Graph pattern matching queries enable flexible graph exploration, similar to what SQL provides for relational databases. In this thesis, we design and improve key components of a distributed in-memory graph querying engine. First, we optimize a distributed depth-first search (DFS) asynchronous pattern matching algorithm by combining it with breadth-first search (BFS), thus improving the overall engine performance by leveraging strengths of both approaches: ability to strictly bound the consumed memory of DFS and better parallelization, locality, and load balancing of BFS. Second, we further extend the distributed pattern matching with a novel solution for reachability regular path queries (RPQs) that supports variable-length patterns based on regular expressions. Our design retains the underlying runtime characteristics, allowing for efficient memory control during path exploration with great performance and scalability. Third, we improve query planning, which is one of the most crucial aspects impacting the performance of any querying system. Choosing the “best” query plan is challenging due to the many aspects influencing the performance, especially in a distributed system. We present a lightweight mechanism for gathering runtime information, which can be used to select the most effective query plan that performs optimally in the actual environment where the engine executes.