Parámetros del *benchmarking*.

Objetivo: Plantear las métricas a usar y la forma de cómo se llevará a cabo el *benchmarking.*

Resultados esperados: Documento con una lista de parámetros que serán usados en el *benchmarking*.

Definiciones

*Benchmarking*: Proceso metódico y continuo para evaluar servicios, productos y/o procesamientos de trabajo de organizaciones o empresas las cuales son bien reconocidas por llevar a cabo las mejores prácticas con el objetivo que mejorar o llevar a cabo estudios de referencia [1].

Plan de ejecución: Secuencia de operaciones en sistemas de administración de bases de datos para llevar cabo las sentencias que se requieran [2].

En el presente documento se explican las bases del *benchmarking* que se realizará una vez completado el módulo de consultas federadas para el software Apache Marmotta.

Michael Shmidt y sus colegas [3] en su *paper FedBench: A Benchmark Suite for Federated Semantic Data Query Processing* proponen un *benchmarking* que mida las capacidades de los gestores de consultas federadas. Fue elaborado basándose en otros *benchmarking* populares y citados*,* los cuales son *SP^2Bench* [4] y *Berlín SPARQL Benchmark* [5]*,* cuyo objetivo fueron evaluar consultas centralizadas. Este será el documento en el se basará para construir el *benchmarking* del actual proyecto.

Tal como lo establece Michael Schmidt y su equipo, no existe una evaluación que cubra todos los aspectos de un sistema de consultas. Sin embargo, ellos abordan la heterogeneidad a nivel de datos el cual, es uno de los retos que los sistemas de consultas tienen que abordar.

Heterogeneidad a nivel de datos se identifican en los siguientes aspectos

* Distribución física: Cómo se comportan los sistemas de consultas frente a *datasets* cuya ubicación física son distintas.
* Interfaz de acceso a datos: El acceso a los datos pueden ser repositorios nativos, *SPARQL endpoints* y datos en la nube del *Linked Data*.
* Existencia de fuentes de datos: Este aspecto solo es posible llevarse a cabo en la nube del *Linked Data* ya que a priori no se puede averiguar si la nube de datos contiene o no las fuentes de datos solicitadas en la consulta. De manera local, si se tiene el conocimiento sobre los campos que hay en el repositorio.
* Estadísticas de datos: De manera local, se puede conocer información de los repositorios mediante sus histogramas sobre propiedades, cantidad de datos y su distribución de datos. Sin embargo, cuando los repositorios no son locales, puede haber poca o nula información acera de ellos.

Aunado a lo anterior, también existen retos a nivel de consultas. Michael y sus colegas identificaron los siguientes aspectos a cubrir en la evaluación de consultas.

* Lenguaje de consulta: La expresividad entre lenguajes de consulta puede variar. Unos son simples, mientras que otros basan su sistemas en lenguajes de consultas muy específicos.
* Completitud de datos: Existen sistemas que pueden lidiar con una gran cantidad de información, pero también hay otros que no. Cabe decir que en la nube del *Linked Data* no es posible determinar la completitud de datos ya que un *triple store* puede desembocar en otros *triple store*, dando como resultado una interminable búsqueda de datos o quizá, un *triple store* no está referenciado correctamente a otros *triple store* resultando en resultados truncados.
* Lista de clasificación: Los sistemas pueden tener establecido una lista de mejores consultas basadas en sus propias métricas.

*FedBench* está basado en 3 componentes los cuales, pueden ser modificados, extendidos y personalizado para cubrir un escenario en específico. Los componentes son

* Múltiples *datasets*: Elaboración de *datasets* y estadísticas. Para este proyecto, los *datasets* serán sobre datos geoespaciales.
* Múltiples conjuntos de consultas: Elaboración de las consultas.
* *Framework* de evaluación

La continuación del *paper* es una profundización de los 3 componentes mencionados, pero con sus propios datos y consultas. Tal y como el mismo *paper* lo propone, para este proyecto terminal se adaptará los componentes para llevarlo a cabio con el software de Apache Marmotta una vez implementado el módulo.

Se determinó que el *paper* anterior no era suficiente para delimitar el *benchmarking* del proyecto actual por lo que se buscó otro.

Gabriela Montoya y su equipo en España desarrollaron un *paper* que cubriera las limitaciones del *benchmarking* que Michael había desarrollado. El *paper* B*enchmarking Federated SPARQL Query Engines: Are Existing Testbeds Enough?* [6]especifica más sobre *FedBench* y sobre los componentes que impactan el rendimiento de sistemas de consultas federadas.

Con base a Montoya y su equipo, existen 2 variables que influyen en el sistemas de consultas federadas y son

* Independientes: Son aquellas características que debe de ser especificadas con el fin de asegurar que sean replicables los escenarios de evaluación. Las variables independientes que se proponen son
  + Consulta: Variable que definen las consultas en términos de su estructura, evaluación y expresividad del lenguaje. Está basada en 3 aspectos
    - Forma del plan de ejecución: La forma en que los planes de ejecución se generan en los sistemas pueden variar, por lo que tiene un importante impacto en las 3 variables dependientes.
    - Número de patrones de tripletas básicas en la consulta: Este aspecto afecta directamente al plan de ejecución debido a que depende del número de tripletas que se recuperan en función de las tripletas especificadas en la consulta.
    - Instancias y posición en las tripletas: Este aspecto está relacionado a cómo están relacionados el sujeto, el objeto y el predicado en las tripletas. Puede tener un gran impacto en la evaluación este aspecto ya que, por ejemplo, el predicado de una tripleta puede no estar instanciado dando lugar a que diversos *endpoints* respondan cuando una consulta requiera información de dicho predicado.
* Dependientes: Son las características que están influenciadas por las variables independientes. Las variables dependientes son las que serán medidas en la evaluación. Las variables dependientes que se proponen son
  + Tiempo de selección del *endpoint:* Tiempo transcurrido entre la realización de la consulta y la generación de la consulta federada, basado en el protocolo *SPARQL* 1.1, con los *endpoint* donde se llevarán a cabo las subconsultas.
  + Tiempo de ejecución: Esta variables se comprende de 3 elementos
    - Tiempo entre la realización de la consulta y la primer respuesta.
    - Distribución de tiempo en la recepción de respuestas de las consultas.
    - Tiempo total de ejecución.
  + Completitud de respuestas: Cantidad de respuestas recibidas comparados a los datos disponibles en los *endpoints* seleccionados.