Task ID: Big-Data-Stream-Evaluation

Design Scenario

An internet company provides popular content and online services to millions of web users. Besides providing information to external users, the company collects and analyzes *massive logs* of data that are generated from its infrastructure (e.g. server logs). The size of the log files is in *Tera-bytes*. To cope with the fast infrastructure growth, the company decided to develop a software application to manage the logs. A high level conceptual model has been designed, which consists of a data stream, which sends its data to a batch layer and a real time view. The data stream component dispatches data from multiple data sources in real-time. The architects of the system are discussing the possible technology choices for implementing the data stream component. Three technology families are identified as alternative architectural solutions:

1) Data collector technologies (e.g. Apache Flume, Fluentd)
2) Distributed message broker technologies (e.g. Apache Kafka, Amazon SQS, Active MQ)
3) ETL/Data Integration engines (e.g. Streamsets, Talend)

Non-functional requirements

- *Performance*: The system shall collect up to 15,000 events/second from web servers.
- *Extensibility*: The system shall support adding new data sources by just updating a configuration file.
- *Availability*: The system shall continue operating with no downtime if any single node or component fails.
- *Deployability*: The system deployment procedure shall be fully automated and support a number of environments (development, test, production).

Constraints

The system shall be composed of primary open source technologies (for cost reasons).

Search goal

The architect would like to compare the three technology families regarding their suitability to the described scenario, non-functional requirements and constraints.

Please consider specially finding information about the benefits of the three technologies, which can motivate the architect to select one of them.

Search and determine the relevance and the type of architectural knowledge of the resulted web pages from Google, which could support the architect fulfilling his request.