ToiletPaper #86

SMACK Stack

Author: Andreas Scharf / Senior Software Architect / Business Division New Business

Definition SMACK stack:

The SMACK stack is a collection of technologies composed to build a resilient and distributed data processing architecture to enable real-time data-analysis and fast deployment. The acronym **SMACK** stands for the **Spark** engine, the **Mesos** manager, the **Akka** toolkit and runtime, the **Cassandra** database and the **Kafka** message broker. All components except for Akka are Apache projects. The software is open source and production-proven at scale. By using this loosely coupled toolchain of technologies, it's possible to create a private cloud platform to handle large amounts of data and avoid cloud vendor lock-in at the same time.

✓ Spark

Spark is an engine for large-scale data processing and makes it easy to build parallel applications or batch jobs. An advantage over Hadoop MapReduce is a superior performance. Users have the ability to query structured data using Spark SQL. Furthermore, Spark Streaming addresses real-time use cases: Incoming data is chunked into micro batches and processed separately.

✓ Mesos

Mesos is a scheduling framework to manage clusters. It provides resources for applications, services, and jobs and abstracts the underlying hardware. The workload is distributed across the cluster. The distributed operating system DC/OS, built on top of Mesos, simplifies deployment and scaling of containerized applications.

✓ Akka

Akka is an implementation of the Actor Model and allows creating message-driven applications in Scala or Java. The toolkit is inspired by the language Erlang and its fault-tolerance is due to actor based concurrency. Actors can modify local state but won't expose it. Instead, they use asynchronous messages to interact with other actors.

✓ Cassandra

Cassandra is a NoSQL database. It is a wide column store and can be seen as a two-dimensional key-value store. Cassandra's query operations are limited in order to guarantee performance and linear, horizontal scaling. There is no single point of failure and it's possible to span a Cassandra cluster across data centers.

✓ Kafka

Kafka is a distributed messaging system which is well-known for low latency and high availability and throughput. Kafka is using a shared commit log internally. It can replay the log which is its unique selling point. Producers publish messages to publish-subscribe message queues. Kafka partitions the data within a topic. Partitions can be distributed to cluster nodes and consumers receive the requested messages.