

COMPLEX EVENT PROCESSING APPROACH ON SUBSCRIBERS' DATA OF TELECOM OPERATOR

ZHUMEKENOV A.

Kazakh-British technical university, 050000, Almaty, Kazakhstan

Abstract. Nowadays, the usage of mobile phones reached extremely large worldwide proportions and increasing dramatically, there is a stronger need to decrypt the important information that is hidden among them. Even all needed information is gained, the processes of companies remain static and can't be changed dynamically to adapt to actual business needs, reducing the advantages that can be achieved. Every second, millions of raw information are generating by mobile users, which are handled by Telecom operators in data servers. By using the Complex Event Processing (CEP) approach in real-time, we can obtain the information that matters to our business and use it to monetize the vast amount of data that is being collected through mobile phone usage. In this paper, we present an internally developed framework that combines the strengths of CEP and business process implementations, which allows us to react to the needs of today's fast-changing environment and requirements. We demonstrate 3 simple use case scenarios to show the effectiveness of the CEP approach in our situation. The importance of implementing the CEP approach on subscribers' data should not be overlooked as means of trying to capitalize on new services but have to be considered as a challenge to allow subscribers get more customized offers and services.

Keywords: Complex Event Processing, Telecom Data Analysis, Information Processing, Targeted campaigns.

БАЙЛАНЫС ОПЕРАТОРЫНЫҢ АБОНЕНТТІК ДЕРЕКТЕРДІ ӨНДЕУДІҢ КЕШЕНДІ ТӘСІЛІ

ЖУМЕКЕНОВ А.

Қазақстан-Британ техникалық университеті, 050000, Алматы, Қазақстан

Аңдатпа. Қазіргі уақытта ұялы телефондарды пайдалану бүкіл әлем бойынша өте үлкен мөлшерге жетті және күрт өсуде, олардың арасында жасырылған маңызды ақпаратты шифрдан шығару қажеттілігі күшейе түседі. Тіпті барлық қажетті ақпарат алынады, компаниялардың процестері тұрақты болып қалады және оларды қолдауға болатын артықшылықтарды азайта отырып, іскери қажеттіліктерге бейімдеу үшін динамикалық түрде өзгерту мүмкін емес. Әр секунд сайын ұялы байланыс пайдаланушылары миллиондаған ішкі ақпараттар шығарады, оларды байланыс операторлары деректер серверлерінде басқарады. Нақты уақыт режимінде іс-шараларды өңдеудің кешенді әдісін қолдану арқылы біз өз бизнесіміз үшін маңызды ақпаратты ала аламыз және оны ұялы телефондарды пайдалану арқылы жиналатын көптеген деректерді монетизациялау үшін қолдана аламыз. Бұл жұмыста CEP пен бизнес-процестерді жүзеге асырудың мықты жақтарын біріктіретін ішкі дамыған құрылымды ұсынамыз, яғни қазіргі заманғы тез өзгеріп отыратын ортаның қажеттіліктері мен талаптарына жауап беруге мүмкіндік береді. Аталған CEP тәсілінің тиімділігін арттыру үшін 3 қарапайым сценарийді көрсетеміз. Абоненттердің мәліметтеріне CEP тәсілін енгізудің маңыздылығы жаңа қызметтерді пайдалануға тырысу құралы ретінде ескерілмейді, ал олай болса, абоненттерге неғұрлым бейімделген ұсыныстар мен қызметтерді алуға мүмкіндік беру мәселесі ретінде қарастырылуы керек.

Түйінді сөздер: абонент деректер анализі, мақсатты науқандар, байланыс оператордың анализі, мақсатты қауқандар

КОМПЛЕКСНЫЙ ПОДХОД К ОБРАБОТКЕ СОБЫТИЙ АБОНЕНТСКИХ ДАННЫХ ОПЕРАТОРА СВЯЗИ

ЖУМЕКЕНОВ А.

Казахстанско-Британский технический университет, 050000, Алматы, Казахстан

Аннотация. В настоящее время использование мобильных телефонов достигло чрезвычайно больших масштабов во всем мире и продолжает стремительный рост, в связи с чем существует большая потребность в расшифровке важных данных, которые скрыты от обычных пользователей. Даже если все необходимые данные по абонентам получены, процессы компаний остаются статичными и не могут динамически изменяться для адаптации к реальным потребностям бизнеса, что снижает достижимые преимущества. Каждую секунду мобильные пользователи генерируют миллионы необработанных и неструктурированных данных (например, информация о звонках, СМС, пользование интернетом), которые хранятся операторами связи на серверах данных. Используя подход комплексной обработки событий (КОС) в режиме реального времени, мы можем получить информацию по абонентам, которая действительно важна для бизнеса телекомоператора, и использовать ее для монетизации. В этом документе мы представляем решение, которое объединяет сильные стороны КОС и методологию автоматизации бизнес-процессов, что позволяет реагировать на потребности сегодняшней быстро меняющейся среды и требований со стороны бизнес-заказчиков. Мы продемонстрируем 3 сценария использования, чтобы показать эффективность подхода КОС. Не следует рассматривать реализацию подхода КОС к данным абонентов как попытку извлечь бизнес-выгоду с помощью монетизации услуг, а следует рассмотреть как возможность, позволяющую абонентам получать таргетированные предложения и сервисы.

Ключевые слова: комплексная обработка событий, анализ телекоммуникационных данных, обработка информации, целевые кампании

Introduction

In 2019, the mobile phone user database reached almost 70% of the global population, with more than 5 billion devices in use [1]. Given the dynamicity of today's business environments, there is a need to continuously adapt and keep up-to-date the business processes in order to respond to the changes in those environments and keep a competitive level among Telecom operators and solution providers. One of the main concerns for Telecom operators' applications is to handle and interpret online raw data. By using the CEP approach and developed framework we can facilitate the solution of this problem by gathering needed information in real-time about the subscribers' different events on the Telecom network in order to determine the necessity of making specific offers and suggesting new promo.

Real-time data processing systems are used widely to provide insights about events as they happen. Many companies have developed their systems: Twitter's Storm [10] and Heron [9], Google's Millwheel [8], LinkedIn's Samza [7] and Facebook's Puma, Swift and Stylus [11].

One of the main challenges for processing data dynamically is the size and the speed of incoming data at which it is being generated. Every second, millions of data are generated by phone mobile users and entire traffic allocates to Telecom operator's data centers. Despite having a common goal, solutions based on CEP differ in a wide range of aspects, including architectures, data models, rule languages, and processing mechanisms. In part, this is due to the fact that they were the result of the research efforts of different communities, business market demands, each one bringing its view of the problem and its background for the definition of a solution [3].

In this paper, an internally developed framework is presented that combines the advantages of the CEP and dynamic business process adaptation, which allows us to respond to the needs of today's rapidly changing environments. Subscribers' information for different business-specific use cases and triggers is handled such as balance information, mobile internet usages, the location of the device to enrich data about subscribers, and give deep

insights on user's interest, lifestyle patterns in a given time period. The picture below illustrates our main idea of implementing the CEP approach on Telecom operator's data traffic.

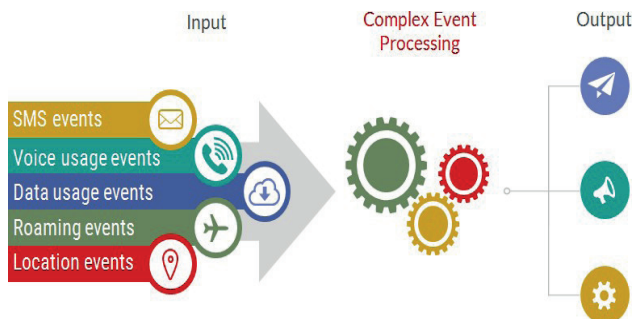


Fig. 1. Main idea

The Framework

Before we start our journey on solution architecture, it is important to say about motivations, limitations and business/technical requirements that faced.

Firstly, we need to mention the current limitations of legacy systems, where transactions per second (TPS), in our case Events Per Second is more precise, was limited by 2000. Taking into account, that a subscriber base reaches 10 million and an active subscriber base is almost 6 million, that threshold is insufficient to cover and handle all network events in real-time. Besides, the system was consuming all resources all the time, which caused performance issues and several critical incidents on the system. Analyzing the data traffic throughout (22.5 TB/month), dynamicity of business requirements, and capabilities of the new solution, challenged us to handle at least 165 000 events per second.

Secondly, from the technical perspective, our approach was to create one job that will run all triggers and business rules. By this approach, we should benefit that all triggers and business rules will run effectively, and manage rules without restarting the system.

Our proposed solution uses a component-based architecture, where each component plays a critical role in a whole system. Figure 2 on page 3 illustrates a high-level overview of the system. On the left, the ingestion layer is responsible for fetching and receiving raw data from different

data sources (SMS events, Voice usage, Internet usage, Location-based triggers) and saving it into the Events hub. Then the data is consumed by the event execution engine, which in turn, based on current trigger definitions, outputs notifications to the Events hub. Finally, the outgestion layer fetches the notification and dispatches them to appropriate subsystems. All main components in the architecture will be described in the next sections to give you a more detailed overview of the solution.

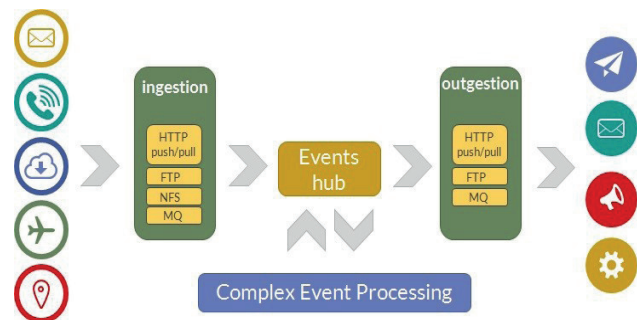


Fig. 2. High-level architecture overview

The Ingestion

The process of collecting data from various data sources, preparing it, and saving it for complex processing and generating triggers. Currently, there are several different approaches to how a data ingestion pipeline can be built:

- The first approach consists of two steps. Firstly, the raw data is fetched from a data source and is a subject for some preliminary processing, e.g., splitting a huge file containing multiple events. After that, the data is being transformed, serialized, and finally pushed to the Events hub, where it waits for consumption by the event execution engine. This approach is preferred whenever the event transformation is complex or the data volume is too big to be processed in the ingestion layer.
- The data is fetched/received, transformed, and serialized within the same pipeline.
- The raw data is immediately pushed to the Events hub. Transformation takes place in the execution engine.

It is worth mentioning that both intermediate results and transformed events are stored in Events hub topics. Each event type (more

precisely, events originating from the same data source) is kept in a separate topic. Despite which ingestion approach suits better, we need to notice what file types and protocols can be ingested in our system. To illustrate how various file types are fetching and transforming to one input format, see figure 3 on page 3

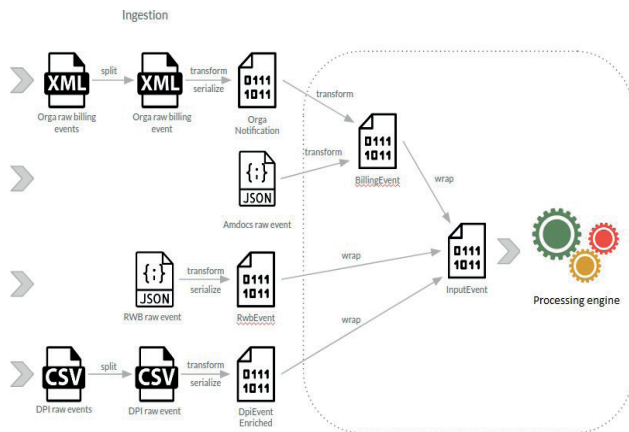


Fig. 3. Ingestion file types

Events Hub

It is a system that keeps all the in-flight data streams in a reliable and persistent way. The data in an Events hub topic is divided into partitions. The partitions are distributed over the servers in the cluster with each server handling data and requests for a share of the partitions. Each partition is replicated across a configurable number of servers for fault tolerance. For a topic with replication factor N, the Events hub tolerates up to N-1 server failures without losing any records committed to the topic.

The processing engine

It is an execution engine for complex event processing applications. In this component, streaming applications are implemented, that consume all the prepared events by the ingestion system and apply trigger procedures defined by business users. The provided management functionalities over the CEP applications, like define, start/stop of the application. Execution jobs are stateful. To make state fault-tolerant, the processing engine makes periodic checkpoints of the state. Checkpoints allow to recover state and positions in the streams in case of a program failure (due to machine, network, or software

failure). Any records that are processed as part of the restarted job are guaranteed to not have been part of the previously checkpointed state. In other words, each event from an input stream affects the data flow state exactly once. However, it does not mean each event is processed only once. More precisely, when the state and input streams' offsets are restored, the processing engine needs to replay the events that have been processed between the last checkpoint and the failure. In consequence, for instance, some notifications might be generated twice. To illustrate how state and checkpoints are working, please see figure 4 below.

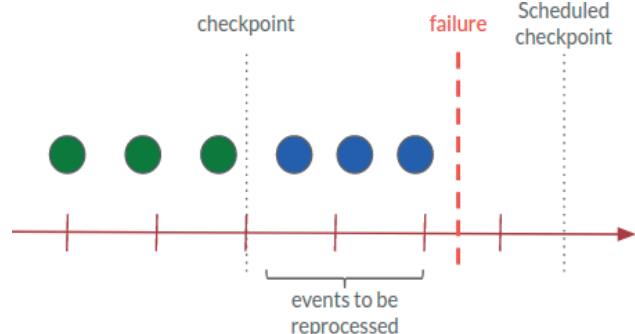


Fig. 4. Checkpoint and failure

The Outgestion

The process of communicating with external layers based on processed data. outgestion layer is pretty straightforward, all Processing job notifications are stored in a single Events hub topic. Then the outgestion pipeline reads them, transforms to the format-specific for the destination system, and pushes them there.

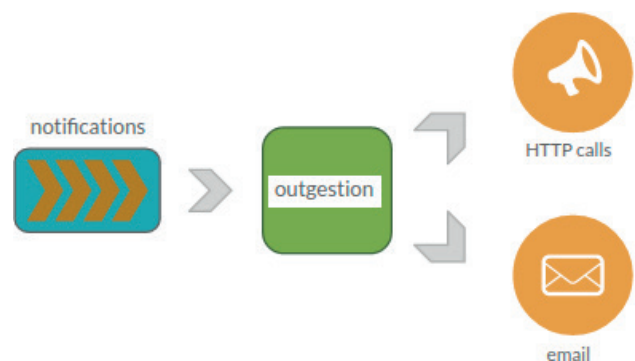


Fig. 5. Outgestion

The technology stacks

Regarding the technology stack that was

used, take a look at the list below, all components are based on a free and an open-sourced Apache product:

- Apache Flink - event processing engine. Provides flexible event time support. Fault-tolerant, high throughput and low latency.
- Apache Hadoop - a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. In the provided solution, Hortonworks distributive is using.
- Apache Kafka - central events hub, where all events come through. Simple and reliable. The ingestion system saves data to it, the outgestion system is reading data from it for sending it further. It is the only component that communicates with Apache Flink directly.
- Apache NiFi - engine for managing ingestion and outgestion pipelines. NiFi collects data from data sources and transfers them to Apache Kafka. With NiFi it is possible to deploy different servers, for example, HTTP, FTP, etc.

The Use Cases

In this section, I want to describe top-3 business use cases, which could be easily configured in the CEP and give win-win results when the whole system starts working.

1. **Balance top-up** – *If subscriber top-ups her balance too often in a short period of time. We can offer her a less expensive tariff or auto-payment services.*

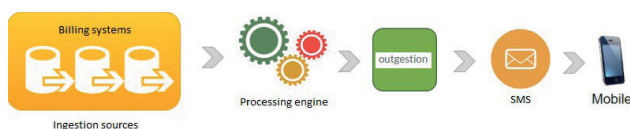


Fig. 6. Balance top-up

2. **Fraud detection** – *Send an email to the anti-fraud unit if a subscriber registered in roaming but his balance at the moment is equal to 0.*

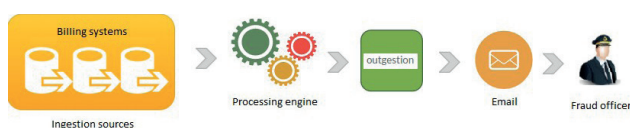


Fig. 7. Fraud detection

3. **Extra bonuses** – *Top-up subscriber balance with extra SMS/Voice usage/Internet usage bonuses when the subscriber fulfills all conditions at promo campaigns.*

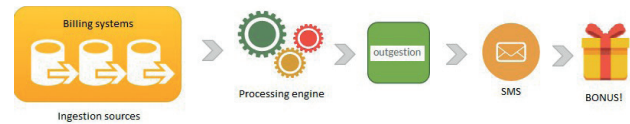


Fig. 8. Extra bonus

Conclusion and future work

In this paper, the solution to how the CEP approach could be implemented in Telecom operator is presented, which helps to solve many technical limitations of our legacy systems and business requirements issues, process, and analyze the vast amounts of subscriber's data in near real-time that was impossible before. Furthermore, we have defined triggers and business rules which could give a win-win effect after project launches. Apache components that are typically used in our architecture provide the possibility to build the CEP ecosystem, which will play a vital role in the next decade for the Telecom operator business. Ongoing work will focus on further optimization and improvement possibilities, some of the directions in which this work can be improved are:

- One of our approaches was to build one job to run all triggers/rules. However, any coin has a reserve side, and in our case, one bad rule could affect all other triggers/rules impacting a whole system. That is the main challenge that should be addressed in future works.

- Response latency time will be always an actual and crucial issue, due to the dynamicity of today's business environments and requirements. We need to pay attention to how effectively ingestion and outgestion pipelines are executing the data traffic.

- Because data traffic is extremely huge and grows dramatically, the challenge is to continuously write and store data in Data Lake/ Data Warehouse and implement an Online Analytical Processing engine for analytical purposes and reports. Separating written data from data traffic should help to avoid performance and sustainability issues.

REFERENCES

1. <https://datareportal.com/reports/digital-2019-global-digital-overview> .
2. Digital 2019: Global digital overview.
3. Using Complex Event Processing for Dynamic Business Process Adaptation.
4. Gabriel Hermosillo, Lionel Seinturier, Laurence Duchien. 2010 IEEE International Conference on Services Computing.
5. Processing Flows of Information: From Data Stream to Complex Event Processing.
6. Alessandro Margara, Gianpaolo Cugola
7. SoLoMo analytics for telco Big Data monetization.
8. H. Cao, W. S. Dong L, S. Liu, C. Y. Ma, W. H. Qian, J. W. Shi, C. H. Tian, Y. Wang, D. Konopnicki, M. Shmueli-Scheuer, D. Cohen, N. Modani, H. Lamba, A. Dwivedi, A. A. Nanavati, M. Kumar.
9. MigCEP: Operator Migration for Mobility Driven Distributed Complex Event Processing.
10. Beate Ottenwälder, Boris Koldehofe, Kurt Rothermel, Umakishore Ramachandran.
11. The 8 Requirements of Real-Time Stream Processing.
12. Michael Stonebraker, Ugur Çetintemel, Stan Zdonik.
13. Samza. <http://samza.apache.org>.
14. T. Akidau, A. Balikov, K. Bekiroglu, S. Chernyak, J. Haberman, R. Lax, S. McVeety, D. Mills, P. Nordstrom, and S. Whittle. Millwheel: Fault- tolerant stream processing at internet scale.
15. PVLDB, 6(11):1033-1044, Aug 2013
16. S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja. Twitter heron: Stream processing at scale.
17. A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham, N. Bhagat, S. Mittal, and D. Ryaboy. Storm@twitter. In SIGMOD, pages 147–156, 2014
18. Realtime Data Processing at Facebook
19. Guoqiang Jerry Chen, Janet L. Wiener, Shridhar Iyer, Anshul Jaiswal, Ran Lei Nikhil Simha, Wei Wang, Kevin Wilfong, Tim Williamson, and Serhat Yilmaz Facebook, Inc
20. The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems.
21. D. C. Luckham Addison-Wesley Longman Publishing Co., Inc., 2001

Information about author:

1. Zhumeckenov A. – Faculty of Information Technology, Kazakh-British Technical University
Email: zhumeckenov@gmail.com