Democratizing Data Analytics: Crowd-sourcing Decentralized Collective Measurements

Evangelos Pournaras, Edward Gaere, Renato Kunz, Atif Nabi Ghulam
ETH Zurich, Zurich, Switzerland
{epournaras,egaere,rkunz,anabi}@ethz.ch

***Abstract*—This paper introduces a Technology Readiness Level (TRL) 6 live demonstrator for socially responsible real-time data analytics: crowd-sourced decentralized collective measurements by preserving privacy. It provides a proof of concept for the democratization of big data running by citizens, for citizens. The demonstrator connects DIAS, the Dynamic Intelligent Aggregation Service, and GDELT, the Global Database of Events, Language, and Tone, to monitor planetary activity in different countries. DIAS remains operational for more than 4 months, exchanging millions of messages to self-adapt to data updates from GDELT. It accurately estimates the total news events providing high responsiveness to mobile devices. The demonstrator comes with a community Slack Bot for status updates of system health.**

***Index Terms*—data analytics, big data, decentralized system, network, self-adaptation, collected measurement, crowd**

I. Introduction

Smart data-intensive services are nowadays the cornerstone for improving citizens' quality of life in several sectors of society, e.g. energy, transport and health. However centralized collection and processing of personal data permits privacy intrusion, profiling, discriminatory and nudging actions. As a result, governmental and corporate bodies with large shares of computing resources and personal data have the power to undermine citizens' autonomy and freedom.

In contrast, this paper studies socially responsible data analytics as follows: (i) Crowd-sourcing citizens' computational resources. (ii) Forming bottom-up networks empowered by blockchain for decentralized computations without a trusted third party. (iii) Maximizing data locality, while sharing data under informational self-determination using differential privacy and homomorphic encryption. The potential and feasibility of such an alternative paradigm in real-world has not been demonstrated so far. This paper introduces a Technology Readiness Level (TRL) 6 live demonstrator to fill this gap.

The proposed demonstrator[1] tackles the following challenge: Decentralized systems are highly complex to develop, debug and test. Only when deployed in operational environments, they may fail due to synchronization flaws, low fault-tolerance, etc. Unpredictable users' interactions perplex even further the capability to self-adapt. A vicious cycle emerges: The system requires real-world usage profiles for verification, while users require a robust system to trust for usage.

The demonstrator breaks this vicious cycle by bringing together (i) DIAS, the *Dynamic Intelligent Aggregation Service* [1] and (ii) GDELT, the *Global Database of Events, Language, and Tone* [2]. DIAS (http://dias-net.org) performs real-time data analytics in a decentralized and privacy-preserving way. GDELT (http://gdeltproject.org) makes openly available via an Application Programming Interface (API) real-world data with which collective measurements are performed by DIAS to verify its TRL 6 level. The live demonstrator confirms the high accuracy of DIAS measurements over the total GDELT news events generated by different countries as well as the high system responsiveness for mobile devices.

The DIAS-GDELT demonstrator comes along with the following contributions: (i) Advancing the distributed prototyping toolkit of Protopeer [3] to support long-term operational lifecycle of TRL 6. (ii) A fully decentralized approach to make collective measurements of planetary news events via a DIAS-GDELT integration. (iii) A community Slack Bot for collaborative status monitoring of system health to coordinate recovery and maintenance actions.

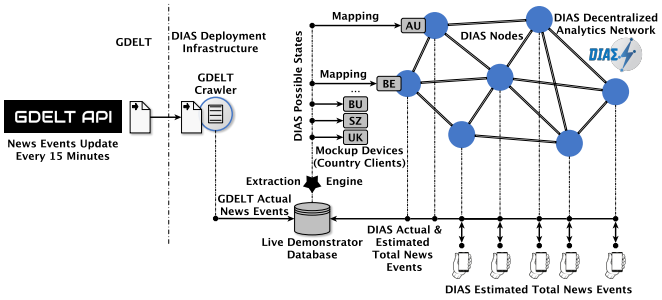
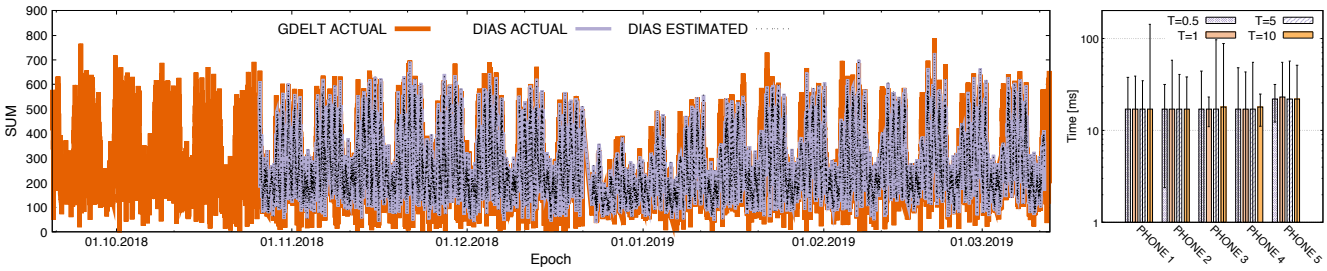


Figure 1: Architecture of the DIAS-GDELT live demonstrator.

II. Background

A DIAS network consists of software agents running on remote mobile devices, community servers or personal computers that share and aggregate data streams [1]. Each DIAS device computes several aggregation functions, e.g. SUM, AVG, MAX, etc. using the same algorithm. The aggregation functions receive as input all shared data from the online remote agents. Computations are accurate even in the following extreme cases: (i) Agents continuously share new data, i.e. data streams [1]. (ii) Agents arbitrarily join and leave the network. (iii) Agents fail. Agents joining, leaving or failing require corrective operations that roll back the performed computations to preserve accurate estimations [4]. DIAS self-adapts to these extreme dynamics based on its following features: (i) Efficient and decentralized network discovery of events using gossiping communication. (ii) A distributed memory system based on Bloom filters with which agents collaboratively reason which data records are new, which require update or which are duplicate and should be ignored.



(a) DIAS can accurately estimate the actual GDELT events in the long term.

(b) DIAS responsiveness.

Figure 2: Proof of concept.

GDELT monitors global human society. It captures news media in real-time at a planetary scale and in over 100 languages, whether news are in print, broadcast or web formats. GDELT data are a result of natural language and machine learning algorithms that extract more than 300 event categories, millions of themes and thousands of emotions as well as the networks that tie them together. Data can be accessed via an API in almost real-time, i.e. every 15 minutes.

III. Decentralized Monitoring of News Events

Figure 1 illustrates the DIAS-GDELT live demonstrator architecture[1]. DIAS nodes are mapped to 28 countries generating GDELT news events. A DIAS node (GDELT country) disseminates the number of news generated during the last 15 minutes and at the same time aggregates the number of news generated by the other GDELT country nodes. All GDELT country nodes accurately compute the total number of news generated by all countries participating in the network.

The DIAS-GDELT live demonstrator consists of the following: (i) *GDELT crawler*: It fetches GDELT news updates every 15 minutes and sends them to the demonstrator database and the mockup devices. (ii) *Live demonstrator database*: It stores all required data. (iii) *Extraction engine*: It uses the raw GDELT data to extract the possible states of the DIAS nodes [1]. (iv) *Mockup devices*: A general-puprose software client connecting DIAS nodes with a data source, i.e. smart phones or GDELT countries in this case. (v) *DIAS nodes*: They perform data analytics. All components are deployed in distributed Hetzner servers (http://hetzner.com). Every DIAS node runs in a separate JVM. Communication is performed via ZeroMQ implementing the network interface of Protopeer [3] to achieve a memory leak-free and reliable communication.

Figure 2a outlines the proof of concept. Three time series data are shown: (i) *GDELT Actual*: The raw baseline values extracted from GDELT. (ii) *DIAS Actual*: 9 representative state values, e.g. low, medium and high profiles, are used as input to DIAS nodes to make a decentralized aggregation computationally feasible. The state profiles are extracted by a sliding a window of 27 observations and uniformly sampling 9 values. (iii) *DIAS Estimated*: The estimated values of the DIAS nodes. The interactive live version found online[1] updates every 15 minutes. The user can turn on or off the three time series as well as the data of each country node.

The performance of four periodic requests[1] for aggregates made by five smart phones[1] is evaluated by measuring Round Trip Times (RTTs). No timeouts are detected and the median RTTs are shown in Figure 2b. Results confirm the high responsiveness, in the order of a few milliseconds, of DIAS when frequently accessed by mobile devices.

The live demonstrator comes with a community Slack Bot that monitors the system health status bidirectionally: The Slack Bot triggers (smart phone) notification in case of errors. Users proactively inquire the system heath status via commands. The Slack Bot relies on two Python monitors. The first monitor queries regularly the live demonstrator database and if there are no updates after 30 minutes, it raises a warning. The second monitor queries the DIAS aggregates.

IV. Conclusion and Future Work

This paper concludes that privacy-preserving collective measurements over decentralized dynamic networks are feasible to prototype at a TRL 6 operational level. Highly accurate and responsive aggregation measurements running for months with GDELT and smart phone data provide a proof of concept. The DIAS-GDELT live demonstrator aspires to convey a blueprint for democratizing data analytics, by citizens, for citizens. Ongoing and future work focuses on the following aspects: Scaling up the deployment and management of the DIAS network by a community. Connecting DIAS with distributed platforms for the Internet of Things. Turning the Slack Bot to a decentralized service integrated within DIAS.

References

[1] E. Pournaras, J. Nikolic, A. Omerzel, and D. Helbing, "Engineering democratization in internet of things data analytics," in *2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2017, pp. 994–1003.

[2] K. Leetaru and P. A. Schrodt, "Gdelt: Global data on events, location, and tone, 1979–2012," in *ISA annual convention*, vol. 2, no. 4. Citeseer, 2013, pp. 1–49.

[3] W. Galuba, K. Aberer, Z. Despotovic, and W. Kellerer, "Protopeer: a p2p toolkit bridging the gap between simulation and live deployement," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, 2009, p. 60.

[4] E. Pournaras and J. Nikolić, "Self-corrective dynamic networks via decentralized reverse computations," in *2017 IEEE International Conference on Autonomic Computing (ICAC)*. IEEE, 2017, pp. 11–20.

[1] **Live Demonstrator**: http://dias-net.org/dias-gdelt-live. **Source**: https://github.com/epournaras/DIAS-GDELT. **Documentation**: https://github.com/epournaras/DIAS-Documentation. **Settings** (in sec and number of requests): T=0.5, 15378, T=1, 4028, T=5, 4034 and T=10, 4370. **Phone 1 & 2**: HTC Desire 12+, Android 8.0. **Phone 3 & 4**: Moto G 5s, Android 7.1.1. **Phone 5**: Motorola Moto G, Android 5.1.