Processing SPARQL Aggregate Queries with Web Preemption

A. Grall, T. Minier, H. Skaf-Molli and **P. Molli**

LS2N, University of Nantes

UNIVERSITÉ DE NANTES

ESWC 2020
Online.
May, 2020

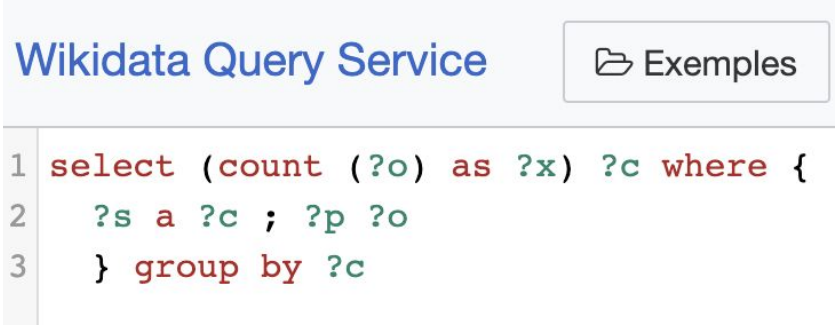
eswc20

How to execute aggregate queries online and get complete results ?

Ex: Number of objects per class

```
1 select (count (?o) as ?x) ?c where {
2   ?s a ?c ; ?p ?o
3   } group by ?c
```

On Wikidata: Timeout




```
SPARQL-QUERY: queryStr=select (count (?o) as ?x) ?c where {
  ?s a ?c ; ?p ?o
  } group by ?c
java.util.concurrent.TimeoutException
        at java.util.concurrent.FutureTask.get(FutureTask.java
```

On Dbpedia: Partial Results

Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

http://dbpedia.org

Query Text

```
 SELECT (COUNT(?o) AS ?x) ?c WHERE {
?s a ?c ; ?p ?o
} GROUP BY ?c
```

x	c
1	http://dbpedia.org/class/yago/WikicatMinesweepersOfTheFijianNavy
6	http://dbpedia.org/class/yago/WikicatParalympicBronzeMedalistsForLatvia
1	http://dbpedia.org/class/yago/WikicatFoundationsBasedInRussia
1	htt

X-SPARQL-MaxRows: 10000

Dumps ??

- Download the dump and compute locally:

dcatap.rdf	07-May-2020 17:21	79345
latest-all.json.bz2	06-May-2020 05:45	55587167799
latest-all.json.gz	06-May-2020 00:21	83468999588
latest-all.nt.bz2	07-May-2020 15:36	113723653769
latest-all.nt.gz	07-May-2020 02:30	144546065089
latest-all.ttl.bz2	07-May-2020 06:00	71897810492
latest-all.ttl.gz	06-May-2020 23:09	86000654891

- Well… Good luck… Tell me when it’s done… ;)
- **Not Live Queries...**

TPF with restricted web servers terminates…

#LD
Linked Data Fragments

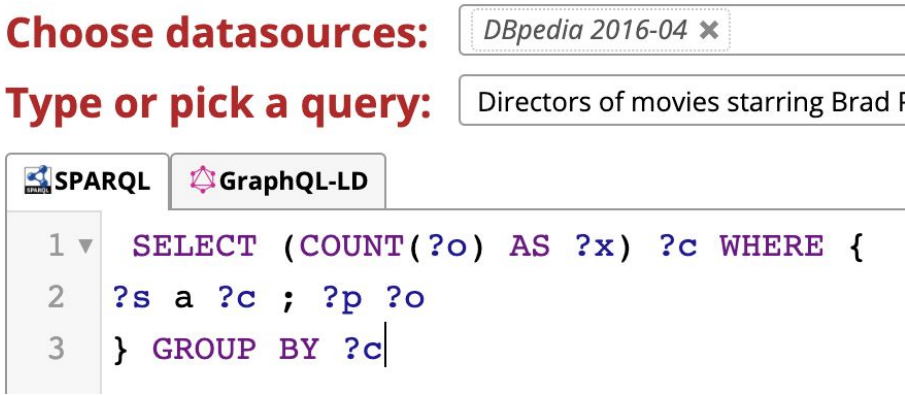
- But, browser executes:

  *For ?s in **http**(?s a ?c):*

  *   **http**(?s ?p ?o)*

  Group by ?c

  count(?o)
- Nearly download SPO (~dump)
- **Too much calls and data transfer. Not realistic**

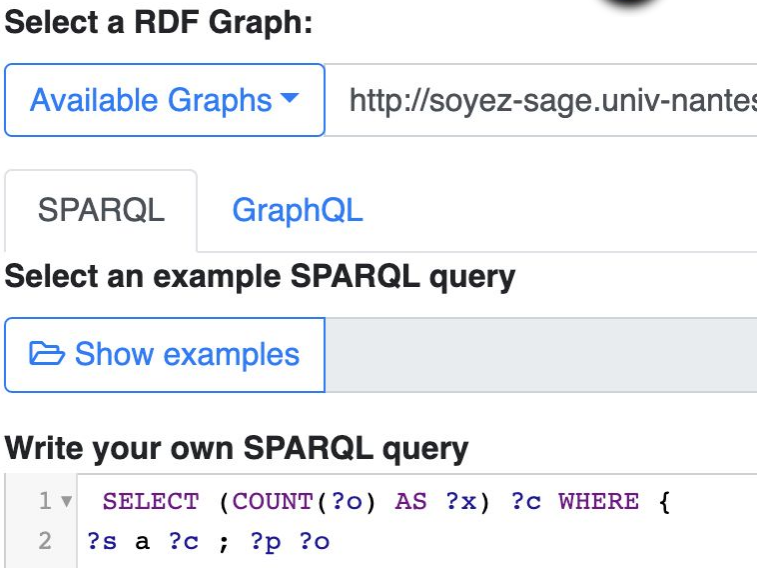


SaGe with web preemption terminates...

- The browser executes:

*?o, ?c = **http**(?s a ?c;?p ?o):*
Group by ?c
count(?o)

Better than TPF,
but still too much data transfer...



Aggregate Queries

SPARQL Endpoints

- Fast when under the quota
- But, no guarantee of termination

Fragment, Web preemption

- Terminates...
- But, prohibitive data transfer, slow

How to compute SPARQL aggregate queries online and get complete results ?

Our approach

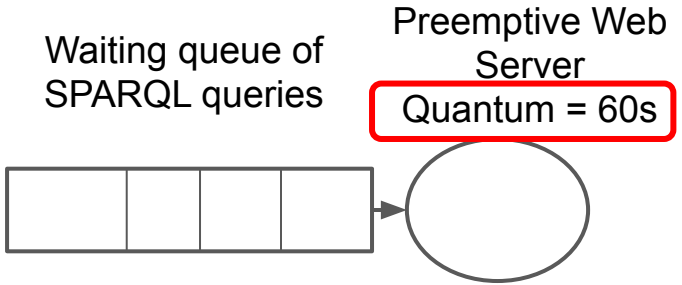
Build partial aggregations distributed in time with web preemption

Web Preemption

“The capacity of a Web server to **suspend** a running query after a **time quantum** with the intention to **resume** it later.”

- **There is no need for a QUOTA if you have a quantum.**

Web Preemption in action

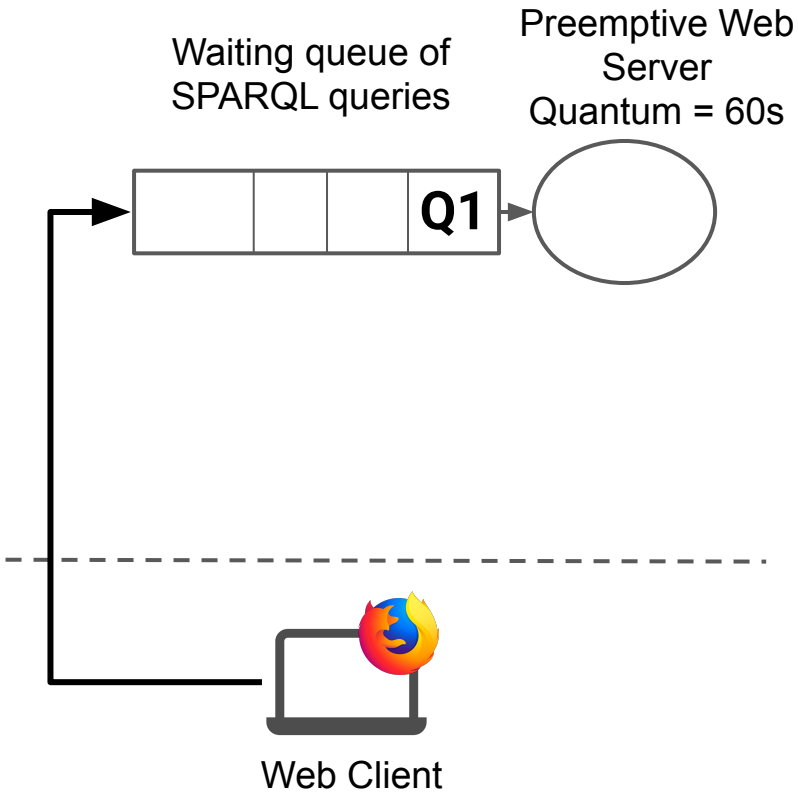


Q1

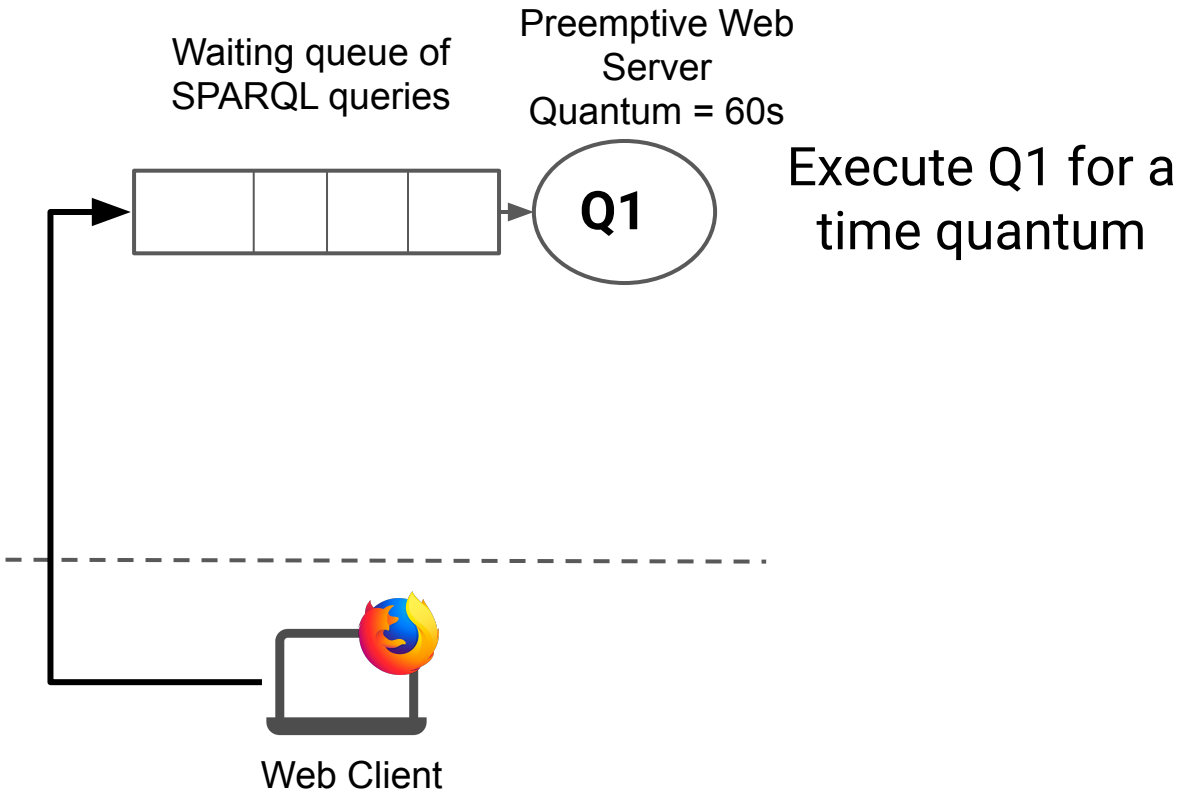
```
SELECT ?c ?o WHERE {
   ?s a ?c ; ?p ?o
}
```

Web Client

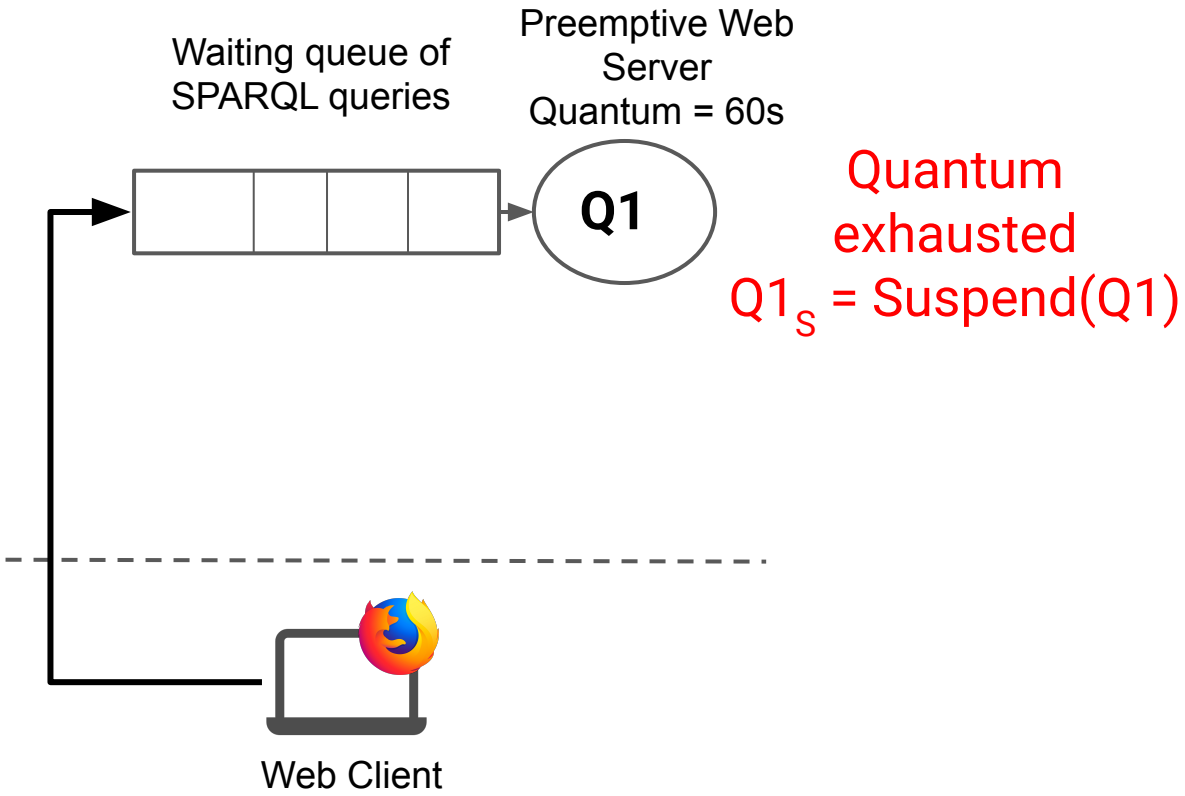
Web Preemption in action



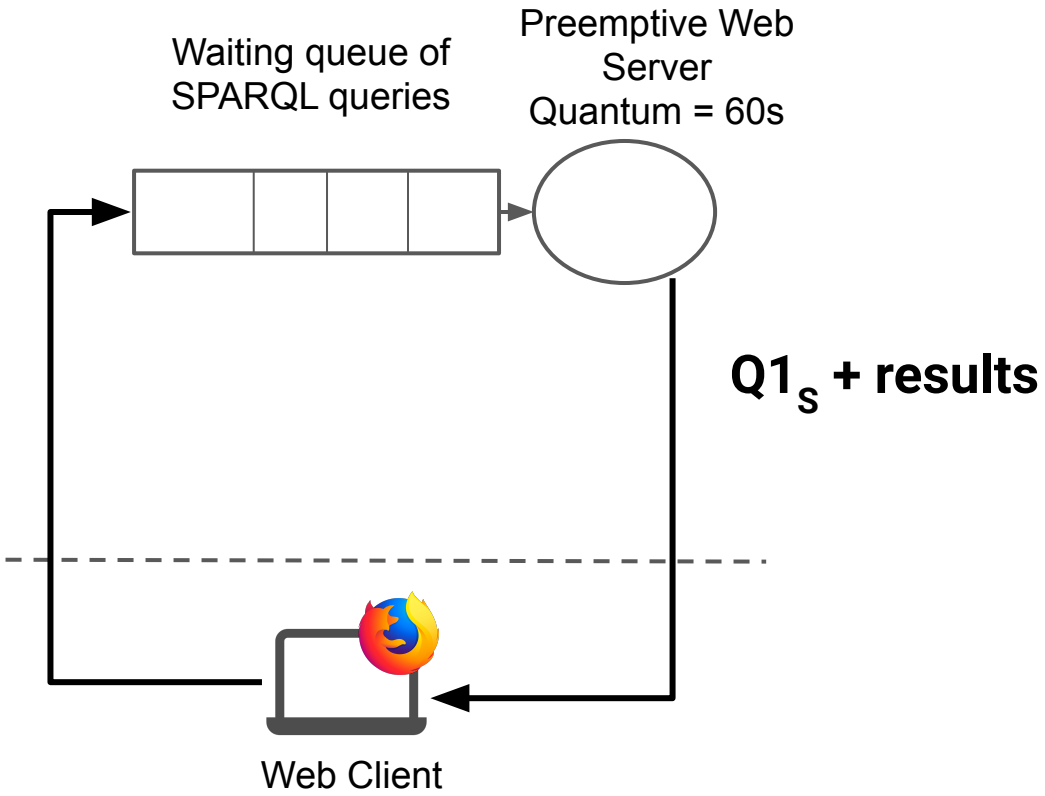
Web Preemption in action



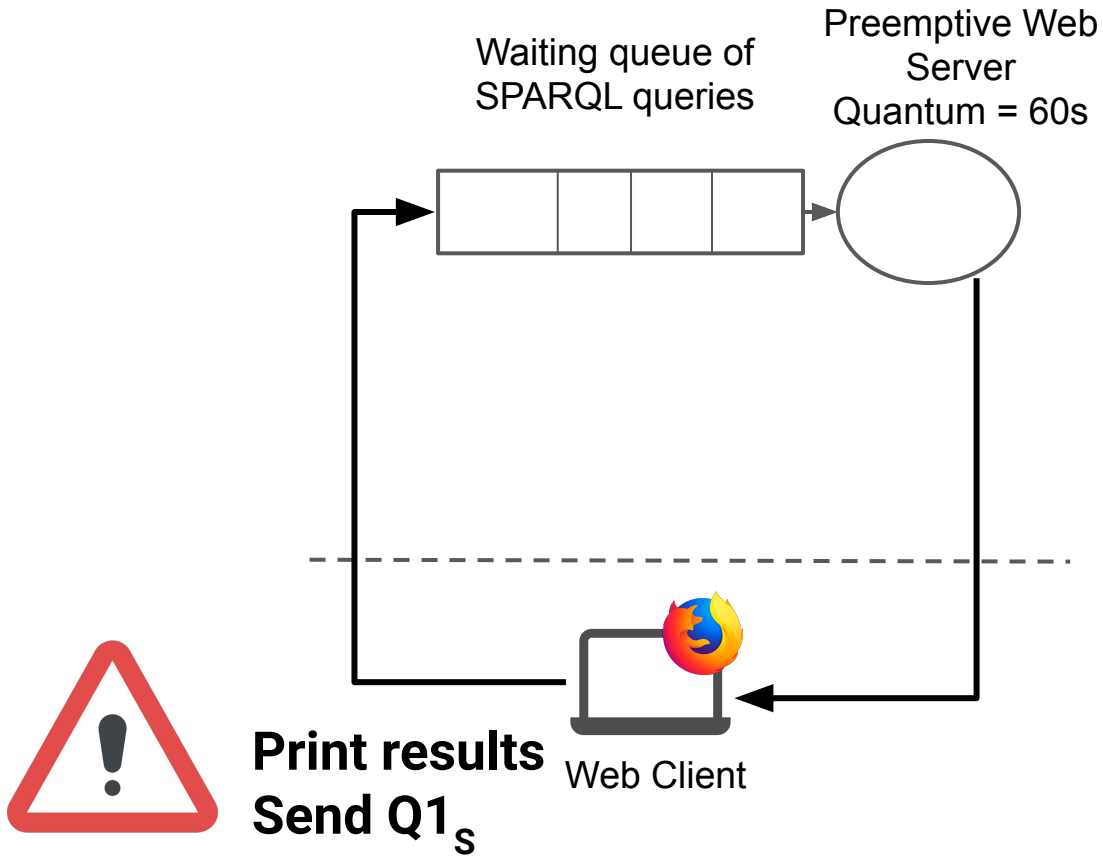
Web Preemption in action



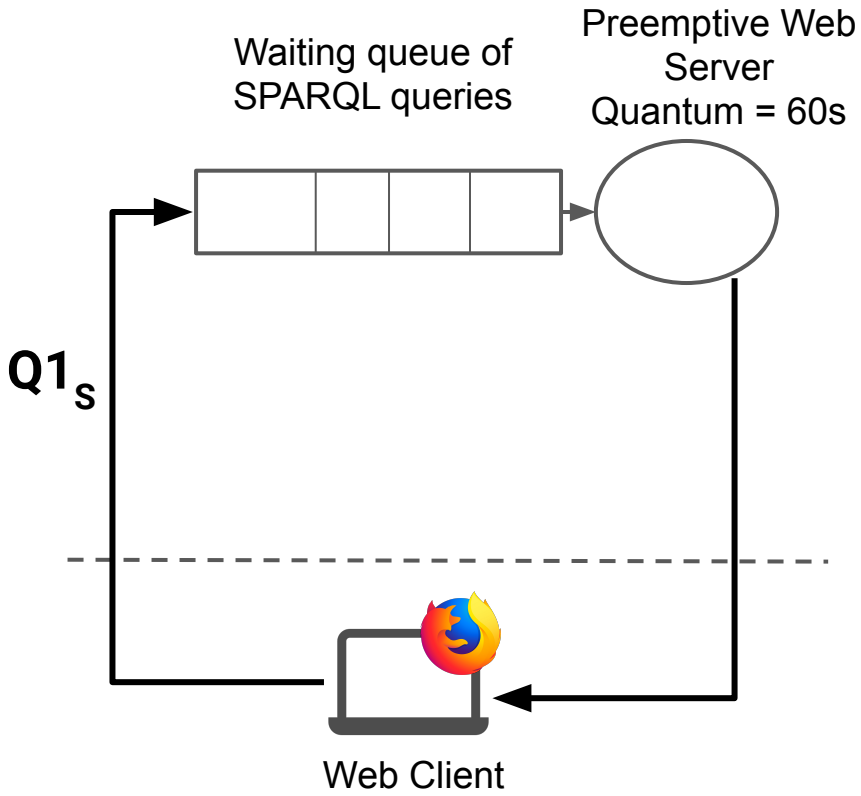
Web Preemption in action



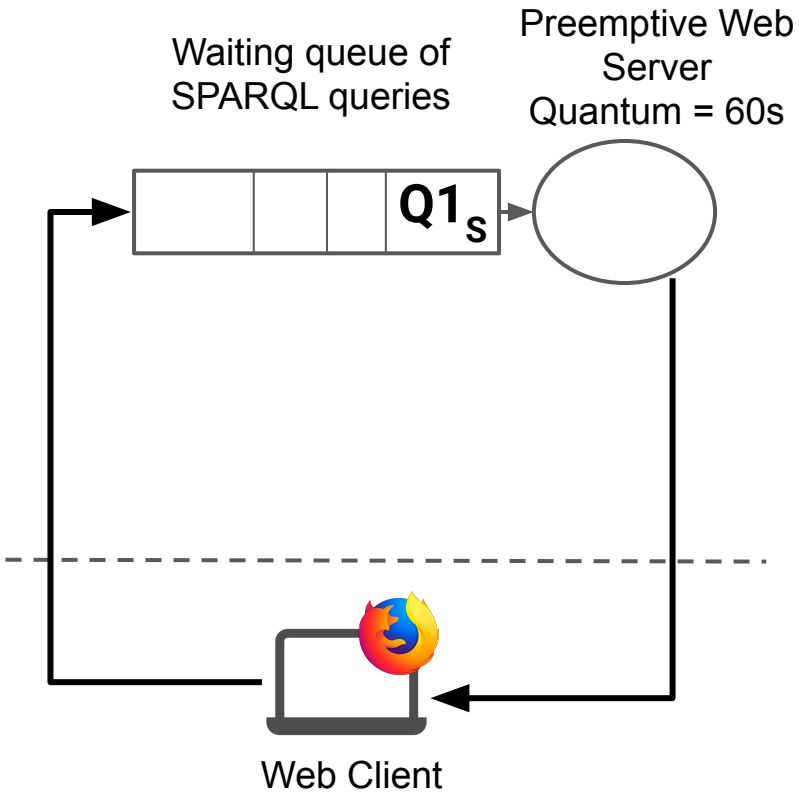
Web Preemption in action



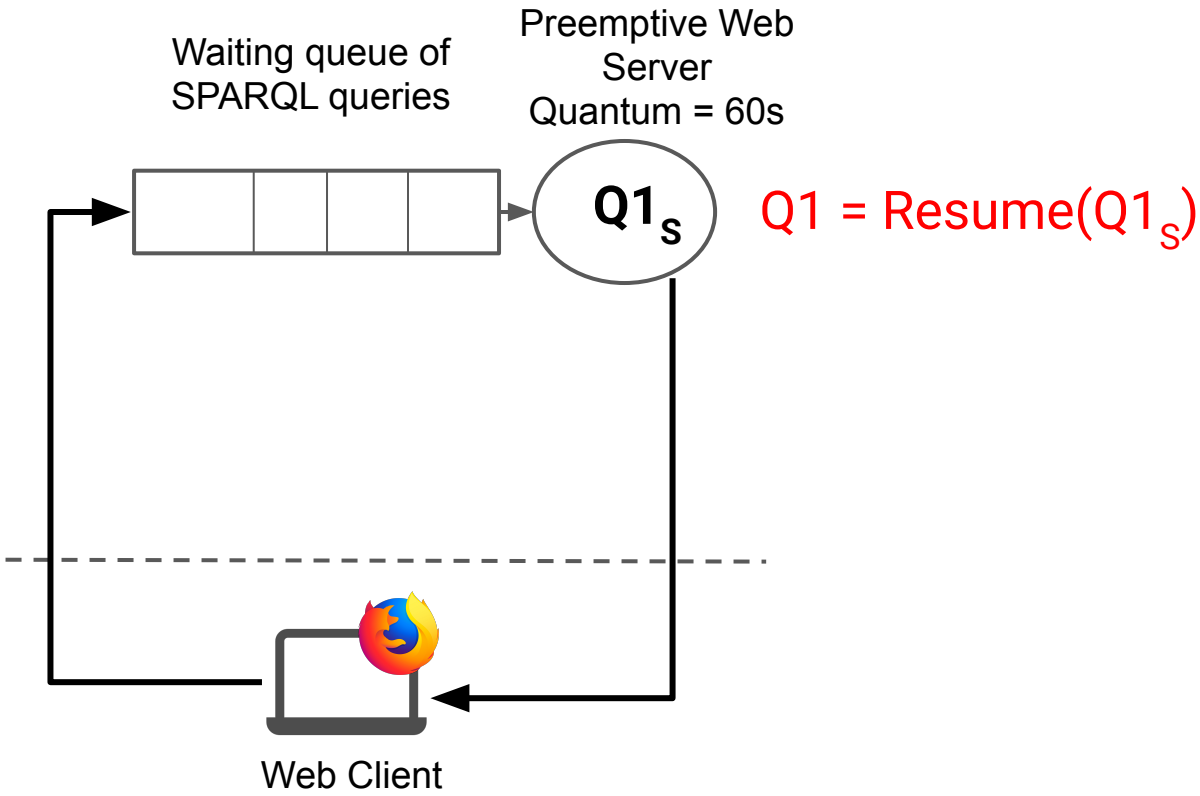
Web Preemption in action



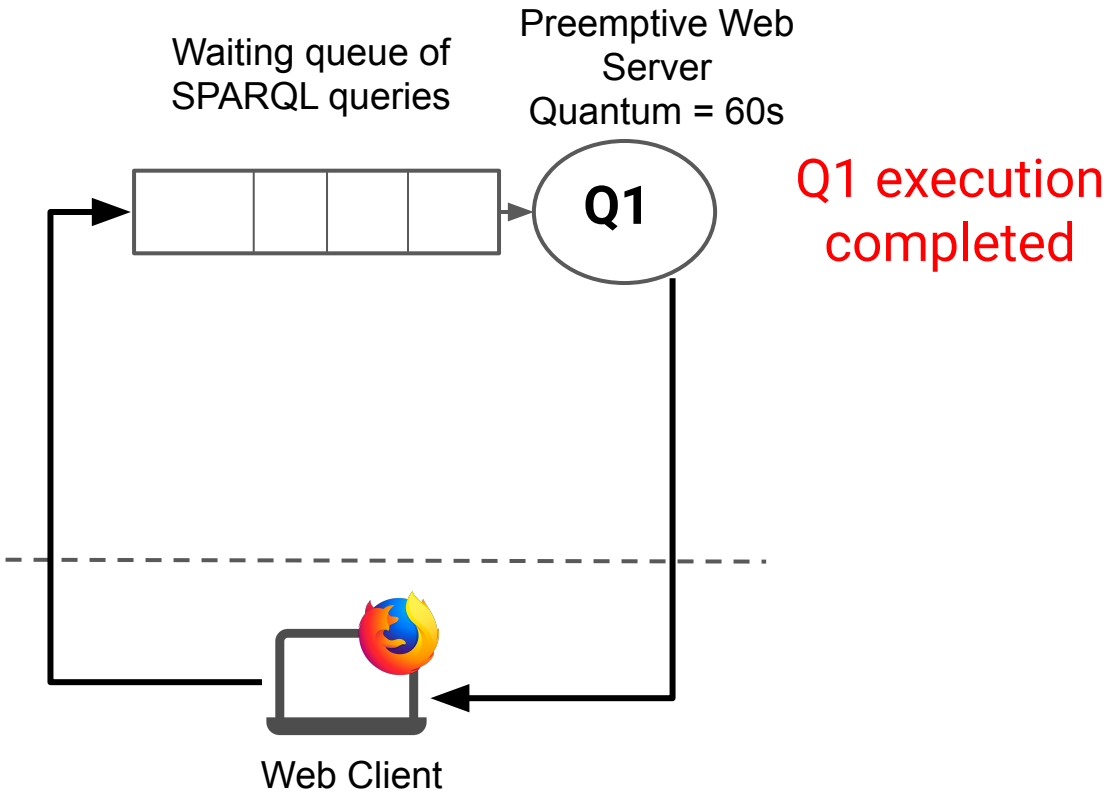
Web Preemption in action



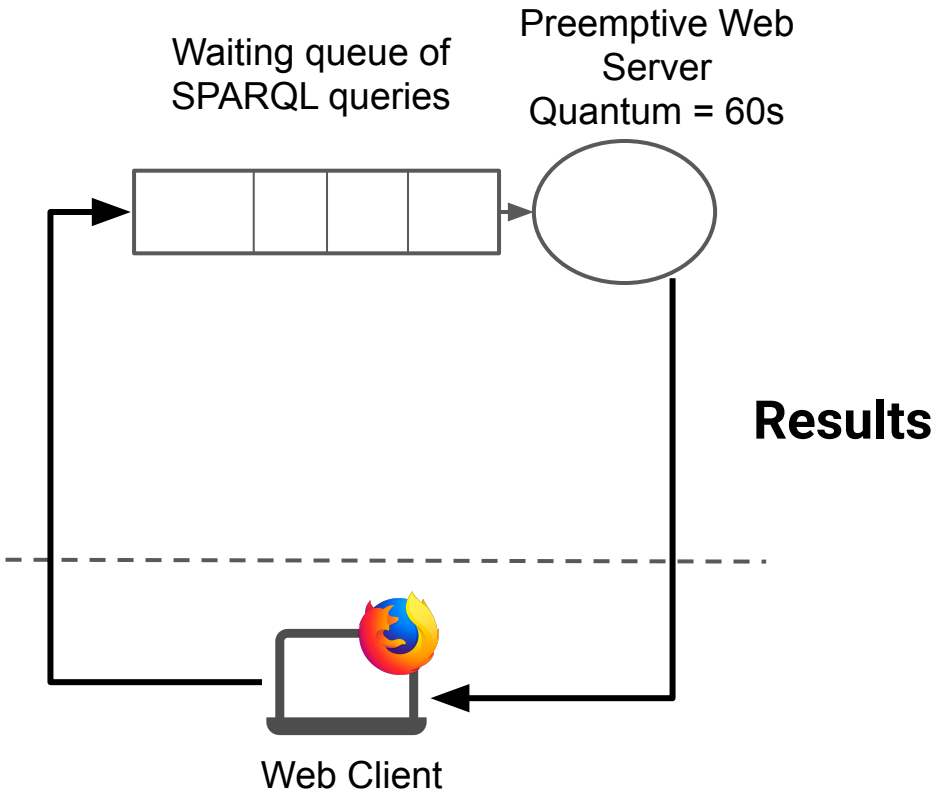
Web Preemption in action



Web Preemption in action

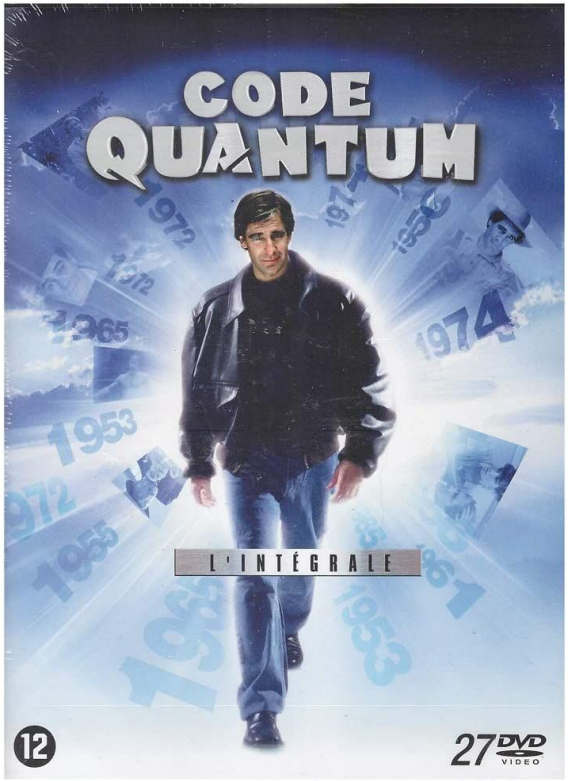


Web Preemption in action

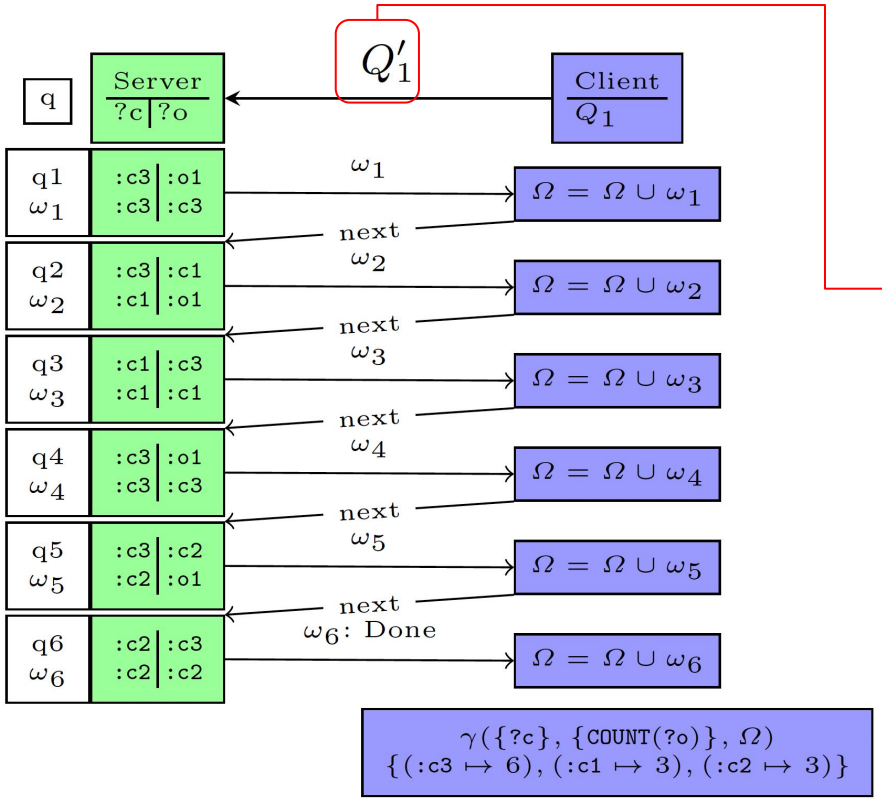


Web Preemption allows...

- A **fair allocation** of web server resources across queries
- Better **average completion time** per query
- Better **time for first results** per query



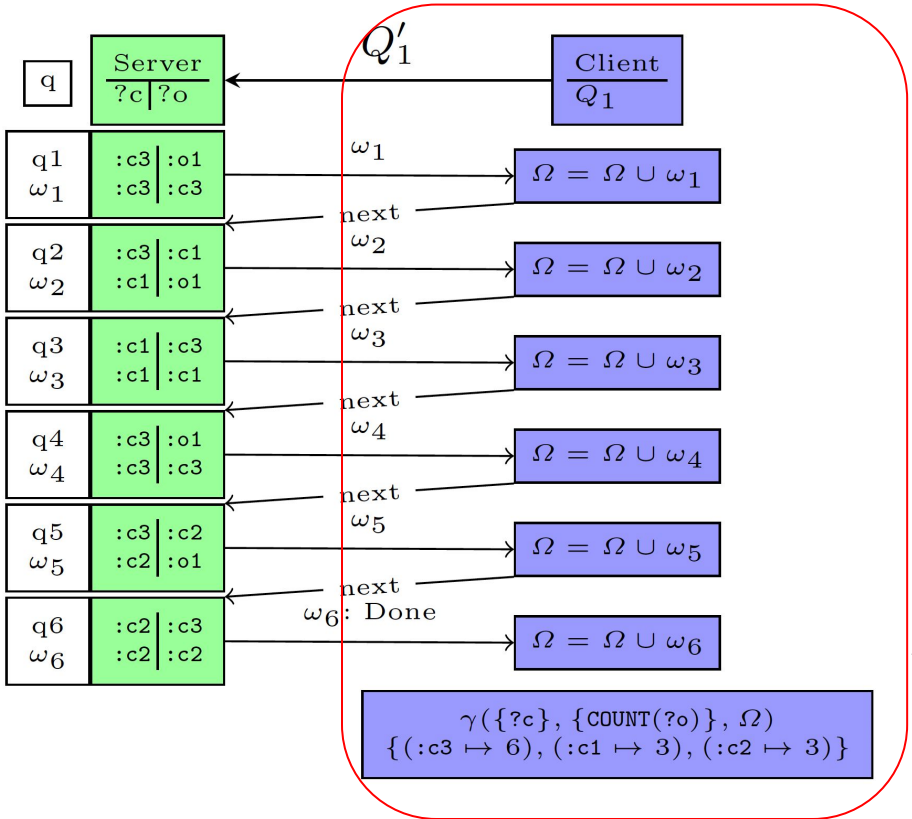
Aggregates on Client with web preemtion




```
SELECT ?c
  (COUNT(?o) AS ?z)
WHERE { ?s :a ?c .
  ?s ?p ?o . ?s :p1 :o1}
GROUP BY ?c
```

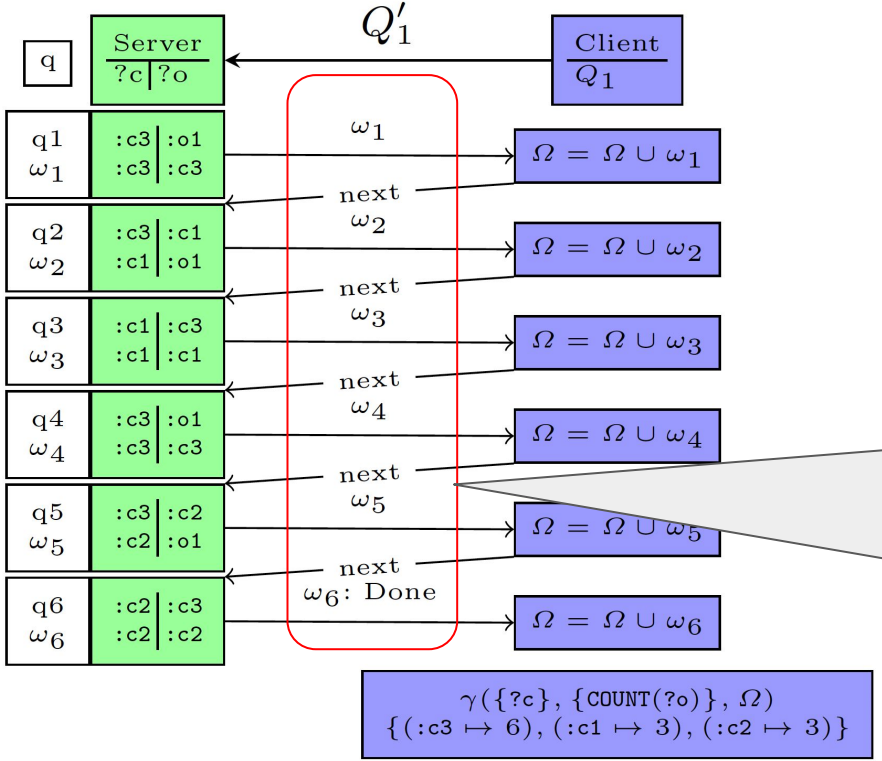
:s1 :p1 :o1 .
:s1 :a :c2, :c3.
:s2 :p1 :o1 .
:s2 :a :c1, :c3.

Current processing of aggregate



Aggregation is done on CLIENT

Current processing of aggregate



So all <?c,?o> are transfered !
-> prohibitive with large datasets

Just execute Aggregation on server...

- **PB: Aggregation is not Preemptable !**
- When computing an aggregate
  - Need to keep a temporary table of group keys.
- O(Suspend/Resume(Aggregate))~ size(Aggregate) != constant time
- **Not preemptable**

city	customer_count
Albany	3
Amarillo	5
Amityville	9
Amsterdam	5
Anaheim	11
Apple Valley	11
Astoria	12
Atwater	5
Auburn	4
Bakersfield	5

Problem statement

- Define a preemptable aggregation operator such that the complexity in time and space of suspending and resuming is bounded in constant time

Key Idea

- Web preemption creates partition of mappings per quantum
  - Compute partial Aggregates per quantum
  - Client merge partial aggregate
- Correct because aggregation functions are decomposable

Decomposability of Aggregation

- A function f is (self) decomposable [1] if:
- $f(X \uplus Y) = f(X) \diamond f(Y)$
    - where $\diamond$ is a merge operator
- Ex: $COUNT(X \uplus Y) = COUNT(X) + COUNT(Y)$
- Ex: $Max(X \uplus Y) = max(MAX(X),MAX(Y))$
- etc...

[1] Yan, W.P., Larson, P.A.: Eager aggregation and lazy aggregation. In: 21th International Conference on Very Large Data Bases, VLDB. pp. 345–357 (1995)

Decomposability of Aggregation

Definition 3 (Decomposable aggregation function). *An aggregation function f is decomposable if for some grouping expressions E and all non-empty multisets of solution mappings Ω_1 and Ω_2, there exists a (merge) operator $\diamond$, a function h and an aggregation function f_1 such that:*

$$\gamma(E, \{f\}, \Omega_1 \uplus \Omega_2) = \{k \mapsto h(v_1 \diamond v_2) \mid k \mapsto v_1 \in \gamma(E, \{f_1\}, \Omega_1), \\ k \mapsto v_2 \in \gamma(E, \{f_1\}, \Omega_2)\}$$

[1] Yan, W.P., Larson, P.A.: Eager aggregation and lazy aggregation. In: 21th International Conference on Very Large Data Bases, VLDB. pp. 345–357 (1995)

Decomposability of Aggregation

$$\gamma(E, \{f\}, \Omega_1 \uplus \Omega_2) = \{k \mapsto h(v_1 \diamond v_2) \mid k \mapsto v_1 \in \gamma(E, \{f_1\}, \Omega_1), \\ k \mapsto v_2 \in \gamma(E, \{f_1\}, \Omega_2)\}$$

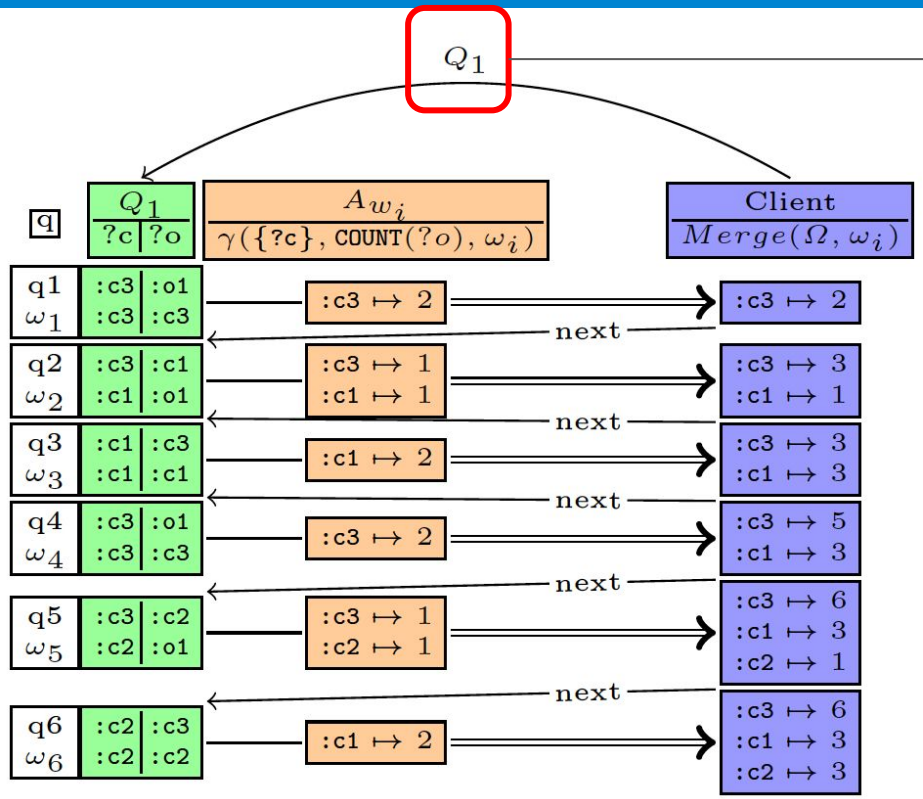
- f = COUNT(?c)
- γ(V,{f},Ω1 ⊎ Ω2) st.
  - γ(V,{f},Ω1) = {{?c → 2}}
  - γ(V,{f},Ω2) = {{?c→ 5}}
- γ(V,{f},Ω1 ⊎ Ω2) = {{?c → 2◇5 = 2+5 = 7}}

[1] Yan, W.P., Larson, P.A.: Eager aggregation and lazy aggregation. In: 21th International Conference on Very Large Data Bases, VLDB. pp. 345–357 (1995)

Decomposability of Aggregation

SPARQL Aggregations functions								
	COUNT	SUM	MIN	MAX	AVG	COUNT_D	SUM_D	AVG_D
f_1	COUNT	SUM	MIN	MAX	SaC	CT		
$v \diamond v'$	$v + v'$		$min(v, v')$	$max(v, v')$	$v \oplus v'$	$v \cup v'$		
h	Id				$(x, y) \mapsto x/y$	COUNT	SUM	AVG

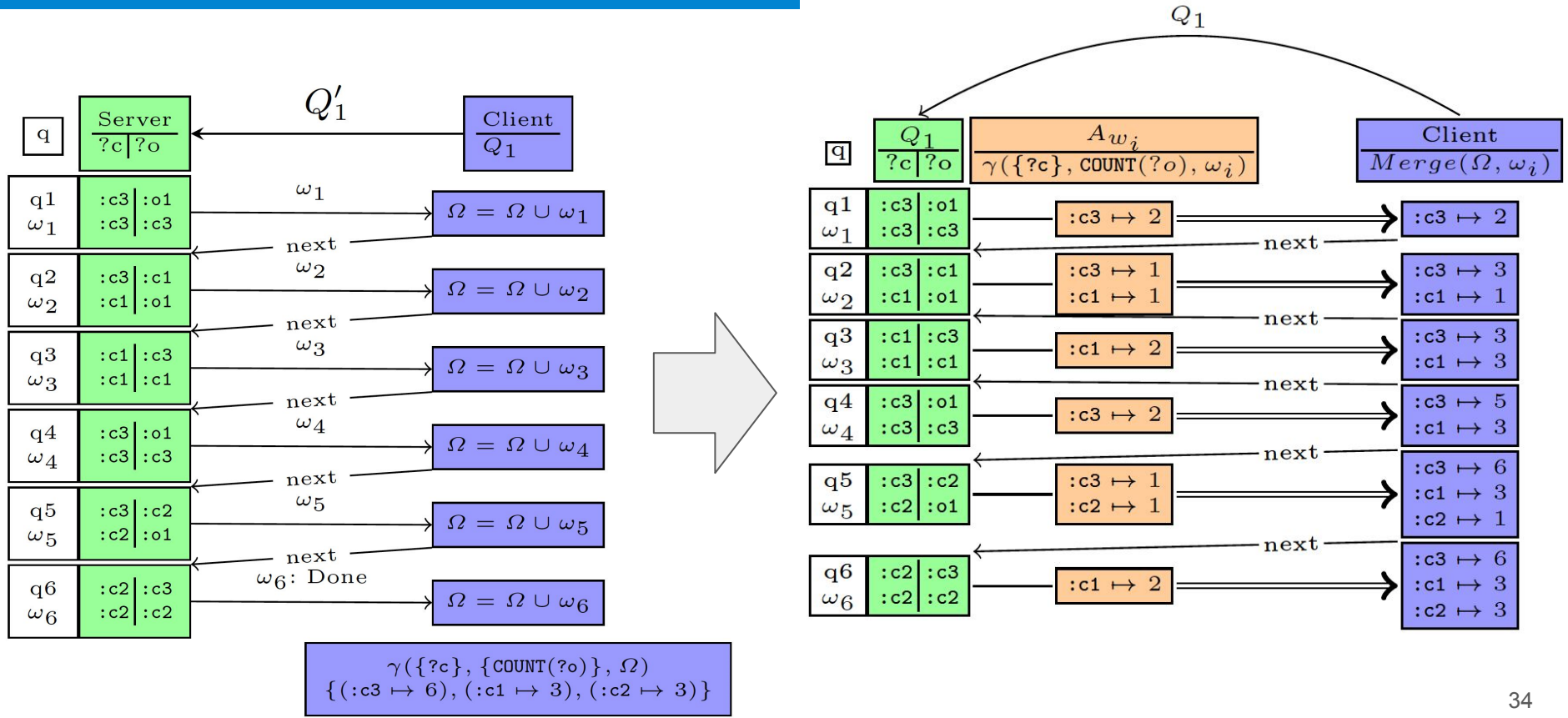
Decomposability of Aggregation



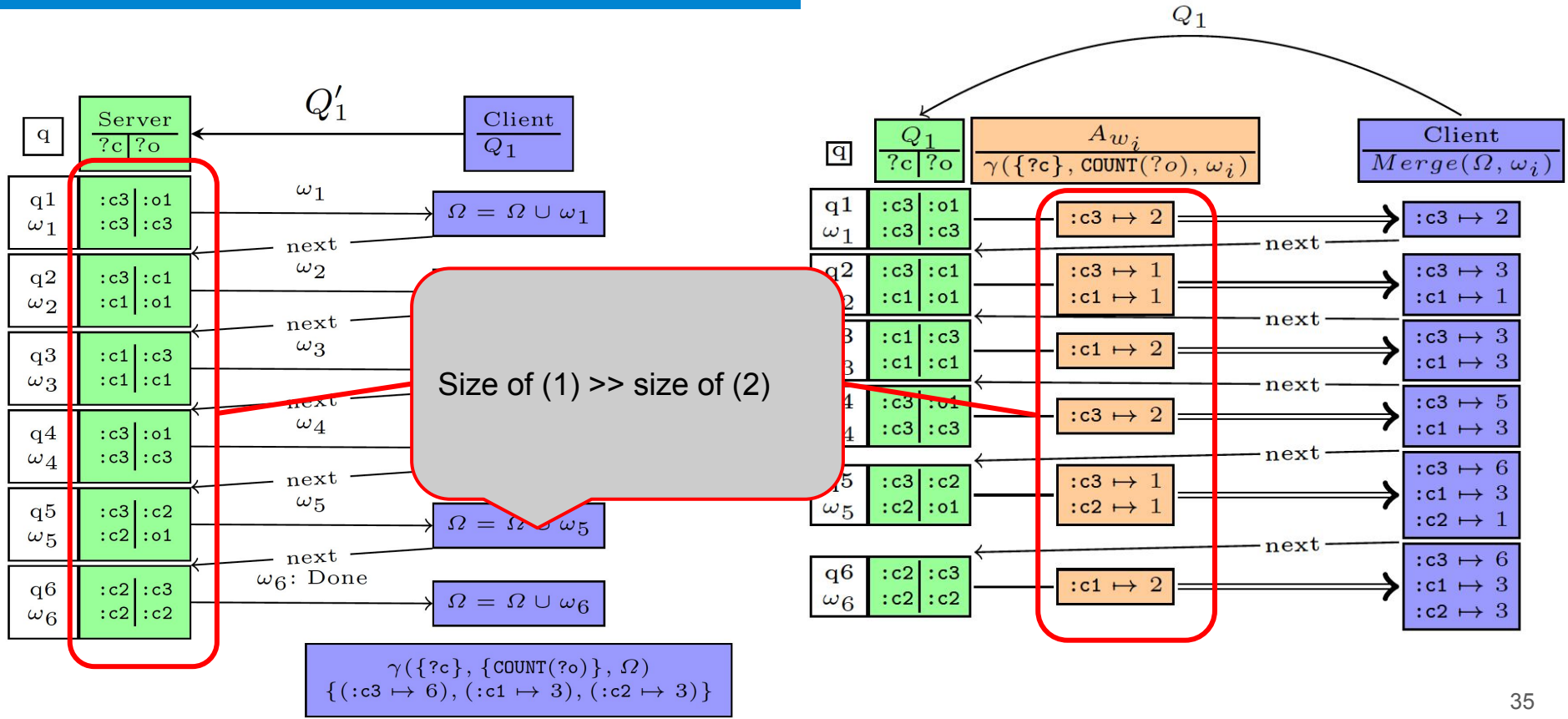

```
SELECT ?c
  (COUNT(?o) AS ?z)
WHERE { ?s :a ?c .
  ?s ?p ?o . ?s :p1 :o1}
GROUP BY ?c
```

```
:s1 :p1 :o1 .
:s1 :a :c2, :c3.
:s2 :p1 :o1 .
:s2 :a :c1, :c3.
```

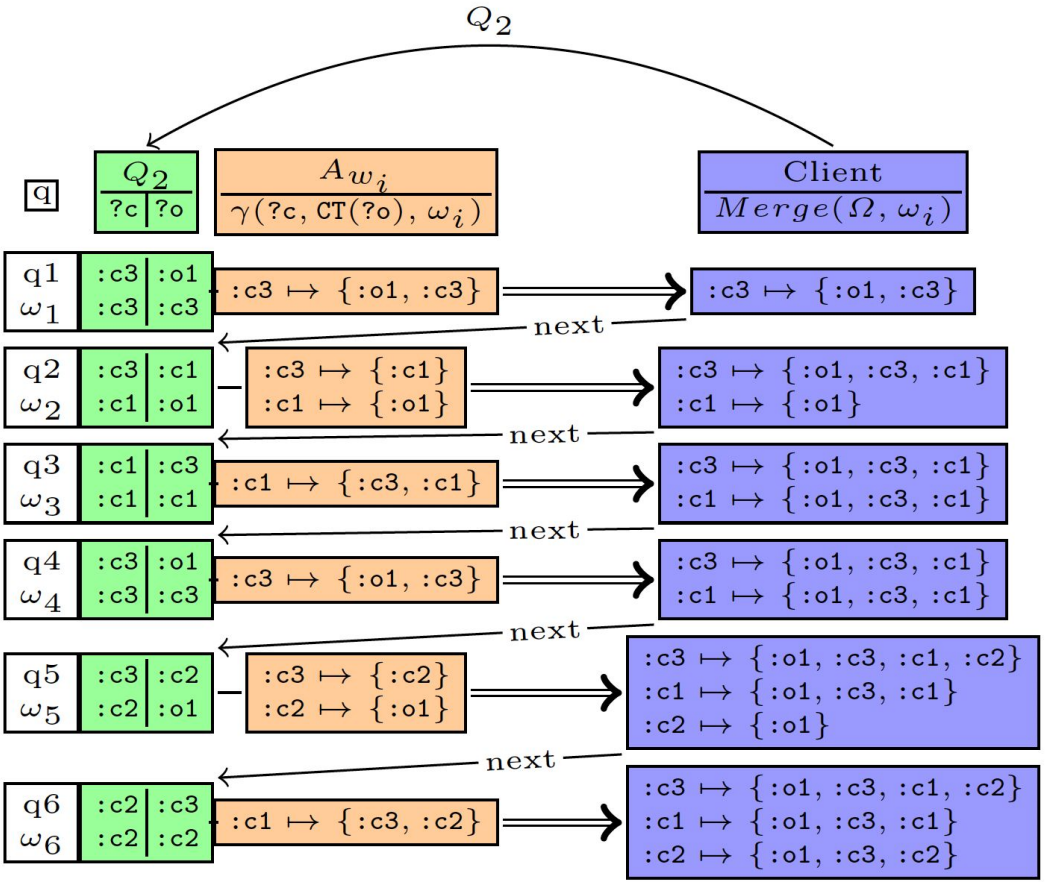
Decomposability of Aggregation



Decomposability of Aggregation



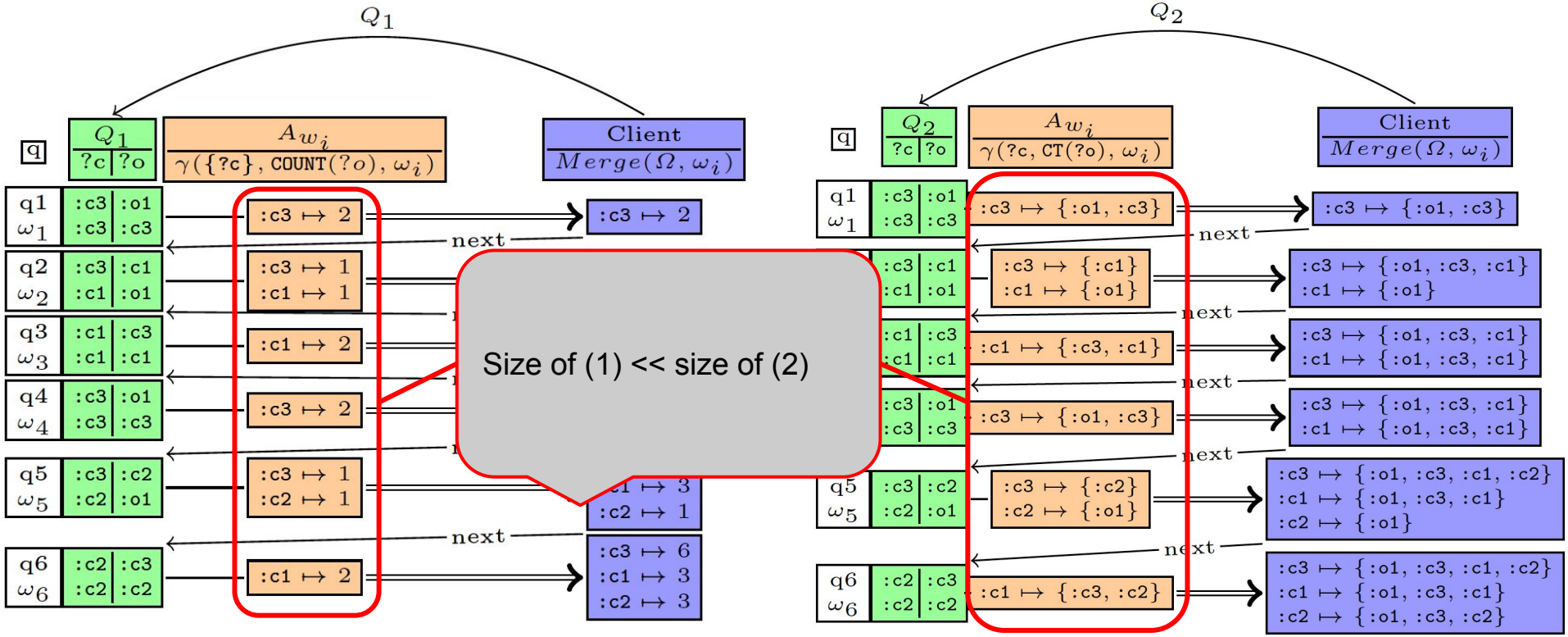
Partial Aggregate with Distinct




```
SELECT ?c
  (COUNT(Distinct(?o)) AS ?z)
WHERE { ?s :a ?c .
  ?s ?p ?o . ?s :p1 :o1}
GROUP BY ?c
```

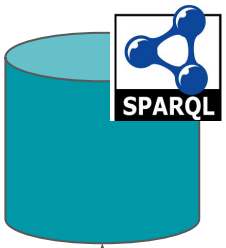
```
:s1 :p1 :o1 .
:s1 :a :c2, :c3.
:s2 :p1 :o1 .
:s2 :a :c1, :c3.
```

No Distinct / Distinct



SaGe: A preemptive SPARQL query engine

SaGe distributes Physical Query Operators between Server and Client



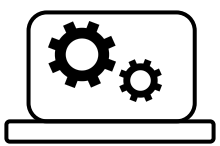
Preemptive Web server

Server language

Triple pattern, ⋈, U,
SELECT, FILTER

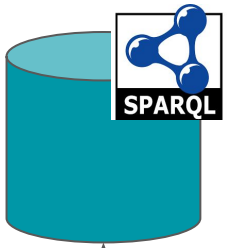
Client language

OPTIONAL
solution modifiers



Smart Web Client

Sage-AGG



Preemptive Web server

Server language

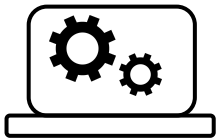
Triple pattern, ⋈, U,

SELECT, FILTER,

Client language

OPTIONAL,

solution modifiers



Smart Web Client

Experimental Study

Experimental Study

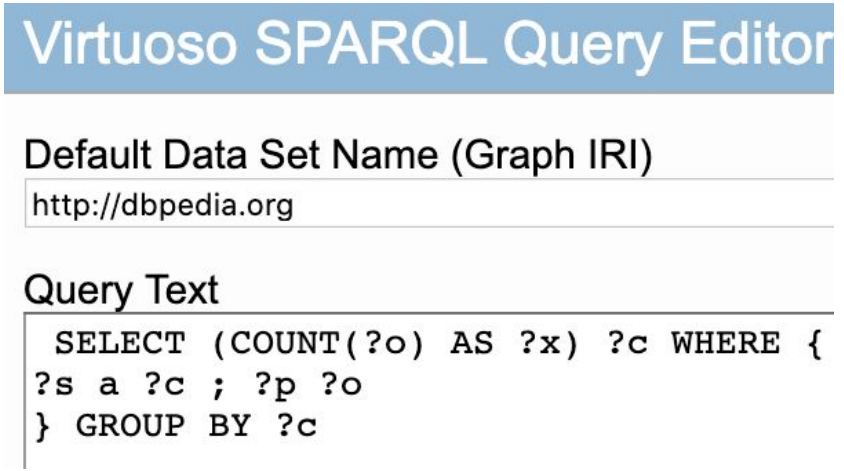
1. What is the data transfer reduction obtained with partial aggregations?
2. What is the speed up obtained with partial aggregations?
3. What is the impact of time quantum on data transfer and execution time?

Data

RDF Dataset	# Triples	# Subjects	# Predicates	# Objects	# Classes
BSBM-10	4 987	614	40	1 920	11
BSBM-100	40 177	4 174	40	11 012	22
BSBM-1k	371 911	36433	40	86202	103
DBpedia 3.5.1	153M	6 085 631	35 631	35 201 955	243

Experimental Setup

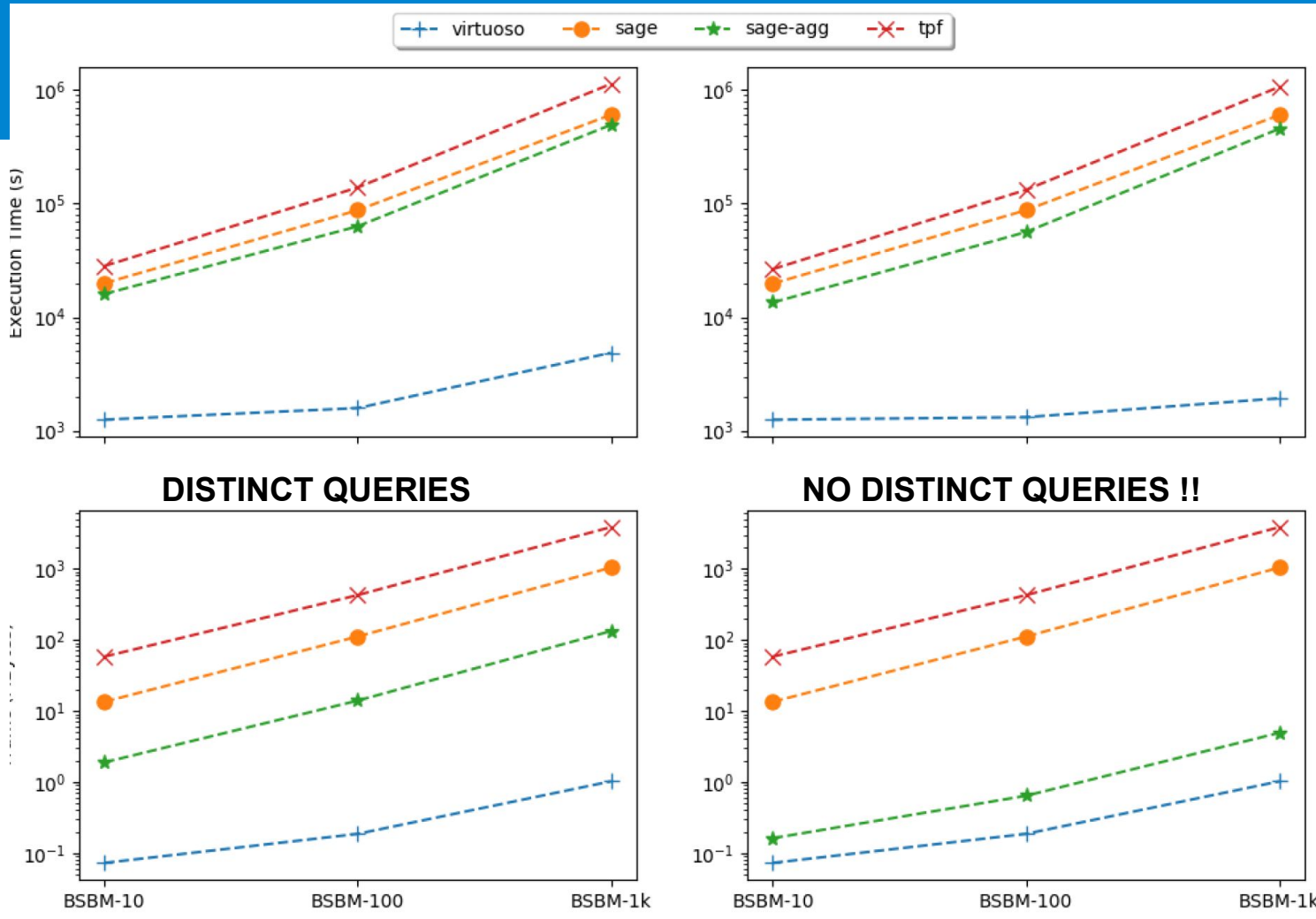
- Workload of 18 queries from SPORTAL queries [1]
  - Most queries don’t terminate under quota
- Engines
  - TPF
  - SaGe
  - SaGe-AGG (our proposal)
  - Virtuoso (as the optimal)



[1] Hasnain, A., Mehmood, Q., e Zainab ang Aidan Hogan, S.S.: SPORTAL: profiling the content of public SPARQL endpoints. Int. J. Semantic Web Inf. Syst. 12(3), 134–163 (2016)

Execution time

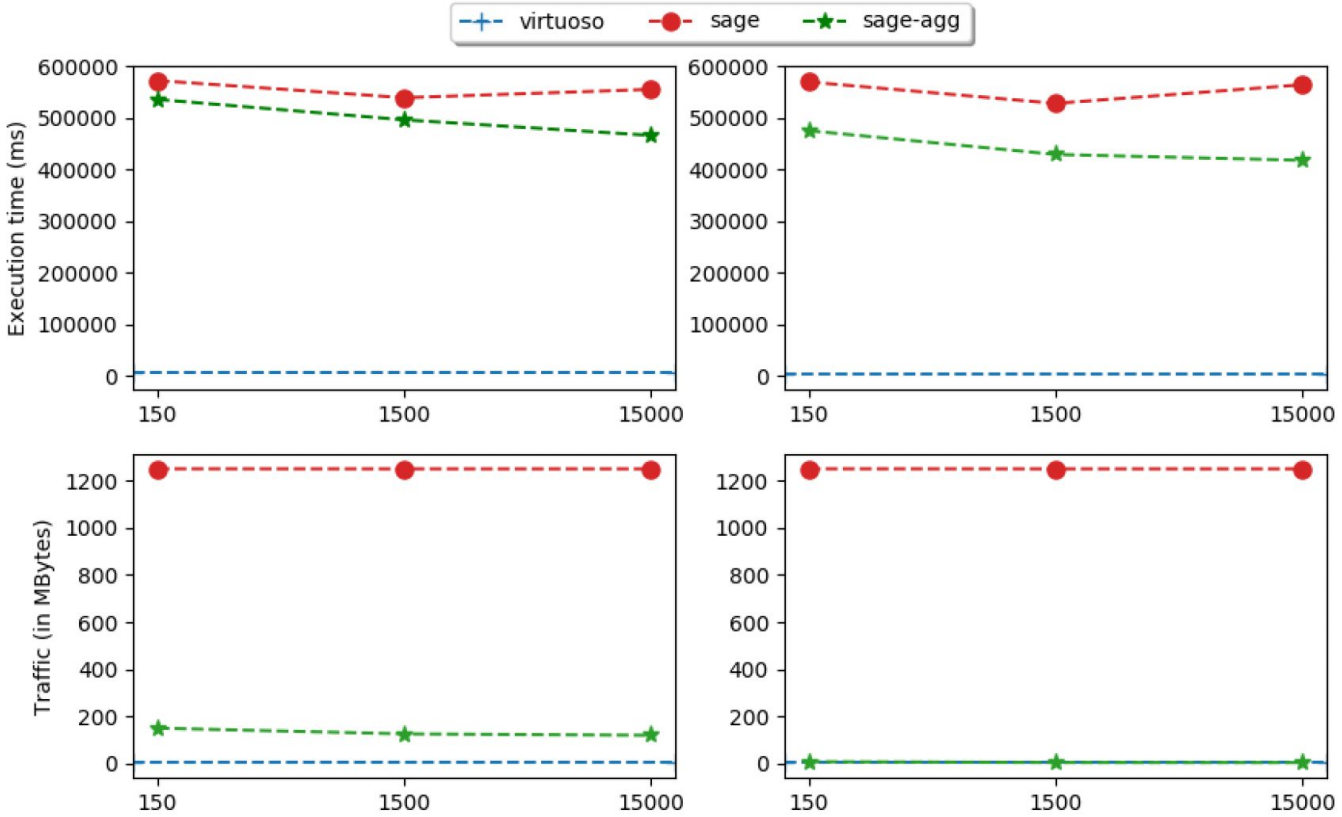
Data Transfer



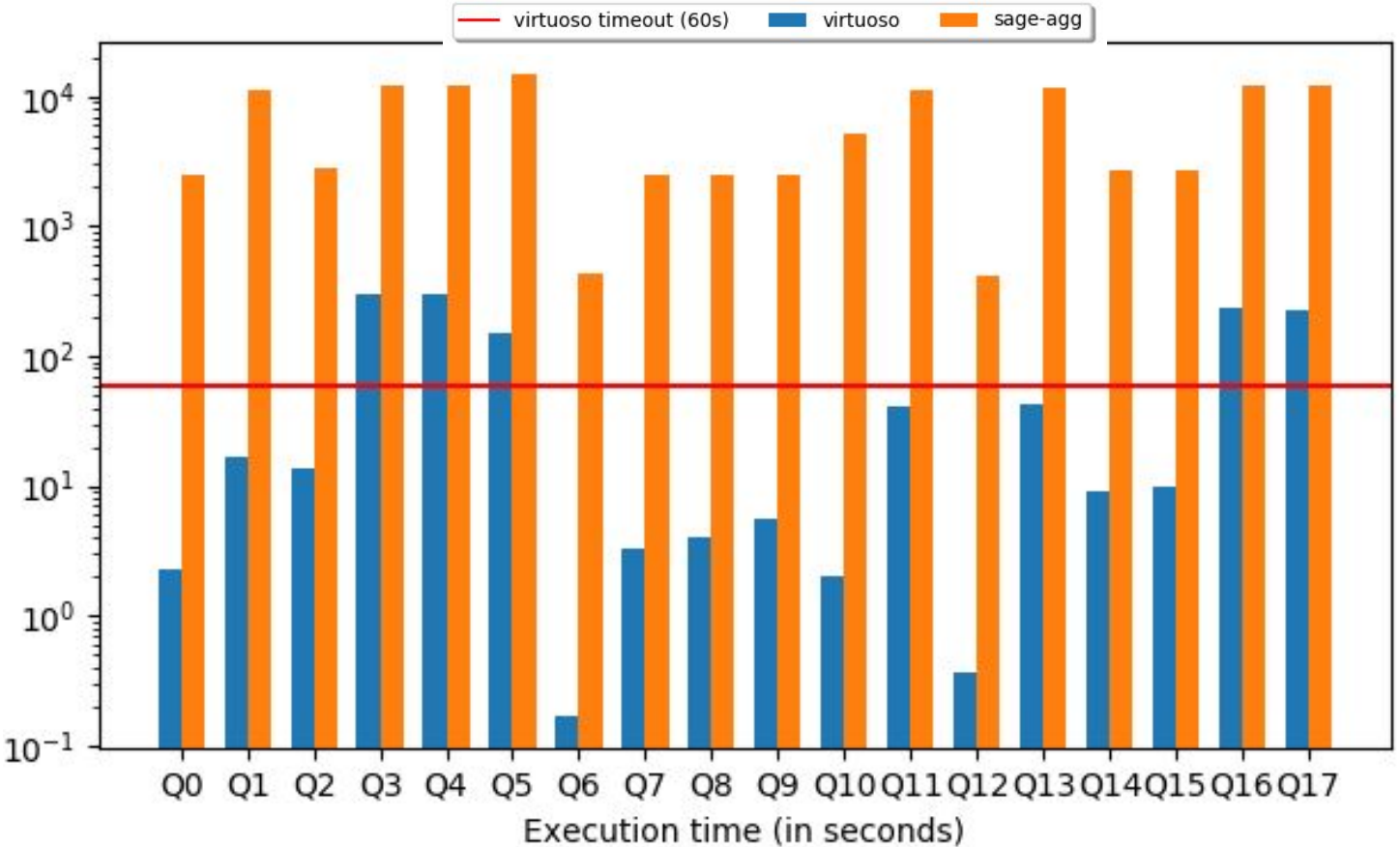
Impact of Quantum, BSBM1K

Execution time

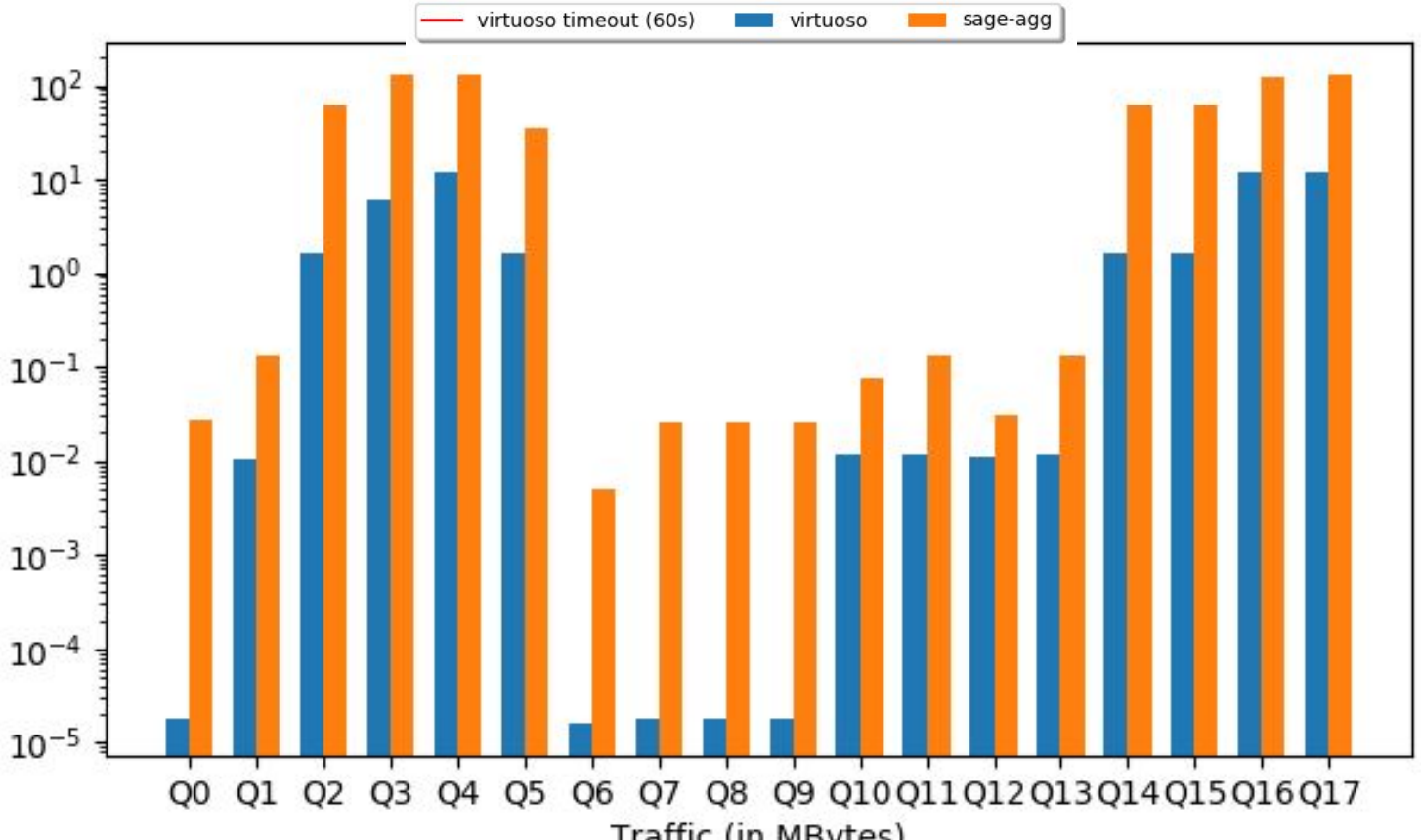
Data Transfer



DBpedia Experiment : Execution Time



DBpedia Experiment : Traffic



Conclusion

- We defined an preemptable aggregate operator for Public SPARQL services
- Allow to execute aggregate queries on public endpoint that terminates
- Allow to compute statistics online, (no dump ;)

Perspectives

- Support for CONSTRUCT and REDUCED
  - Same approach
- Speed up execution time with parallelism
  - Require range partitioning of data

Processing SPARQL Aggregate Queries with Web Preemption

A. Grall, T. Minier, H. Skaf-Molli and **P. Molli**

LS2N, University of Nantes

ESWC 2020
Online.
May, 2020

DBpedia Experiment