# Query Optimizations over Decentralized RDF Graphs

Ibrahim Abdelaziz, Essam Mansour, Mourad Ouzzani, Ashraf Aboulnaga, Panos Kalnis

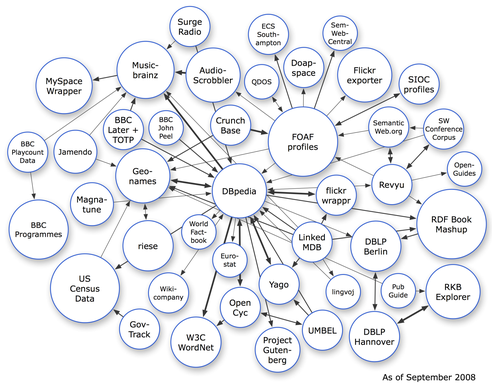
King Abdullah University of Science and Technology

Qatar Computing Research Institute

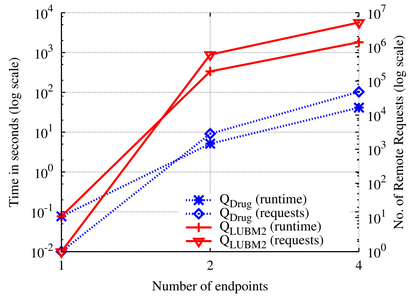
Hamad Bin Khalifa University

## Motivation

- Linked Open Data (LOD) contains >1,000 datasets with 130 billion triples and more than 500M interlinks.
- Emerging Applications in life sciences, decentralized social networks, and Internet of Things need to integrate data from multiple RDF datasets via SPARQL queries.
- Existing RDF systems have been proposed to support a small number of data sources by utilizing schema information.
- Objective: decompose queries to avoid unnecessary data communication and inefficient computation.

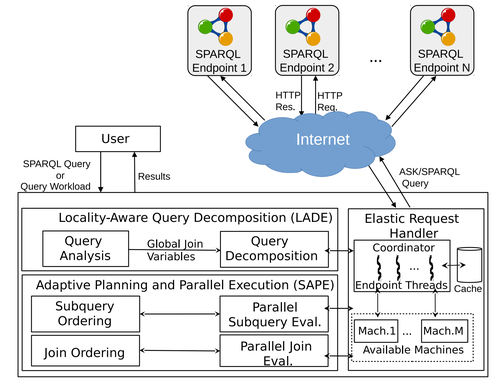



As of September 2008



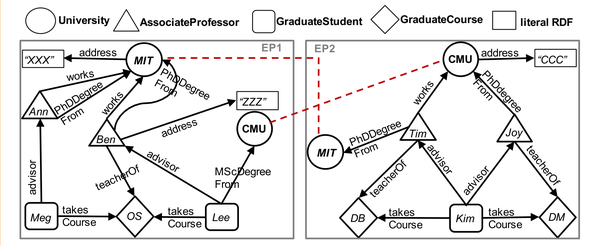

Scalability limitations of existing systems for querying linked data using FedX, the state-of-the-art system.

## Lusail



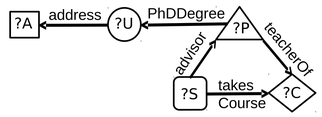

Lusail: a scalable and efficient federated RDF engine for querying linked data at scale.

## Example Data and Query



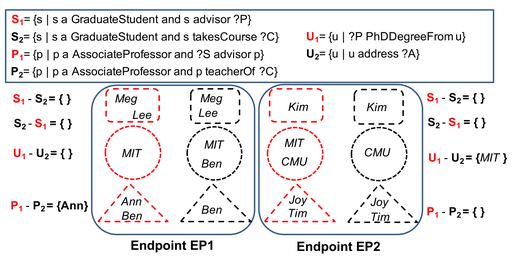

Decentralized graphs for two universities managed by geo-distributed SPARQL endpoints.

Example Query $Q$

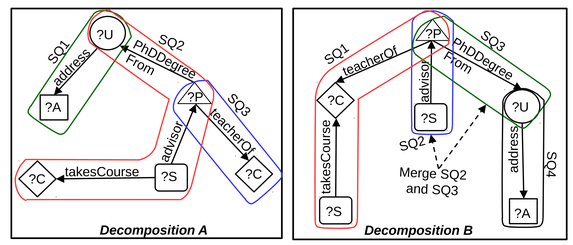



## Running Example

$S_1$= {s | s a GraduateStudent and s advisor ?P}
$S_2$= {s | s a GraduateStudent and s takesCourse ?C}
$P_1$= {p | p a AssociateProfessor and ?S advisor p}
$P_2$= {p | p a AssociateProfessor and p teacherOf ?C}
$U_1$= {u | ?P PhDDegreeFrom u}
$U_2$= {u | u address ?A}




Locality analysis of data instances matching ?S, ?U, and ?P in in $Q$



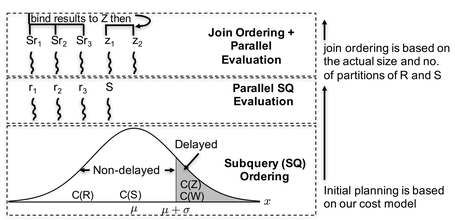

Two possible decompositions for $Q$ using detected global join variables ?U and ?P

## Locality Aware Decomposition

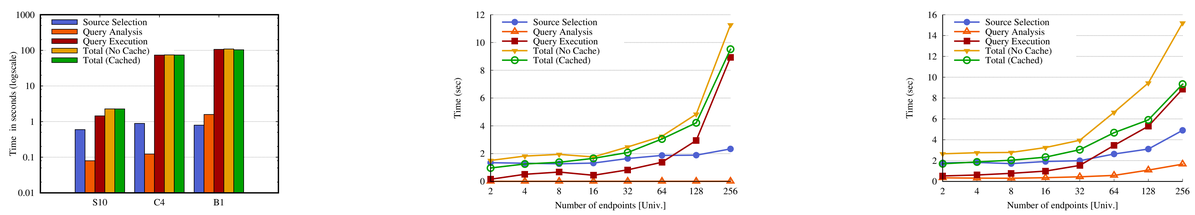
- Maximizes the computations at the endpoints and minimize intermediate results.
- Utilizes the knowledge of the locations of the actual RDF triple instances matching a query variable.
- Determines triple patterns that can be sent together to an endpoint.
- Decomposes the input query into a set of independent subqueries.

## Selectivity-Aware Execution

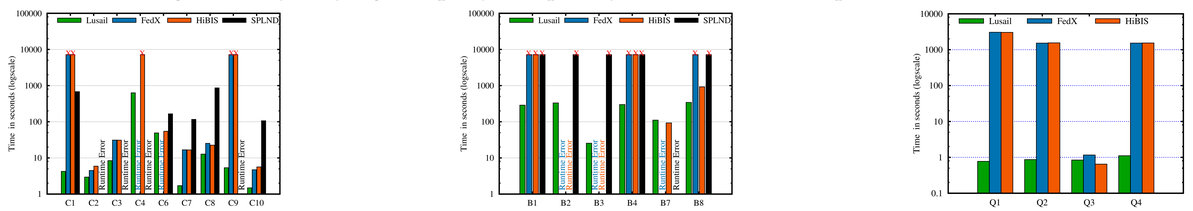
- Cost model delays subqueries expected to return large results.
- Achieve a high degree of parallelism while minimizing the communication cost.




## Evaluation




Profiling Lusail by varying the query complexity, the number of endpoints, and the data size.




Geo-distributed federation: endpoints deployed in 7 different regions of the Azure cloud.

## References

- I. Abdelaziz, E. Mansour, M. Ouzzani, A. Aboulnaga, P. Kalnis, "*Lusail: A System for Querying Linked Data at Scale*", PVLDB 11(4), 2018
- E. Mansour, I. Abdelaziz, M. Ouzzani, A. Aboulnaga, P. Kalnis, "*A Demonstration of Lusail – Querying Linked Data at Scale*", SIGMOD 2017 (Demo).