Cloud Computing for Big Data Analytics in the Process Control Industry

E. Goldin[1], D. Feldman[1], G. Georgoulas[2], M. Castaño[2], and G. Nikolakopoulos[2]

Abstract— **The aim of this article is to present an example of a novel cloud computing infrastructure for big data analytics in the Process Control Industry. Latest innovations in the field of Process Analyzer Techniques (PAT), big data and wireless technologies have created a new environment in which almost all stages of the industrial process can be recorded and utilized, not only for safety, but also for real time optimization. Based on analysis of historical sensor data, machine learning based optimization models can be developed and deployed in real time closed control loops. However, still the local implementation of those systems requires a huge investment in hardware and software, as a direct result of the big data nature of sensors data being recorded continuously. The current technological advancements in cloud computing for big data processing, open new opportunities for the industry, while acting as an enabler for a significant reduction in costs, making the technology available to plants of all sizes. The main contribution of this article stems from the presentation for a fist time ever of a pilot cloud based architecture for the application of a data driven modeling and optimal control configuration for the field of Process Control. As it will be presented, these developments have been carried in close relationship with the process industry and pave a way for a generalized application of the cloud based approaches, towards the future of Industry 4.0.**

I. INTRODUCTION

For many years SCADA systems have been used to collect sensor data in order to control industrial processes, usually in real time [1]. The topological complexity of these systems (see [2]) involves large costs associated to scaling and adapting to the vast amount of signals gathered for allowing a general reconfiguration on the control structure for the process plant (see [3]). It should be also mentioned that the majority of these SCADA systems, up to now, have been utilized mainly for providing an overview of the controlled process, while having the ability to perform Process Analyzer Techniques (PAT) mainly for the statistical processing of the received data for an off line analysis.

However, the recent innovations in online PAT and wireless embedded technologies have created a new era in which almost all stages in the industrial process can be recorded, stored and analyzed. This process is producing a massive amount of sampled data that need to be stored and processed in real time for allowing an overall reconfiguration of the control plant and for achieving a continuous operational optimality against the variations of the production stages.

Towards this vision, the industrial processes require an IT infrastructure that could efficiently manage massive amounts of complex data structures collected form disparate data sources, while providing the necessary computational power and tools for analyzing these data in batch, near and hard real-time approaches. The overall problem becomes more complex, because of the diversity of acquired data mainly due to the: different data and sensors types, data reliability levels, measurement frequencies and missing data. Moreover in every case, the acquired data needs to be filtered, stored and often aggregated before any meaningful analysis can be performed.

With the explosion of the "*Internet of Things*" [4] in the last decade, a world of new technologies has become readily accessible and relevant for the industrial process. Nowadays, with relatively low costs, it is possible to send torrents of data to the "cloud" for storage and analysis. Cloud computing encompasses, cloud storage, and batch and streaming analysis of data using the latest Machine Learning (ML) algorithms. The potential benefits of using cloud computing for dynamic optimal control in the industrial plants include:

- Dramatically reduced costs of storing and analyzing large amounts of data
- Low levels of complexity relative to existing systems
- Enabling the use of advanced ML algorithms in batch and real time
- Reduces the industry entry level costs, for implementing advanced control systems
- Enabling large scale implementation with many low cost sensors
- Very easy to manage from the cloud
- Easy to scale or modify storage capacities

Inspired by these capabilities of the cloud infrastructure and the reachability of these technologies nowadays, the proposed architecture aims to combine the existing PAT based analysis of process that is carried in most of the times off line, or in a batch of time samples, with the multiple streams of sensory data describing the process and product states. The low-dimensional data should be robust against infrequent updates of PAT measurements and missing data, while handling largely varying measurement intervals. The model should also be able to handle the multivariate and auto correlated nature of process data and the high quantities of data from regular on line measurements. Principles from wireless sensor networks, estimation and statistical signal

The work has received funding from the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No.636834, DISIRE.

Accepted version of paper with the same title published in the 25th Mediterranean Conference on Control and Automation. Link to the published paper: http://ieeexplore.ieee.org/document/7984310

[1]GSTAT, Israel

[2] Robotic Team, Division of Signal and Systems, Electrical Engineering Department, Luleå University of Technology, Luleå, Sweden.

processing will be integrated and evaluated with real process data in order to create a novel and reliable PAT based swarm sensing and data analysis that would drive the changes in the Integrated Process Control (IPC) industry. Based on such an architecture it will be for the first time feasible to acquire and process online huge streams of data, improve the process models and correspondingly perform an online reconfiguration or re-tuning of the control scheme, in order to meet the changing demands of the process under investigation and apply platwide control techniques (see [5], [6]). Towards this vision, the corresponding architecture of the cloud computing for the big data analytics will be presented that forms the major contribution of this article. Furthermore, the proposed technological platform will be adjusted to the use case of a walking beam furnace.

The rest of this article is structured as it follows. In the Section II the architecture and components of cloud computing will be introduced, while in Section III a use case of a dynamic optimal design problem that can be implemented using the described architecture will be analyzed. Finally, Chapter IV will conclude the article by summarizing the benefits and limitations in using the described architecture in the industrial process.

II. Architecture for Cloud Computing

In batch computing, data is first stored in a Big Data Repository where it can be properly cleaned, aggregated or transformed before being analyzed by the process managers (see [7]). Often this includes saving the data in Parquet format that can reduce the size of the data up to 90% of its original size.

In the proposed prototype architecture for batch processing over the Cloud, users (industrial processes) were given access to an Amazon web portal for S3 storage services. All users were encouraged to contribute their raw batch data to the S3 repository. From the S3 storage service it is feasible to collect the data onto virtual computers (”instances”) implemented over the EC2 Amazon elastic computing framework, for data analysis and cleaning. On these virtual computers the Hadoop cluster [8] has been installed with a Spark engine [9] for computing and an RStudio Server [10] as an analytic access point for the end-users. Further access is also provided to the virtual computers via the RStudio Server IDE, through which they can perform ML algorithms and a vast array of statistical analysis on the data. The overall architecture of the proposed cloud architecture is presented in Figure 1.

In the architecture depicted in Figure 1, historical data collected from sensors embedded in the industrial process, are uploaded to the S3 storage on the Amazon Web Service (AWS). After the upload the data are cleaned and prepared for analysis on the big data framework. The process managers can access this data via local computers where they can send, develop and test their algorithms, including dynamic optimal control algorithms on the cloud of the monitored process.

Historical Data Repository - Users were given access to an Amazon S3 storage facility to which they were able to

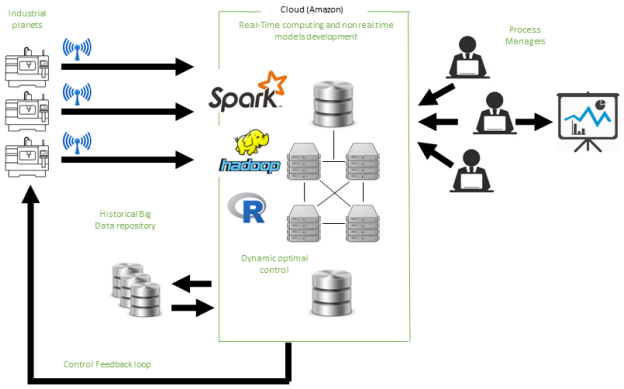

Fig. 1. Schematic Diagram of the Cloud Based Architecture

upload their historical/batch data in various formats (csv, Json, etc.). Amazon Simple Storage Service (S3) is a web storage interface that can facilitate storage of virtually unlimited data bucketed into 5 terabytes in size. Furthermore, the analytic architecture on the cloud is comprised of a ”big data” infrastructure, where the files are distributed over several machines for storage and parallel computing and a statistical software from which the data can be transformed and analyzed.

A. Cloud Storage

Amazon Web Services (AWS) offers a suite of over 70 services that form an on-demand computing platform. The two core services offered are:

1) *Amazon Elastic Compute Cloud (EC2)* - a virtual computer rental service through which users can run any software they desire and tailor the computer specifications to their specific needs. The payment scheme is per hour of actual usage - where computers can be ”stopped” and ”started” on demand.
2) *Amazon Simple Storage Service (S3)* - a web storage interface which can facilitate storage of virtually unlimited data bucketed into 5 terabytes in size.

In the presented architecture, the utilized Amazon on-demand platform allowed for higher flexibility in pricing and almost instantaneous setup of our prototype architecture. It also served as a platform where the different partners could easily upload and access their data for further analysis.

B. Hadoop Cluster (HDFS)

Apache Hadoop is the leading open-source software framework for distributed storage and processing of Big Data [8]. While *Hadoop* encompasses a suite of Apache software programs that help manage the tasks on the distributed system, the two core components of *Hadoop* are:

1) *Hadoop Distributed File System (HDFS)* - The system that takes very large data, breaks it down into separate pieces and distributes them to different nodes (servers) in a cluster.
2) *MapReduce* - The computational engine that can perform analysis on the cluster.

HDFS was designed to store Big Data with a very high reliability and flexibility to scale up by simply adding commodity servers.

In the presented prototype architecture it has been utilized *Hadoop* as a framework for setting up the *HDFS* cluster on which the sensor data are stored.

C. Apache Spark Engine

The main feature of *Apache Spark* is its in-memory cluster computing that increases of the processing speed much faster than the Hadoop's MapReduce technology. *Spark* uses *HDFS* for storage purpose, where calculations are performed in memory on each of the nodes. Aside from the increased speed in computation, the *Spark* engine is able to:

- Provide built-in APIs for multiple languages: Java, Scala, Python and R
- Spark-SQL for querying big data with SQL liked code
- Spark-MLlib [11] for big data parallel machine learning algorithms like linear and logistic regression, clustering K-means, decision trees, random forest, neural network, recommendation engine and more
- Spark-Streaming for calculating machine learning algorithms on streaming data

D. Process Managers

At the other end of the proposed architecture are the *process managers* who, through local computers, can access and perform machine learning algorithms on the data stored in the Hadoop cluster. The two leading programs that serve as an interface for conducting statistical analysis using the Spark engine are:

1) R - An open-source statistical language used widely both in the industry in academia.
2) Python - An open-source all around language which has a vast library of functions for implementing machine learning algorithms.

As mentioned above, both of these coding languages have APIs that pass commands to the Spark engine. The process managers access and run these programs through a number of web-based development environments and notebooks such as the *Jupyter notebook*, which is popular in the Python community and *RStudio*, which is the leading IDE amongst R users.

E. Control Feedback Loop

After the process managers have performed their analysis, they can set up dynamic models for implementation in the cloud that can push back responses to the industrial processes. This process is explained further in the *Near Real-Time Computing* subsection.

F. Historical Big Data Repository

In the cloud, the raw data and the process manager's recommendations will be stored at the historical big data repository (AWS S3). AWS offers great flexibility in storage plans that have the merit to be easily scaled as needed.

G. Near Real-time Computing

Apache Kafka [12] is a publish-subscribe messaging application that enables sending and receiving streaming information between the plants and the Spark engine on the cloud. On the local computers (in the plants) a Kafka API (which consists of a few Java libraries) sends streaming data to a Kafka Server set up on AWS that manages the queue of information passed on to the Spark engine. The Spark engine then performs the streaming analysis and pushes back the results to the Kafka server and from there back to the plants. The analysis can be either cleaning of the data, searching for outliers or implementing a ML algorithm in real-time. In addition, every 10 minutes the Spark server sends the accumulated data to the *Historical Big Data Repository* for future use or for *batch computing*.

H. Batch Computing

In batch computing, the data are initially stored in the *Historical Big Data Repository* where it can be properly cleaned, aggregated or transformed before being analyzed by the *process managers*. In many cases, this step includes saving the data in the *Parquet* format which can reduce the size of the data by using the R or Python languages. In general, the *process managers* can choose from a vast array of ML algorithms that can be implemented on the cluster through the *Spark engine*.

III. The use case of the walking beam furnace

The walking beam furnace is used to re-heat slabs (large steel beams) to a specific temperature before their refinement in the steel industry (see [13]). The slabs are walked from the feed to the output of the furnace by the cyclic movement of so-called walking beams. During this passage, the items are directly exposed to the heat produced by burners located inside the furnace. Since the heat distribution affects the quality of the finished product, a natural optimal control problem in this context is to regulate pre-assigned temperatures at specific points of the furnace, while minimizing the energy expenditure for the heat generation (see [14], [15]).

The walking beam furnace at MEFOS is an experimental furnace and lacks some of the features of an industrial furnace. Specifically, the temperatures throughout the furnace are not feedback controlled (as it is otherwise customary in the industry), i.e., the furnace operates open loop. Currently, a human operator configures the furnace set-points manually (the set-point values are, however, computed numerically) and then measures the slabs temperature at the furnace exit using a pyrometer. In fact, under normal operating conditions, the open-loop control can be tuned to work well. Additionally, this industrial installation is affected by stops and other variations that influence the control performance and correspondingly the need for a feedback control loop. In the described use case the main variables that need to be controlled are thus: a) the furnace temperatures in several zones of the furnace and b) the temperature of slabs at the output (the target temperature). Furthermore, the main objective is to reduce the operating costs through the reduction

of energy consumption. In this respect, a small decrease in energy consumption such as 0.5% translates into a saving of 2kWh per ton of heated product, while optimal control strategies could lead to quality improvements as well. The overall schematic diagram of the WBF with the indicative control loops, the sensors and the different heating zones is depicted in Figure 2.

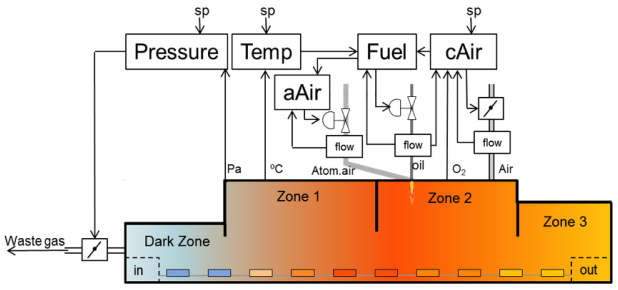

Fig. 2. Schematic Diagram of the Walking Beam Furnace

To achieve these goals there is a need to gather more information about the process on-line, while the optimal controls output would optimize the process by controlling the following variables: 1) the fuel supply rate at the burners, one burner at each zone, total of three burners, 2) the fuel atomization air supply rate, one for each burner, 3) the combustion air flow, one at each zone, total of three zones, and 4) the exhaust flow, e.g. exhaust damper position, one exhaust damper in the furnace.

In this use case, MEFOS has installed a dedicating PC in the WBF site for managing the flow of the measurements data. Figure 3 presents the flow of the sensory data from ABB control system to the connectivity server and from there to the corresponding PC and in the sequel to the cloud.

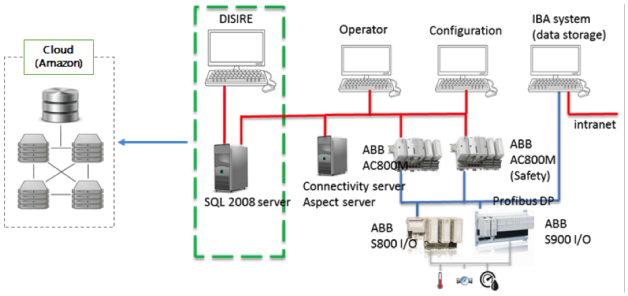

Fig. 3. Cloud Based Implemented Architecture of the WBF

In the presented use case it is intended to stream the data on-line, near real-time from the process by using the Kafka-producer component, to Kafka service in the cloud, while the Apache Kafka publishes-subscribes messaging applications. In the cloud the data will be pulled by the Kafka-consumer that will be implemented at the Spark cluster. At the cluster, the data will be verified, cleaned, aggregated, organized and sent to the optimal control system to determine recommendations. Afterwards the optimizer's recommendations will be pushed back to Kafka, while the corresponding gateway will determine the fuel supply rate at the burners, the fuel atomization air supply rate, the combustion air flow and the exhaust flow. In the cloud the raw data and the optimizers recommendations will be stored at historical big data repository (AWS S3). The overall schematic representation of the presented architecture is depicted in Figure 4.

For this usecase, the variables required from the optimal control module are the following ones in Figure 5:

The minimum data input for the optimal control is 200 past values of the averages every 10 seconds of the above parameters (one value every 10 seconds in the last 2,000 seconds, i.e. 33 minute and 20 seconds) is required.

A. Transferring data from the sensors to the cloud

For transferring data from the sensors to the cloud, a computer connected to the WBF process is utilized that is able to manage and update the site metadata, i.e. a MefosService method which run preliminary for synchronization of factory list, zone list, sensor list, bath list and model list. Furthermore, this method create a file in json structure with 3 fields: FactoryID, ZoneID, SensorID in Every possible values, while the posted data can be either a single message or array. The input messages are processed at the Kafka

TABLE I
MESSAGE TYPES

Message Type 1 - Process Status Change	
Factory ID	F Key [Predifined Integer]
Batch ID	F key [Predefined Integer]
Status ID	P key [running Integer]
Date time	[Time Stamp]
Current Status	[Predefined String:Idle/Start/Stop/Pause/Restart]
Message Type 2 - Measurements	
Factory ID	F key [Predefined Integer: -1 / 1 / 2 / 3 /]
Zone ID	F key [Predefined Integer]
Sensor ID	F key [Predefined Integer]
Batch ID	F key [Predefined Integer]
Date Time	[Time Stamp]
Measurement value	[Double]
Measurement unit	[Char: C/ % / m^3/h / kg/h / MMWC / Boolean]
Quality	[Integer]

server by using a specific topic that it is known by both sides as the MefosService and the MefosSpark, while it requires suitable configurations e.g "ToSpark". The Kafka API provides a callback method which verifies the input streaming received on Kafka server. The POST method "/ SendMeashurements" uses this API to evaluates any loss, if there is some.

B. The Cloud side

On the clouds side (AWS) there will be the Kafka server which will receive a streaming of data and will manage the queue. Overall the data will be routed through the Kafka server into the Spark cluster and from there back to Kafka. As mentioned before, the Kafka server will be held responsible for managing the messages that arrive from MefosService. The Spark streaming process consumes measurements data from the Kafka server, store it in the memory, and feeds the relevant process models at 10 sec. In every batch intervals the process receives the recommendations per measurement type from each model and sends

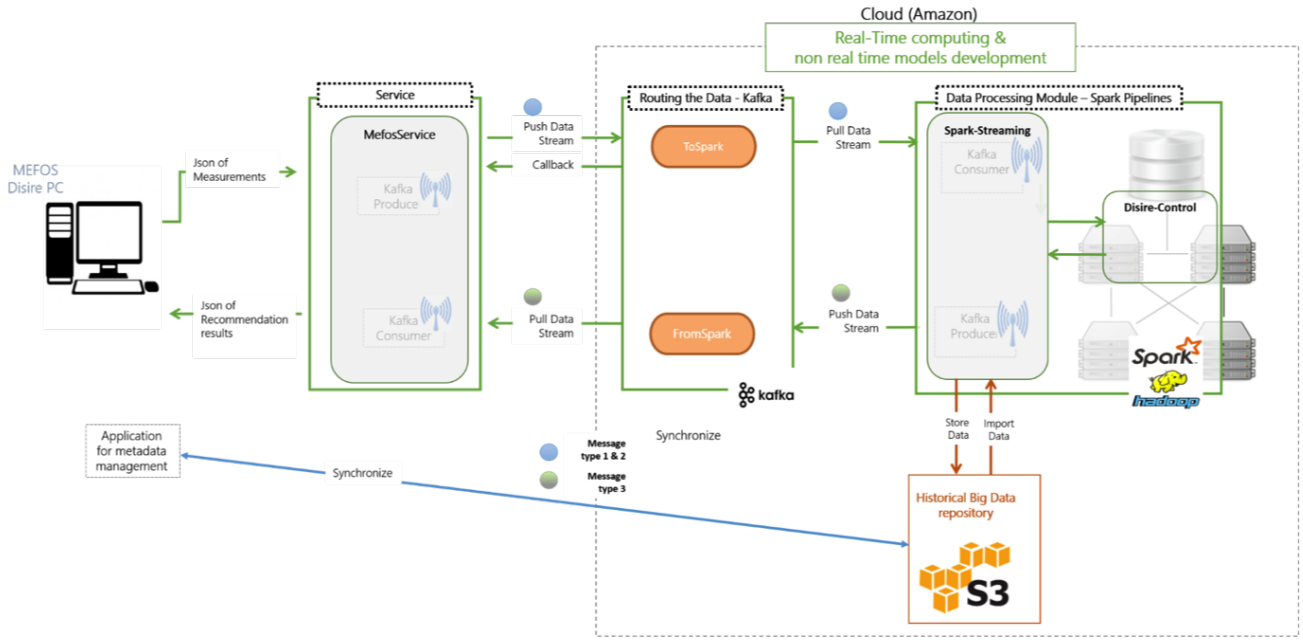

Fig. 4. Schematic description of the Architecture

Name	Tag name	Unit
Fuel	WBF_Z01_OilControl_FIC:HSI.MV	KG/H
Fuel	WBF_Z02_OilControl_FIC:HSI.MV	KG/H
Fuel	WBF_Z03_OilControl_FIC:HSI.MV	KG/H
Air-flow	WBF_Z01_AtomAir_FIC:HSI.MV	M^3/H
Air-flow	WBF_Z02_AtomAir_FIC:HSI.MV	M^3/H
Air-flow	WBF_Z02_AtomAir_FIC:HSI.MV	M^3/H
Zone temperature	WBF_Z01_ZoneTemp:HSI.MV	°C
Zone temperature	WBF_Z02_ZoneTemp:HSI.MV	°C
Zone temperature	WBF_Z03_ZoneTemp:HSI.MV	°C
Pressure	WBF__PC027:HSI.MV	MMWC
Air-flow	WBF_Z01_CombAir_FIC:HSI.MV	M^3/H
Air-flow	WBF_Z02_CombAir_FIC:HSI.MV	M^3/H
Air-flow	WBF_Z03_CombAir_FIC:HSI.MV	M^3/H
Oxygen	WBF_Z01_O2_QIC:HSI.MV	%
Oxygen	WBF_Z02_O2_QIC:HSI.MV	%
Oxygen	WBF_Z03_O2_QIC:HSI.MV	%
Air-flow	WBF_MainExhaust_ExhaustFlow_FIC:HSI.MV	M^3/H
status of the entrance door	SU_IML_GB5_SGU:HSI.Value	Boolean
status of the entrance door	SU_IML_GB6_SGN:HSI.Value	Boolean
status of the exit door	SU_UML_GB29_SGU:HSI.Value	Boolean
status of the exit door	SU_UML_GB30_SGN:HSI.Value	Boolean
recirculation of cold air	WBF_ColdCAir_OutputHSI:MV	M^3/H

Fig. 5. Variables required by the optimal control module

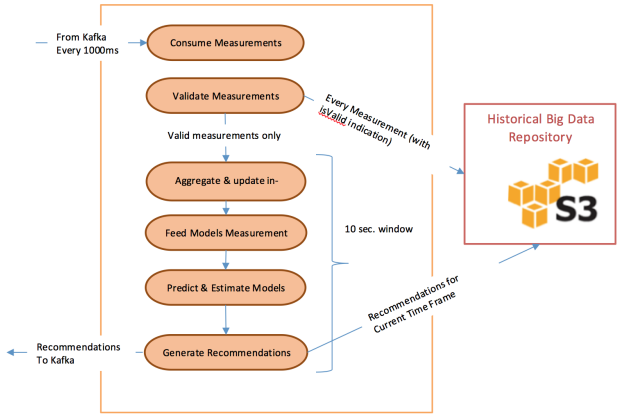

Fig. 6. Overview of the Streaming Process

the recommendations to the Kafka server. In the sequel, the Spark streaming process saves the measurements data along with the recommendations to AWS S3. Overall, the streaming process is depicted in Figure 6.

The Kafka server will also keep and be responsible for the recommendations data queue that it is arrived from the Spark cluster. For the transfering of the results from the cloud back to the process, the Kafka server keeps the controls recommendations data and streams them on a specific output topic to some consumer, while the ”MefosService” includes the Kafka-consumer feature that pulls the recommendations data from the output topic, e.g. ”FromSpark”. Finally, the output recommendations are reaching to the Web-API of the process by a provided URL.

For the big data repository, the Spark-Streaming process metadata are synchronized and pre-processed. After this step the data are being pushed from the Mefos-Service PC into the Kafka server and from there are pulled by the Spark cluster. At the Spark-Streaming, the initial data are accumulated in the memory and afterwards are saved at a historical Big Data repository. The Controls recommendations data are also accumulated at the memory and are saved at the historical Big Data repository that relies at the AWS S3 (Amazon Simple Storage Service), while the files will be saved as Parquet file type with the following benefits: 1) The structure of the table, i.e. the number of the columns, their types and the delimiter between columns, will be saved, 2) the data are compressed, a fact that saves about 60% of its volume compared to text file type, and 3) it enables

the straight upload into Spark in memory data storage, no conversions will be needed. Furthermore, the historical Big Data repository will enable deep investigation of the data in case it is required for the development of new models, such as the BI reports, etc.

IV. Conclusions

In this article an example of a novel cloud computing infrastructure for big data analytics in the Process Control Industry has been presented. The current technological advancements in cloud computing for big data processing, open new opportunities for the industry, while acting as an enabler for a significant reduction in costs, making the technology available to plants of all sizes. The main contribution of this article has been the presentation for the fist time ever of a pilot cloud based architecture for the application of a data driven modeling and optimal control configuration for the field of Process Control. These developments have been carried in close relationship with the process industry, since it has been presented a use case at the walking beam furnace of the Steel Industry MEFOS in Sweden. Part of the future work includes the full extended experimentation and validation of the proposed scheme in WBF campaigns.

References

[1] D. Bailey and E. Wright, *Practical SCADA for industry*. Newnes, 2003.

[2] O. Sporns and G. Tononi, "Classes of network connectivity and dynamics," in *Complexity*, vol. 7, pp. 28–38, 2001.

[3] M. van de Wal and B. de Jager, "Control structure design: a survey," in *Proceedings of the 1995 American Control Conference*, vol. 1, pp. 225–229 vol.1, Jun 1995.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[5] S. Skogestad, "Plantwide control: the search for the self-optimizing control structure," *Journal of Process Control*, vol. 10, pp. 487–507, October 2000.

[6] W. L. Luyben, B. D. Tyreus, and M. L. Luyben, *Plant-wide process control*. McGraw-Hill, 1998.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pp. 13–16, ACM, 2012.

[8] T. White, *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.

[9] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets.," *HotCloud*, vol. 10, no. 10-10, p. 95, 2010.

[10] J. Allaire, "Rstudio: Integrated development environment for r," *Boston, MA*, 2012.

[11] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, *et al.*, "Mllib: Machine learning in apache spark," *Journal of Machine Learning Research*, vol. 17, no. 34, pp. 1–7, 2016.

[12] N. Garg, *Apache Kafka*. Packt Publishing Ltd, 2013.

[13] H. S. Ko, J.-S. Kim, T.-W. Yoon, M. Lim, D. R. Yang, and I. S. Jun, "Modeling and predictive control of a reheating furnace," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 4, pp. 2725–2729, IEEE, 2000.

[14] B. Leden, "A control system for fuel optimization of reheating furnaces," *Scand. J. Metall.*, vol. 15, no. 1, pp. 16–24, 1986.

[15] J. Srisertpol, S. Tantrairatn, P. Tragrunwong, and V. Khomphis, "Estimation of the mathematical model of the reheating furnace walking hearth type in heating curve up process,"