : if suddenly the speed of the wind increases and changes direction while intense rain falls, there is a need for (1) updating the short-term weather predictions, but also for (2) deploying appropriate supervision of the now-in-danger zone, so that, in case of a flooding risk of the new targeted zone, assuming the new zone is an inhabited one, the system gets able to trigger alerts towards the right actors: if now a flooding can reach the hospital, evacuation effectively should start, whereas it was not necessary few minutes before when wind was still soft and given its direction the risky zone was a field full of corn; now, supervising also electricity delivery is required to make sure any blackout during evacuation risk can be anticipated. Another example of anticipated analytics is in multimedia stream processing: for example, a data journalist that is extracting data from its preferred social network and runs analytics on videos to learn about happenings in a specific city area prior to writing its press article, suspects from the broadcasted content to recognize someone possibly involved in a crime scene shown in a concomitant published video; and, accordingly, decides to adapt its current analytics to combine the two information sources around some common data which is the relatives whose accommodation lies in the specific city area, and continues its search focusing now on these relatives acts while still monitoring the initially recognized suspect.

From these exemplary scenarios, we clearly foresee that one trend in big data technologies is in predictive analytics [16][17][18] which by nature requires a strong capability of dynamic adaptation given partial already gained results. In this trend, our claim is that the analytics require to adapt (**even better self-adapt**) to what is happening, given of course some previously user-defined rules dictating which adaptation it could be relevant to decide to trigger.

State of the art:

Several big data platforms geared at real-time analytics have emerged recently: Spark Streaming [4], Twitter's Storm, S4[5]. These platforms allow one to define a program as taking eventually, after a compilation process, the form of a DAG (directed acyclic graph) but to our knowledge, none allows to adapt the program at runtime with respect to its functional/business nature. This is because these languages such as StreamSQL, CQL, StreamIt, and IBM's SPL [7] are generally declarative. As a result, developers focus on expressing data processing logic, but not orchestrating runtime adaptations (i.e. in which situation functional adaptation should happen, how to monitor the adaptation needs, how to effectively modify the program without redeploying a new one from scratch). Some data workflow engines start to appear in the community of large-scale stream processing but without yet an adequate behavioural/functional adaptation capability, as the so far only focus was to allow adaptation to face requirements for non-functional changes dictated by scalability, fault-tolerance, performance needs through node replication or elasticity [6]. From our literature survey done so far, only [8] sketches a solution for functional adaptivity in stream processing languages of big data platforms, even if some anterior works on stream platforms or active databases are a good starting point [9]. Also, an open question remains about the way data streams have to be managed (stopped, paused,…) during the reconfiguration process. It depends about what properties about the data are seek (should all tuples be handled, or can the application afford to lose some of them, etc). Ensuring the needed guarantees requires relying upon a sound data stream analytics programming model and support.

Work (starting point):

As a result from some years of research in the Scale team, Grid Component Model (GCM) [10] is a component model for applications to be run on distributed infrastructures, that extends the Fractal component model. Fractal defines a component model where components can be hierarchically organized, reconfigured, and controlled offering functional server interfaces and requiring client interfaces. GCM extends that model providing to the components the possibility to be remotely located, distributed, parallel, and deployed in a large-scale computing environment (cluster, grid, cloud, etc), and adding collective communications (multicast and gathercast interfaces). Autonomic capabilities can be expressed in the membranes of GCM components, that can drive their reconfiguration at functional level, in a totally **autonomous** manner [10]. If the DAG of a streaming application translates into a component oriented program, then it can naturally benefit from its intrinsic reconfiguration properties, and on the soundness properties[11]. This is one of the expected research question to be addressed in the scope of this PhD: how to benefit from autonomic, high-expressivity, clear functional versus non-functional separation of concerns features of the GCM component-oriented approach in order to support dynamic adaptation of the analytics the streaming application corresponds to. We have started to define a streaming platform, based upon GCM. The goal of this thesis is to pursue this preliminary work [12], and as such innovate even more in the context of sound and self-adaptation of big data stream analytics. Specifically for ensuring guarantees properties, it will strongly rely upon theoretical results got on the formalization of the GCM model ([13],[14]).

References :

[1] Shivnath Babu and Jennifer Widom. 2001. Continuous queries over data streams. *SIGMOD Rec.* 30, 3 (September 2001), 109-120

[2] Scheuermann, Peter, and Goce Trajcevski. "Active Database Systems." *Wiley Encyclopedia of Computer Science and Engineering* (2008).

[3] Eugene Wu, Yanlei Diao, and Shariq Rizvi. 2006. High-performance complex event processing over streams. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data* (SIGMOD '06)

[4]Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, and Ion Stoica. 2013. Discretized streams: fault-tolerant streaming computation at scale. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles* (SOSP '13)

[5] Neumeyer, Leonardo, et al. "S4: Distributed stream computing platform." *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010.

[6] Raphaël Barazzutti, Thomas Heinze, Andre Martin, Emanuel Onica, Pascal Felber, Christof Fetzer, Zbigniew Jerzak, Marcelo Pasin, Etienne Riviere:
Elastic Scaling of a High-Throughput Content-Based Publish/Subscribe Engine. ICDCS 2014: 567-576

[7] Martin Hirzel, Henrique Andrade, Bugra Gedik, Vibhore Kumar, Giuliano Losa, Mark Mendell, Howard Nasgaard, Rboert Soulé, Kun-Lung Wu - *SPL Stream Processing Language Specification* - IBM, 2009

[8] Gabriela Jacques-Silva, Bugra Gedik, Rohit Wagle, Kun-Lung Wu, Vibhore Kumar - *Building User-defined Runtime Adaptation Routines for Stream Processing Applications* - The Very Large Data Base Endowment Journal (VLDB), 2012

[9] Trajcevski, Goce, et al. "Evolving triggers for dynamic environments." *Advances in Database Technology-EDBT 2006*. Springer Berlin Heidelberg, 2006. 1039-1048.

[10] F. Baude, L. Henrio, C. Ruz *Programming Distributed and Adaptable Autonomous Components - the GCM/ProActive Framework* Software: Practice and Experience, Wiley, In Press, 2015

[11] Tatiana Aubonnet, Ludovic Henrio, Soumia Kessal, Oleksandra Kulankhina, Frédéric Lemoine, Eric Madelaine, Cristian Ruz, Noëmie Simoni: Management of service composition based on self-controlled components. J. Internet Services and Applications 6(1): 15:1-15:17 (2015).

[12] F. Baude, L. El Bèze, M. Oliva *Towards a flexible data stream analytics platform based on the GCM autonomous software component technology* In HPCS'2016, workshop on Autonomic HPC. To appear. IEEE, July 2016.

[13] Rabéa Ameur-Boulifa, Raluca Halalai, Ludovic Henrio, and Eric Madelaine - *Verifying Safety of Fault-Tolerant Distributed Components* - FACS 2011 - LNCS – Springer

[14] Ludovic Henrio, Oleksandra Kulankhina, Siqi Li, Eric Madelaine. *Integrated environment for verifying and running distributed components.* 19th Int. Conference on Fundamental Approaches to Software Engineering (FASE'16), LNCS. Springer, 2016.