FRI-2G.302-1-CSN-15

BIG DATA LIFE CYCLE IN MODERN WEB SYSTEMS [15]

Svetalana Stefanova, PhD
Department of Computer science,
“Angel Kanchev” University of Ruse
Tel.: +359 888 309 579
E-mail: sstefanova@ecs.uni-ruse.bg

Iliya Draganov
Department of Computer science,
“Angel Kanchev” University of Ruse
Phone: +359 883 476 542
E-mail: idraganov@uni-ruse.bg

Abstract: *The paper is reviewing proposed or existing life cycles of Big Data, and analyses their universality and versatility when applied in modern web-based systems. Every single web application or service is generating on its own the actual life cycle of the data that it is processing. Those cycles have in common a lot of stages, which are in turn organised in groups, in order to form the web system's software architecture. With the technological advancement of web software and hardware, new points of view are emerging regarding the connection between web systems' architecture and Big Data life cycles. The number of web systems, which automatically process data and communicate with other systems similar to them, is growing significantly, which demands reviewing the approach used when handling Big Data.*

Keywords: *Big Data, Lifecycle, Web Systems, Review, Data Processing*

INTRODUCTION

There are many proposals for structuring a web-based system and directing the life cycle of data after that. Most of them start with the generating, collecting and capturing of the data, followed by the processes of filtering, analysing, encrypting, sorting and redirecting. The most significant part of the cycle is its completion – the saving of the data, which involves storing and securing the availability of Big Data. Last, but not least, comes a stage of publishing, sharing and visualising of requested data from the storage. Because of the integration of many end-user services into the World Wide Web, there is a huge amount of web systems with the potential for fast growth and the possibility of reaching the point of becoming Big Data proprietors. For some organizations, this might be as close as few dozens of terabytes of data, while for others it may be hundreds of petabytes. Therefore, this paper will focus on analyzing the data life cycle in the small and mid sized web systems, as they are most common.

EXPOSITION

Initial phase of the data life cycle

There are three basic ways known for data capturing that are very important – acquisition of existing data, entry of new data, and capturing (Chisholm, M., 2015). They represent almost 50% of the whole cycle's timespan, and mainly start the system's work process. Data collection or generation is generally the first phase of each data life cycle. Large amounts of data are created in the forms of log file data and sensor data, mobile equipment, satellites, laboratories, supercomputers, searching entries, chatting records, posts on Internet forums, and microblog messages (Nawsher, K., 2014). Acquisition is the process of receiving already existing data, formatted and accepted in a certain convinient way. If the data is formatted according to the

[15]

system's recommendations, the process of acquiring it is much faster. Information entry is the newly created data from an end user, which in some cases could be extremely valuable, for example, when it is a media type, because of its uniqueness. This, however, may require additional reformatting and/or reorganising, because of the difference in the user's and the software engineer's perspectives. The third approach is only used in very specific situations or mainly for monitoring someone's or some device's activity over time, because it focuses more on the data time occurrence than the information itself. This phase of the life cycle does not have any ideological differences, but rather has software impementation ones. Visualisation of the process is mandatory when data is created by the user, it is almost always present while monitoring, and it is optionable when adding to already existent data. Table 1 offers a brief comparison and shows the differences between the known approaches.

Table 1 Data gathering approaches comparison

	Origin	**Additional Processing**	**Purpose**	**Visual Presentation**
Entry	**New Data**	**Imposed by the software architecture**	**Populate**	**Always**
Acquisition	**Existing Data**	**Imposed by the software architecture**	**Populate**	**Optional**
Capture	**Timestamps**	**None**	**Monitor**	**Optional, but recommended**

Decision phases of the data life cycle

The next stages in the life cycle of Big Data's pieces involve filtering, analysis and distribution to storage spaces. In an ideal scenario, it is possible to easily handle vast amounts of data if only distribution action is required, but in most cases, due to security issues, software architecture or project directing decisions, additional processing operations are required. Very often the reformatting and structural reorganisation of the received data is the performance "difficulty" in such web systems. In order to avoid or skip this stage, it is preferable that the interface or the "connecting mechanism" takes care of those actions, and if applicable – it also includes filtration. That way, the layer for receiving data (first phase of the life cycle) by the system becomes interchangeable in the aspect of software technology, and it is able to connect with other software products only if it requires certain formatting specifications.

With the entry approach it is easier to exclude the stages of filtration and analysis, because those functions can easily be implemented in the user's interface. All the technologies used in the user-to-computer communication naturally manage the type and structure of the content that is inserted, such as mobile applications and browser forms.

Acquisition of existing data is more likely to be difficult and dangerous, thus filtration and securing the system is almost always imperative, and fully mandatory when the documentation is public. There is always a chance of acquiring data that is harmful to the system, even when the system is gathering information from a source that is well-known and considered risk free. When a web system is receiving data, it is usually in an encrypted way with public security certificates in which the sender is known, or an authentication mechanism is used - those are usually a sort of a text string keys or credentials. There is also the problem with the analysis of the data destination, which is why good documentation of the receiving/gathering part of the system is required, so that the data entering the web system can be formatted and structured accordingly.

The data-filtering phase is optional and it is mainly used when there is a risk of data floodage, which creates "noise" in the environment and if all of the data is recorded, a fast depletion of storage space can occur. Such phases can be ignored if the network's infrastructure filtrates the data itself.

Even though the distribution phase is brief, it is very crucial, because it represents the final step before saving the data, and it can send the data into an inexisting or unreachable destination. It is often then, that the software products are not able to handle a failure, meaning that when an unsuccessful transfer occurs, the data is not stored anywhere for it to be recovered or processed with human intervention. The log files however, if such are available, allow for the "lost" data to be stored temporarily, by showing what data has not been processed or transferred correctly. In the case of web systems, a broken connection to a destination or a database query failure can lead to a continuous data loss, which means an end to the life cycle. The software layer should be designed and structured in a way that it allows to be interchanged when needed with fewer modifications to the interface or database parts. Usually, that kind of modification is done when a system is increasing in its scale. Many of the software tools are module based, so that flexibility can be achieved if there is a need to make a change to the web system.

Analysis of the incoming data can be implemented before and between the filtering and distributing stages, just before the actual saving in the storage. The process itself can be a heavy task, which is why it should be well planned and placed at a correct position in the life cycle. With the increasing complexity of the information, the probability of need for an analysis stage before every other stage is increasing. Complicated systems are executing various types of analysis – data mining algorithms, cluster analysis, correlation analysis, statistical analysis and regression analysis. Having an analysis done of the data after it has been stored is only recommended when doing reports, or when there are plenty of computing resources left unused.

Combining the initial types of stages with the ones described above, can lead to conclusions on which phases can be omitted, depending on the situation.

	Entry	**Acquisition**	**Capture**
Filter	**Not needed**	**Mandatory**	**Mandatory if not available from the environment.**
Analysis	**Not needed**	**Mandatory before storing complex data.** **Optional after storing any data.**	**Used after storing the data.**
Distribution	**Mandatory**	**Mandatory**	**Mandatory**

Storing phase in the data life cycle

In order to use the collected data further, the step that every modern web system is taking is to store and secure the processed information. This phase can take place on the same computing machine or on several others (clusters), depending on the system's scale. In most of the cases, a database engine does this job, but it needs to be configured properly, so data loss can be avoided in an event of an attack or a hardware failure. A good practice is to schedule a creation of a backup copy of the data and transfer it to another location. The following factors must be considered when using a distributing system to store Big Data - consistency, availablility and partition tolerance. This means that the data must always be available in its latest version, and that it could be accessed when part of the storage system is down or unavailable. For smaller scale systems, a part of the solution is keeping cached information somewhere in the nodes that are executing other phases of the data life cycle.

Data usage phase

This phase is inevitable, because it is the essential reason for Big Data web systems existence. From a software engineering point of view, this phase can occur in the same environment as the data capture phase. Stored data can be sent to another system, or visualised to the end user. Most of the time, only the second option is chosen, but with the development of data providing software technologies and tools, both options are combined in order to strive for diversity in the types of information exchanged between web services. This stage can appear as the initial stage of the lifecycle of another system, or the restart of the current one. If data is only presented to an end user or system, without enabled editing or deleting rights, it is considered to be simply reported or just proved to exist; otherwise it is clear that a reset of the lifecycle is intended.

Data visualisation is necessary in order to present the answer in a clear and simple way so that a human can easily understand and visualize it. It is at this stage in the data life cycle when we need to consider, along with functionality, aesthetics and human visual perception to convey the results of data analysis (Wing, J., 2018). There is a huge variety of types of devices that display information to the end users, which is why the software technology that does the work needs to be very flexible and must have a rich set of tools in order to present any kind of data in an understandable format. Browser based technologies are universal, but lack in performance when dealing with large volumes of data. Software technologies, which are based on the type of the operating system on the end user machine, usually require additional software installed, and are prone to failures during the data’s phase of distribution. For the time being, software technologies, which are working on operating systems designed for mobile platforms, are the most balanced in performance and flexibility. Mobile devices are most commonly used in a human’s everyday life. Almost every Big Data web system is targeting primarily the mobile platforms as a “data acquisition” node, which is why every end user is a comprehending witness of this finalising or restarting stage of the data life cycle.

CONCLUSION

This paper introduces one of the possible data life cycles, which is suitable not only for Big Data, but for small and mid range data size as well. A comparison analysis is also introduced between the known ways for data capturing and the phases which can be omitted in different situations. The suggested data life cycle can be useful for any software system that works with data.

ACKNOWLEDGMENTS

This report is published with the cooperation of the 2018-EEA-01 “Methods of collection, organisation, access, process and analysis of large volumes of structured and unstructured data” project, financed by the “Scientific researches” fund of the “Angel Kanchev” University of Ruse.

REFERENCES

Chisholm, M. (2015). *7 phases of a data life cycle*, Data Governance. Bloomberg Professional Services. URL: https://www.bloomberg.com/professional/blog/7-phases-of-a-data-life-cycle/ (Accessed on 17.08.2018)

Nathan, B., & Raju, P. (2018) *The Data Life Cycle*, Strategic Finance Magazine, URL: http://sfmagazine.com/post-entry/july-2018-the-data-life-cycle/ (Accessed on 14.09.2018).

Nawsher, K., (2014). *Big Data: Survey, Technologies, Opportunities, and Challenges*. The Scientific World Journal Volume 2014, Article ID 712826, 18 pages

Wing, J., (2018). *The Data Life Cycle*, Data Science Institute. Columbia University in the City of New York. URL: https://datascience.columbia.edu/data-life-cycle (Accessed on 17.08.2018)