Strategies for Big Data Analytics through Lambda Architectures in Volatile Environments

Veith, Alexandre da Silva; **Anjos, Julio C. S. dos;** de Freitas, Edison Pignaton; Geyer, Claudio F.

Federal University of Rio Grande do Sul (UFRGS), Porto Alegre -RS

Lampoltshammer, Thomas J.

Danube University Krems, Department for E-Governance and Administration, Dr.-Karl-Dorrek-Str. 30, 3500 Krems, Austria

Contact: jcsanjos@inf.ufrgs.br

INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

GPPD

4th IFAC Symposium on Telematics Applications.
November 6-9, 2016, UFRGS, Porto Alegre, RS, Brazil

Outline

1. Introduction
2. Related Work
3. SMART Model
4. Dispatcher Strategies
5. Conclusion
6. Future Work

Outline

1. Introduction
2. Related Work
3. SMART Model
4. Dispatcher Strategies
5. Conclusion
6. Future Work

Introduction

- Internet of Things (IoT)
  - **Fusion of virtual environments** and contained objects with their real-world counterparts (Uckelmann et al., 2011)
  - The challenge to handle vast amounts of data - **data analytics**
  - Total increase of data driven projects by **125% during the period 2014-2015**[1]
- Stream-processing or Oriented-to-events
  - **Huge increase** in volume and availability (Tudoran et al., 2014)
  - **Overwhelming collection rates**
  - Apache Storm, Spark, Flink or S4

[1] IDG - http://www.idgenterprise.com/

Introduction

- Lambda Architecture (Marz, 2013)
  - **Handle vast amounts of data**
  - **4th Generation** of Data Processing Engines (Ewen et al., 2013)
  - **Robustness**, fault tolerance, low latency of reading and updating, **scalability**, **generalization**, extensibility, **ad hoc queries**, and minimal maintenance
- The improvement of the decision-making engine of the Dispatcher module
  - **SMART** (Anjos et al., 2015)
  - **Large variety** of data sources
  - Several policies to the **managing data and tasks**
- Study and application of different strategies on the SMART-Sent environment
  - SMART is an **Ubilytics** environment

[1] IDG - http://www.idgenterprise.com/

Outline

1. Introduction
2. Related Work
3. SMART Model
4. Dispatcher Strategies
5. Conclusion
6. Future Work

Related Work

Heterogeneous Infrastructures

- **JetStream (Tudoran et al., 2014)** is a set of **strategies** for efficient transfers of events **between cloud data centers**
- **SMART (Anjos et al., 2015)** is a platform that offers an efficient architecture for Big Data analysis applications for **small and medium-sized organizations**
- **(Pham et al. , 2016)** is a generic, extensible, scalable, fine-grained, and **re-configurable multi-cloud framework**

Hybrid Infrastructures

- **BIGhybrid (Anjos et al., 2016)** summarizes **the main features of a Hybrid MR**
- **HybridMR (Tang et al., 2015)** is a model for the execution of MapReduce on **hybrid computation environments** (Cloud and DG)

Related Work

Hybrid Engines

- **Apache Spark (Zaharia et al., 2012)** is a framework that uses **resilient distributed datasets (RDDs)** and **enables efficient data reuse**
- **Apache Flink (Alexandrov et al., 2014)** enables massively parallel **in-situ data analytics**, using a programming model based on second order functions
- **Summingbird (Boykin et al., 2014)** integrates **batch and online analyses** with the aid of a hybrid processing model

Related Work

❑Open Opportunities

- Stream processing only has been performed in **heterogeneous environments**
- Generally the engines were designed to run in clusters and cloud computing environments – using **Round Robin policies** to deploy the tasks. This is not suitable for heterogeneous and dynamic environments (i.e., R-Storm (PENG et al., 2015), P-Scheduler (ESKANDARI; HUANG; EYERS, 2016) and (LIAO et al., 2015))
- Optimize the utilization of the infrastructure – **idle resources**
- Take advantage of the **BIGHybrid Simulation**

Outline

1. Introduction
2. Related Work
3. SMART Model
4. Dispatcher Strategies
5. Conclusion
6. Future Work

SMART Model

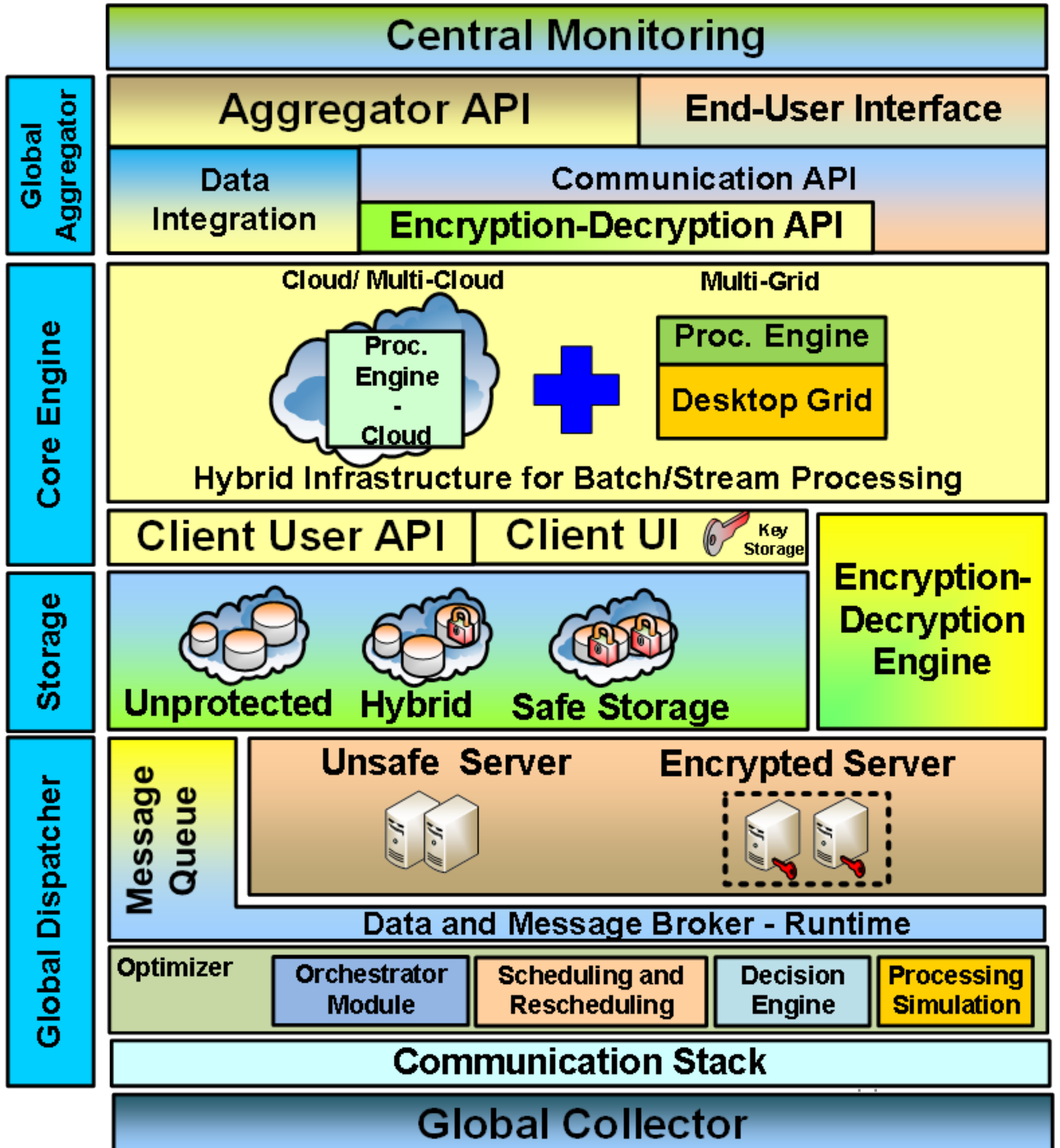
Global Collector - layer handles the **management and coordination** of the sensing modules

Global Dispatcher - **the data is decoupled** from the lower layers in the message queue mechanism. It is put in a FIFO queue so that it can be distributed to severs in accordance to the availability of their resources

- The **optimization layer analyses** the volume of input data and employs the Decision Engine to make decisions about scheduling tasks and data through distinct environments.
- A **simulation process** implements an execution time prediction that will be used by the Decision Engine to improve the accuracy of the scheduling mechanism.

Core Engine - must support hybrid systems, i.e., **provision of streaming and batch** computations at the same time

Global Aggregator is a module that **orchestrates the results** of the aggregation and maintains the safety data mechanism for the end-users.



Outline

1. Introduction
2. Related Work
3. SMART Model
4. Dispatcher Strategies
5. Conclusion
6. Future Work

Dispatcher Strategies

- Applied on the **Global Dispatcher** of the **SMART platform**
- Regarding data distribution in **real time applications,** via solutions presented by Righi et al. (2015)
- Global Dispatcher will carry out the tasks of **load balancing, and latency control** (data stream processing and network bandwidth), **provision of scalability, while reducing the costs for improvement of availability of resources**
- The role of the Decision Engine is to select the **computing resources needed for carrying out a task**
- The task definition will be achieved by **simulation**, which will use **BIGhybrid**, and at the same time, to evaluate the environment for the re-scheduling processes (data placement and data movement)

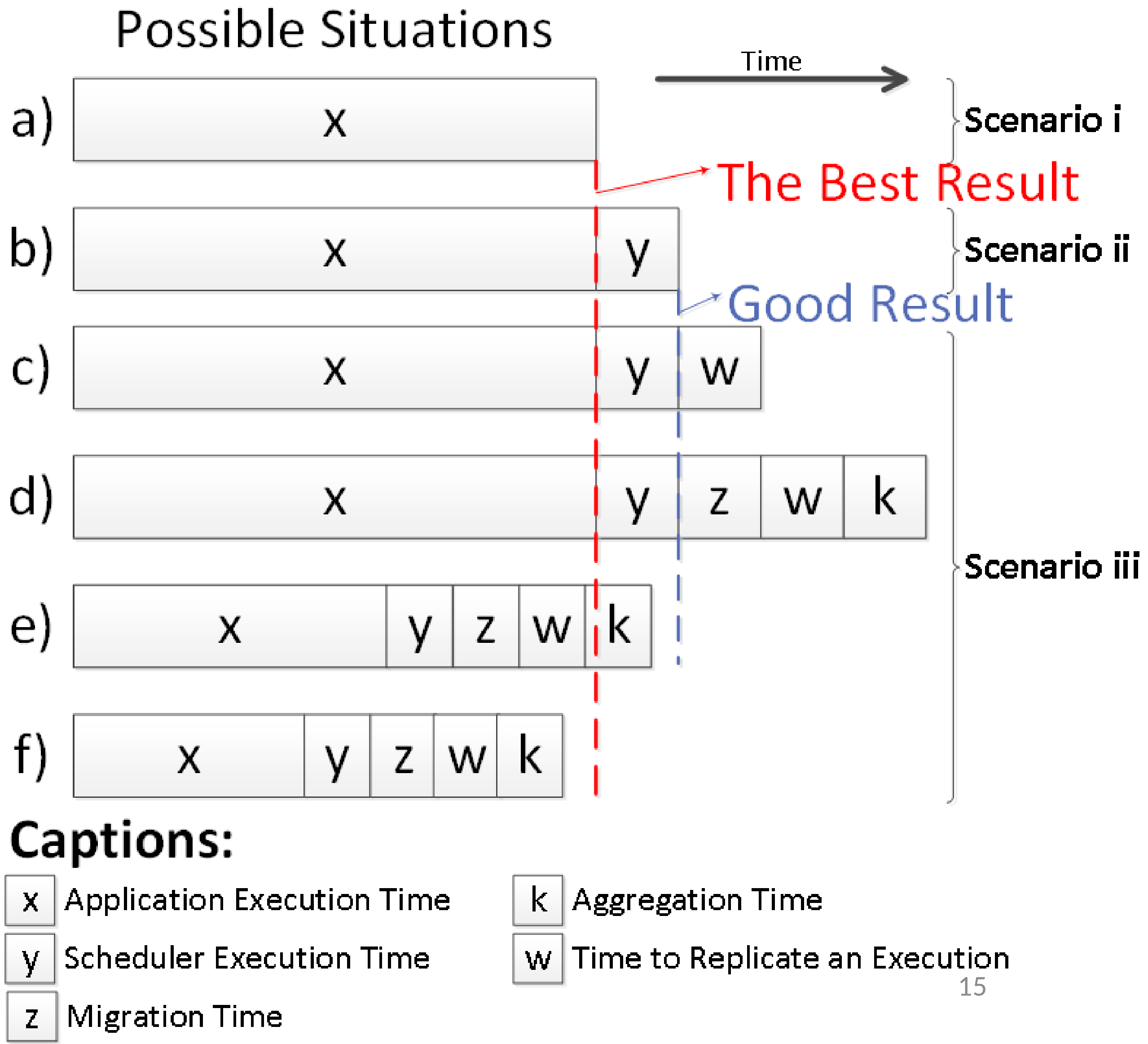
Dispatcher Strategies

- **Asynchronously create batch views** in the background
- Create the **stream views**, in which the latest logs (nodes and tasks) will be collected for the control
- **Combination of views** will achieve a performance gain due to the fact that most of the information required will already have been generated when it is needed
- The clustering data method and processing in **small batches** will be used to **obtain low latency** for the stream-based processing (Das et al. (2014))
- Through the batch-sizing of stream processing, it will reduce the latency, make it easier for the processing flow (i.e., by **processing simulations**) and facilitate the **scheduling and rescheduling of tasks and data**.

Dispatcher Strategies

- The differential approach is highlighted by the adoption of task simulation, availability of resources, network availability, costs, aggregation time and so on) with the strategies (i.e. batch and stream) to decide where to allocate the task to (node or Cloud).
  - ***Scenario i***: represents the simple execution, disabling all other services
  - ***Scenario ii***: adds the scheduling calculus
  - ***Scenario iii***: this scenario enables all the services

Situation f is **the best execution**, because beside have all services running the time is smaller then the situation a. Although situation e has a larger time when compared to situation a, it was computed



Dispatcher Strategies

- Best Result or a Good Result
  - Include the time to **movement data and the task migrations**
  - Time to data aggregate when the data is shared between
    a) the computing resources
    b) time estimated to execute a task
    c) data placement
    d) computing resource rating. In this way the overall execution time can be reduced and allow a simple fail control (volatile)

Dispatcher Strategies

- The reasons for applying this feature to the architecture are set out below:
  - **Migration**: Since there is a **dynamic**, the nodes might be in or out of the network (i.e., churn). This means that a certain task may be at an overload or a slow node, and there might be a machine that runs in less time (when the movements are counted). In this case, it is worth migrating the task
  - **Aggregation**: Distributing the tasks belongs to the network and forms a part of a set; thus it will be necessary to **group the results**. The distribution will be able to obtain time and make use of the idle resources
  - **Replication**: Replicating the task belongs to the network and means that it will be necessary to ensure the correct execution and reduce the time when a fault occurs. A fault generally occurs because of the **volatile environment (Desktop Grid)**
  - **Computing Resource Rating**: The rate will evaluate the resource data, network data and past execution data
  - **Time Estimation**: Knowing the time before executing a task will make it easier to ensure a correct scheduling

Dispatcher Strategies

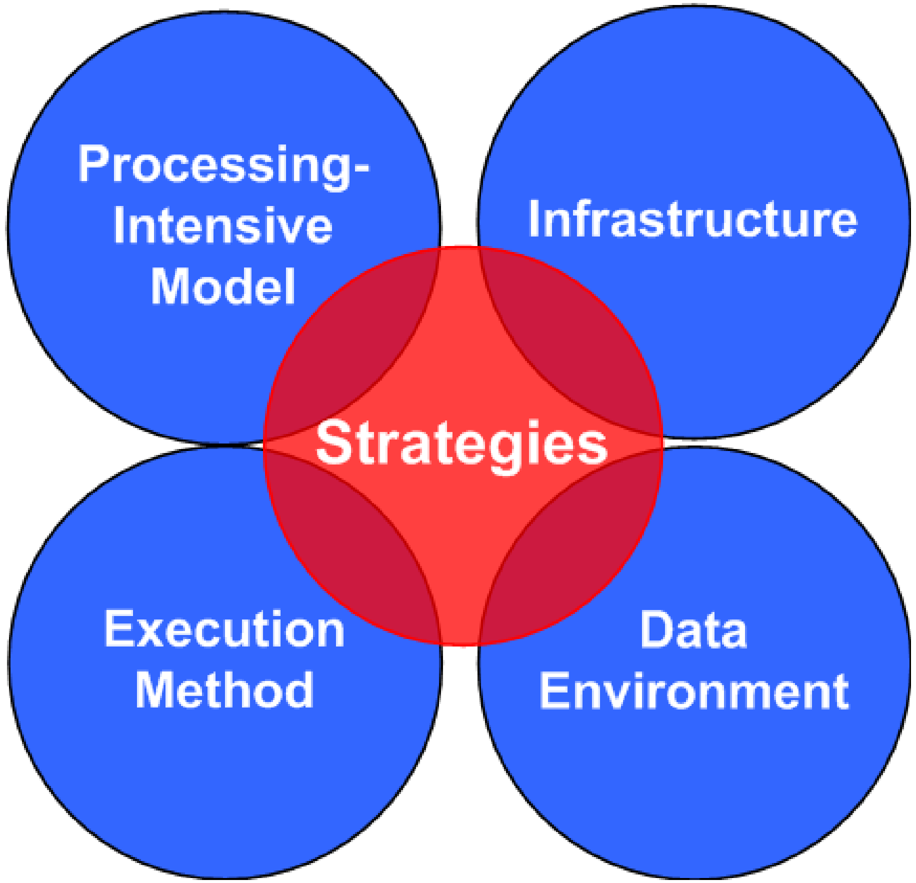
- The **combination of batch and stream strategies** will be able to reduce the Dispatcher time. The work of De Francisci Morales and Bifet (2015) provides some evidence that there is a significant reduction with the result of this merge. The batch-based method can reduce the decision making time and the management time and, in addition, can be used to the stream processing monitoring. However, all this must be in accordance with the Lambda Architecture paradigm.
- **BIGhybrid simulator** could be employed to estimate an execution time. The BIGhybrid will use the computational resources found in the environment, such as hardware performance, network performance, and tasks costs. A weight rating of computing will be defined through the simulation to dene some thresholds. The restriction will aid to control overhead levels of run time
- **Estimating an approximate execution time**, will make the scheduling and re-scheduling easier, because this makes it possible to know when a task will probably end at a particular computing resource before it starts

Outline

1. Introduction
2. Related Work
3. SMART Model
4. Dispatcher Strategies
5. Conclusion
6. Future Work

Conclusion

- Stream processing applied **to volatile and heterogeneous environments** is currently a significant subject for research
- The proposed solution will be applied at a **complex infrastructure (i.e., geographical distributed)** to study its issues and validate the model
- **The migration, scheduling** (MapReduce simulation and heuristics), and replication features will treat the problems of its volatility, heterogeneity and dynamic
- Through **the strategies to combine the desirable features**, the proposed model will provide a Good or The Best Result on the scheduling settlements



Outline

1. Introduction
2. Related Work
3. SMART Model
4. Dispatcher Strategies
5. Conclusion
6. Future Work

Future Work

- Study the impact of decouple environment in a stream-processing infrastructure
- Define dynamically the batched-size of the streams
- Control the flow of the environment
- Applied the Lambda Architecture on the SMART
- Evaluate the model

Strategies for Big Data Analytics through Lambda Architectures in Volatile Environments

Veith, Alexandre da Silva; **Anjos, Julio C. S. dos;** de Freitas, Edison Pignaton; Geyer, Claudio F.

Federal University of Rio Grande do Sul (UFRGS), Porto Alegre -RS

Lampoltshammer, Thomas J.

Danube University Krems, Department for E-Governance and Administration, Dr.-Karl-Dorrek-Str. 30, 3500 Krems, Austria

Contact: jcsanjos@inf.ufrgs.br

INTERNATIONAL FEDERATION OF AUTOMATIC CONTROL

4th IFAC Symposium on Telematics Applications.
November 6-9, 2016, UFRGS, Porto Alegre, RS, Brazil

Bibligraphy

- Alexandrov, A., Bergmann, R., Ewen, S., Freytag, J., Hueske, F., Heise, A., Kao, O., Leich, M., Leser, U., Markl, V., Naumann, F., Peters, M., Rheinl⊖ander, A., Sax, M.J., Schelter, S., H⊖oger, M., Tzoumas, K., and Warneke, D. (2014). The Stratosphere platform for big data analytics. VLBD Journal, 23(6), 939{964. doi: 10.1007/s00778-014-0357-y.
- Anjos, J.C.S., Assuncao, M.D., Bez, J., Carissimi, A., Costa, J.P.C.L., Freitag, F., Markl, V., Fergus, P., Pereira, R., de Freitas, E.P., Fedak, G., and Geyer, C.F.R. (2015). SMART: An Application Framework for Real Time Big Data Analysis on Heterogeneous Cloud Environments. In In proceedings of 15th IEEE International Conference on Computer and Information Technology (CIT-2015), Liverpool, England, UK, October 2015. IEEE Computer Society. Anjos, J.C.S., Fedak, G., and Geyer, C.F.R. (2016). BIGhybrid: a simulator for MapReduce applications in hybrid distributed infrastructures validated with the Grid5000 experimental platform. Concurrency and Computation: Practice and Experience, 28(8), 2416{2439. doi:10.1002/cpe.3665. Cpe.3665.
- Boykin, O., Ritchie, S., O'Connell, I., and Lin, J. (2014). Summingbird: A Framework for Integrating Batch and Online MapReduce Computations. Proc. VLDB Endow., 7(13), 1441{1451.
- Das, T., Zhong, Y., Stoica, I., and Shenker, S. (2014).Adaptive Stream Processing Using Dynamic Batch Sizing. In Proceedings of the ACM Symposium on Cloud Computing, SOCC '14, 16:1{16:13. ACM, New York, NY, USA. doi:10.1145/2670979.2670995.
- De Francisci Morales, G. and Bifet, A. (2015). SAMOA: Scalable Advanced Massive Online Analysis. J. Mach. Learn. Res., 16(1), 149{153.
- Delamare, S., Fedak, G., Kondo, D., and Lodygensky, O. (2012). SpeQuloS: A QoS Service for BoT Applications Using Best E
ort Distributed Computing Infrastructures. In Proceedings of the 21st International Symposium on High-Performance Parallel and Distributed Computing, HPDC '12, 173{186. ACM, New York, NY, USA. doi:10.1145/2287076.2287106.
- Eskandari, L., Huang, Z., and Eyers, D. (2016). PScheduler: Adaptive Hierarchical Scheduling in Apache Storm. In Proceedings of the Australasian Computer Science Week Multiconference, ACSW '16, 26:1{26:10. ACM, New York, NY, USA. doi:10.1145/2843043.2843056.
- Ewen, S., Schelter, S., Tzoumas, K., Warneke, D., and Markl, V. (2013). Iterative parallel data processing with stratosphere: an inside look. In Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22-27, 2013, 1053{1056. doi:10.1145/2463676.2463693.

Bibligraphy

- Hu, H., Wen, Y., Chua, T.S., and Li, X. (2014). Toward Scalable Systems for Big Data Analytics: A Technology Tutorial. IEEE Access, 2, 652{687. doi:10.1109/ ACCESS.2014.2332453.
- Liao, X., Gao, Z., Ji, W., and Wang, Y. (2015). An enforcement of real time scheduling in Spark Streaming. In IGSC, 1{6. IEEE.
- Marz, N. (2013). Big data : principles and best practices of scalable realtime data systems. O'Reilly Media, [S.l.].
- Peng, B., Hosseini, M., Hong, Z., Farivar, R., and Campbell, R.H. (2015). R-Storm: Resource-Aware Scheduling in Storm. In Proceedings of the 16th Annual Middleware Conference, Vancouver, BC, Canada, December 07 - 11, 2015, 149{161. doi:10.1145/2814576.2814808.
- Pham, L.M., El-Rheddane, A., Donsez, D., and de Palma, N. (2016). CIRUS: an elastic cloud-based framework for Ubilytics. Annals of Telecommunications, 71(3), 133{140.
- Righi, R.R., Veith, A., Rodrigues, V.F., Rostirolla, G., da Costa, C.A., Farias, K., and Alberti, A.M. (2015). Rescheduling and Checkpointing As Strategies to Run Synchronous Parallel Programs on P2P Desktop Grids. In Proceedings of the 30th Annual ACM Symposium on Applied Computing, SAC '15, 501{504. ACM, New York, NY, USA. doi:10.1145/2695664.2695979.
- Tang, B., He, H., and Fedak, G. (2015). HybridMR: a new approach for hybrid MapReduce combining desktop grid and cloud infrastructures. Concurrency and Computation: Practice and Experience, 27(16), 4140{4155. doi:10.1002/cpe.3515.
- Tudoran, R., Nano, O., Santos, I., Costan, A., Soncu, H., Bouge, L., and Antoniu, G. (2014). JetStream: Enabling High Performance Event Streaming Across Cloud Datacenters. In Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems, DEBS '14, 23{34. ACM, New York, NY, USA. doi:10.1145/2611286.2611298.
- Uckelmann, D., Harrison, M., and Michahelles, F. (2011). An architectural approach towards the future internet of things. In Architecting the internet of things, 1{24. Springer.
- Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M.J., Shenker, S., and Stoica, I. (2012). Resilient Distributed Datasets: A Faulttolerant Abstraction for In-memory Cluster Computing. In Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, NSDI'12, 2{2. USENIX Association, Berkeley, CA, USA.