### **Benchmarking Modern Graph Databases: Methodology and Feasibility Study**

#### **1. Introduction**

* **Purpose of the Document:**
  + To identify and evaluate benchmarking methods for modern graph databases.
  + To assess the resources required for implementing these benchmarks.
  + To evaluate the feasibility of utilising these benchmarks in our study.
* **Scope:**
  + Overview of modern graph databases.
  + Description of benchmarking methods from existing studies.
  + Resource requirements and feasibility analysis.

#### **2. Overview of Modern Graph Databases**

* **Brief Description of Graph Databases:**
  + Examples: Neo4j, Amazon Neptune, GraphBase, Dgraph, MemGraph
  + Importance in industry and research.

#### **3. Benchmarking Methods**

* **3.1 LDBC Social Network Benchmark (SNB)**
  + **Description:**
    - Realistic social network scenario.
    - Workloads: Interactive, Business Intelligence, Graph Analytics.
  + **Metrics:**
    - Queries per second (QPS), query latency, throughput, resource utilisation.
  + **Feasibility:**
    - In an existing study, the researchers used a machine with the following specifications:  
      AMD Ryzen 5 5600H CEZANNE processor, 3.3 Ghz;16Gb of RAM memory; NVidia Ampere Geforce RTX 3060 graphics card; 512Gb SSD disk;
    - We currently don’t possess such high level specifications on our machines and will have to request the Projects Lab team to assign us one of the computers.
* **3.2 HPC Scalable Graph Analysis Benchmark (HPC-SGAB) with R-MAT**
  + **Description:**
    - Evaluates graph database performance using R-MAT.
    - Includes various kernels for comprehensive analysis.
  + **Metrics:**
    - Query execution time, throughput, resource utilisation.
  + **Reference:**
    - Original experimental setup and relevance.
  + **Feasibility:**
    - High-performance servers with multiple cores, ample RAM, and fast storage (SSD preferred).
    - Example: Two Quad Core Intel Xeon E5410 at 2.33 GHz, 11 GB RAM, 2.25 TB disk.
  + **Currently We Have This:**
    - Normal computers with likely lower specifications (e.g., single/dual-core processors, 8 GB RAM, smaller HDD).
  + **We'll Need to Ensure:**
    - Sufficient processing power and memory.
    - Sufficient storage space, potentially using SSDs for better I/O performance.
* **3.3 LFR-Benchmark Generator**
  + **Feasibility:**
    - Machines available to us have the specifications required
    - However Java Virtual Machine needs to be installed
  + **Description:**
    - Designed for testing community detection algorithms, it produces networks with power-law degree distribution and implanted communities within the network
    - The benchmarks include traversal operations such as computing connected components using in- and out-going edges and breadth-first search (BFS) from a vertex​
    - The benchmark investigates whether storing intermediate state information in memory or within graph elements (using the property mechanism provided by the graph database) is feasible​
* **3.4 Graph Database Benchmark**
  + **Feasibility:**
    - Most feasible method as the framework is open source
    - Uses Blueprint-compliant databases which means that the lower level implementation details of each specific type of graph are hidden and the same interface can be used to run operations on multiple graph databases
  + **Drawbacks:**
    - Blueprints does not offer all types of functionalities to interact with graph databases. As discussed in the last meeting, we want to incorporate novelty into our research by comparing other capabilities of graphs as well such as link predictions through machine learning pipelines
    - Hence we can use this approach but will have to integrate certain other aspects for benchmarking the features we are interested in as wel