



HAL
open science

TYPHON: Hybrid Data Lakes for Real-Time Big Data Analytics – An Evaluation Framework in the Telecom Industry

Antonios Misargopoulos, George Papavassiliou, Christos A. Gizelis, Filippos Nikolopoulos-Gkamatsis

► **To cite this version:**

Antonios Misargopoulos, George Papavassiliou, Christos A. Gizelis, Filippos Nikolopoulos-Gkamatsis. TYPHON: Hybrid Data Lakes for Real-Time Big Data Analytics – An Evaluation Framework in the Telecom Industry. 17th IFIP International Conference on Artificial Intelligence Applications and Innovations (AIAI), Jun 2021, Hersonissos, Crete, Greece. pp.128-137, 10.1007/978-3-030-79157-5_12 . hal-03788999

HAL Id: hal-03788999

<https://inria.hal.science/hal-03788999>

Submitted on 27 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

TYPHON: Hybrid Data Lakes for Real-time Big Data Analytics – An Evaluation Framework in the Telecom Industry

Antonis Misargopoulos¹[0000–0003–3019–0207], George Papavassiliou²[0000–0003–0138–9671], C.A.Gizelis¹[0000–0001–5127–3891], and F.Nikolopoulos-Gkamatsis¹[0000–0003–2858–9227]

¹ Hellenic Telecommunications Organization S.A. (OTE), Marousi Attiki, Greece, amisargopo@ote.gr, cgkizelis@ote.gr, fnikolop@ote.gr

² Avlos LLC, Cambridge MA 02139, USA, george@avlos.ai

Abstract. In this paper we aim to present outcomes from the evaluation and deployment of the TYPHON H2020 project¹ in a container-based production ecosystem². The main objective is to implement a successful production-ready deployment following best practices in terms of infrastructure, monitoring, and high availability while achieving analysis in real-time of massive streams of data. Moreover, to evaluate the TYPHON architecture when integrated by a Telecom Industry user and present the benefits of its adaptation by IT Units for predictive maintenance purposes.

Keywords: Hybrid Persistence Architectures · Big Data Analytics · Evaluation Framework and Measures · Adaptation to Telecom Industry

1 Introduction

The Hellenic Telecommunications Organization S.A. (OTE) [1] as the largest telecommunications provider in Southeast Europe provides its customers with a wide variety of ICT services addressing a large customer base. OTE owns repositories of big data sets from a variety of sources such as call centers, billing, mobility, traffic data, etc., and concentrates huge amounts of both operational by means its infrastructure as well as consumer-related data through its customers in daily basis. In collaboration with Avlos [2], a technology company providing custom and innovative technology solutions to enterprises worked together in order to evaluate the TYPHON [3] driven approach and methodology towards its adaptation by Telecom industries.

¹ The TYPHON project receives funding under the European Union’s Horizon 2020 Research and Innovation Programme under grant agreement No. 780251.

² Supported by Hellenic Telecommunications Organization S.A. (OTE).

2 Related Work & Current Status

Predictive maintenance is the art of anticipation of equipment failures to allow for in advance scheduling of corrective actions, thereby preventing unexpected downtime and improving the customer service quality. The main goal of predictive maintenance is to enable proactive scheduling of corrective work and thus, prevent unanticipated failures. Predictive maintenance requires insights into the running conditions of equipment and software. In a corporate environment, there is a diversity of computing systems and software, which are utilized by employees given their diversity of tasks.

Nowadays, the use of logs to estimate system level-failures with a high-confidence poses many challenges. As logs are mainly used for debugging purposes, the following challenges emerge:

- they rarely contain explicit information on failure prediction unless structured properly
- they contain heterogeneous data including symbolic sequences, numeric time-series, categorical variables, unstructured text; and
- they produce massive amounts of data, posing computational challenges.

To make use of log data, we first had to interpret them, filter out a large amount of noise (i.e. data irrelevant to our goal), and extract predictive features. Next, we had to collect known failure cases for learning/evaluating models, transform the problem into an appropriate learning scenario and determine performance indicators that reflect real-world needs. The effort and cost of modern database architectures assisting the above process are often discouraging, given that the volume, variety, veracity, and complexity of the logs, lead to heterogeneous data sets that are very hard to even be queried.

In this context the TYPHON project offers a concrete methodology to interact with a number of relational and non-relational databases and perform modeling activities on both open and closed data sets. Using TYPHON capabilities OTE is given the opportunity to overcome the following limitations:

- handle large volumes of data that come from different database sources
- analyze large amounts of diversely structured data
- run analytics (i.e. statistical algorithms) measures on massive amounts of data in near real-time to extract insights.

For OTE, the TYPHON architecture offers an environment to interact with a diversity of data sources, minimizing the effort and time required to analyze such complex data sets. The following sections present how TYPHON capabilities can be exploited as well as the evaluation framework implemented to assess the level of adaption efficiency with respect to the challenges emerge in a Telecom business environment like OTE.

3 TYPHON & OTS Solutions at OTE

TYPHON platform offers solid framework for managing big volume of data stored in heterogeneous persistent back-ends to calculate advanced KPIs in near to real-time basis; i.e.,:

- Technologies for designing hybrid *polystores* [4] are agnostic of the structure of the data, the availability, partitioning, and consistency requirements of different subsets of the data, and the available deployment resources
- Novel algorithms offer hybrid polystore design models transformation into pre-configured optimized virtual machines which can be deployed on cloud infrastructure
- An extensible high-level language for querying and modifying data stored in hybrid polystores is available, as well as capabilities for translating high-level queries into efficient native queries
- A high-performance framework for publishing and processing data access and updating events to facilitate real-time monitoring and predictive analytics
- Technologies for evolving the organization and distribution of data in hybrid polystores, along with tools for monitoring use of polystores for enhanced evolution

Besides the interest of the academic community, an open source innovative solution like TYPHON could inspire companies/organizations of diverse large-scale industry areas to gain from. The adoption though, by an industry environment where business need is fulfilled by an off-the-self (OTS) solution/product is out of discussion a great challenge. In particular, at OTE a problem like this can be described as follows:

- The Dynatrace monitoring tool [5], at the moment running in-house, analyses streams of logs from CRM services and evaluates in real-time the user experience of its CRM services. The software utilized, requires a data store and ETL pipelines to be set in place; and its cost as well as deployment burdens the Operations department.
- A time-consuming process is followed to amend data on already stored records.
- At the moment the user experience of CRM services can hardly be predicted. Data is spread across data stores without any point of communication; moreover, performing any statistical analyses techniques on top of this multitude and diverse set of data stores is hard.

In the next section it is described the experience in deploying and utilising each of the TYPHON components within the context of OTE, the degree to which OTE requirements have been satisfied, the level of efficiency on how TYPHON technologies can cover all business related measures, and finally what benefits TYPHON as solution can bring to OTE in terms of service quality, time or cost.

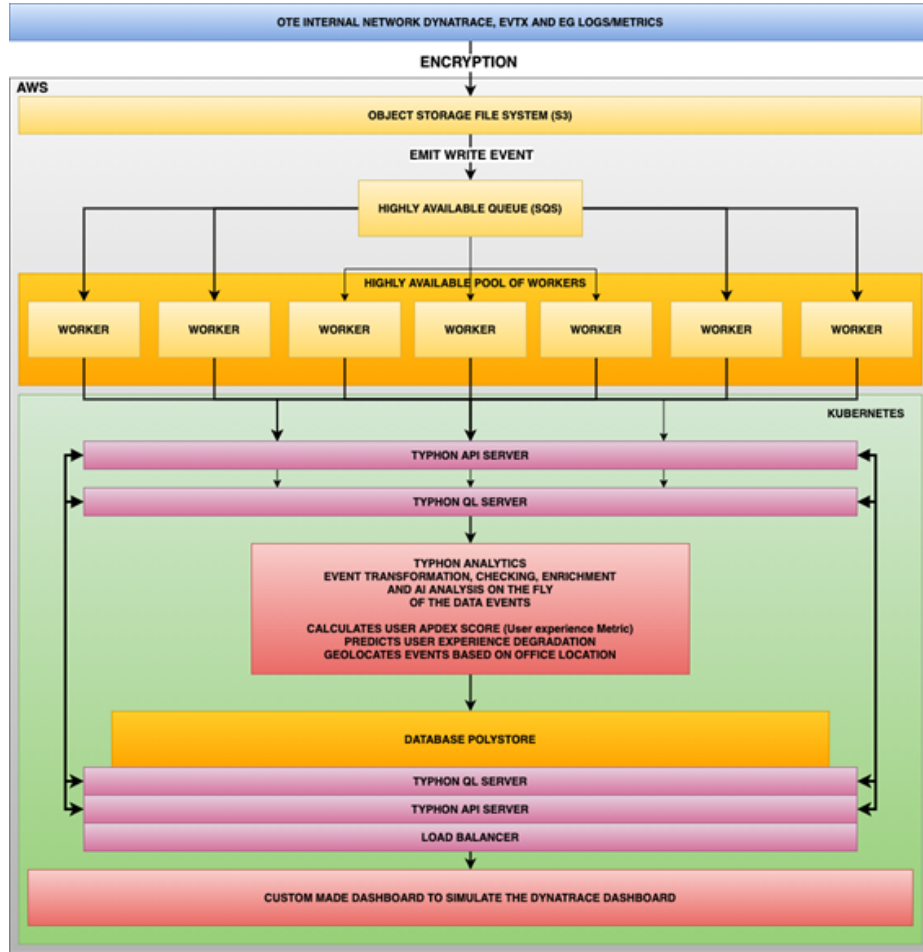


Fig. 1. Our E2E Infrastructure Architecture

4 Our Evaluation Framework

In this section we present our approach for the assessment of TYPHON solution within OTE organization. I.e., the end-to-end (E2E) infrastructure architecture we deployed to run the evaluation, and the use cases and measurement scenarios themselves.

4.1 Deployment Architecture

Fig.1 depicts the E2E infrastructure architecture we deployed that is based on following steps:

1. Data ingestion into Object Storage (AWS S3) [6] after Security Encryption Pass; i.e., anonymization³ procedure
2. Once the encrypted file is written, an event is emitted and the file write-event is put into a queue (AWS SQS) [7]
3. A pool of workers reads the file, parses it line by line, and then adds each line as an event back in the queue (AWS SQS)
4. The same pool of workers reads the event lines and starts posting them to the TYPHON API for ingestion
5. Each event log line passes through TYPHON Analytics component [8] to be authorized, amended with data and then stored

Queries on the data are performed through external tools.

4.2 Use Cases

To validate the functionality of the TYPHON platform, the following two use cases have been identified and examined thoroughly within the context of OTE participation as an end user and evaluator at TYPHON project:

Descriptive Reporting OTE IT Department injected Customer Relationship Management (CRM) session information with virtual machine environment details. More specifically, it has enriched the user’s session data source with extra data fields such as the server name, server farm name, the user’s real IP address as well as other relevant fields that will help in the analysis. It is thus possible, for example, to run statistical analyses and classifications of which subsets of users are connected via the virtualized infrastructure or not; and what the consequences of such connections are with respect to user’s experience. Exploiting the TYPHON Analytics Component capabilities, OTE used a unified framework to inject the data running an Apache Flink [9] pipeline rather than ad-hoc python scripts that intercepts connection on the infrastructure machines and later store it in the TYPHON Polystore.

Predictive Maintenance for improved Quality of Service (QoS) In this use case as a first step data is gathered and enriched in massive amounts within the TYPHON polystore, and as a second step large queries run across all underlying databases where the data is modeled to generate enriched data that will serve as the basis of a prediction model capable of forecasting a service degradation based on faults and errors counts in the back-end infrastructure that supports the CRM system. This type of prediction modeling and forecasting had lead to proactive, and even automated, measures to be taken by the IT Department for maintenance tasks and will, as a result, improve the quality of the provided service.

³ A type of information sanitization for privacy protection by removing personally identifiable information from a data set.

4.3 Scenarios

User Experience Apdex Scoring and threshold determination Application Performance Index (Apdex⁴) [10] is an open standard for measuring performance of software applications in computing. It converts measurements into insights about user satisfaction, by specifying a uniform way to analyze and report on the degree to which measured performance meets user expectations. The Apdex method converts many measurements into one number on a uniform scale of 0 to 1 (0 = no users satisfied, 1 = all users satisfied). The resulting Apdex score is a numerical measure of user satisfaction with the performance of enterprise applications. The Apdex formula is the number of satisfied samples plus half of the tolerating samples plus none of the frustrated samples, divided by all the samples:

$$\text{Apdex}_t = \frac{\text{SatisfiedCnt} + (\text{ToleratingCnt} * 0.5) + (\text{FrustratedCnt} * 0)}{\text{TotalSamplesCnt}}, \quad (1)$$

where the sub-script t is the target time, and the tolerable time is assumed to be 4 times the target time.

The Dynatrace system at OTE calculates Apdex scoring to determine whether a user of the application is satisfied with their experience. We used the Analytics component of the TYPHON to produce the same or similar metric. Note that scoring for the user experience of our application is highly influenced by the underlying hardware they connect with. Using Apache Flink to process buckets of data in real time and determining the Apdex for a user session within a certain time window, we derive a similar Apdex scoring pattern as that which the Dynatrace platform currently provides for the internal operations team.

Importing Java Statistics functions into Apache Flink for calculating probabilities and quantiles for the gamma distribution [11, 12], which defines an Erlang distribution [13, 14] when the shape value is positive, OTE used TYPHON itself to identify the appropriate thresholds to establish whether the user experience is “satisfying”, “tolerable”, “intolerable”. In the end we should have an Apdex metric timeseries that represents the user experience per connected machine cluster. By doing this, we can infer the discrepancies in response time and session duration between users that connect directly to the application and those that are connected through a virtualized server middle ware cluster. The reporting performed inside the TYPHON Platform will be compared and evaluated with respect to the one reporting, which is currently performed outside TYPHON.

Data Preparation and Annotation as a Platform Using the TYPHON Platform’s Analytics and its APIs we are able to amend and filter data appropriately, similar to the data injection that is happening currently by the Internal Operations Unit. This will happen instantly as the data comes into TYPHON and with the appropriate tables in the Data Model used, metadata will be added

⁴ <https://www.apdex.org/>

to the current data streams entering the databases. In detail, a table is going to be created in the TYPHON polystore with a bijection of IP addresses to retail shop locations which will be used by the Analytics Component. The Apache Flink consumer will analyse the IP address of the incoming data point and then query this internal table. In case that the IP exists in the table, it will resolve and amend the geolocation of the retail shop in the data point; otherwise the record will be rejected.

Using TYPHON as a Data lake to extract massive amounts of data for Machine Learning (ML) training By directly accessing data components through the TYPHON Query Language component (TyphonQL) API [15], we evaluated the querying of massive amounts of data to use as the training set of a ML Prediction Model. In this scenario the ultimate objective was to conclude how TYPHON can assist in the training of ML Models. To do this, the TyphonQL and TYPHON Analytics components would be utilized to hook various ML frameworks into the platform and perform the training. Thus, the platform succeeded with respect to its performance when tasked with running oversized data queries as well as being able to process these queries in a paginated manner.

5 Evaluation Measurements Insights

In this section it is presented the evaluation KPIs and the full prototype test measurements for the TYPHON capabilities within the scope of OTE as Use Case of Telecom industry.

Support of Highly Available Data Stores TYPHON uses technologies underneath that have been battle tested in production for years, i.e., MongoDB [16], MariaDB [17] etc. It has thus managed to expose their highly available/clustered versions for a fail safe deployment of stateful services (i.e. databases) in a cloud native environment (Kubernetes [18]).

Support for a wide Variety of Data Stores Given the current support for a variety of data stores (MariaDB, MongoDB, Cassandra [19], ArangoDB [20]) the platform provides us with the versatility not only to use currently existing databases but also to extend the data sources to others not previously thought of or considered.

Accuracy of Apdex Scoring The TYPHON Analytics component is based on a streaming database solution (Kafka [21]) which is also highly available and has been run in production systems for years by a multitude of organizations. Using the TYPHON Analytics Component we were able to reproduce the mathematical calculations that extract meaningful statistical data from TYPHON with 100% accuracy, in fact we can implement any mathematical function operating within a

time window of data. Dynatrace has its own proprietary algorithm for calculating Apdex and as expected, our results are not identical since we used one of the open formulas of the Apdex equations. We essentially managed to prove though that we are able to develop any mathematical/statistical function in the TYPHON Analytics API to reproduce any statistically significant result or measure we are interested in.

Cost of Maintenance We have observed that running TYPHON on Kubernetes gives a significant cost reduction on the operations and maintenance of the platform. Not only we were able to switch between cloud providers or on-premise deployments, but we had also observed a significant cost reduction running the platform in this cloud native manner compared to other systems. This is a direct result of its open source nature, containerized and cloud native deployments and the cost of ownership.

Deprecation of Currently Running Systems Assuming that TYPHON can be used both as a real time analytics solution as well as a data lake (polystore) system, the platform is able to replace a number of systems that are running internally. For example, it could replace the variety of proprietary and costly stores used to gather multidimensional data; i.e. aggregators and analyzers like Dynatrace and Vertica [22], as well as internally provided bulky Data lake systems like Hadoop.

6 New Capabilities Adopting TYPHON Technologies

6.1 Benefits for Users/Customers

The aforementioned measures provide a very promising scenario for using TYPHON in a diversity of areas within the organization to optimize the analysis of data with our own customer base as well as internal operations. For example, using TYPHON platform OTE can gather information from a variety of sources, generate insights and act on maintenance of machines within our stores proactively, even automatically, to replace parts, machines, software, before they malfunction.

There are areas though, that we have not yet explored; i.e., using TYPHON as a data lake of numerous stores running underneath it, we should be able to connect data sources that we had not yet thought of and improve the User Experience (UX) of our customers in our digital stores or generate customer behavior analysis and optimisation metrics. The experience and technical knowledge gained from an H2020 project like TYPHON could be used in our industrial environment for exploiting Big Data Analytics or ML capabilities, i.e., alerting notification based on KPI threshold violations or proactive warning upon potential service degradation based on past measures.

6.2 Benefits for OTE

For internal operations, the realm of possibilities widens even more. Most significantly in the areas of cost of ownership, operational resilience and staff productivity.

Cost of Ownership Based on the measures extracted above the cost of ownership on an open source solution like TYPHON significantly decreases in various contexts. Given the amount of different solutions already running internally that it can replace, by acting both as a real time analytics solution as well as a hybrid data lake without the need to change the underlying data sources, we have found a cost reduction in significant scale. Additionally, operating the solution on Kubernetes provides significant cost and agility benefits. Not only there is no need by Operation Engineers to retrain on a different platform as they are already familiar with the Kubernetes framework, but the deployment platform itself also enables us to migrate quickly from one cloud solution to another; it can even be moved on premises if its deployed in the cloud and vice versa.

Operational Resilience The significant power that the deployment of TYPHON on a cloud native solution like Kubernetes provides is that of resilience. That is, the highly available nature of a platform like Kubernetes allows the TYPHON platform to provide fail safe guarantees for a seamless deployment and run-time configuration as well as blue-green deploy scenarios. By using the highly available versions of the underlying data stores we can be assured of significant uptimes because:

- restoring and roll-back at any stage when a failure happens at run time or during deployments remains seamless, quick and safe
- the clustered version of the databases and their replication allow for a node to fail without data loss.

Staff Productivity Since the deployment and operational overhead is reduced dramatically, data engineers' productivity increases significantly with respect to the business case at hand. Instead of worrying about the operations and the performance of their queries since the deployment is taken care by a scheduler like Kubernetes, the data engineers are now able to focus solely on analysing the business problem at hand and mining the data needed with ease to derive the appropriate insights that will finally help them resolve any issue or incident comes to front.

7 Conclusion

Summarizing the evaluation parameters presented in the previous section, it is now clear that TYPHON architecture and components follow an open source-like

methodology which is the demand of the IT Unit within a Telecom Organization. Such an industry owns, handles and processes an extremely large diversity of data stored in various and heterogeneous databases and systems. Open source solutions generally speaking follow a more cost-effective approach than a proprietary one. Moreover, it is favored for industry players, since it utilizes open standards in order to reduce the cost of ownership of systems or delay system obsolescence.

TYPHON Polyglot system managed to meet and fully satisfy the requirements of OTE with respect to the ability to handle a wide variety of data stores with high availability. This is crucial for a company with a large size of several hundreds of retail stores and millions of customers/subscribers. In this context the need for a solution that will contribute to time efficiency and cost reduction will always be of high priority. The flexibility that TYPHON architecture offers in terms of easiness in integration as well as the wide diversity of supported heterogeneous persistence back-ends optimize resource allocation and improve staff productivity to handle human-expert tasks more efficiently.

References

1. OTE Group of Companies, Research & Innovation, https://www.cosmote.gr/cs/otegroup/en/ereuna_kai_kainotomia.html/
2. Avlos.ai, <https://avlos.ai/>
3. TYPHON Project, Polyglot and Hybrid Persistence Architectures for Big Data Analytics, <https://www.typhon-project.org/>
4. Jérme Fink, Maxime Gobert, Anthony Cleve: Adapting Queries to Database Schema Changes in Hybrid Polystores. In Proceedings of IEEE International Workshop on Source Code Analysis and Manipulation (2020)
5. The Dynatrace software intelligence platform, <https://www.dynatrace.com/platform/>
6. Amazon Simple Storage Service S3, <https://docs.aws.amazon.com/s3/index.html/>
7. Amazon Simple Queue Service SQS, <https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/welcome.html>
8. Athanasios Zolotas, Konstantinos Barmpis, Fady Medhat, Patrick Neubauer, Dimitris Kolovos, and Richard F. Paige: An Architecture for the Development of Distributed Analytics based on Polystore Events, Poly'20 Workshop Proceedings of the 46th International Conference on Very Large Databases VLDB Endowment Archive, Vol. 14, No. 1, (2020)
9. Apache Flink, <https://flink.apache.org/>
10. Apdex (Application Performance Index), <https://en.wikipedia.org/wiki/Apdex>
11. Gamma distribution, <https://mathworld.wolfram.com/GammaDistribution.html>
12. Gamma distribution, https://en.wikipedia.org/wiki/Gamma_distribution
13. Erlang distribution, <https://mathworld.wolfram.com/ErlangDistribution.html>
14. Erlang distribution, https://en.wikipedia.org/wiki/Erlang_distribution
15. TyphonQL Compilers and Interpreters, Centrum Wiskunde & Informatica (CWI) and SWAT.engineering, TYPHON Project D4.5 Final deliverable, Public distribution, <https://4d97e142-6f1b-4bbd-9bbb-577958797a89.filesusr.com/ugd/d3bb5c-4349c798cf42409d93a4d588ab1f4162.pdf>
16. MongoDB, A source-available cross-platform document-oriented database, <https://www.mongodb.com/>

17. MariaDB Relational Database, <https://mariadb.org/>
18. Kubernetes K8s, An open-source system for automating deployment, scaling, and management of containerized applications, <https://kubernetes.io/>
19. Apache Cassandra, An open-source, distributed, wide-column store, NoSQL database management system <https://cassandra.apache.org/>
20. ArangoDB, An open-source multi-model database for graph and beyond, <https://www.arangodb.com/>
21. Apache Kafka Connector, <https://ci.apache.org/projects/flink/flink-docs-stable/dev/connectors/kafka.html>
22. Vertica, A unified advanced analytics database, <https://www.vertica.com/v9/>