Estimation (theoretical and experimental) of the computational complexity of a query execution to a multilevel and one-level TN model relative to a parallel SPARQL requests execution

Request #1 – Select node using the identifier

Request #2 – Select of related elements (obtaining elements taking into account the hierarchy)

Request #3 – Select of related items using a filter

Request #4 – Select of related items with filtering and grouping

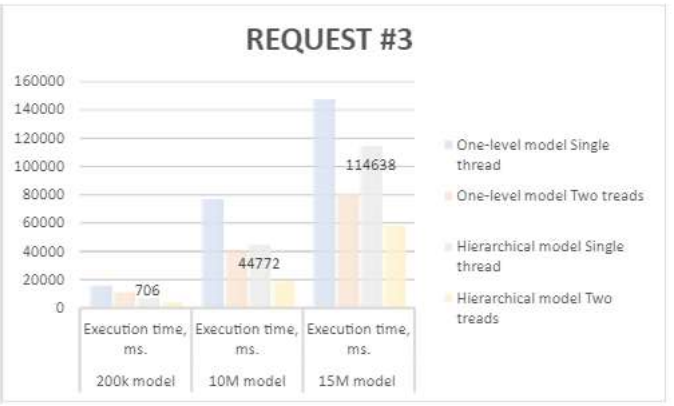
Request #5 – Substring search

Number of KG nodes	Model type	Model parameters		Single thread / Parallel threads	Request #1 Execution time (mean/mean-square deviation), ms.	Request #2 Execution time (mean/mean-square deviation), ms.	Request #3 Execution time (mean/mean-square deviation), ms.	Request #4 Execution time (mean/mean-square deviation), ms.	Request #5 Execution time (mean/mean-square deviation), ms.
		Number of levels	Source models linked levels						
200к	One-level model			Single	210±4	92±4	1586±65	995±47	595±32
				Parallel	127±3	78±3	1104±57	958±39	565±23
	Hierarchical model	3 levels	2-2 (U)*	Single	73±2	67±3	706±38	578±32	318±22
			2-2 (U)*	Parallel	38±3	39±4	418±21	471±18	271±11
10M	One-level model			Single	361±8	112±6	77010±297	72234±309	12145±82
				Parallel	231±5	91±4	39732±105	39562±179	7231±74
	Hierarchical model	3 levels	2-2 (U)*	Single	143±4	77±4	44772±277	57316±286	517±24
			2-2 (U)*	Parallel	78±2	41±2	19232±112	27561±93	395±18
15M	One-level model			Single	384±10	128±7	147667±772	110514±512	17525±95
				Parallel	274±7	86±4	81245±299	62818±301	8513±43
	Hierarchical model	3 levels	2-2 (U)*	Single	235±6	84±7	114638±579	96903±437	621±36
			2-2 (U)*	Parallel	112±4	43±5	58423±66	44264±254	542±25

* (U) - Uniform distribution, (L) - Linear distribution, (Q) - Quadratic distribution, (E) - Exponential distribution.


REQUEST #1
450
400
350
300
250
200
150
100
50
0
73
143
235
Execution time, ms.
Execution time, ms.
Execution time, ms.
200k model
10M model
15M model
One-level model Single thread
One-level model Two treads
Hierarchical model Single thread
Hierarchical model Two treads


REQUEST #2
140
120
100
80
60
40
20
0
67
77
84
Execution time, ms.
Execution time, ms.
Execution time, ms.
200k model
10M model
15M model
One-level model Single thread
One-level model Two treads
Hierarchical model Single thread
Hierarchical model Two treads


REQUEST #3
160000
140000
120000
100000
80000
60000
40000
20000
0
706
44772
114638
Execution time, ms.
Execution time, ms.
Execution time, ms.
200k model
10M model
15M model
One-level model Single thread
One-level model Two treads
Hierarchical model Single thread
Hierarchical model Two treads


REQUEST #4
120000
100000
80000
60000
40000
20000
0
578
57316
96903
Execution time, ms.
Execution time, ms.
Execution time, ms.
200k model
10M model
15M model
One-level model Single thread
One-level model Two treads
Hierarchical model Single thread
Hierarchical model Two treads


REQUEST #5
20000
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
318
517
621
Execution time, ms.
Execution time, ms.
Execution time, ms.
200k model
10M model
15M model
One-level model Single thread
One-level model Two treads
Hierarchical model Single thread
Hierarchical model Two treads

The parameters of the experiment

Elements distribution by levels

1. Uniform distribution

3 level model

Model #1		Model #2	
Level 1	1 / 1 / 1	Level 1	1 / 1 / 1
Level 2	33333 / 1666667 / 2500000	Level 2	33333 / 1666667 / 2500000
Level 3	33333 / 1666667 / 2500000	Level 3	33333 / 1666667 / 2500000
Level 4: objects	33333 / 1666667 / 2500000	Level 4: options	33333 / 1666667 / 2500000

The graph structure from Level 0 to Level 4 is tree. The objects and options are linked to levels 2-3 according to the experiment conditions.

Every SPARQL request is executed 10 times and average time and deviation values are taken as result. To avoid SPARQL requests answers caching, the particular literals values were changed for per request.

Requests

One-level model

Request #1	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
	PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
	PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
	PREFIX my: <http://127.0.0.1/bg/ont/test1#>
	SELECT ?Object
	WHERE
	{
	?Object my:has_id "Object_10000"
	}

Request #2	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
	PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
	PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
	PREFIX my: <http://127.0.0.1/bg/ont/test1#>
	SELECT ?Object
	WHERE
	{
	?Object my:has_option_id "Option_10" .
	}

Request #3	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
	PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
	PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
	PREFIX my: <http://127.0.0.1/bg/ont/test1#>
	SELECT ?Object
	WHERE
	{
	?Object my:has_option_id ?Option_id .
	FILTER (?Option_id = "Option_9" || ?Option_id = "Option_10")
	}

Request #4	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema> PREFIX my: <http://127.0.0.1/bg/ont/test1#> SELECT (count(distinct ?Object) as ?count) WHERE

	{
	?Object my:has_option_id ?Option_id .
	FILTER (?Option_id = "Option_9" || ?Option_id = "Option_10")
	}
	GROUP BY ?Option_id

Request #5	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
	PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
	PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
	PREFIX my: <http://127.0.0.1/bg/ont/test1#>
	SELECT ?Object
	WHERE
	{
	?Object my:has_option_id ?Option_id .
	FILTER contains(?Option_id, "_200")
	}

3-level model, the source connected levels: 2-2

Request #1	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
	PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
	PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
	PREFIX my: <http://127.0.0.1/bg/ont/test1#>
	SELECT ?Object
	WHERE
	{
	?Object my:has_id "Object_10000"
	}

Request #2	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema> PREFIX my: <http://127.0.0.1/bg/ont/test1#> SELECT * WHERE { ?Object my:has_parent_id/my:has_parent_id/my:linked_to ?Core_2_Level_2 . ?Option my:has_parent_id/my:has_parent_id ?Core_2_Level_2 . ?Option my:has_id "Option_11" .

	} LIMIT 100

Request #3	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
	PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
	PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
	PREFIX my: <http://127.0.0.1/bg/ont/test1#>
	SELECT *
	WHERE
	{
	?Object my:has_parent_id/my:has_parent_id/my:linked_to ?Core_2_Level_2 .
	?Option my:has_parent_id/my:has_parent_id ?Core_2_Level_2 .
	?Option my:has_id ?Option_id .
	FILTER (?Option_id = "Option_8" || ?Option_id = "Option_11")
	}
	LIMIT 100

Request #4	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
	PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
	PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
	PREFIX my: <http://127.0.0.1/bg/ont/test1#>
	SELECT (count(distinct ?Object) as ?count)
	WHERE
	{
	?Object my:has_parent_id/my:has_parent_id/my:linked_to ?Core_2_Level_2 .
	?Option my:has_parent_id/my:has_parent_id ?Core_2_Level_2 .
	?Option my:has_id ?Option_id .
	FILTER (?Option_id = "Option_8" || ?Option_id = "Option_11")
	}
	GROUP BY ?Option_id

Request #5	PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> PREFIX xsd: <http://www.w3.org/2001/XMLSchema> PREFIX my: <http://127.0.0.1/bg/ont/test1#> SELECT ?Object WHERE { ?Object my:has_parent_id/my:has_parent_id/my:linked_to ?Option_2 . ?Option my:has_parent_id/my:has_parent_id ?Option_2 . ?Option my:has_id ?Option_id .

	FILTER contains(?Option_id, "_201") } LIMIT 100