

Auteurs

BOUBEKEUR Tahar
JOUHRI Anass

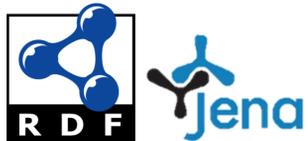
Encadrants

RADDAOUI Badran
ALBILANI Mohamad

Partenaires

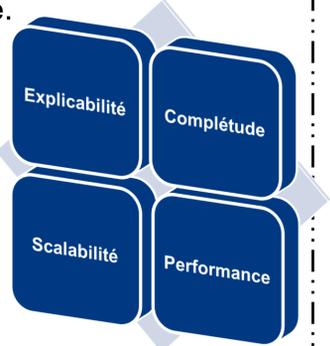


Technologies phares



Contexte :

- Les bases de connaissances **Triplestore** permettent le stockage, la modélisation et la liaison d'un volume important de données du web.
- Le manque de visibilité sur le schéma de la base rend l'exploitation de ces données très difficile.
- Les requêtes utilisateur peuvent renvoyer des **résultats vides ou incomplets** sans indiquer le motif d'échec.
- La recherche manuelle des causes d'échec peut être un processus fastidieux, long et frustrant.
- Plusieurs solutions ont été élaborées, mais elles ne satisfont pas les postulats suivants :



Objectifs :

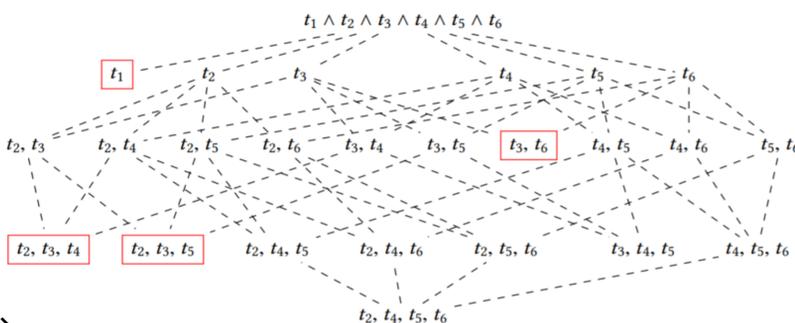
- Chercher les raisons derrière l'échec de la requête initiale (**Minimal Failing Subqueries** ou **MFSes**) ainsi que de les proposer à l'utilisateur comme explications.
- Retourner toutes les relaxations possibles (**Maximal Succeeding Subqueries** ou **XSSes**).
- Traiter des requêtes complexes sur des grandes masses de données.
- Optimiser le temps de réponse en minimisant l'interrogation de la base de données.

Notre solution :

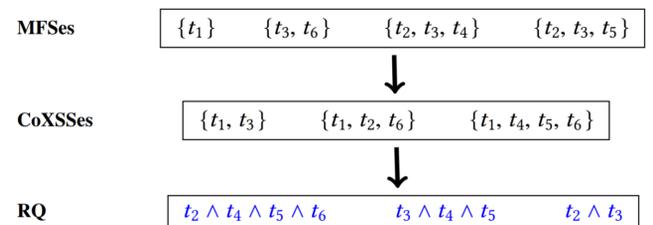
Recherche des MFSes :

Heuristique de recherche des sous-requêtes responsables de l'échec en :

- décomposant la requête initiale en sous-requêtes.
- évaluant quelques sous-requêtes :
 - **Échec :** Pas besoin de tester ses super-requêtes.
 - **Succès :** Continuer l'évaluation des super-requêtes.



Calcul des requêtes relaxées :



- Calculer les **Hitting Sets** des MFSes.
- Ces **Hitting Sets** sont le complément de toutes les relaxations possibles (**CoXSSes**).
- Déterminer les requêtes relaxées (**RQ**) en prenant le complément de chaque **Hitting Set**.

Evaluation expérimentale :

Notre solution comparée aux solutions existantes (LBA, MBA) :

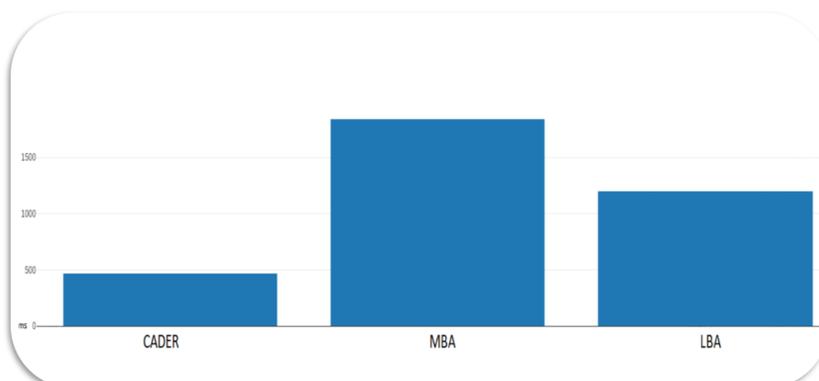


Fig 1 : Comparaison du temps d'exécution de la relaxation sur LUBM1k

Query	#Triples	LBA			MBA			CADER			
		#XSS	#MFS	#Ex. queries	#XSS	#MFS	#Ex. queries	#RQ	#MFS	#Ex. queries	
Star	Q ₁	3	1	2	90	1	2	90	1	2	6
	Q ₂	5	2	4	654	2	4	654	2	4	12
Chain	Q ₁	3	1	2	156	1	2	156	1	2	3
	Q ₂	5	1	3	540	1	3	570	1	3	8
Composite	Q ₁	3	1	1	102	1	1	54	1	1	4
	Q ₂	5	1	2	396	1	2	168	1	2	9

Fig 2 : Comparaison du nombre de requêtes exécutées sur LUBM10k