

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

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

The problem

The problem

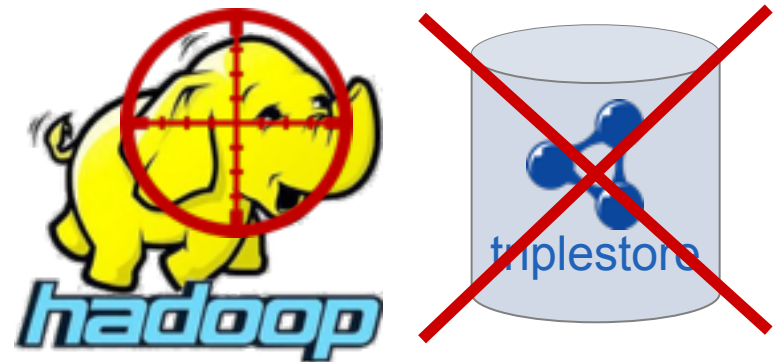
- perform relevant RDF processing tasks
 - TBox and statistics extraction
 - data filtering
 - data transformation
 - inference materialisation
 - smushing
 - ...

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

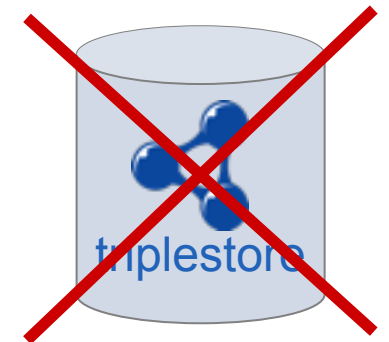
The problem

- 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



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

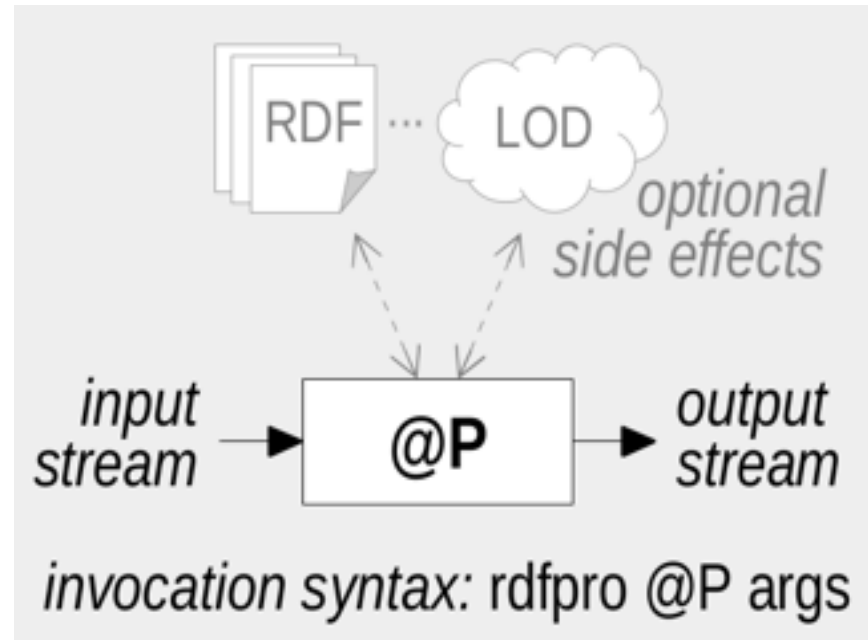


Our Contributions

- **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

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

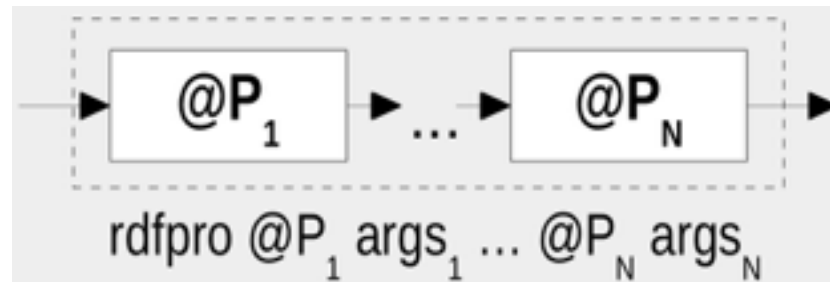
- offered to processors as a **primitive** to arbitrarily sort selected data during a pass
 - implemented via **external sorting** (unix sort + smart data encoding)
 - **effectively exploits** available hardware resources
- **enables tasks** not feasible with streaming alone:
 - duplicates removal
 -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

RDF_{pro}: on-board RDF processors

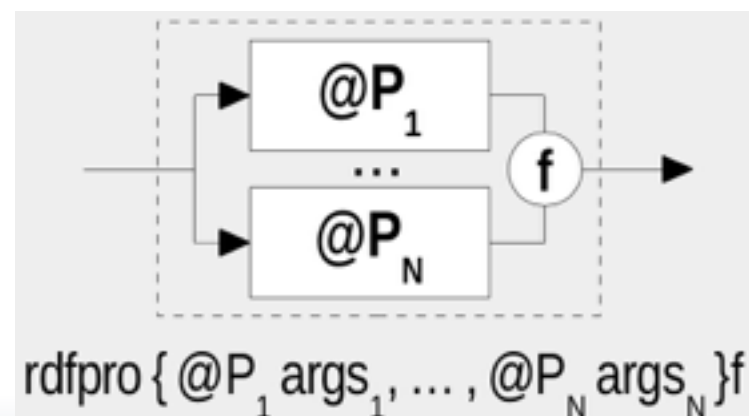
- **move data around**
 - @read / @write **files**
 - @download **from** / @upload to SPARQL endpoints
- **transform data**
 - arbitrary data @transform while streaming on triples (via Groovy scripts)
 - @infer the RDFS closure
 - @smush data, merging owl:sameAs URIs into canonical URIs
 - extract @tbox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

RDF_{pro}: processor composition

- processors can be derived by (recursively) applying **sequential**

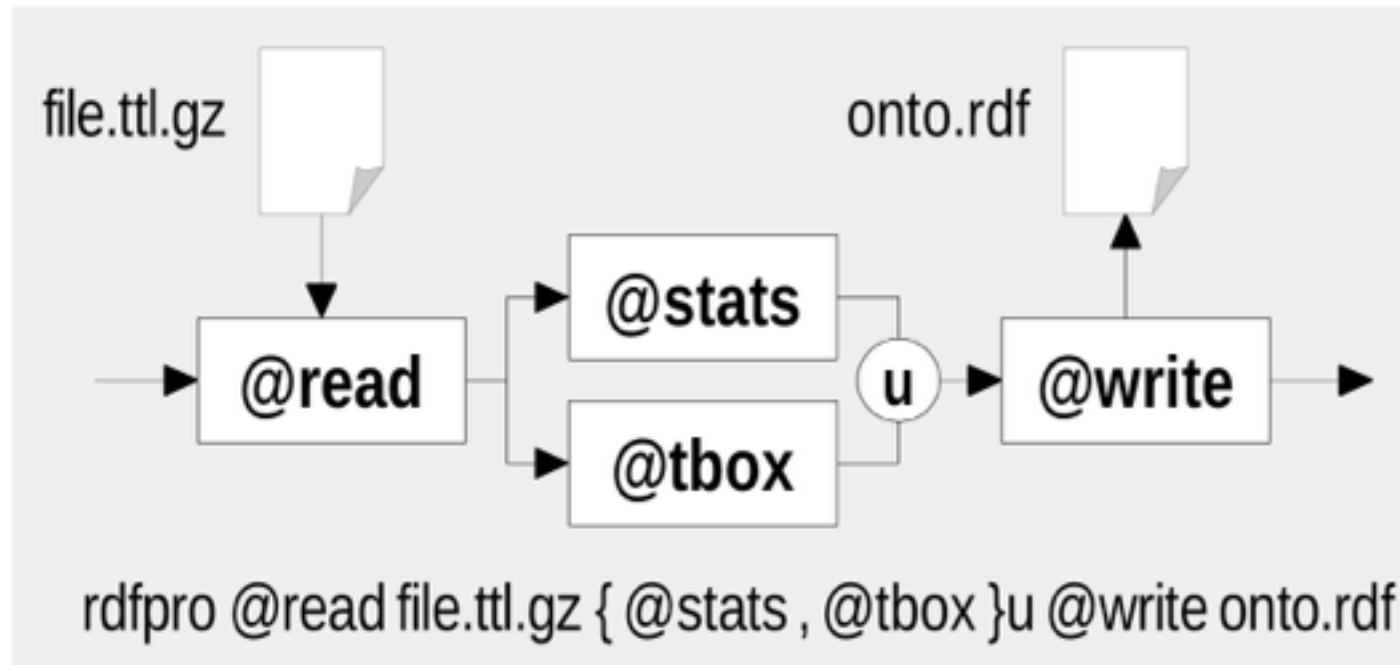


and **parallel** compositions



RDF_{pro}: processor composition

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

- 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

Empirical Evaluation

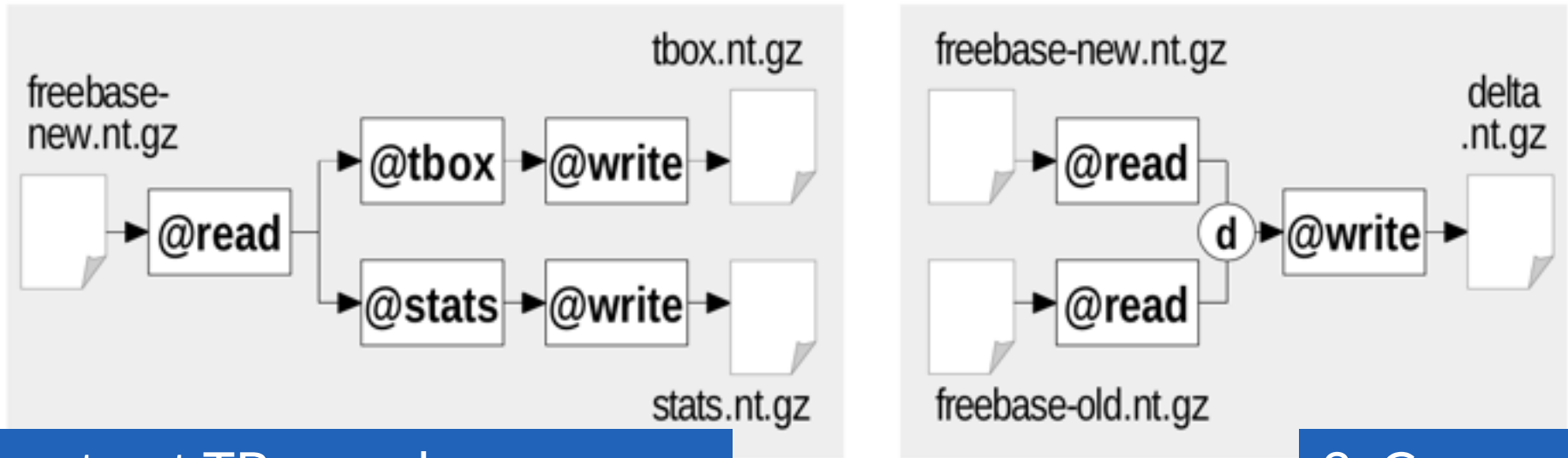
4 usage scenarios

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

- 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

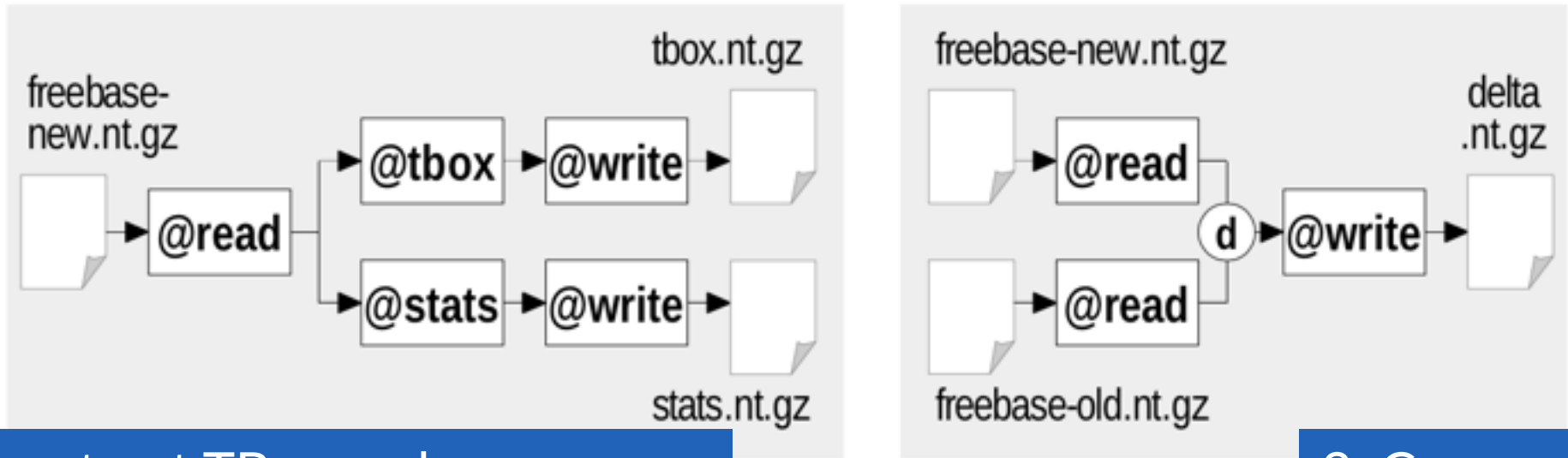


1. extract TBox and
2. compute ABox data Statistics

3. Compare
datasets

Task	Input		Output		Throughput		Time [s]
	[MQ]	[MB]	[MQ]	[MB]	[MQ/s]	[MB/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

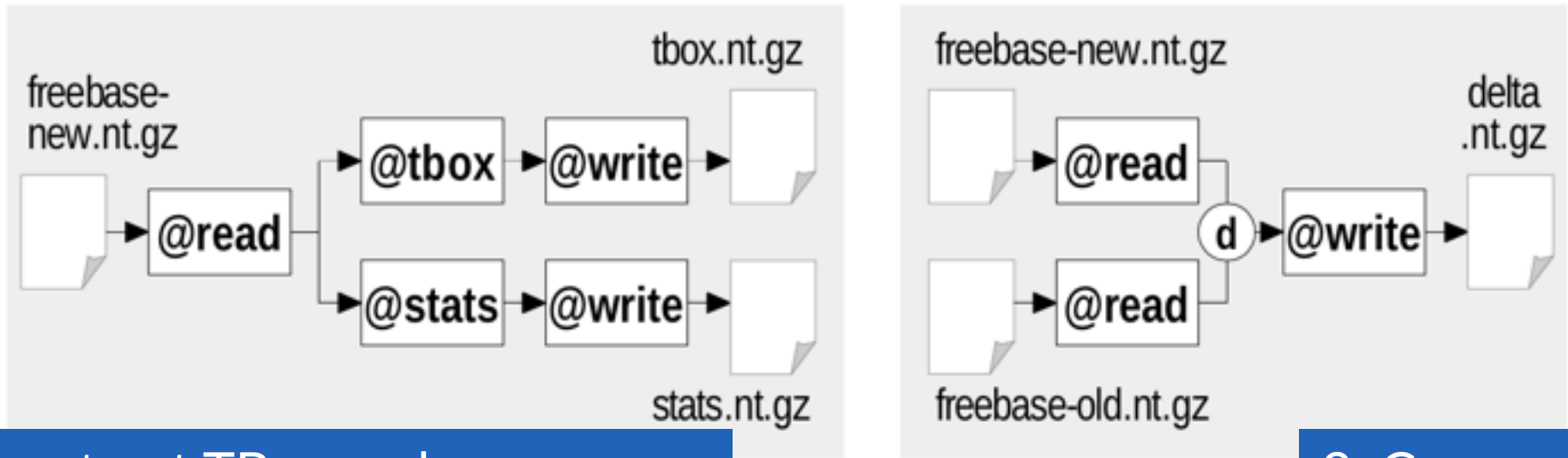


1. extract TBox and
2. compute ABox data Statistics

3. Compare
datasets

Task	Input		Output		Throughput		Time [s]
	[MQ]	[MB]	[MQ]	[MB]	[MQ/s]	[MB/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

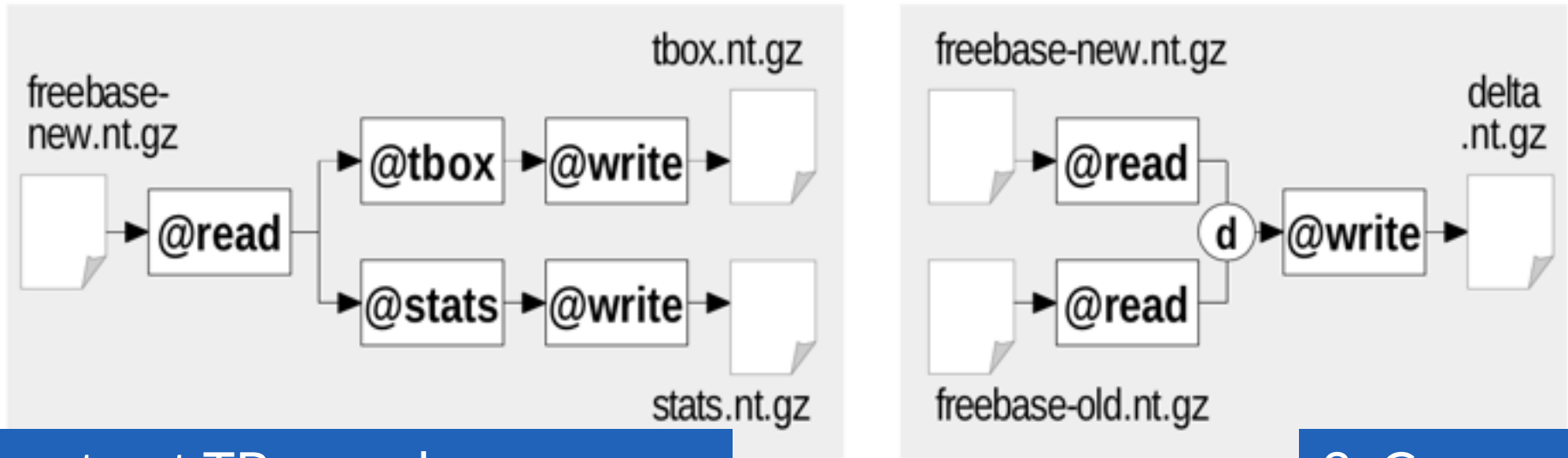


1. extract TBox and
2. compute ABox data Statistics

3. Compare
datasets

Task	Input		Output		Throughput		Time [s]
	[MQ]	[MB]	[MQ]	[MB]	[MQ/s]	[MB/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		Output		Throughput		Time [s]
	[MQ]	[MB]	[MQ]	[MB]	[MQ/s]	[MB/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

OntologyID(Anonymous-2) : [/mnt/wind/work/rdfprostats/stats2/stats.rdf]

File Edit View Reasoner Tools Refactor Window Help

OntologyID(Anonymous-2) Search for entity

Entities Classes Object Properties Data Properties Annotation Properties Individuals Explore

Class hierarchy: 'fb:music.drummer (275)'

- 'fb:music.conducted_ensemble (202)'
- 'fb:music.conducting_tenure (1029)'
- 'fb:music.conductor (2325)'
- 'fb:music.drummer (275)' **1**
- 'fb:music.engineer (8466)'
- 'fb:music.featured_artist (3385)'
- 'fb:music.festival (1107)'
- 'fb:music.genre (2245)'
- 'fb:music.group_member (235010)'

Object property hierarchy: 'fb:music.album.release_date'

- 'fb:music.artist.concert_tours (1276, 0)'
- 'fb:music.artist.concerts (150, 0)'
- 'fb:music.artist.contribution (21865, 01)'
- 'fb:music.artist.genre (217635, 0)'
- 'fb:music.artist.home_page (17641, 0)'

Data property hierarchy:

Annotations: 'fb:music.drummer (275)'

Annotations +

void:example [type: string] **3**

fb:m.0ktf04
 rdfs:label "Eric Harland"@en;
 rdf:type fb:base.type_ontology.agent, fb:people.person, fb:music.artist,
 fb:base.type_ontology.physically_instantiable, fb:music.group_member,
 fb:music.drummer, fb:base.type_ontology.animate, fb:music.featured_artist,
 fb:common.topic, fb:music.composer;
 fb:common.topic.description "Eric Harland is an American jazz drumm...", ... ;
 fb:common.topic.article fb:m.0ktf07;
 fb:common.topic.image fb:m.04pk_d, ... ;
 fb:common.topic.notable_for fb:g.I25c_41ry;
 fb:common.topic.notable_types fb:m.0kpync;
 fb:common.topic.official_website <<http://www.ericharland.com/>>, ... ;
 fb:common.topic.webpage fb:m.0qw1mkf, ... ;
 fb:music.artist.album fb:m.01jx9t2;
 fb:music.artist.genre fb:m.03_d0;
 fb:music.artist.track_contributions fb:m.0q98k_r;
 fb:music.artist.track fb:m.0q5nwbT;
 fb:music.composer.compositions fb:m.0wzkw6;
 fb:music.featured_artist.recordings fb:m.0wzktkh;
 fb:music.group_member.membership fb:m.0nkq6qm;
 fb:people.person.date_of_birth "1978-11-08"^^xsd:datetime;
 ...

void:globalStats

◆ 'stats:music.drummer (275, C)'

Annotations: music.drummer

Annotations +

void:class

- 'fb:music.drummer (275)'

void:entities [type: long]

275

void:triples [type: long]

38099

void:averageProperties [type: double]

37.24

void:typeTriples [type: long]

2670

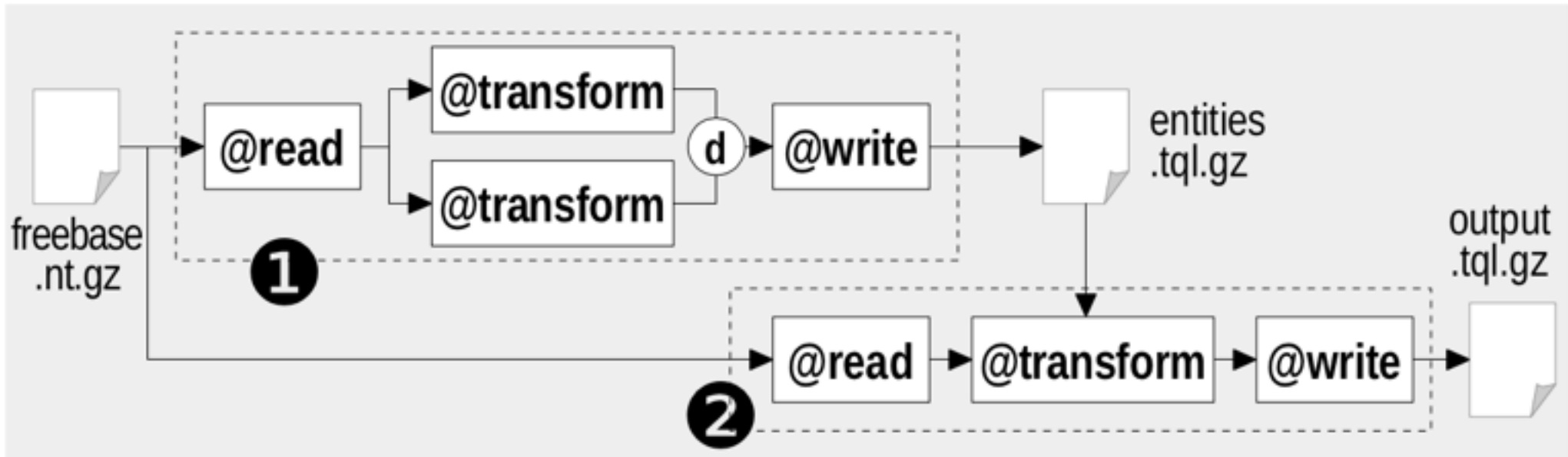
4

To use the reasoner click Reasoner->Start reasoner Show Inferences

Scenario 2: Dataset Filtering

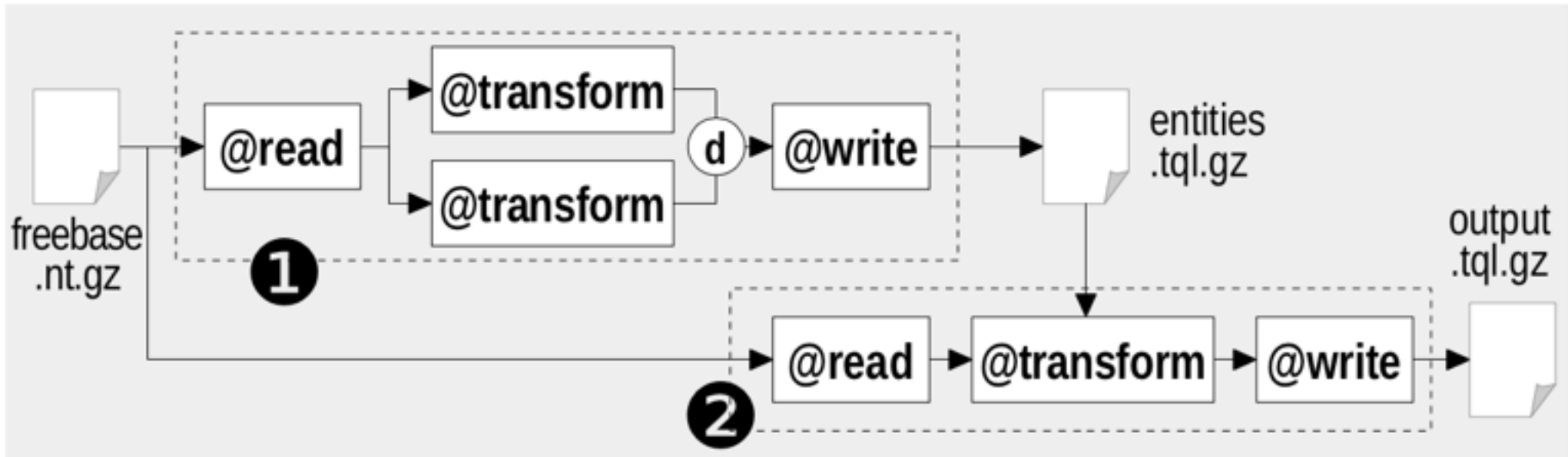
- TASK: **extract** a subset of data, by
 1. **identifying the entities of interest** in the dataset (selection conditions on their URIs, `rdf:type` or other properties)
 2. **extracting selected quads** about these entities
- EXPERIMENT: extract from Freebase (2014/07/10, 2863 MQ):
 - **entities of 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		Output		Throughput		Time [s]
	[MQ]	[MB]	[MQ]	[MB]	[MQ/s]	[MB/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

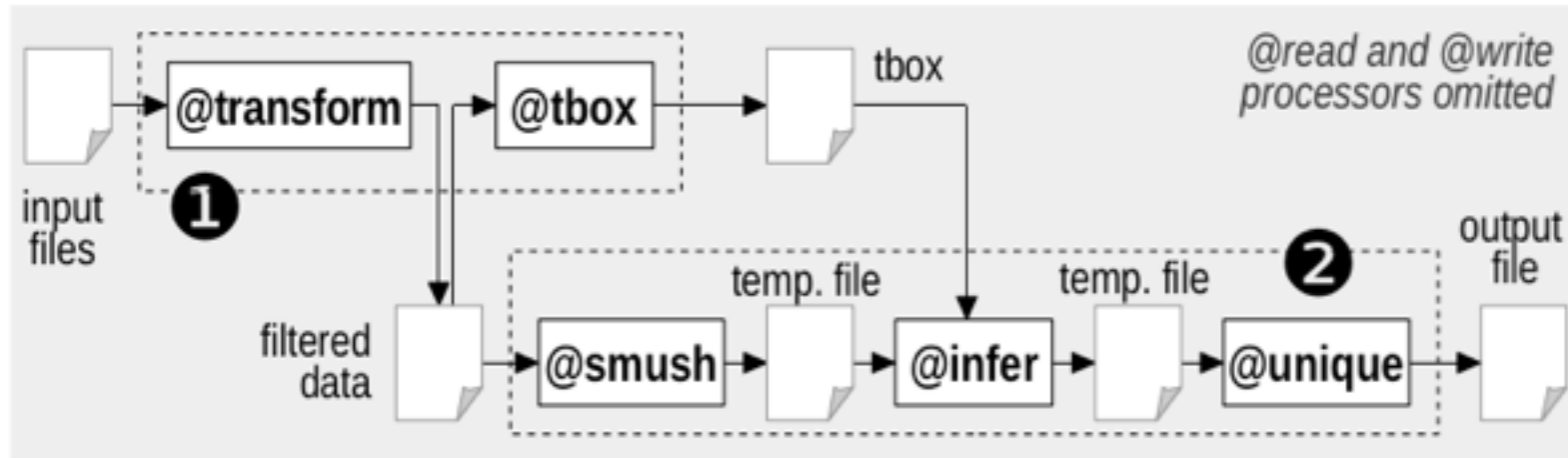


Task	Input		Output		Throughput		Time [s]
	[MQ]	[MB]	[MQ]	[MB]	[MQ/s]	[MB/s]	
1 Select entities	2863	28339	0.20	0.73	1.36	13.4	2111
2 Extract quads	2863	28339	0.42	5.17	1.15	11.4	2481

Scenario 3: Dataset Merging

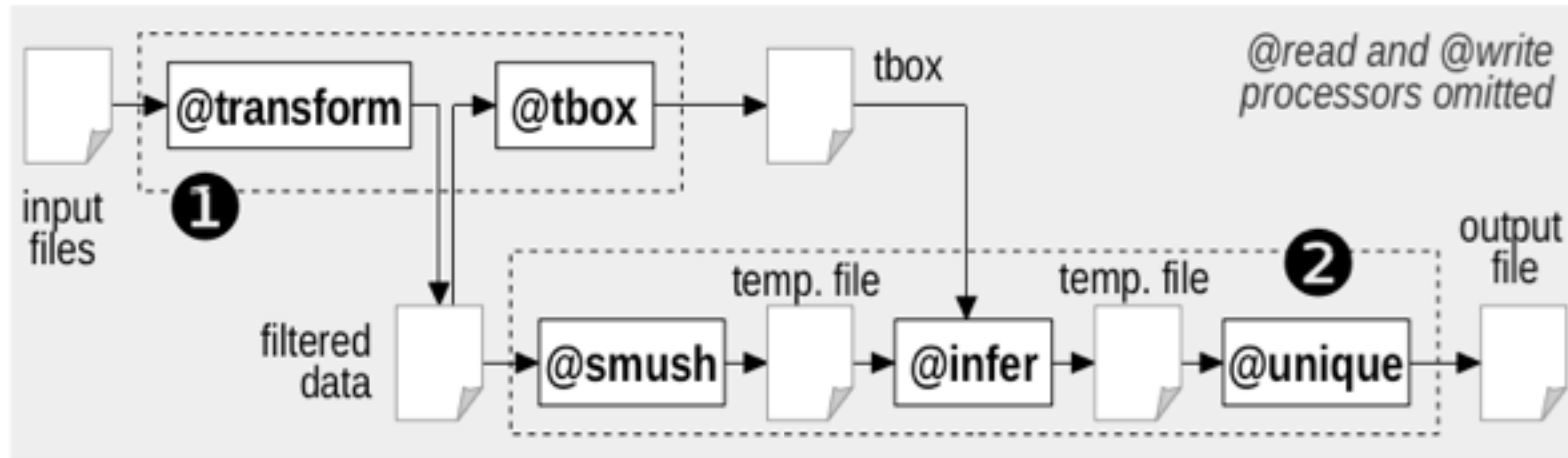
- TASK: multiple RDF datasets are integrated and prepared for application consumption
 - comprises tasks such as smushing, inference materialization and data deduplication
 - EXPERIMENT: merging of
 - Freebase (2014/07/10, 2863 MQ)
 - GeoNames (2013/08/27, 125 MQ)
 - 4 DBpedia subsets (EN, ES, IT, NL - version 3.9, 406 MQ)
- Total: 3394 MQ

Scenario 3: Dataset Merging



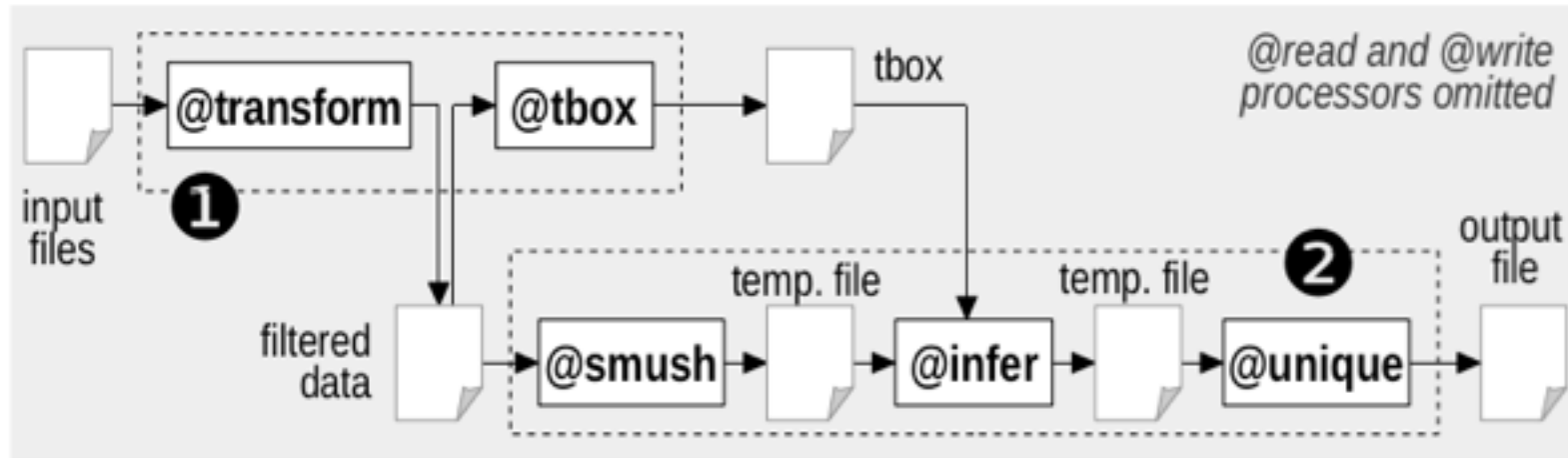
Step	Input		Output		Throughput		Time
	[MQ]	[MB]	[MQ]	[MB]	[MQ/s]	[MB/s]	[s]
@transform	3394	33524	3394	36903	0.42	4.12	8137
@tbox	3394	36903	<1	4	1.28	13.9	2656
@smush	3394	36903	3424	38823	0.37	3.98	9265
@infer	3424	38823	5615	51927	0.32	3.66	10612
@unique	5615	51927	4085	31297	0.33	3.03	17133
1 Aggregated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

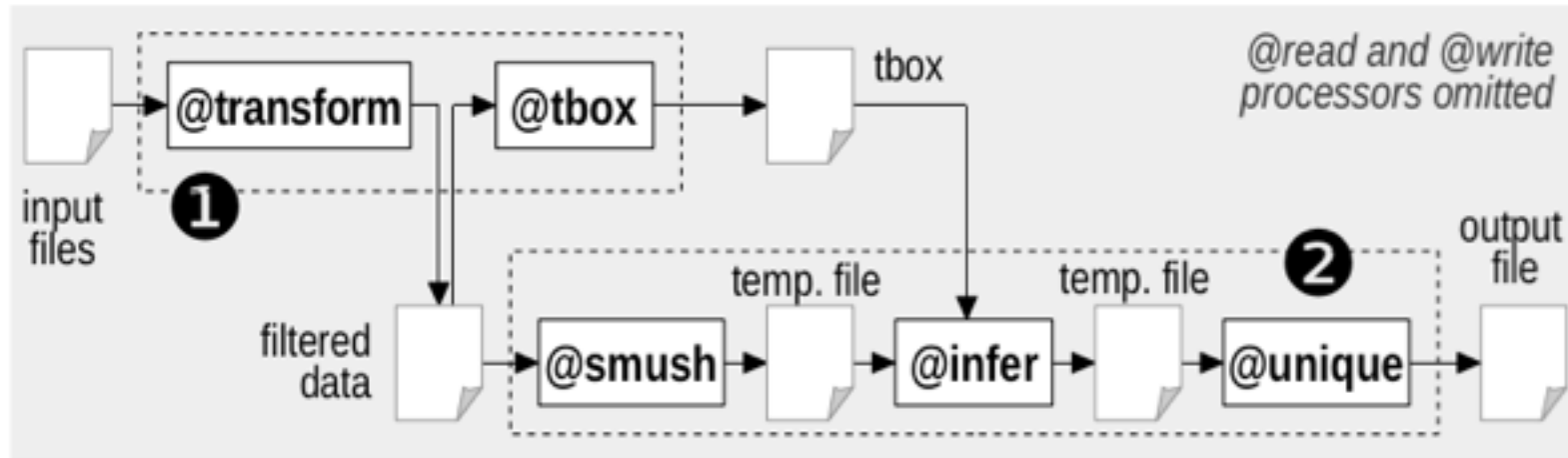
Scenario 3: Dataset Merging



Step	Input		Output		Throughput		Time
	[MQ]	[MB]	[MQ]	[MB]	[MQ/s]	[MB/s]	[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		Output		Throughput		Time	
	[MQ]	[MB]	[MQ]	[MB]	[MQ/s]	[MB/s]	[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

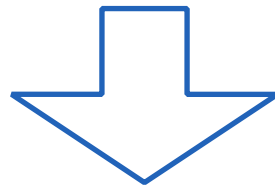
- TASK: ad-hoc **transformations** necessary to make data better suited to a particular use
 - **data repackaging**: preserve data content, but affect the way data is packaged (e.g., changing of RDF syntax)
 - **data sanitization**: fixing or removing the RDF terms or quads that prevent any further processing 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

- 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

- 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} ...

- ... 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](https://twitter.com/marcorospocher)
Fondazione Bruno Kessler
Trento, Italy