BIG DATA MANAGEMENT TOOLS FOR SMART FARMING DATA

Michal Stočes, Pavel Šimek, Edita Šilerová, Jan Masner and Jan Jarolímek
Department of Information Technologies of Faculty of Economics and Management at Czech University of Life Sciences Prague, Czech Republic
stoces@pef.czu.cz, simek@pef.czu.cz, silerova@pef.czu.cz , masner@pef.czu.cz, jarolimek@pef.czu.cz

ABSTRACT

This paper deals with the issue of role of middleware in the process of data transformation, aggregation, storage and analysis. Data used in the concept of precision agriculture is very diverse. Not only in terms of sources but also formats. Farms use small sensor data as well as large files from satellite systems for various analyzes. Selected middleware services will be evaluated using a multiple-criteria decision analysis.

In agriculture, the collected data need to be continuously analyzed and worked with. Data processing procedures, especially with regard to Big Data, are not yet properly addressed in agriculture. As the volume of data collected grows, the demands for efficient storage grows as well. It is necessary to deal with this issue. The farmer's data sources can be divided between the data acquired by the farm from its own internal, private data source and data obtained externally. External data can be used from public open data databases or purchased.

On the basis of a multiple-criteria decision analysis of the weighted sum method, key criteria, selected weights for the use of a small start-up company, ThingsBoard was released as a compromise option with the highest value of 0.9. The second was DeviceHive with 0.833 and Mainflux, WSo2 IoT and Thinger.io.

Keywords: middleware, internet of things, big data, sensor data, data transformation, data aggregation, data store, data analytics

1. INTRODUCTION

The paper focuses on the issue of middleware as a tool for managing data from smart agriculture. The paper continues on previous authors research in the field of IoT, Big data and smart agriculture. (Stočes et al., 2018) and (Stočes et al., 2018).

1.1 Internet of Things and Middleware

Atzori et al. (2013) divided the Internet of Things paradigm into three groups in his work. The first group, such as Internet-based Vision, which includes web stuff, Internet 0 and IP (Internet Protocol address) for smart devices, is just connecting the device to the network. The second group is a semantically oriented vision that can be summed up as knowledge of working with information. This group includes data analysis and thus the middleware discussed in this work. The intersection of these two groups is then smart connected middleware. The last group is a device-oriented vision that includes the devices themselves. The intersection of all three groups is then the Internet of Things. The whole scheme is shown in Figure 1 below (Dastjerdi et al., 2016).

Copyright © 2019 for this paper by its authors.

Middleware is a layer or set of sub-layers of software between different levels of an application. The aim of middleware is to hide details of various technologies, protocols, network environments (Stočes et al., 2018), duplicate data, as a higher goal is to shield the programmer from problems that are not directly relevant to its scope and mission-specific development (Jarolímek et al., 2019). In addition, middleware masks the heterogeneity of computer architectures, operating systems, programming languages, and network technologies to facilitate application programming and management (Stočes et al., 2019). The reasons for using middleware in IoT are much from the security discussed in the later chapter, as well as its abstraction from the devices themselves (Kumarage et al., 2016). The middleware discussed in this work is not an abstraction of the devices themselves, their I / O systems, or the abstraction of networking (Cimino et al., 2019).

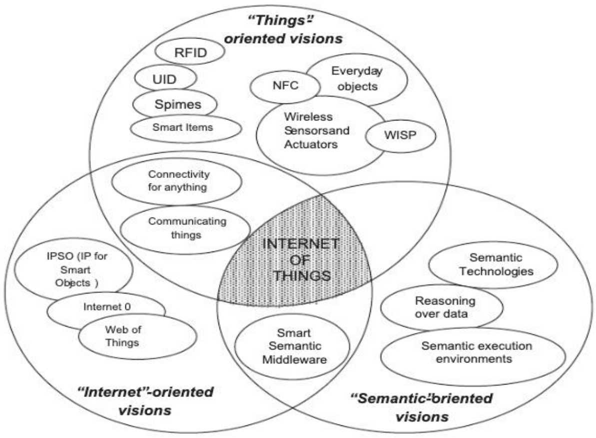

Figure 1. Internet of things (Atzori et al., 2010)

1.2. Middleware architectural designs

Middleware can be divided (Razzaque et al., 2016) into seven groups according to their design: event driven, service oriented, virtual machine driven, agent driven, tuple space paradigm, database oriented, and application specific. Some middleware is a combination of several groups that take advantage of selected groups. The three most used and basic ones from which others can be derived will be described. Event-based middleware, where middleware plays an active role in the data flow and sends the data itself to the application layer. Service-oriented middleware where it is passive and the data provides and exposes the API (Application Programming Interface). The last described pattern will be a virtual machine-controlled middleware in which the middleware is passive in terms of application layer and data only provides, but active against the physical layer in which it deploys virtual machines (Liangzhao et al., 2004.).

2. METHODOLOGY

2.1 Middleware evaluation

Middleware services were selected, individual solutions for specific use cases were evaluated. The main case is a small farm that wants to penetrate the smart farming industry and collect data from them (10 data user). They are interested in further developing this project and have the capacity of developers to develop their own solutions. Middleware services will be evaluated using a multiple-criteria decision analysis (MCDA) (Fiala et al., 1994), for which points will first be determined on a scale from 1 to 10, where 10 is the best value. Subsequently, according to the use case, suitable scales will be selected for literary research, the chosen use case as well as a number of experts dealing with the issue of the Internet of Things at the Czech University of Life Sciences Prague and the Farms dealing with smart agriculture. After the evaluation, a suitable compromise option will be chosen, or the development options based on the results for individual use cases will be described. Based on the survey, the following 8 evaluation criteria have been selected by the experts: Licensing, Documentation & Community, Development, User & Device Management, Format Support, Database, Data Distribution, Visualization (da Cruz et al., 2018).

2.1 Selected Middleware

The main selection criterion for the selection of middleware services was their availability and the preferred option was open source, the possibility of testing in a time limited trial version or limited demo functionality - the program was available for trial without credit card insertion. This excluded options like AWS - Amazon Web Services with its IoT Core, Microsoft Azure - IoT Hub and the ability to create its own middleware service using Elastic's ELK file - ElasticSearch for search, database and analysis, LogStash for logging and data collection and Kiban for visualization. (Šimek, et al., 2019) These solutions are very robust and are not needed for small business use. Middleware services were also selected according to current developments, the latest update could not be older than a few months and at the same time being developed by a company or group, not by an individual, this requirement will exclude a small, unsearchable middleware that will be many. These small projects can bring value, but in the case of using a small business that wants to continue its long-term development, and the product is actively used, the likelihood of closing a product that is covered by an individual is too high (Somani et al., 2019).

Five middleware services were selected for the final evaluation: Thinger.io, WSo2, IoT Mainflux DeviceHiv, ThingsBoard. All selected middleware are available online.

3.Results

For the weighted sum method, scores were assigned for each criterion of each middleware service (Table 1.) All points for the weighted sum method are listed in the table below (Table 2.).

Table 1. Weighted sum method I.part.

Criteria	Middleware				
	Thinger.io	WSo2 IoT	Mainflux	DeviceHive	ThingsBoard
Licensing	3	6	10	10	10
Documentation & Community	7	3	6	10	10
Development	6	3	6	8	10
User & Device Management	7	2	7	8	10
Format Support	3	10	8	8	3
Data Distribution	4	4	8	7	10
Visualization	5	7	8	8	9
Database	3	3	7	7	9

The next step after filling the weighted table is to calculate the total benefit of the variant. This is calculated by the scalar product of the values in the row with the values selected by the balance. The result is a table of total benefits below (Table 3.)

Table 2. Weighted sum method part II.

Criteria	Criterium weight	Middleware				
		Thinger.io	WSo2 IoT	Mainflux	DeviceHive	ThingsBoard
Licensing	**0.15**	0	0.43	1	1	1
Documentation & Community	**0.2**	0.57	0	0.43	1	1
Development	**0.2**	0.43	0	0.43	0.71	1
User & Device Management	**0.05**	0.63	0	0.63	0.75	1
Format Support	**0.15**	0	1	0.71	0.71	0
Data Distribution	**0.1**	0	0	0.67	0.5	1
Visualization	**0.15**	0	0.5	0.75	0.75	1
Database	**0.05**	0	0	0.67	0.67	1

Table 3. Result of multi-criteria analysis of variants.

Middleware	Thinger.io	WSo2 IoT	Mainflux	DeviceHive	ThingsBoard
Overall benefit of the variants	0.231	0.289	0.672	0.833	0.9
Order	5.	4.	3.	2.	1.

On the basis of a multiple-criteria decision analysis of the weighted sum method, key criteria, selected weights for the use of a small start-up company, ThingsBoard was released as a compromise option with the highest value of 0.9. The second was DeviceHive with 0.833 and Mainflux, WSo2 IoT and Thinger.io.

4. DISCUSSION

The multiple-criteria decision analysis is based on the best-rated compromise variant of the middleware service ThingsBoard. If a small start-up company is taken into account, this service is very interesting for it. Initially, the company does not have the money to buy a robust ready-made solution, and for this free middleware they would have the opportunity to free up their 10 users. The suitability of ThingsBoard would be enhanced by the user interface, which is much more accessible to non-technical users and offers suitable presentation materials for potential clients. The second proposed option could be to implement a custom solution that is being developed by a large multinational corporation with library support in 4 major languages. Both variants have great documentation with a detailed description of all key aspects, this fact supports better support on GitHub, on which ThingsBoard is actively responding. ThingsBoard would be particularly useful when there is a need for rapid prototyping of a solution, both to demonstrate an idea, a feasibility test, or to validate a concept. Fast prototyping is enabled by the graphical scripting framework Rule Engine.

In the case of recommendations for medium to large farms, a different situation needs to be analyzed. Large companies always prefer to develop a customized system tailored to their specific requirements, a second DeviceHive compromise option would be appropriate for this example. By covering the programming languages with their libraries - Python, Java, JavaScript, Go - they cover most of the common market in finding experts.

Other options had a greater gap in the multiple-criteria decision analysis than the first two variants among themselves, which was due to the prevailing lack of evaluation in the core criteria, Documentation and Community and Development. The last two variants of WSo2 IoT and Thinger.io had poorly processed visualization and limited connectivity to external databases.

Comparing the results of multiple-criteria decision analysis with other sources and articles is not possible because it cannot be found through a common search for similar comparisons. Online are to find the introduction of individual middleware services, but not their comparison. There were lots of resources on the Internet that compare middleware services in terms of their architectures and implementations.

5. Conclusions

The aim of this work was to evaluate middleware services used for aggregation, transformation and distribution of data obtained from smart agriculture sensor networks. In the first part of paper, the key requirements for middleware services resulting from the literary research were analyzed. Each chosen criterion was presented, described on the basis of what knowledge the method and procedure of the evaluation follows and defined. The extent to which middleware has been addressed as a topic of work has been precisely defined and the case of using a small start-up company for which criteria and their weights have been described. The documentation and method of developing a particular middleware service were collectively the most important criteria as key parameters for middleware solution development.

In the next chapter, the middleware services were selected according to the basic criteria: available online, open source, it is possible to try them in a time limited trial version or demo without the need to insert a credit card. Middleware services also had to be actively supported by developers and the latest update could not be older than a few months. Based on these initial requirements, 5 services were selected - Thinger.io with UI development, WSo2 IoT with multi-format support and UI development environment, Mainflux custom solution implementation service, DeviceHive with detailed documentation, and ThingsBoard middleware that offered custom UI scripting via Rule Engine framework. The services were presented and further described, their functionality, architecture and other characteristics according to the selected criteria.

In the last part, selected middleware services were evaluated using multi-criteria analysis and suitable compromise variants applied to individual cases of use were evaluated. The results are as follows ThingsBoard with the highest value of 0.9. The second was DeviceHive with 0.833 and Mainflux, WSo2 IoT and Thinger.io. For the case of using a small start-up company, ThingsBoard appeared to be a suitable compromise option, which had the most benefit in DeviceCritical analysis and DeviceHive. ThingsBoard is suitable for rapid prototyping with the UI Rule Engine framework, which can be implemented with its own implementation for this robust tool, offering asset management, user and group management as an Asset Management feature. DeviceHive is suitable for both a small farm and larger companies that want to work on their own solution with the support of an open source middleware service that is being developed by a multinational company. DeviceHive has its own library in multiple languages - Python, JavaScript, Java and Go. These results and findings can be followed by a real project implementation, which will collect data from the Internet of Things devices and help in selecting the appropriate middleware service.

Acknowledgments

The results and knowledge included herein have been obtained owing to support from the following institutional grants. Internal grant agency of the Faculty of Economics and Management, Czech University of Life Sciences in Prague, grant no. 2019B0009 – "Life Sciences 4.0".

References

Atzori, L., Iera, A. and Morabito, G. (2010). 'The Internet of Things: A Survey'. Computer networks, 54(15), pp. 2787-2805. doi: 10.1016/j.comnet.2010.05.010

Cimino, L.D., de Resende, J.E.E., Silva, L.H.M. et al. (2019) 'A middleware solution for integrating and exploring IoT and HPC capabilities', Software: Practice and Experience, 49(4), pp. doi: 10.1002/spe.2630

Dastjerdi, A. V. and Buyya.,R. (2016). 'Fog Computing: Helping the Internet of Things Realize Its Potential'. Computer, 49(8), pp. 112-116. doi: 10.1109/MC.2016.245

da Cruz, M. A. A., Rodrigues, J. J. P. C., Al-Muhtadi,J., Korotaev,V. V. and de Albuquerque, V. H. C. (2018) 'A Reference Model for Internet of Things Middleware' IEEE Internet of Things Journal 5(2), pp.871-883. doi:10.1109/JIOT.2018.2796561

Fiala, P., Jablonský, J. and Maňas, M. (1994) 'Vícekriteriální rozhodování' Vysoká škola ekonomická v Praze, ISBN 80-7079-748-7

Jarolímek, J., Pavlík, J., Kholova, J. and Ronanki, S. (2019) 'Data Pre-processing for Agricultural Simulations', AGRIS on-line Papers in Economics and Informatics, 11(1), pp. 49-53. doi: 10.7160/aol.2019.110105.

Kumarage, H., Khalil, I., Alabdulatif, A., Tari, Z. and Yi, X. (2016) 'Secure Data Analytics for Cloud-Integrated Internet of Things Applications' IEEE Cloud Computing, 3(2), pp.46-56. doi:10.1109/MCC.2016.30.

Liangzhao Zeng, B. Benatallah, Ngu, A. H. H., Dumas, M., Kalagnanam, J. and Chang., H. (2004) 'QoS-aware middleware for Web services composition' IEEE Transactions on Software Engineering 30(5), pp. 311-327. doi: 10.1109/TSE.2004.11

Razzaque, M. A., Milojevic-Jevric, M., Palade, A., and Clarke, S. (2016) 'Middleware for Internet of Things: A Survey' IEEE Internet of Things Journal, 3(1), pp. 70-95. doi: 10.1109/JIOT.2015.2498900

Šimek P., Vaněk J., Stočes M., Jarolímek J. and Pavlík J. (2017) 'Mobile accessibility expense analysis of the agrarian WWW portal' Agricultural Economics, 63, pp. 197-203. doi: 10.17221/313/2015-AGRICECON

Somani, G., Zhao, X. Srirama, S. N.and Buyya, R. (2019) 'Integration of Cloud, Internet of Things, and Big Data Analytics' Software: Practice and Experience, 49(4), pp 561-564. doi: 10.1002/spe.2664.

Stočes, M., Masner, J., Jarolímek, J., Šimek, P. (2018) 'Internet of Things and Big Data Processing in Agriculture.' In AFITA/WCCA 2018 Proceedings - Research Frontiers in Precision Agriculture 24.10.2018, Powai, Mumbai, pp. 49-51.

Stočes, M., Šilerová, E., Vaněk, J., Jarolimek, J. and Šimek, P. (2018) 'Possibilities of using open data in sugar & ugar beet sector' Listy cukrovarnické a řepařské, 134(3), pp. 117-121. ISSN: 1210-3306.

Stočes, M. – Masner, J. – Kánsaká, E. – Jarolímek, J. (2018) 'Processing og Big Data in Internet of Things and Precision Agriculture'.International Scientific Conference Agrarian Perspectives XXVII. Food Safety – Food Security. pp. 353-358. ISSN 2464-4781