Die Lambda Architektur als Grundmuster für Big Data Projekte

Lambda Architecture: A blueprint for Big Data Applications

Karl-Heinz Sylla, Dr. Michael Mock
Fraunhofer IAIS
53757 Sankt Augustin
karl-heinz.sylla@iais.fraunhofer.de
michael.mock@iais.fraunhofer.de

GI Workshop
Architekturen 2016
Hildesheim 23.6.2016
Big Data Projects – Big Data Components

"Big Data"-topic is the cause to
• understand, select, apply and experience analytic algorithms
• learn the features and the impacts of new technology
• start pilot projects for proof of concept

Select
• Requirements, Ideas, Tasks
• Data
• Tools
• Big Data Components

Understand Big Data Technology
Setup a justifiable, useful application architecture
Lambda Architecture

Batch view = Function(All Data)
Realtime view = Function(Realtime view, New data)
Query = Function(Batch view, Realtime view)

Lambda Architecture

Nathan Marz with James Warren
Big Data: Principles and best practices of scalable real-time data systems
Manning 2015
Answer queries consistently and fast by using precomputed views

**Master dataset**

**Transformation** → **precomputed Batch view** → **Query**

**Batch view** = Function(All Data)

**Query** = Function(Batch view)

**Information**: General collection of knowledge relevant for the Big Data System.

**Data**: Information that can't be derived from anything else.

**Query**: Questions that are asked about the data.

**Views**: Information that has been derived from the base data. Views help to **answer questions fast and consistently**

The master dataset holds base data-records that are:

- **raw**
- **immutable**
- **timestamped and eternally true**
Information available for queries has high latency

- **Master dataset**
- **Batch view**
- **Query**

Batch transformation takes time

Views are out of date

Included within Batch View

- Not included

Included within Batch View

- under process by batch workflow
- new

Time now
Bridge the latency gap of batch processing by immediate processing of incoming data.

New Data \rightarrow Stream Processor \rightarrow Realtime view #1 \rightarrow Realtime view #2

Information that become available in Batch view may be removed from Realtime view.

Stream processing may compute an approximation of exact values, that are computed in batch runs.

"Eventual Accuracy"
Avoid complexity in core components
"Complexity Isolation" into the Real-time view

Create / Read - append only
Event Sourcing
Immutable
Scalable
Simple

Master dataset

Batch view

Realtime view

Query

New data

Random write/update/read access
is very complex for scalable data stores.
(eventually consistent – vector clocks …)
Components in a Big Data "Lambda" Architecture

- **Master dataset**
- **Batch function**
- **Batch view**
- **Speed function**
- **Realtime view**

New Data
- log messages
- sensor data
- process data

Message passing

merge
query
Component Selection / Horizontal Scalability

- > volume
- > parallelization
- > users, volume

Master dataset

Batch function

Batch view

Batch function

New Data

Speed function

Realtime view

Apache Kafka

Cassandra

log messages
sensor data
process data

> volume, velocity

> velocity

> users
ETL Extract-Transform-Load + flexible Analysis

- **Master Dataset**
- **Batch Function**
- **Speed Function**
- **Realtime View**
- **Batch View**
- **In Memory / RDBMS**

New Data
- log messages
- sensor data
- process data

message passing

In Memory / RDBMS

process  query
Reporting

Master dataset → Batch function → report 2016-01-12
report 2016-01-13
report 2016-01-14
...

message passing

New Data

log messages
sensor data
process data

report shipment
Monitoring

New Data
- log messages
- sensor data
- process data

Speed function

message passing

RDBMS

control and analysis panel
Model learning and realtime model-application

- **Master dataset**
  - Batch function
  - (predictive) models
  - Realtime view
  - Monitor and control dashboard

New Data
- log messages
- sensor data
- process data

Message passing
New Data acquisition, filtering, cleaning ... and ingestion is stream processing
Lambda Architecture or Stream Processing

Massive Data Streams Create Bottlenecks
Anonymized CDR Data from HT (Croatia Telekom)

Fraud Rules provided by HT

Standard Approach: CDR data is imported into separate database where fraud rules are applied

=> import delays analysis

Challenge: Apply Rules to Data Streams

=> faster detection of fraud

=> Express fraud rules with Complex Event Patterns!
In-Situ and In-Stream Processing

Use Cases:
- Fraud Mining
- Health-Monitoring

Express a "global condition" but monitor by local conditions
Complex Event Processing

streaming with higher level abstractions

Event Producers → Complex Event Processing System → Event Consumers


Event Processing in Action, O. Etzion and P. Niblett, Manning 2010
Complex Event Processing Principles

Input (simple) Events
- Timestamp
- Event Source
- Attributes

Event Processing Logic
Set of patterns that define conditions on one or multiple events. If events occur that match a pattern, a new event is generated (derived event).

Output (complex, derived) Events
- Timestamp
- (Event Sources)
- Attributes
Input Events and Example Complex Event for Fraud

- Input Event: **CDR** (call detail record) data
  - subscriber number, called number, timestamp, duration, start cell, end cell, ...

- Complex Event 1: **LongCallAtNight**
  A long call to premium distance is made during night hours.

- Complex Event 2: **FrequentLongCallsAtNight**
  At least three of these **LongCallAtNight**-events per calling number

- ...

© Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS
FERARI: towards In-Situ Complex Event Processing

CEP Model
Application
Rules

CEP Optimizer

logical plan

physical plan

cost

event stream
analyzer

Site Configurations

runtime statistics

Fraud Detection Dashboard

Realtime view

runtime statistics

Push

real-time input streams

Fraud Detection Dashboard

Pull

CEP Model
Application Rules

CEP Optimizer

logical plan

physical plan

cost

event stream analyzer

Site Configurations

runtime statistics

Fraud Detection Dashboard

Realtime view

runtime statistics

Push

real-time input streams

Fraud Detection Dashboard

Pull
Proton on Storm

- Proton - IBM Proactive Technology Online
  - research asset developed by IBM Research Haifa
    https://github.com/ishkin/Proton
  - Used in many EU Projects (FIWARE, Fispace, Psymbiosis, SPEEDD)
  - Patterns defined by Event Processing Networks (EPN)

- Proton on Storm
  - Developed by IBM in the FERARI EU Project Grant No 619491
    http://www.ferari-project.eu
  - Distributed, scalable CEP on Storm
  - Use case example: fraud mining on telekom data
    https://bitbucket.org/sbothe-iais/ferari
Selected FERARI References

http://www.ferari-project.eu

Open Source Repository
https://bitbucket.org/sbothe-iais/ferari


N. Giatrakos, A. Deligiannakis, M. Garofalakis: Scalable Approximate Query Tracking over Highly Distributed Data Stream, ACM SIGMOD’2016

wrap up

- Big Data - Proof of concept - Experimental applications
- The Lambda Architecture
  - Principles of data processing
  - Architecture template to recognize Big Data issues
  - Guide to Component Selection
- Examples of Architecture instantiations
- Streaming and the Internet of things - Industrie 4.0
  - Big Data processing becomes Stream Processing
- In-Situ complex event processing