RDF$_{pro}$
Processing Billions of RDF Triples on a Single Machine using Streaming and Sorting

Francesco Corcoglioniti, Marco Rospocher, Marco Amadori, Michele Mostarda

Fondazione Bruno Kessler-IRST
Trento, Italy

http://rdfpro.fbk.eu

SAC2015
Salamanca, 14 April 2015

The problem

FONDAZIONE BRUNO KESSLER

Are relevant RDF processing tasks on large datasets practically feasible on a single commodity machine by using streaming and sorting techniques?

The problem

FONDAZIONE BRUNO KESSLER

The problem

FONDAZIONE BRUNO KESSLER

- perform relevant RDF processing tasks
  - TBox and statistics extraction
  - data filtering
  - data transformation
  - inference materialisation
  - smushing
  - ...

The problem

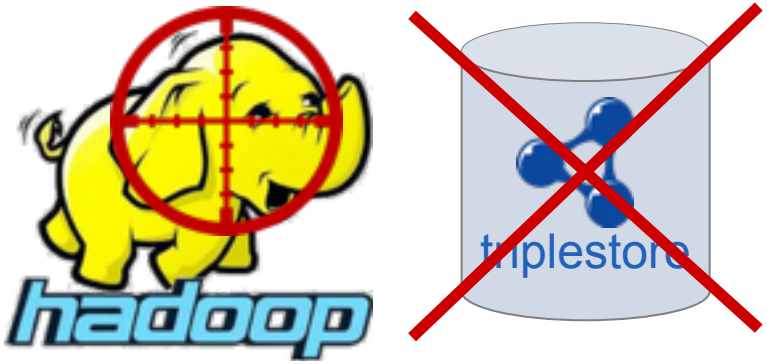
FONDAZIONE BRUNO KESSLER

- perform relevant RDF processing tasks
  - TBox and statistics extraction
  - data filtering
  - data transformation
  - inference materialisation
  - smushing
  - ...
- on large datasets
  - LOD-sized: billions of triples
  - quads, not just triples

The problem

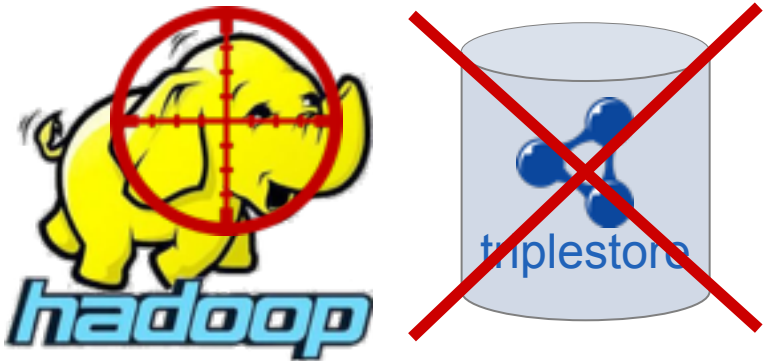
FONDAZIONE BRUNO KESSLER

- perform relevant RDF processing tasks
  - TBox and statistics extraction
  - data filtering
  - data transformation
  - inference materialisation
  - smushing
  - ...
- on large datasets
  - LOD-sized: billions of triples
  - quads, not just triples
- on a single commodity machine
  - no cluster / distributed computing
  - no triplestore or other data index



The problem

- perform relevant RDF processing tasks
  - TBox and statistics extraction
  - data filtering
  - data transformation
  - inference materialisation
  - smushing
  - ...
- on large datasets
  - LOD-sized: billions of triples
  - quads, not just triples
- on a single commodity machine
  - no cluster / distributed computing
  - no triplestore or other data index



- using streaming and sorting
  - data processing primitives managing large amounts of data with constrained resources

Our Contributions

FONDAZIONE BRUNO KESSLER

- RDF_{pro}: an extensible tool for building RDF processing pipelines based on streaming and sorting

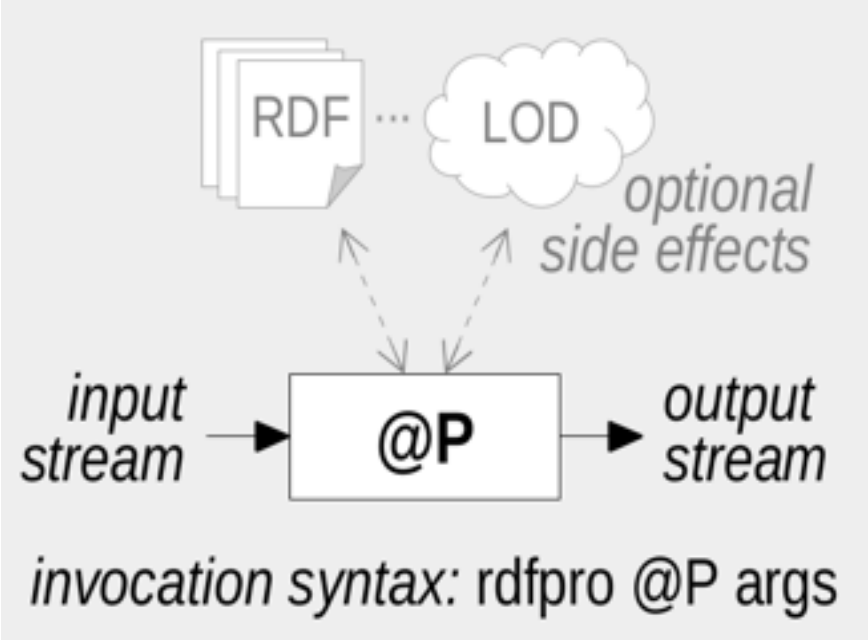
- Empirical Evaluation on 4 usage scenarios, positively answering our research question

FONDAZIONE
BRUNO KESSLER

RDF pro

http://rdfpro.fbk.eu

RDF$_{pro}$ at its core: RDF processor



- Based on Streaming:
  - quads from the input stream are processed one at a time
  - multiple passes can be performed
  - may have an internal state / side effects (e.g., writing)

RDF$_{pro}$: sorting

FONDAZIONE BRUNO KESSLER

- offered to processors as a primitive to arbitrarily sort selected data during a pass
  - implemented via external sorting (unix sort + smart data encoding)
  - effectively exploits available hardware resources
- enables tasks not feasible with streaming alone:
  - duplicates removal
  - set operations
  - any task that need to group together scattered information

RDF$_{pro}$: on-board RDF processors

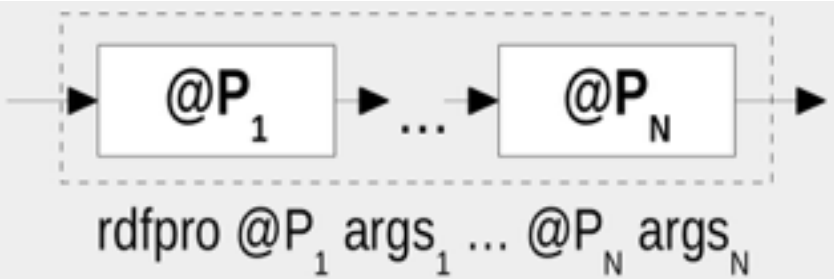
FONDAZIONE BRUNO KESSLER

- move data around
  - @read / @write files
  - @download from / @upload to SPARQL endpoints
- transform data
  - arbitrary data @transform while streaming on triples (via Groovy scripts)
  - @infer the RDFS closure
  - @smush data, merging owl:sameAs URIs into canonical URIs
  - extract @tbox and VOID @stats
  - @unique discards duplicates

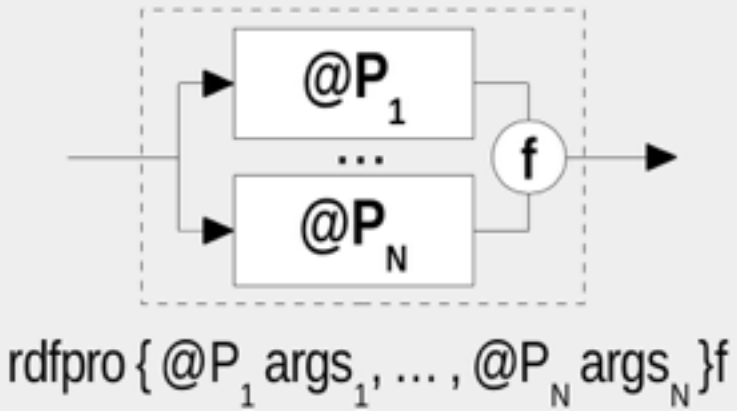
RDF$_{pro}$: processor composition

FONDAZIONE BRUNO KESSLER

- processors can be derived by (recursively) applying sequential



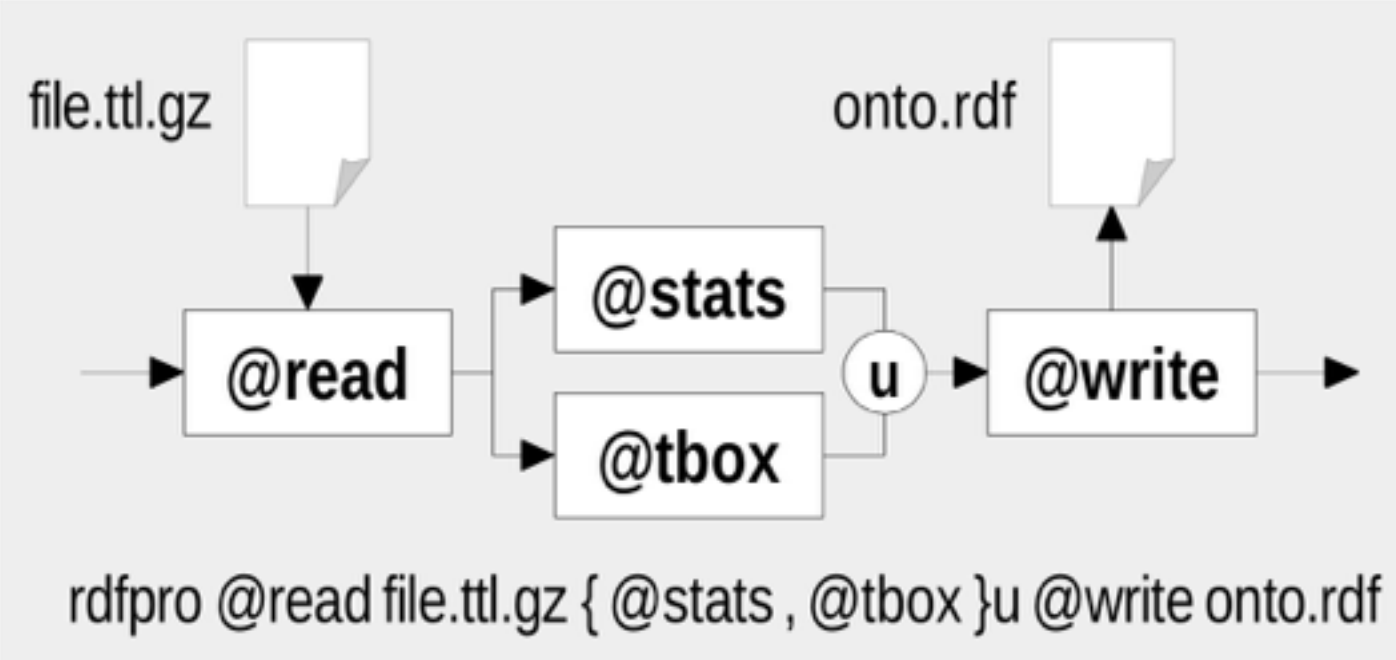
and parallel compositions



RDF$_{pro}$: processor composition

FONDAZIONE BRUNO KESSLER

Example



- read a Turtle+gzip file (file.ttl.gz)
- TBox and VOID statistics are extracted in parallel
- union written to an RDF/XML file (onto.rdf)

RDF$_{pro}$: further details

FONDAZIONE BRUNO KESSLER

- Offered as:
  - Java command line tool
  - embeddable Java library
- Built using a multi-thread design to fully exploit CPU resources
- Built on top of Sesame RDF library
- Extendable with new processors

- Web-site: http://rdfpro.fbk.eu/
- Code
  - available at: https://github.com/dkmfbk/rdfpro
  - CC0 license

FONDAZIONE
BRUNO KESSLER

Empirical Evaluation

4 usage scenarios

Commodity machine used in all the scenarios:
Intel Core I7 860 CPU (4 cores, hyper-threading)
16 GB RAM
500 GB 7200 RPM hard disk
Linux 2.6.32

Scenario 1: Dataset Analysis

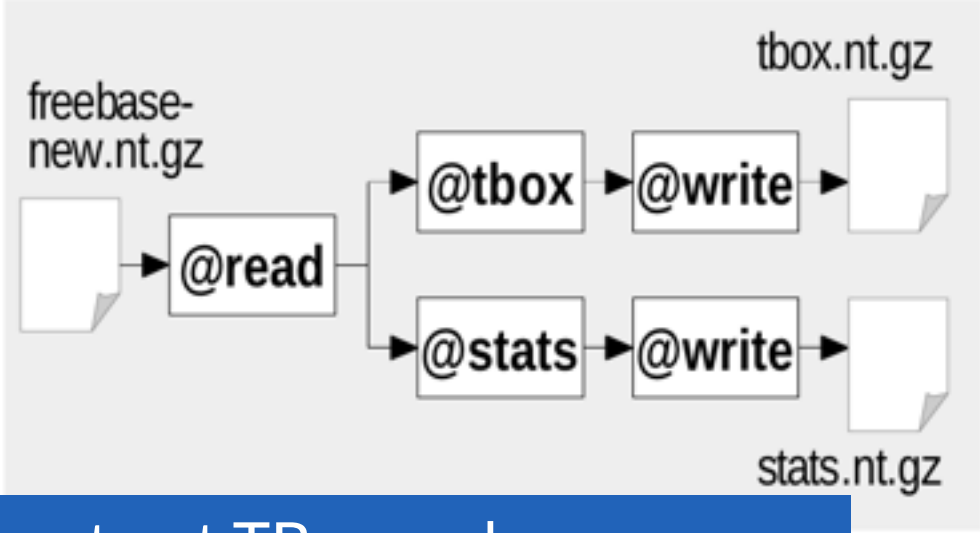
FONDAZIONE BRUNO KESSLER

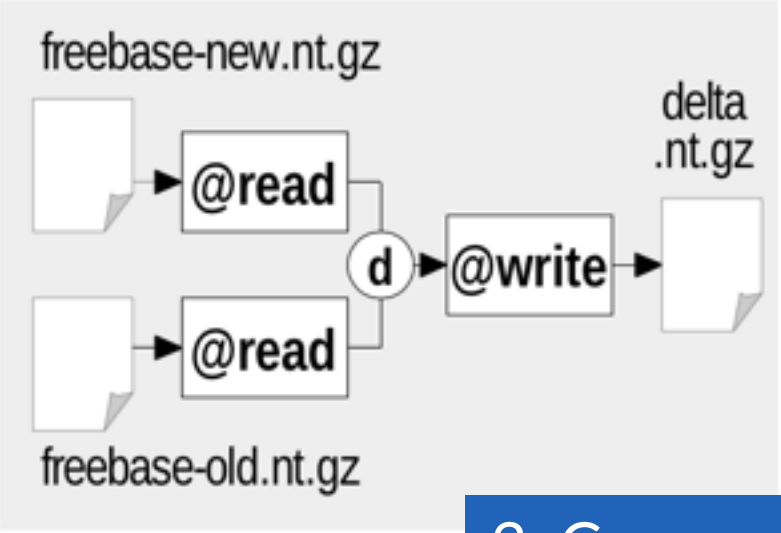
- TASK: provide a qualitative and quantitative characterisation of the contents of an RDF dataset (e.g., extract TBox or compute ABox data statistics)
  - to identify relevant data, pre-processing needs
  - to characterise a dataset for validation / documentation

- EXPERIMENT: extract TBox and statistics from a version of Freebase
  - 2014/09/10 dump, 2863 millions of quads (MQ)

  and compare it with an older version
  - 2014/07/10 dump, 2623 MQ

Scenario 1: Dataset Analysis



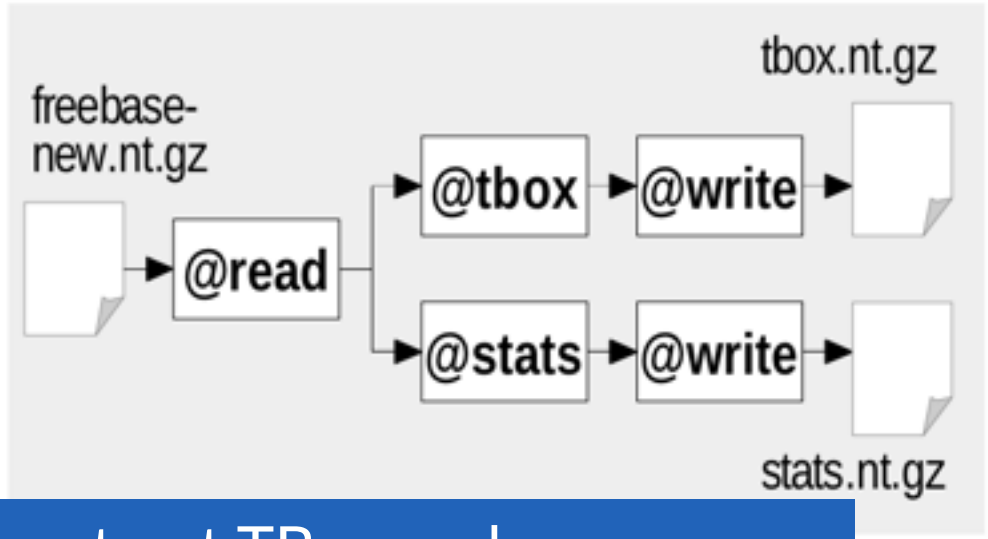


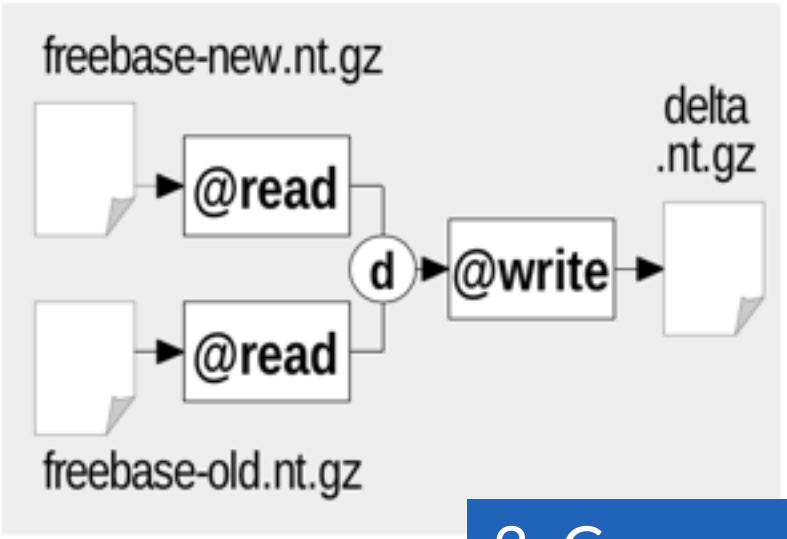
1. extract TBox and
2. compute ABox data Statistics

3. Compare datasets

Task	Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]
1. TBox	2863	28339	0.23	3.01	1.43	14.12	2006
2. Statistics	2863	28339	0.13	1.36	0.34	3.36	8443
1-2 Aggregated	2863	28339	0.36	4.35	0.34	3.36	8426
3. Comparison	5486	55093	260	1894	0.42	4.25	12955

Scenario 1: Dataset Analysis



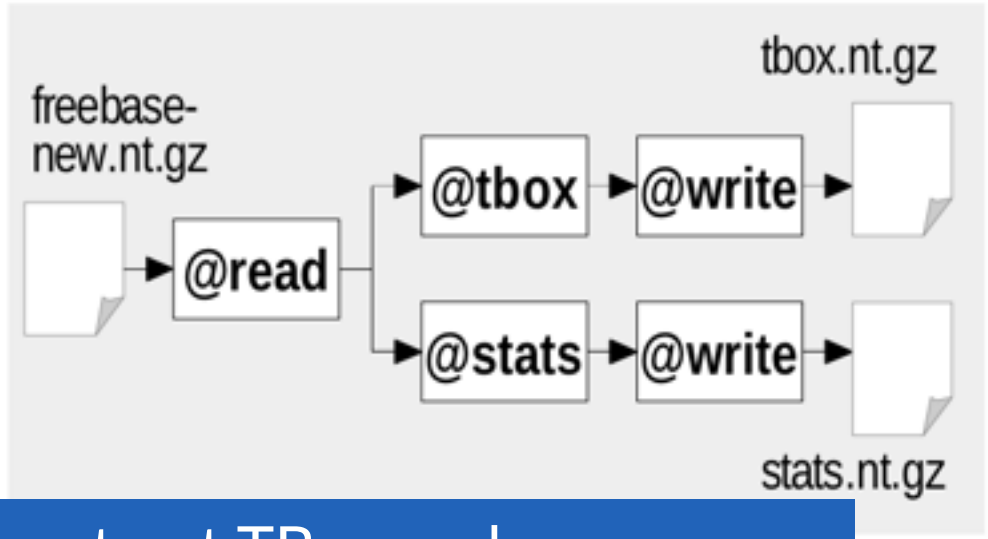


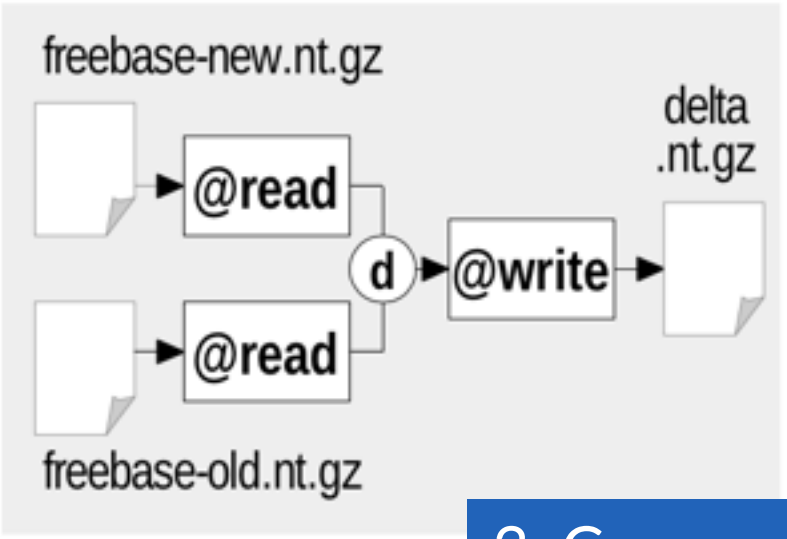
1. extract TBox and
2. compute ABox data Statistics

3. Compare datasets

Task	Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]
1. TBox	2863	28339	0.23	3.01	1.43	14.12	2006
2. Statistics	2863	28339	0.13	1.36	0.34	3.36	8443
1-2 Aggregated	2863	28339	0.36	4.35	0.34	3.36	8426
3. Comparison	5486	55093	260	1894	0.42	4.25	12955

Scenario 1: Dataset Analysis



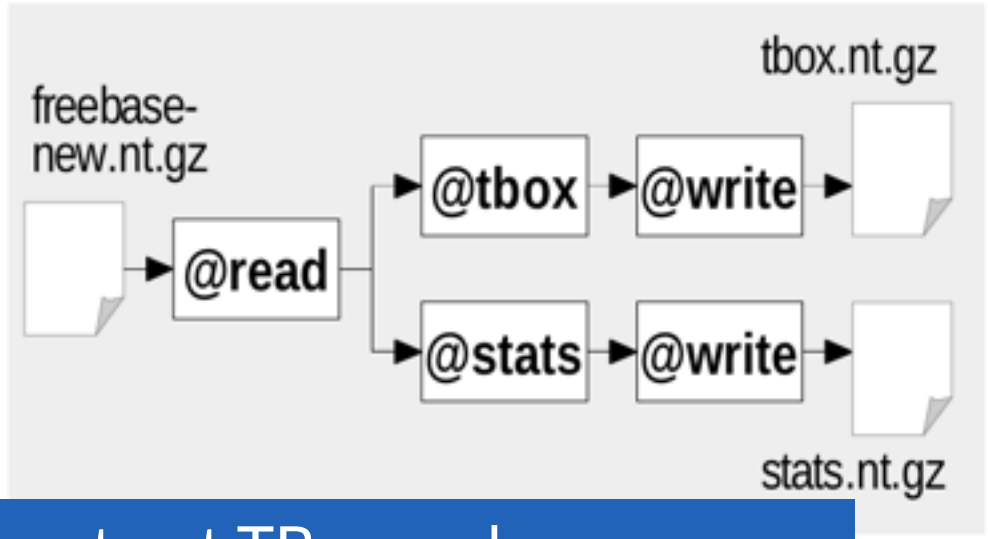


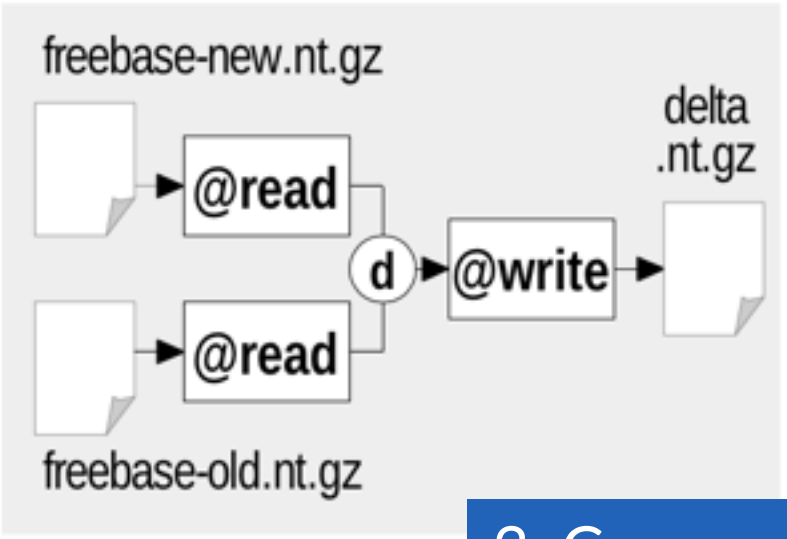
1. extract TBox and
2. compute ABox data Statistics

3. Compare datasets

Task	Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]
1. TBox	2863	28339	0.23	3.01	1.43	14.12	2006
2. Statistics	2863	28339	0.13	1.36	0.34	3.36	8443
1-2 Aggregated	2863	28339	0.36	4.35	0.34	3.36	8426
3. Comparison	5486	55093	260	1894	0.42	4.25	12955

Scenario 1: Dataset Analysis





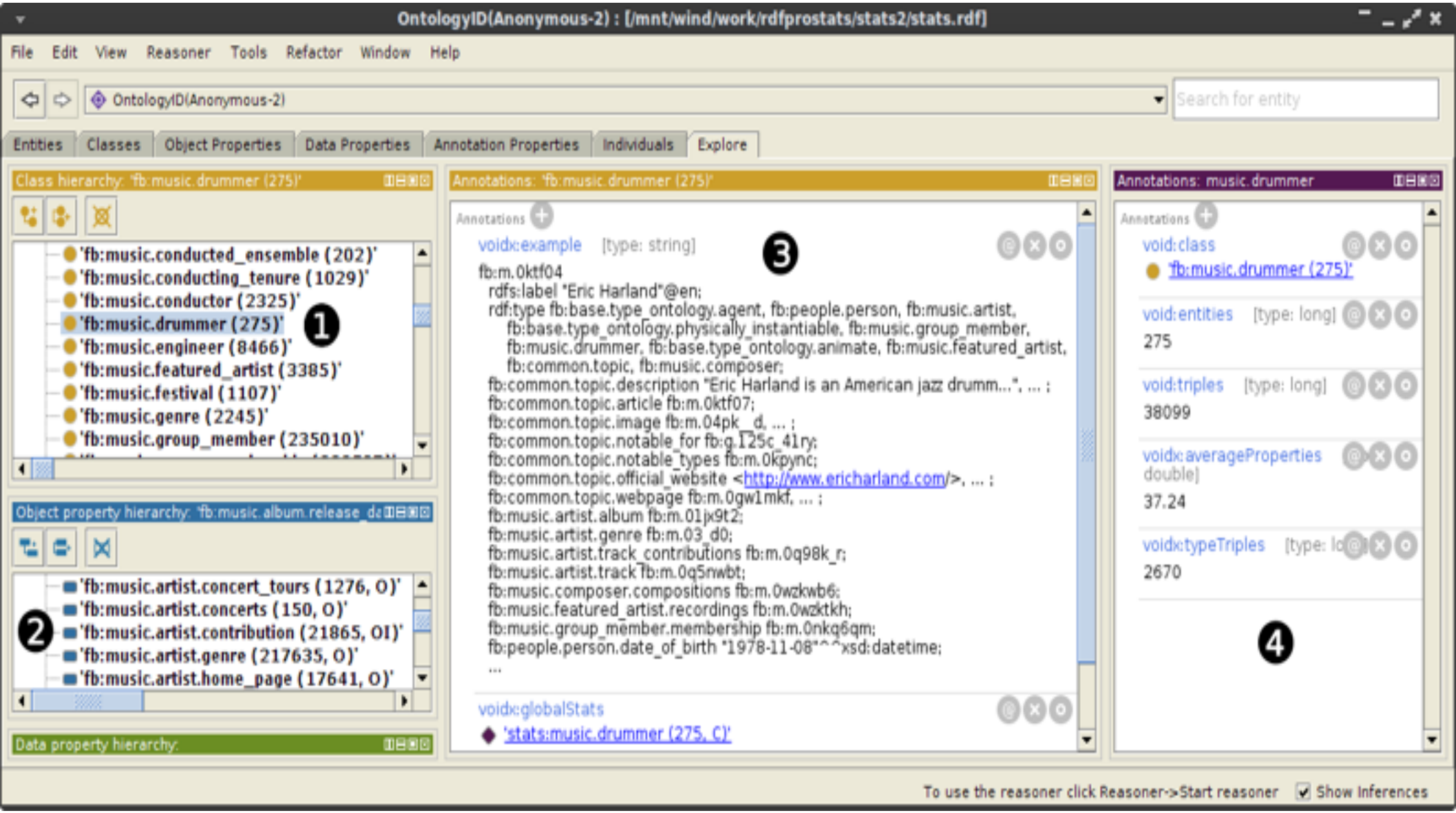
1. extract TBox and
2. compute ABox data Statistics

3. Compare datasets

Task	Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]
1. TBox	2863	28339	0.23	3.01	1.43	14.12	2006
2. Statistics	2863	28339	0.13	1.36	0.34	3.36	8443
1-2 Aggregated	2863	28339	0.36	4.35	0.34	3.36	8426
3. Comparison	5486	55093	260	1894	0.42	4.25	12955

Scenario 1: Dataset Analysis

FONDAZIONE BRUNO KESSLER



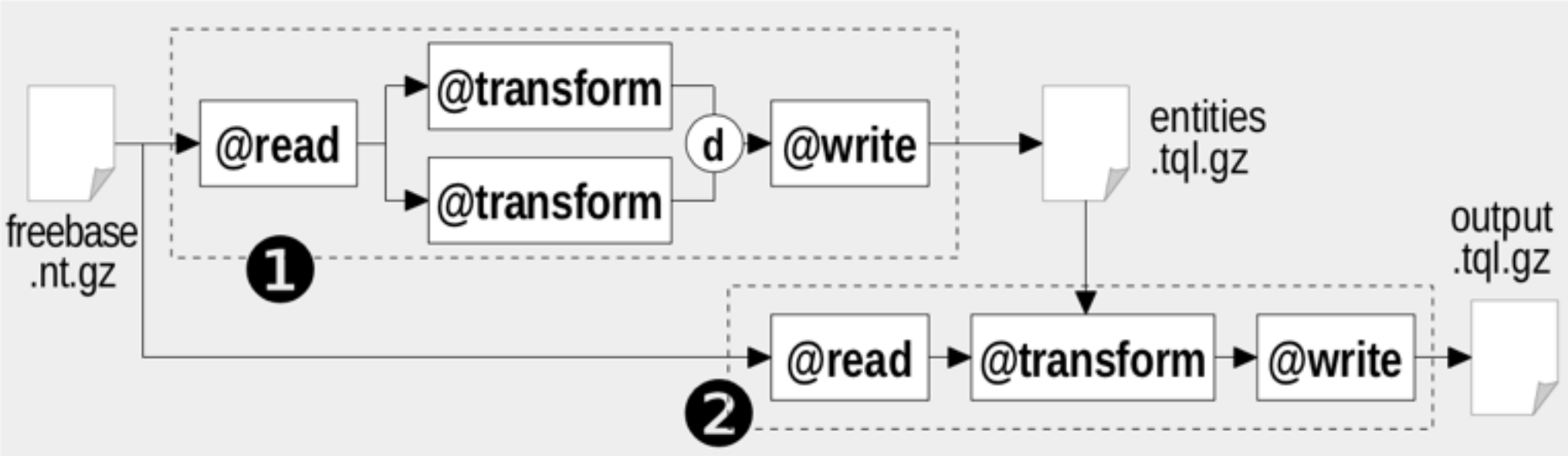
Scenario 2: Dataset Filtering

FONDAZIONE BRUNO KESSLER

- TASK: extract a subset of data, by
  1. identifying the entities of interest in the dataset (selection conditions on their URIs, rdf:type or other properties)
  2. extracting selected quads about these entities

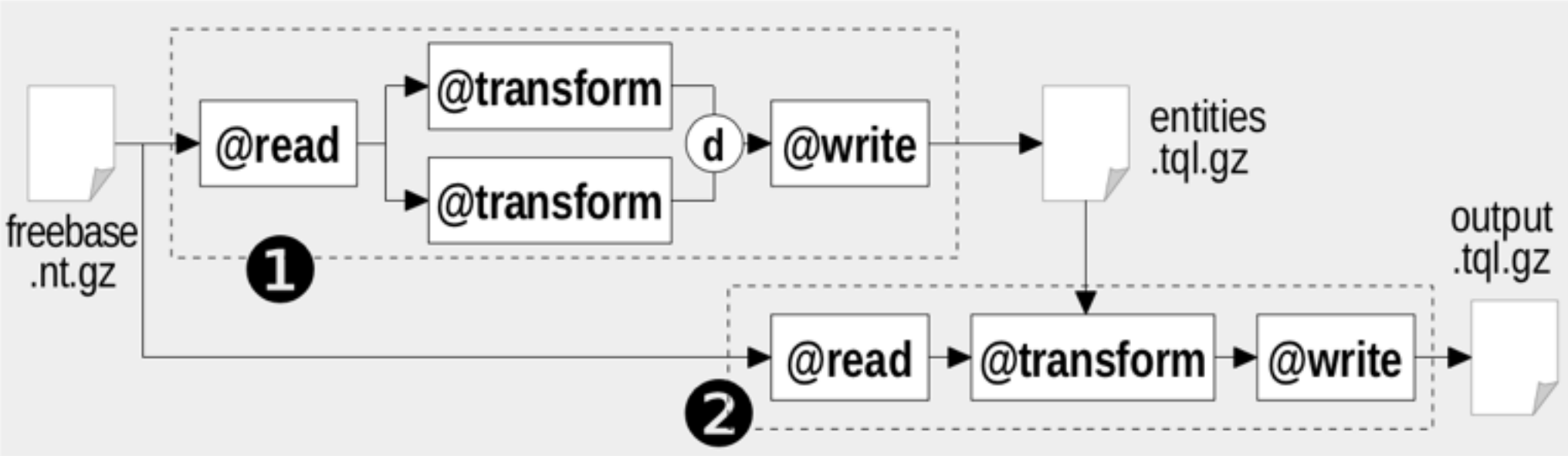
- EXPERIMENT: extract from Freebase (2014/07/10, 2863 MQ):
  - **entities of interest: musical groups** (rdf:type = fb:music.musical_group) **that are still active (having no** fb:music.artist.active_end **triples)**
  - **properties to extract: group name** (rdfs:label), **genre** (fb:music.artist.genre) **and place of origin** (fb:music.artist.origin)

Scenario 2: Dataset Filtering



Task	Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]
1 Select entities	2863	28339	0.20	0.73	1.36	13.4	2111
2 Extract quads	2863	28339	0.42	5.17	1.15	11.4	2481

Scenario 2: Dataset Filtering



Task	Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]
1 Select entities	2863	28339	0.20	0.73	1.36	13.4	2111
2 Extract quads	2863	28339	0.42	5.17	1.15	11.4	2481

Scenario 3: Dataset Merging

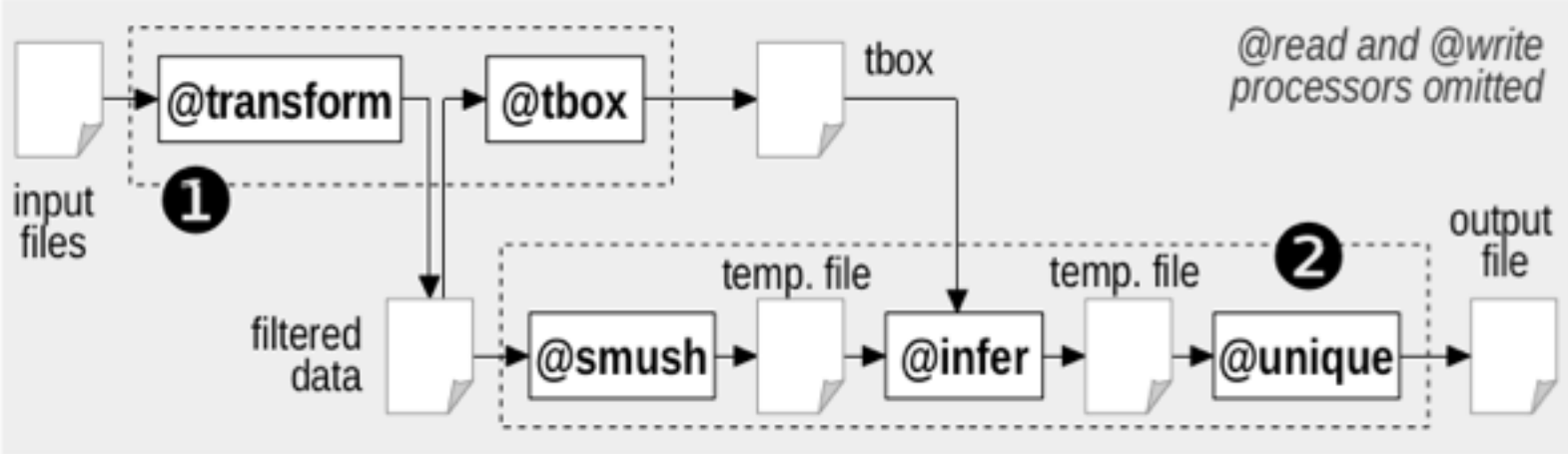
FONDAZIONE BRUNO KESSLER

- TASK: multiple RDF datasets are integrated and prepared for application consumption
  - comprises tasks such as smushing, inference materialization and data deduplication

- EXPERIMENT: merging of
  - Freebase (2014/07/10, 2863 MQ)
  - GeoNames (2013/08/27, 125 MQ)
  - 4 DBpedia subsets (EN, ES, IT, NL - version 3.9, 406 MQ)

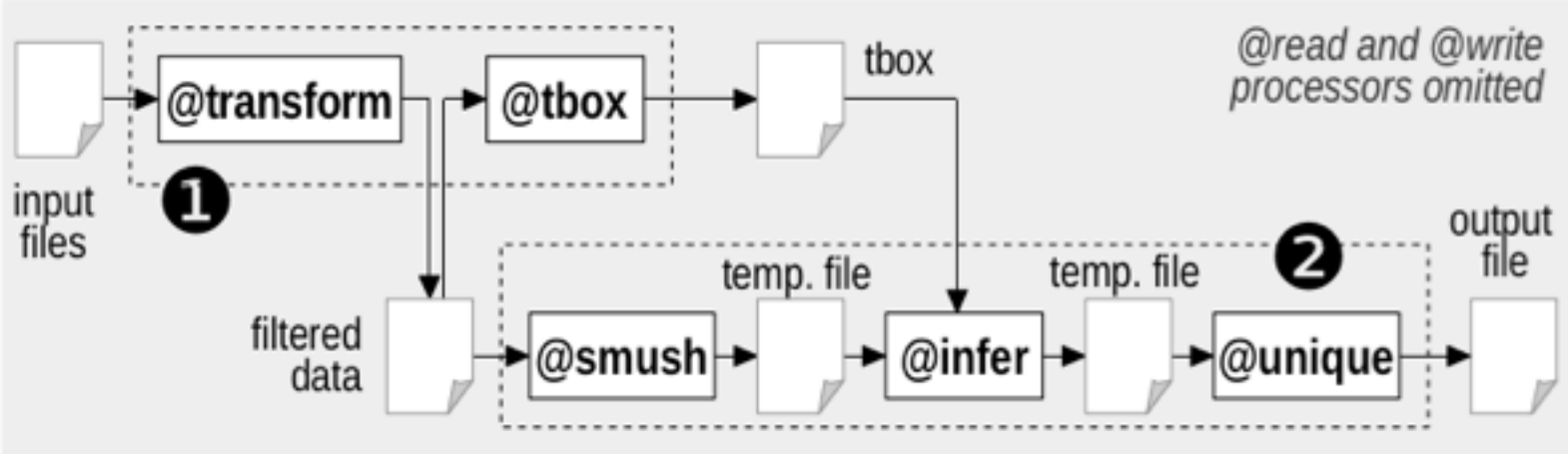
  Total: 3394 MQ

Scenario 3: Dataset Merging



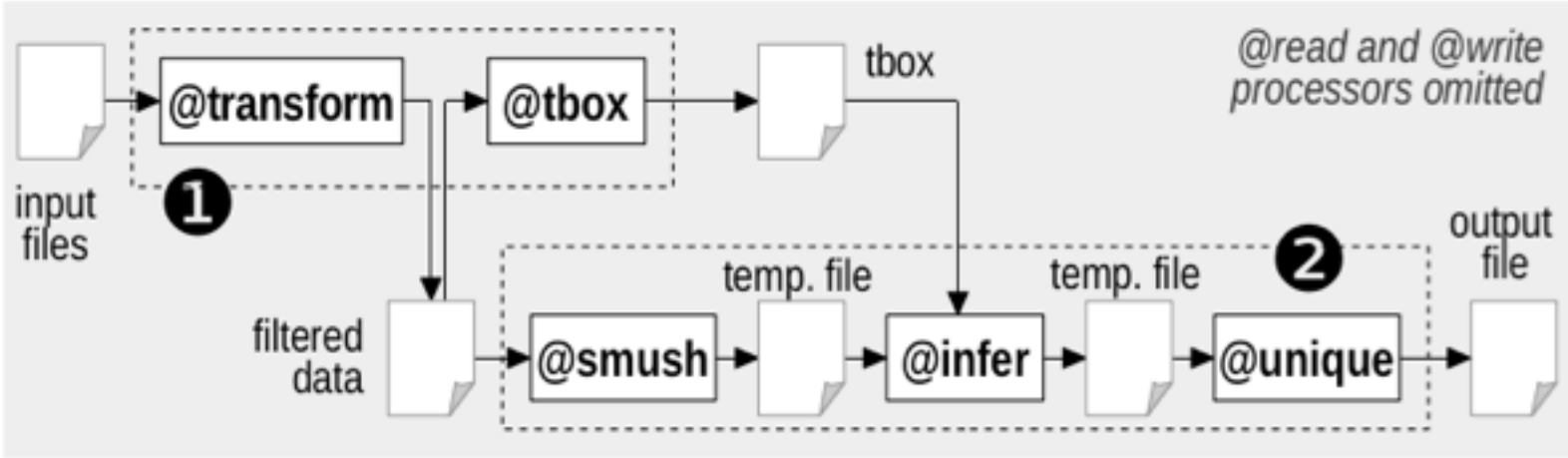
Step		Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]
	@transform	3394	33524	3394	36903	0.42	4.12	8137
	@tbox	3394	36903	<1	4	1.28	13.9	2656
	@smush	3394	36903	3424	38823	0.37	3.98	9265
	@infer	3424	38823	5615	51927	0.32	3.66	10612
	@unique	5615	51927	4085	31297	0.33	3.03	17133
1	Aggregated	3394	33524	3394	36903	0.41	4.06	8247
2	Aggregated	3394	36903	4085	31446	0.14	1.56	23734

Scenario 3: Dataset Merging



Step	Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]
@transform	3394	33524	3394	36903	0.42	4.12	8137
@tbox	3394	36903	<1	4	1.28	13.9	2656
@smush	3394	36903	3424	38823	0.37	3.98	9265
@infer	3424	38823	5615	51927	0.32	3.66	10612
@unique	5615	51927	4085	31297	0.33	3.03	17133
1 Aggregated	3394	33524	3394	36903	0.41	4.06	8247
2 Aggregated	3394	36903	4085	31446	0.14	1.56	23734

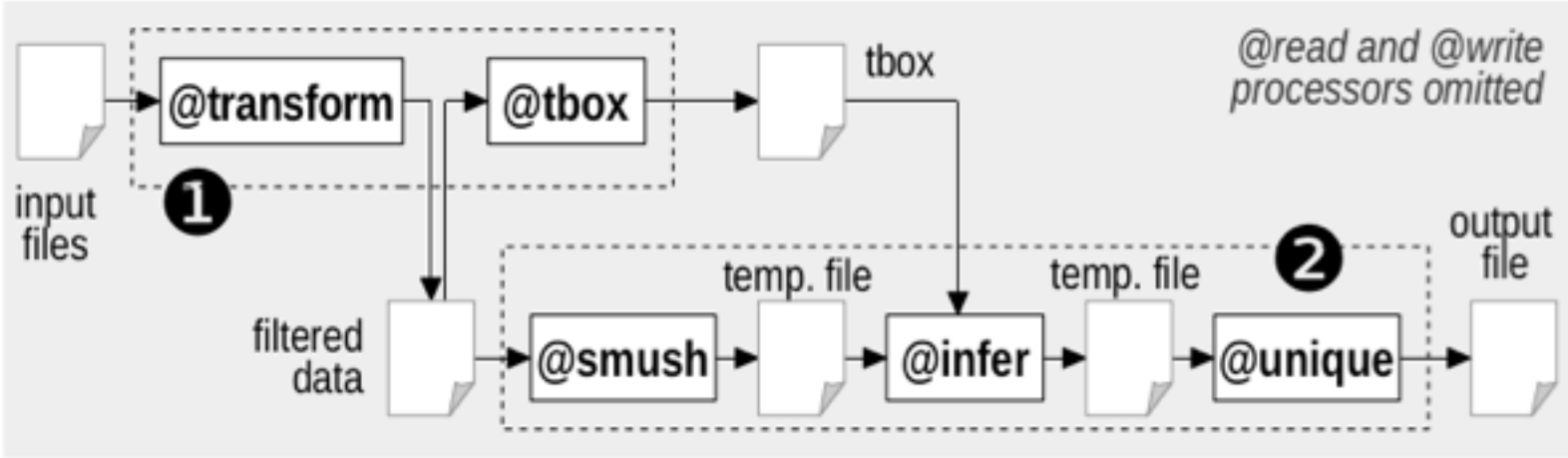
Scenario 3: Dataset Merging



Step	Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]
@transform	3394	33524	3394	36903	0.42	4.12	8137
@tbox	3394	36903	<1	4	1.28	13.9	2656
@smush	3394	36903	3424	38823	0.37	3.98	9265
@infer	3424	38823	5615	51927	0.32	3.66	10612
@unique	5615	51927	4085	31297	0.33	3.03	17133
1 Aggregated	3394	33524	3394	36903	0.41	4.06	8247
2 Aggregated	3394	36903	4085	31446	0.14	1.56	23734

-24%

Scenario 3: Dataset Merging



Step	Input [MQ]	Input [MB]	Output [MQ]	Output [MB]	Throughput [MQ/s]	Throughput [MB/s]	Time [s]	
@transform	3394	33524	3394	36903	0.42	4.12	8137	
@tbox	3394	36903	<1	4	1.28	13.9	2656	
@smush	3394	36903	3424	38823	0.37	3.98	9265	
@infer	3424	38823	5615	51927	0.32	3.66	10612	
@unique	5615	51927	4085	31297	0.33	3.03	17133	
1 Aggregated	3394	33524	3394	36903	0.41	4.06	8247	-24%
2 Aggregated	3394	36903	4085	31446	0.14	1.56	23734	-36%

Scenario 4: Dataset Massaging

FONDAZIONE BRUNO KESSLER

- TASK: ad-hoc transformations necessary to make data better suited to a particular use
  - data repackaging: preserve data content, but affect the way data is packaged (e.g., changing of RDF syntax)
  - data sanitization: fixing or removing the RDF terms or quads that prevent any further processing of data (e.g., conversion of datatype, URI rewriting, normalisation of literals)
  - data derivation: augmenting a dataset with quads computed from original data (e.g., conversion of a numeric value, counting the occurrences of a certain property for an entity)
- typically implemented in RDFpro using @read, @write and @transform in a single pass without sorting (~0.45 MQ/s)

Evaluation Re-cap

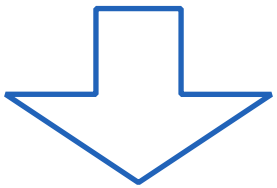
FONDAZIONE BRUNO KESSLER

- RDF_{pro} implementation of the processing tasks succeeds in managing billions of quads / RDF triples on a commodity machine
- execution times are in the order of hours
  - processing times are negligible if compared to load times in SOA triple stores
    - Virtuoso 7, on same machine, 9h08m for loading 1B triples
  - definitely a winner in one-time processing

Evaluation Re-cap

FONDAZIONE BRUNO KESSLER

- RDF_{pro} implementation of the processing tasks succeeds in managing billions of quads / RDF triples on a commodity machine
- execution times are in the order of hours
  - processing times are negligible if compared to load times in SOA triple stores
    - Virtuoso 7, on same machine, 9h08m for loading 1B triples
  - definitely a winner in one-time processing



Positively answer our research question!

Conclusions: RDF$_{pro}$ …

FONDAZIONE BRUNO KESSLER

- … shows that RDF processing tasks on billions of quads can be performed on a single machine using streaming and sorting
- … a "swiss-army-knife" for exploring and manipulating RDF datasets
- … is actively used in the NewsReader EU project
- … is open-source released under the terms of CC0
- … potentially extendable (future work) to implement restricted versions of OWL 2 inference, SPARQL query answering and SPARQL-based data massaging

RDF$_{pro}$

Thank you! Questions?

Marco Rospocher
rospocher@fbk.eu :: @marcorospocher
Fondazione Bruno Kessler
Trento, Italy