Sample Application: A system to track and display the Top 10 best-selling items on an e-commerce website

Requirements

* Millions of End-users, generating thousands of queries per second
  + it MUST be fast - page latency is important
  + So, we need some distributed NoSQL solution
  + Access pattern is simple: “Give me the current top N sellers in category X”
* Hourly updates probably good enough (consistency not hugely important)
* Must be highly available (customers don’t like broken websites)
* So - we want to partition-tolerance and availability more than consistency

Sounds like Cassandra or Amazon DynamoDB

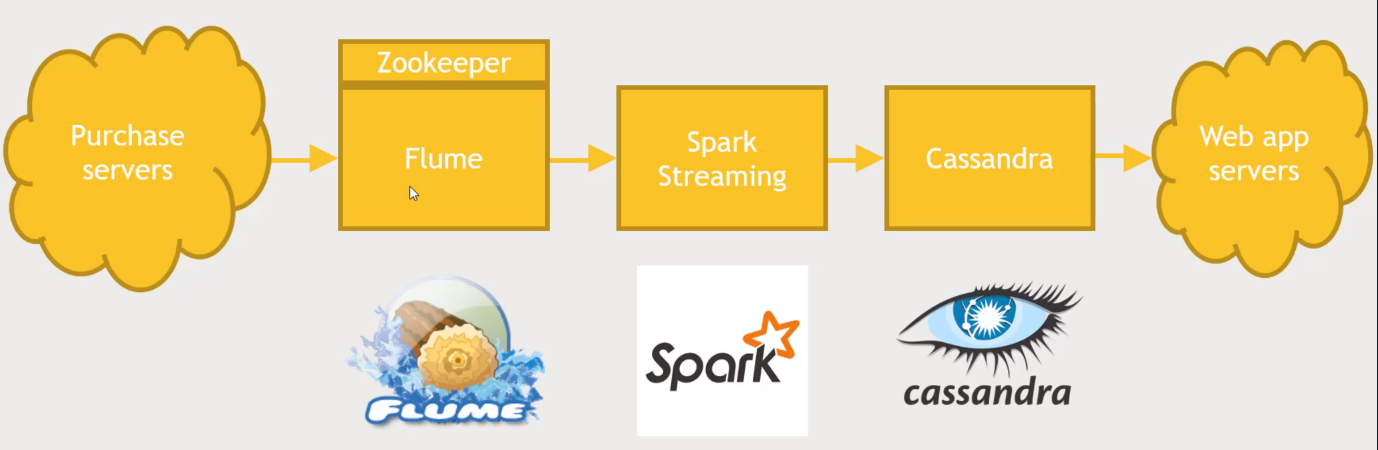
How can data get into Cassandra?

* Spark can talk to Cassandra
* And Spark Streaming can add things up over windows.

How data get into Spark Streaming?

* Kafka or Flume - either works
* Flume is purpose-built for HDFS, which so far we haven’t said we need
* But Flume is also purpose-built for log ingestion, so it may be a good choice
  + Log4j interceptor on the servers that process purchases?

Security

* Purchase data is sensitive - get a security review
  + PII
  + strip out data you don’t need at the source
* Security considerations may even force you into a totally different design
  + Instead of ingesting logs as they are generated, some intermediate database or publisher may be involved

