MA-INF 4223-Lab Distributed Big Data Analytics

SANSA
Scalable Semantic Analytics Stack

23/05/2017
Dr. Hajira Jabeen, Gezim Sejdiu, Prof. Dr. Jens Lehmann
Lesson objectives

After completing this lesson, you should be able to:

- Understand the usage of SANSA
- List and understand the libraries offered by SANSA
SANSA: Motivation

- Abundant machine readable structured information is available (e.g. in RDF)
  - Across SCs, e.g. Life Science Data
  - General: DBpedia, Google knowledge graph
  - Social graphs: Facebook, Twitter

- Need for scalable querying, inference and machine learning
  - Link prediction
  - Knowledge base completion
  - Predictive analytics
SANSA: Motivation

- Over the last years, the size of the Semantic Web has increased and several large-scale datasets were published.
- Now days hadoop ecosystem has become a standard for BigData applications.
- We use this infrastructure for Semantic Web as well.
A New Vision Combining **Semantic Technologies** and **Distributed Machine Learning**

**Scalable Semantic Analytics Stack (SANSA)**

- Machine Learning
- Inference
- Querying
- Knowledge Distribution & Representation
- Distributed In-Memory Processing

**Distributed Analytics**

- Machine Learning Libraries
- In-Memory Computing Framework
- Distributed Filesystem

**Semantic Technology Stack**

- Query: SPARQL
- Ontology: OWL, RDFS
- Rule: RIF
- Data interchange: RDF
- XML
- URI/IRI

- manual data integration
- often simple input formats
- data formats often not standardized
- measurable benefits
- horizontal scalability

- powerful data integration
- expressive modelling
- W3C standardised formats
- benefits only indirectly measurable
- usually no horizontal scalability
SANSA Stack

- It’s core is a **processing data flow** engine that provides data distribution, and fault tolerance for distributed computations over RDF large-scale datasets.

- SANSA includes **several libraries** for creating applications:
  - Read / Write RDF / OWL library
  - Querying library
  - Inference library
  - ML- Machine Learning core library

SANSA: Read Write Layer

- Ingest RDF and OWL data in different formats using Jena / OWL API style interfaces
- Represent data in multiple formats (e.g. RDD, Data Frames, GraphX, Tensors)
- Allow transformation among these formats
- Compute dataset statistics and apply functions to URIs, literals, subjects, objects → Distributed LODStats
SANSA: Query Layer
SANSA: Query Layer

- To make generic queries efficient and fast using:
  - Intelligent indexing
  - Splitting strategies
  - Distributed Storage

- SPARQL query engine evaluation
  (SPARQL-to-SQL approaches, Virtual Views)

- Provision of W3C SPARQL compliant endpoint
SANSA: Inference Layer
SANSA: Inference Layer

- W3C Standards for Modelling: RDFS and OWL
- Parallel in-memory inference via rule-based forward chaining
- Beyond state of the art: dynamically build a rule dependency graph for a rule set
- → Adjustable performance levels
SANSA: ML Layer

- Classification
- Link prediction
- Clustering
- Anomaly Detection
- Latent Embeddings

Analytics
- Inference
- Querying

Distribution
- Knowledge Distribution and Representation
- Distributed In-Memory Processing
SANSA: ML Layer

- Distributed Machine Learning (ML) algorithms that work on RDF data and make use of its structure / semantics

Work in Progress:
- Tensor Factorization for e.g. KB completion (testing stage)
- Simple spatiotemporal analytics (idea stage)
- Graph Clustering (testing stage)
- Association rule mining (evaluation stage)
- Semantic Decision trees (idea stage)
- Inference in Knowledge Graph Embeddings (idea stage)
SANSA = Scala / Maven Repositories based on Spark / Flink
Easy to include both in BDE platform and any Spark / Flink environment
December: **First Release**
- Realise **simple scenarios** e.g. read RDF/OWL files, compute statistics, simple queries, lightweight inference, graph clustering, rule mining
- It is **not** very stable / mature

Subsequent releases **every 6 months**
- **SANSA 0.2** release planned in June 2017.
Conclusions and Next steps

◎ A generic stack for (big) Linked Data
  o Build on top of a state-of-the-art distributed frameworks (Spark, Flink)
◎ Out-of-the-box framework for scalable and distributed semantic data analysis combining semantic web and distributed machine learning for (1) querying, (2) inference and (3) analytics of RDF datasets.
◎ Next steps
  o Refinement of data structures (RDF/OWL Layer)
  o Add support for SPARQL 1.1 and other backend strategies (Query Layer)
  o Define a ML pipelines for Structured ML (ML Layer).
References


[2].