Targeted Advertisement case-study: a LeanBigData benchmark

Jorge Teixeira[1], Miguel Biscaia[1], Iván Brondino[2,3], Mario Moreira[1]

[1] Altice Labs, Aveiro, Portugal
{jorge-teixeira, miguel-b-pias, moreira}@alticelabs.com
[2] Universidad Politécnica de Madrid, Madrid, Spain
[3] LeanXcale, Madrid, Spain
ivan.brondino@leanxcale.com

Abstract. In this paper we present Targeted Advertisement case-study, a big data case-study for LeanBigData project[1]. PT (Portugal Telecom) sells multi-platform ads online covering the whole spectrum of web, mobile and TV, allowing advertisers to define their own campaigns, set their campaign goals and budget, their choice of paid words, as well as many other constraints including geographic and demographic of the targeted customers. To reliably provide efficient, contextualised and targeted advertisements to final users, the current architecture of PT AdServer relies on in-house developed tools for handling the high-throughput stream of data and to deal with analysis and visualisation. We present a benchmark study performed on LeanBigData platform tested with real PT needs in terms of data throughput and scalability.

Keywords: Big data, OLTP, OLAP, SQL, Databases, Advertisement, LeanBigData

1 Introduction

The Big Data hype is said to present huge opportunities for Communication Service Providers (CSP). Most trends – in technology development, consumer behaviour, regulation and investments – seem to corroborate that Big Data is already playing an non-neglectable role in revenue streams. Big Data *per se* has little value, but data processing applied to societal, business or organizational challenges can open a whole new business stream to CSPs: from business predictive analysis and advanced business intelligence to e-health and context aware marketing, among many others. Nevertheless, the current moment is still an early stage in the Big Data hype: companies are still experimenting innovative Big Data based solutions and testing market acceptance. It is estimated that in the near year of 2017, "60% of big data projects will fail to go beyond piloting and experimentation and will be abandoned" [7].

According to a study held by The Economist Intelligence Unit [8], the top two key data challenges in businesses are the quality, reliability or comprehensiveness of data and the lack of effective systems to gather and analyse data.

The same study results show that one of the top three data insights critical to decision-making is "qualitative" (e.g. customer experience). Targeting the best content (advertisement) for each client is thus critical for improved customer experience. As stated by Matthew Keylock [8], "If you don't engage with your best customers, they won't want to engage with you". And this is where LeanBigData project [1] comes into play.

PT sells multi-platform ads online covering the whole spectrum of web, mobile and TV, as depicted in Figure 1. Similar to other industry standards, such as Google AdSense and AdWords, it allows advertisers to define their own campaigns, set their campaign goals and budget, their choice of paid words, as well as many other constraints including geographic and demographic of the targeted customers. Recent trends point to a maximization of convergence synergies between all the distribution channels by allowing profiling to happen across the whole spectrum of data so that ads are served in a much more targeted fashion.

Fig. 1. Ads being served on three different platforms: web, mobile and TV

Decisions on which ads to show in which client need to be made in a fraction of a second and should be informed by all the batch processing associated with profiling. To cope with these large streams of data, PT currently uses a hodgepodge of big data technologies, leading to high communication overheads and undesired operational complexity. The major goals of this case study in the project are thus the following: (i) simplification of the operational complexity by sorting out the current heterogeneous mixture of technologies; (ii) improvement of the overall efficiency of the advertisement system and bid data infrastructure; (iii) improvement of the throughput capacity in at least on order of magnitude and (iv) improvement of cross-domain analytics, made possible through the convergence of the various data streams.

2 Use-cases Description

To reliably provide efficient, contextualised and targeted advertisements to final users, both web services and application users, the current architecture of

AdServer relies on in-house developed tools for handling the high-throughput stream of data and to deal with analysis and visualisation. The AdServer includes of a hodgepodge of technologies, including Hadoop for batch processing of aggregation and behaviour analytics, MonetDB and PostgreSQL for ad-hoc analysis and various small custom solutions, Esper as open source CEP engine, Storm as a scalable stream processing engine and, with a highly negative impact on the entire system, custom code to glue all of these technology blocks together. Data goes through a distributed message queue broker architecture relying on a producer/consumer paradigm and Hadoop and SOLR clusters are used to store and query the relevant consumed data. A Service Delivery Broker hosts all of the internal web services as well as third party web services with a Marketplace for subscribing such services.

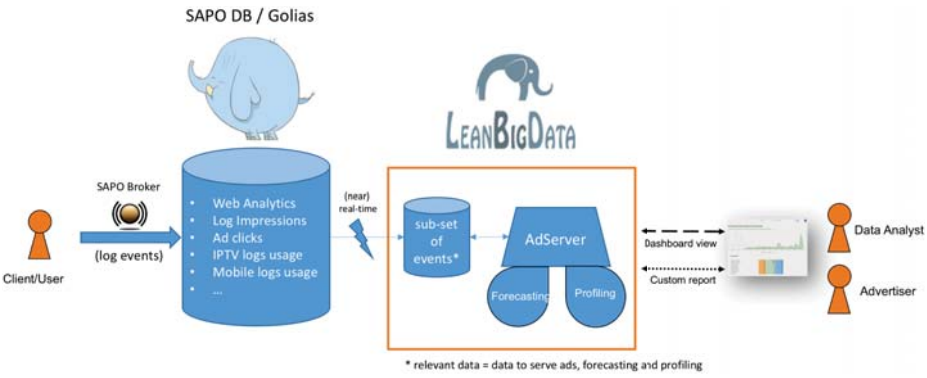

Fig. 2. AdServer architecture with the LBD platform

A high-level overview of the integration of LeanBigData platform with the current infrastructure is depicted in Figure 2. This integration will allow a simplification of the mixture of technologies actually being used, as will allow significant improvements in both the throughput of data analysis as well as in the forecasting and profiling capabilities of the system. From the database perspective, the goal will be to take advantage of several improvements at the SQL-engine level [4][3] as well as the new scalable fault-tolerant transactional platform specifically designed for OLTP workloads [5] to enhance the global throughput of the system. Additionally, recent work on a new approach for Key-Value Data Stores [2] will significantly improve OLAP workloads.

Based on the AdServer architecture, on the LBD platform capabilities and on PT advertisement needs, five different use cases were designed: Ad Serving (UC1), Forecasting (UC2), Dashboards (UC3), User Profiling (UC4) and Campaign Setup (UC5). These use-cases will be described in the following subsections.

2.1 Use-cases Requirements

Current SAPO infrastructure processes about 90 million requests per day, in the scope of multi-platform advertisement. Such requests can go up to 150 million requests per day in peak days, representing over 3600 requests per second. Additionally, 90% of these requests need to be served in less than 30 mili-seconds. In now-a-days market, demands are exponentially increasing and both technical and infrastructures setups, as well as user needs have to evolve accordingly. In this sense, targeting advertisement embraces such challenge by enabling the possibility of improving PT current big data infrastructure, and, more important from the business perspective, it introduces new and enhanced mechanisms of serving more contextualized and better advertisement to PT users.

2.2 UC1: Ad Servering

Ad Serving use-case is focused on serving targeted and contextualised ads to both clients and ad server actors. A client uses any of PT's multiplatform services, whether mobile, web or TV, and receives a targeted contextualised ad which takes into consideration the user profile and specific campaign requirements by the advertiser together with general inferred clusters/models from previous usage of the platform. The expected effect of this UC is the update of the platform with logging information on the ad served and the client request.

Due to high risk management reasons, the AdServer platform was, at the time of writing, emulated in a dedicated cluster of virtual machines, with the support of LeanBigData framework at an SQL boundary, and with a usage scenario as close as possible to the live system.

We collected a sample of one week of web usage, including user-agents used to access the webpages, geographical distribution of the users and hostnames accessed. Using this large sample of real data, together with artificially generated data, we were able to emulate the AdServer environment. Specifically, real-data included (i) approximately one thousand different user-agents combinations, including information about the device, the OS, its version, etc.; (ii) more than two thousand different geographic origins (detailed to city) of web requests and (iii) more than thirty thousand different visited hostnames. Based on this information, the AdServer dataset was designed with three major data inputs: banners' ads, users' profiles and impressions.

- **Banners' Ads**: represent a form of advertising, typically on the web, but also on different platforms such as mobile applications and IPTV, such as depicted in Figure 1. Ads comprehend information ranging from banner tags and hostname to target devices and user locations. The number of banners vary according to the benchmarks, as detailed in section 3.
- **Users' profiles**: comprehend information regarding users' gender and age, mainly corresponding to authenticated users, and does represents scarce and not very rich information. For the sake of this use-case, users' profiles data was randomly generated, and profiles are evenly distributed across age and gender.

– **Impressions**: a single impression, in the context of online advertising, is when an ad is fetched from its source, and is countable. Whether or not the ad is clicked is not taken into account. Impressions are generated according to each user and banner request, and include information such as timestamps (date and time of the request), banner metadata and, if available, user profile.

AdServer emulation : The workload for the AdServer emulation consisted of 95% of impressions, 3% of dashboard queries and 2% of Forecasting. These indicators were extracted from the workload of the live AdServer, and they indicate that 95% of the requests to the AdServer are related to impressions: as expected, most of the AdServer effort concerns with fetching banners based on user requests from multi-platforms. Most typical queries are "inserts" with logging information from each impressiom. Second, dashboard queries embrace a significant 3% of the its workload. Such queries are used for batch and real-time dashboards, allowing technicians, advertisers, data scientists and business analyst to visualize, explore and customize ads behaviours and campaigns. These are typical OLAP queries with reading and aggregation operators, which complexity largely depends on the users needs. Finally, 2% of the workload is targeted to forecasting. This information, typically integrated into dashboards, inform business analyst and advertisers of the ongoing and predict the success of ad campaigns. Similar to dashboard queries, these one are also represented by OLAP queries.

2.3 Other use-cases

PT has designed and is currently working on four additional use-cases, although no final results are yet available to be presented.

UC2: Forecasting: Given the history of a campaign including all the associated specific metadata (e.g. data about the users and kind of user profiles who have consumed the ads), the goal is to compute the expected behaviour of a campaign in time in terms of prints and user clicks. The forecast will be useful for Advertisers to review their campaign specifics, Sales people to predict revenues and Ad Analysts to possibly revise their ad strategy. The forecasting module will be implemented with direct support on LeanBigData CEP component [6].

UC3: Custom Reports and Dashboards: Any set of features/values should be prone to visualisation via insightful dashboards. Sales Persons, Ad Analysts and Data Analysts will be able to create and save their own dashboards based on ad-hoc queries which are executed on the fly from available data streams. Dashboards are currently being implemented by connecting LeanBigData framework to Kibana[4] using the JBDC provided drivers.

UC4: User Profiling: Users profile use case is based on the profiling component, a module built over the Lean Big Data platform, which clusters users into characteristic profiles according to their behaviour and past history with the platform.

[4] Kibana is an "elastic" product for data exploration and visualization, available at `https://www.elastic.co/products/kibana`

UC5: Campaign Setup: Campaign Setup is a simple use case consisting of an Advertiser inputting campaign properties via a friendly user interface and providing that information to the Ad Server. Campaigns typically have delivery restrictions based on user profile (age and gender, for example) and request features such as User Agent (e.g.: device types) and IP address (for geo-reference).

3 Benchmarks

The benchmarks goal is to test LeanBigData platform performance agains current PT AdServer infrastructure. As previously mentioned, due to high risk management reasons, benchmarks can not be performed on PT production environment. Moreover, three different scenarios need to be taken into account for the benchmarks:

- **Production AdServer**: Live system currently being used at PT to serve ads according to the multi-platform scenario needs.
- **Emulated AdServer**: An emulated version of the Production AdServer in a dedicated (non-production) cluster, but with similar workloads and hardware specification.
- **LeanBigData AdServer**: A testing version of PT AdServer use-case in LeanBigData platform, also with similar hardware specification and workloads regarding the Production AdServer.

Workloads applied on the Emulated AdServer and on the LeanBigData AdServer were previously detailed in sub-section 2.2 and aim to guarantee similar test conditions for the benchmarks. Also, hardware specifications are equally equivalent for both scenarios. While for the emulated AdServer we have 2 nodes, each with 2x CPUs E5-2670 (32 virtual CPUs / threads), 132GB RAM, 2x 800GB HDD (10k rpm) and 2x 10Gbps network, for the LeanBigData AdServer we have each node with a CPU E5-2630 (24 virtual CPUs / threads), 128 GB RAM, 0.5 TB SSD + 4 TB HDD and 1Gbps network.

The conditions for the benchmarks are based on four different parameters. The number of users, set to 200.000, represents the total number of different users that may request an ad, an aspect that is taken into account regarding the number and complexity of user profiles. The number of banners, set to 100.000, represents the possible number of banners each request can deliver. The number of interactions (100.000) concerns with the total number of impressions generated during the tests, with each impression being represented by an interaction with the AdServer. Finally the number of connections, varying from 50 to 250, represents the number of concurrent requests to the AdServer.

3.1 Emulated AdServer *versus* Production AdServer

The first benchmark aimed at comparing the Emulated AdServer with the Production AdServer, by measuring both the response time and throughput of these

setups. For the first test (refer to Figure 3, left side), the average response time vary – according to the number of concurrent clients – from 12.6ms to 19.4ms, with a minimum of 5.4ms, which is bellow the average response time for the Production AdServer, 13ms.

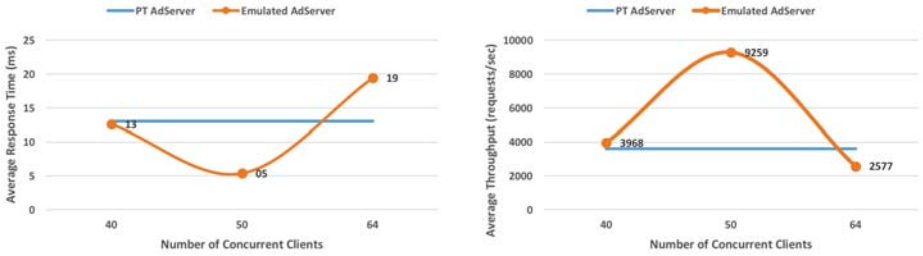

Fig. 3. Emulated versus Production

Regarding the throughput test (refer to Figure 3, right side), the throughput vary from 3668 *requests/sec* to 2577 *requests/sec*, with a maximum of 9259 *requests/sec*, almost three times more that the average throughput at Production AdServer. There results indicate that the Emulated AdServer achieved similar results compared against the Production AdServer, and it thus suitable for further benchmarks with LeanBigData platform.

From these results we assume that the Emulated AdServer is a suitable representation and replacement of the Production AdServer, in the scope of these benchmarks. Following benchmarks are thus performed against Emulated Adserver.

3.2 Emulated AdServer *versus* LeanBigData AdServer

The second benchmark focus on comparing the Emulated AdServer against the LeanBigData AdServer. This benchmark is supported on two distinct metrics: the throughput and response time. The throughput results achieved (refer to Figure 4, left side) show that LeanBigData platform performed better that the Emulated AdServer from PT, with a different of approximately 1500 *requests/sec*. This improvement is considerably interesting, since it represents almost 50% of the average throughput of events at Production AdServer (3600 *requests/sec*).

Regarding the results on the response time metric, presented on the right side of Figure 4, these indicate that LeanBigData is still above the average response time for PT Emulated Adserver, with 5.4ms. Nevertheless, ongoing work in LeanBigData project [1] is expected to bring significant improvements in this matter, in particular with the integration of the new key-value data store[2] and other improvements in the SQL engine[4][3].

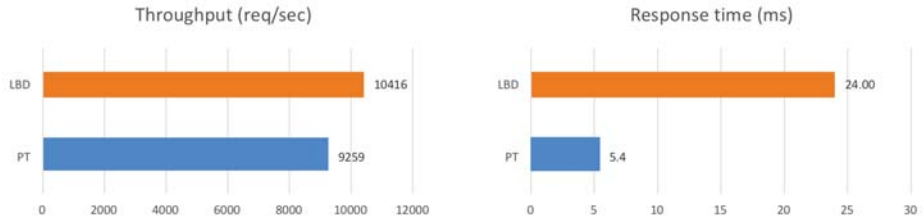

Fig. 4. Emulated versus LeanBigData

3.3 LeanBigData AdServer scalability

The last benchmark goal is to test the scalability of LeanBigData platform, by progressively increasing the number of nodes from 1 to 5. The metrics for this benchmark are the average response time and the average throughput. As depicted in Figure 5 (left side), the average response time stabilizes when adding 2 more nodes to the AdServer configuration, indicating that this result is not affected by the increasing number of nodes and the potential overhead for management.

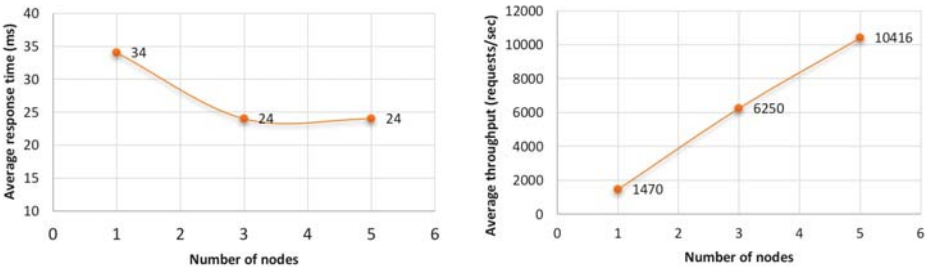

Fig. 5. LeanBigData scalability

Moreover, when testing the LeanBigData AdServer and measuring its throughput, results presented in Figure 5 (right side) indicate that the system is capable of linearly increase its overall throughput capability with the increase of nodes in the AdServer configuration. Both tests are strong indicators that LeanBigData platform can in fact horizontally scale (by increasing the number of nodes) without affecting its performance.

4 Conclusions and Future Work

We presented Targeted Advertisement case-study in the scope of LeanBigData project[1]. Five different use-cases were designed, aiming to cover all Portugal

Telecom needs and expectations regarding a fully targeted and contextualized advertisement platform. Benchmarks were performed on LeanBigData platform for the first use-case (Ad Serving), supported on an emulated scenario. Results have shown that current LeanBigData platform outstands PT in terms of throughput, although response time using LeanBigData platform in still bellow PT results. Nevertheless, LeanBigData roadmap includes several improvements in the key-value data store and SQL engine which are expected to positively impact on these results. Concerning scalability tests, results have shown that LeanBigData was able to linearly scale from 1 to 5 nodes, based on the throughput responses achieved.

Future work includes both the final implementation and validation of all use-cases designed as well as improvements and extensions on benchmarks tests against LeanBigData platform.

Acknowledgments

This work has been partially funded by the European Commission under projects CoherentPaaS and LeanBigData (grants FP7-611068, FP7-619606), the Madrid Regional Council, FSE and FEDER, project Cloud4BigData (grant S2013TIC-2894), and the Spanish Research Agency MICIN project BigDataPaaS (grant TIN2013-46883).

References

1. Leanbigdata project. [Online], available at `http://leanbigdata.eu`
2. Ahmad, M.Y., Kemme, B., Brondino, I., Patiño-Martínez, M., Jiménez-Peris, R.: Transactional failure recovery for a distributed key-value store. In: Middleware 2013, pp. 267–286. Springer Berlin Heidelberg (2013)
3. Coelho, F., Vilaça, R., Pereira, J., Oliveira, R.: Holistic shuffler for the parallel processing of sql window functions. In: Proceedings of the international conference on Distributed applications and interoperable systems. DAIS'16, Springer-Verlag (2016)
4. Gonçalves, R.C., Pereira, J., Jimenez-Peris, R.: An rdma middleware for asynchronous multi-stage shuffling in analytical processing. In: Distributed Applications and Interoperable Systems (2016)
5. Jimenez-Peris, R., Patino-Martinez, M., Kemme, B., Brondino, I., Pereira, J.O., Vilaça, R., Cruz, F., Oliveira, R., Ahmad, M.Y.: Cumulonimbo: A cloud scalable multi-tier sql database. IEEE Data Eng. Bull. 38(1), 73–83 (2015)
6. Jimenez-Peris, R., Vianello, V., Patino-Martinez, M.: Paas-cep - a query language for complex event processing and databases. DataDiversityConvergence, Workshop "Towards Convergence of Big Data, SQL, NoSQL, NewSQL, Data streaming/CEP, OLTP and OLAP", held in conjunction with the 6th International Conference on Cloud Computing and Services Science - CLOSER 2016 (2016)
7. Nick, H., Randall, L., Edjlali, R., Buytendijk, F., Laney, D., Casonato, R., Beyer, M., Adrian, M.: Predicts 2015: Big data challenges move from technology to the organization. Gartner (2014)
8. Stone, M.: Big data for media. Reuters Institute for the Study of Journalism (2014)