CHARACTERIZING BIG DATA SOFTWARE ARCHITECTURES

This briefing reports scientific evidence on the state of the art related to big data software architectures addressing basic requirements, modules, and architectural patterns.

FINDINGS

Findings are grouped into three different groups.

Architectural Requirements for Big Data Systems

- *Scalability:* Big data systems must be able to increase and support different amounts of data, processing them uniformly, allocating resources without impacting costs or efficiency.
- *High Performance Computing:* Big data systems must be able to process large streams of data in a short period of time, thus returning the results to users as efficiently as possible.
- *Modularity:* Big data systems must offer distributed services, divided into modules, which can be used to access, process, visualize, and share data, and models from various domains, providing flexibility to change machine tasks.
- *Consistency:* Big data systems must support data consistency, heterogeneity, and exploitation. Moreover, systems must be flexible to accommodate data exchange formats and achieve the best accuracy to perform these operations;
- *Security:* Big data systems must ensure security in the data and its manipulation in the architecture, supporting integrity of information, exchanging data, multilevel policy-driven, access control, and prevent unauthorized access.
- *Real-time operation:* Big data systems must be able to manage the continuous flow of data and its processing in real time, facilitating decision making;
- *Interoperability:* Big data systems must be transparently intercommunicated to allow exchanging information between machines and processes, interfaces, and people [46];
- *Availability:* Big data systems must ensure high data availability, through data replication horizontal scaling, i.e., spread a data set over clusters of computers and storage nodes, avoiding bottlenecks.

Architectural Modules for Big Data Systems

- *Data Sources:* This module identifies all sources of data and all possible expected formats and structures. The specifications of data sources must be clearly described and documented, detailing information of the different databases and files.
- *Data Integration:* This module supports the process of integration of multiple data and knowledge representing the same real-world object into a consistent, accurate, and useful representation.
- *Data Storage:* This module supports mechanisms to organize and optimize storage space usage, localization, availability, and reliability.
- *Data Analytics:* This module processes and analyze data. Analysis is a very computing intensive and time consuming process and must be optimized for best performance.
- *Data Visualization:* It must support data visualization by making it more accessible, understandable, and usable for users to identify patterns and trends.

Architectural Patterns for Big Data Systems

- *Pipe and filters:* This pattern deals with how streams of data are successively processed or transformed by components. Its importance is justified because the sequence of data analysis is essential, for example, it is not possible to process a data that has not been still integrated.
- *Orchestration:* It configures and combines other modules by integrating activities into a cohesive system. It includes an optimizer for managing the data flow in all pipeline stages.
- *Layered Architectural Pattern:* It focuses on grouping related functionalities required in a system application into distinct layers that are stacked on top of each other. Following are the main layers:
  - *Producers and Consumers:* This layer is responsible for representing all big data producers and consumers, i.e. customers, suppliers, managers, among others.
  - *Security, management, and monitoring:* It includes components that provide base functionalities needed in the other layers, ensuring the proper operation of the whole infrastructure. Some components needed are destined to monitoring, bottlenecks detection, performance improvement by adjusting parameters of the adapted technologies, authentication, data protection, among others.

Middlewares are used to assure quality in the data exchange and improve the communication among modules. However, it should not be a bottleneck.

Who is this briefing for?

Software engineering practitioners who want to make decisions about architectural patterns, selection of components and quality attributes for big data systems based on scientific evidence.

Where the findings come from?

All findings of this briefing were extracted from the systematic mapping study conducted by Sena et al.

What is included in this briefing?

The main findings of the original systematic mapping.

Evidence characteristics through a brief description about the original systematic review and the studies it analyzed.

What is not included in this briefing?

The systematic steps to achieve these findings.

Details about the selected studies that describe these findings.

To access other evidence briefings on software engineering:

http://cin.ufpe.br/eseg/evidence-briefings

For additional information about START:

http://www.labes.icmc.usp.br

ORIGINAL RESEARCH REFERENCE

Bruno Sena et al. ***Characterizing Big Data Software Architectures: A Systematic Mapping***. SBCARS, 2017.