Analytics Vidhya

Parallel Steps Execution in Agentic Systems in LangGraph

Dipanjan Sarkar

Head of Community & Principal AI Scientist at Analytics Vidhya

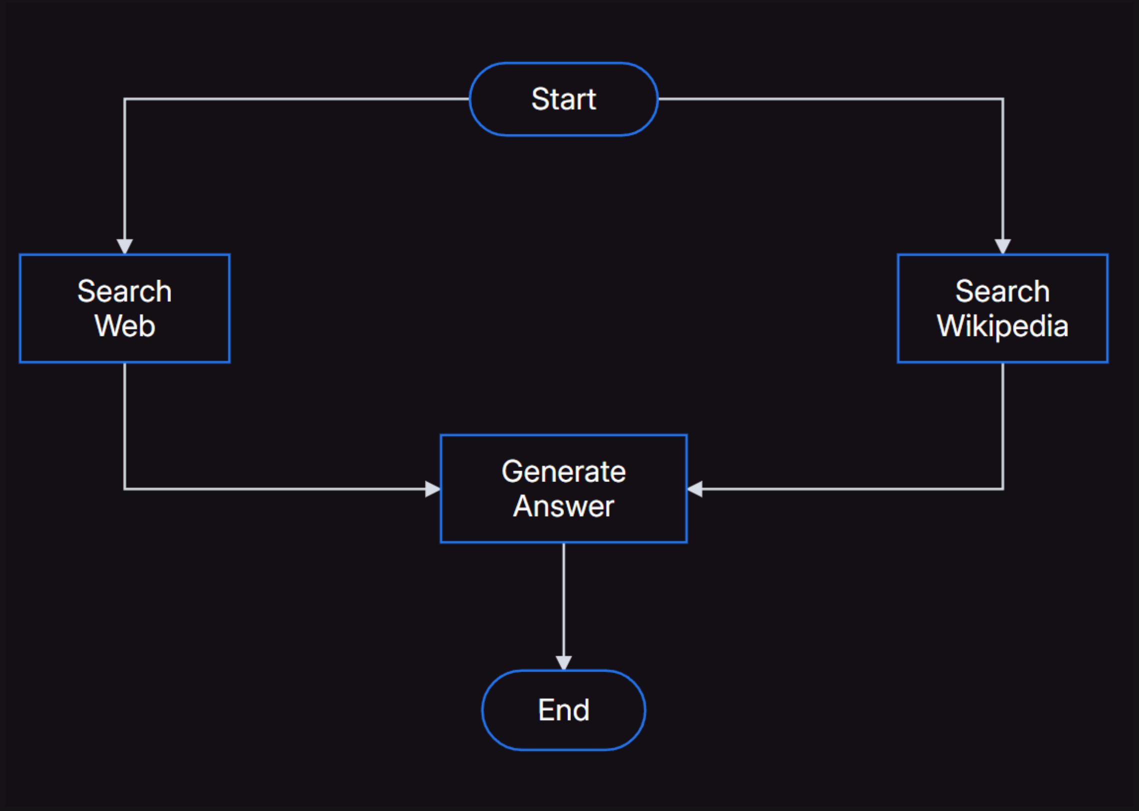
Google Developer Expert - ML & Cloud Champion Innovator

Published Author

Parallel Steps Execution in LangGraph

LangGraph provides robust support for parallel execution of nodes, enhancing the efficiency and performance of graph-based agentic workflows.

This parallelization is achieved through fan-out and fan-in mechanisms and can utilize standard edges or conditional edges.

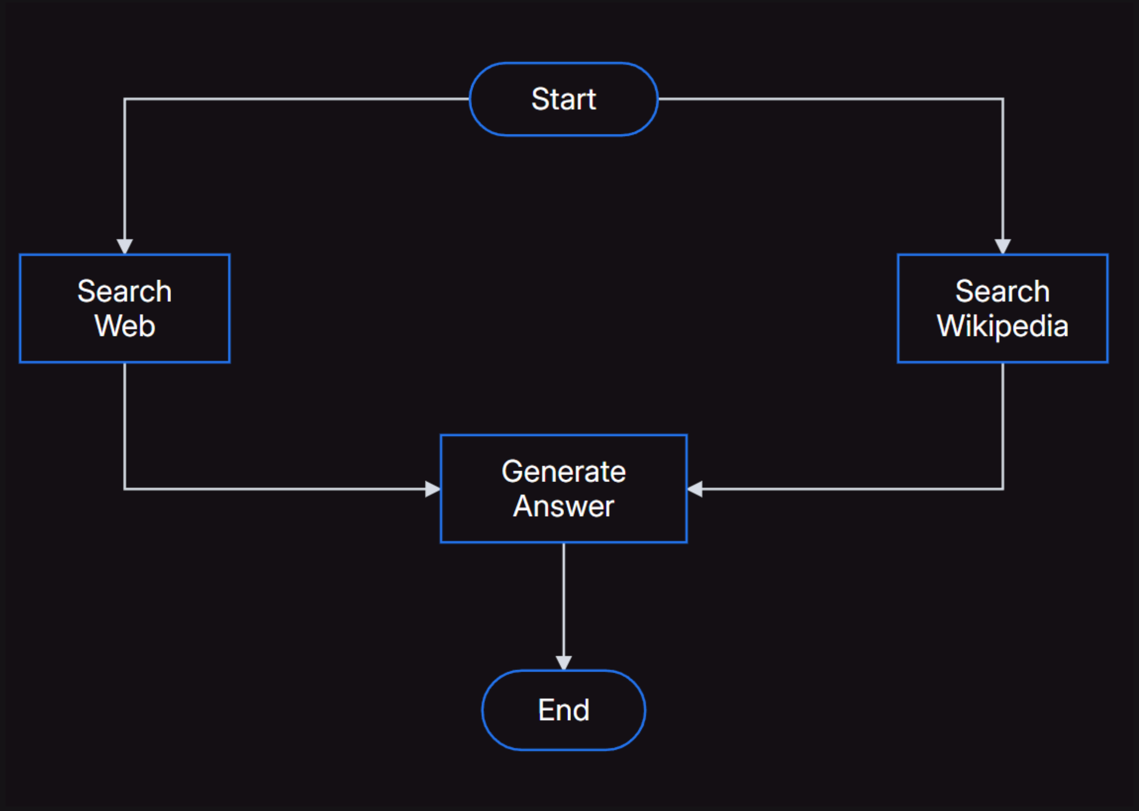


Analytics Vidhya

Parallel Steps Execution in LangGraph

Key Concepts

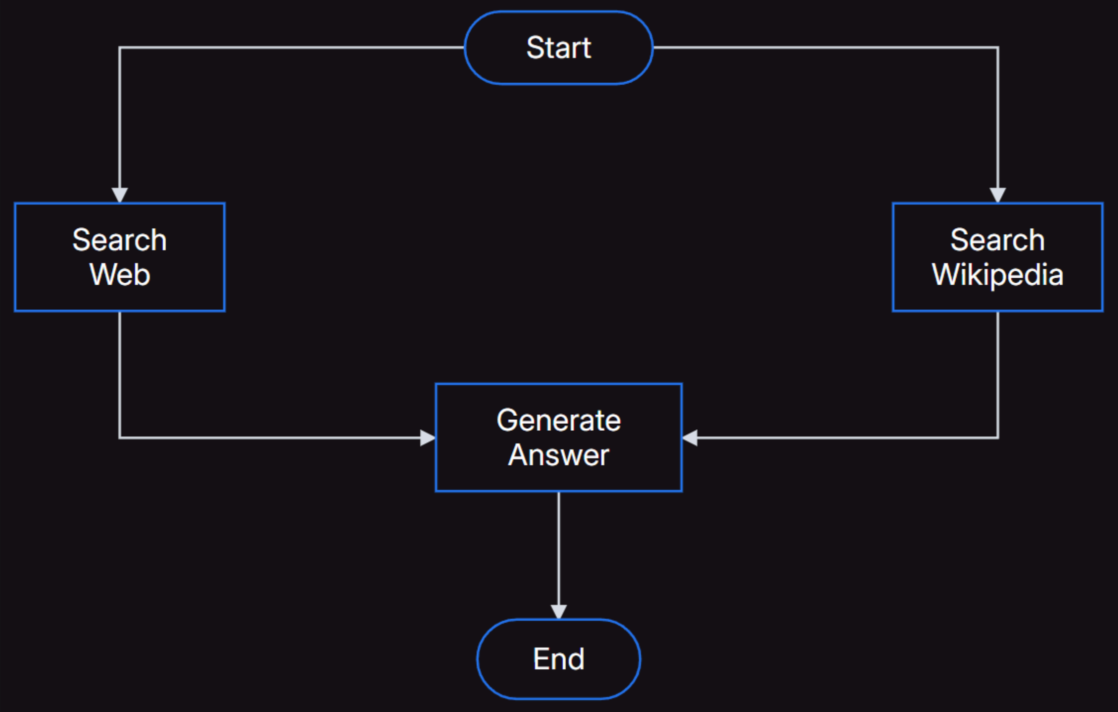
- Fan-Out
  - A process where a single node branches out to multiple nodes, allowing simultaneous execution of tasks. (Like a router node in router agents)
- Fan-In:
  - A process where multiple nodes converge back into a single node, aggregating the results from parallel tasks.



Parallel Steps Execution in LangGraph

Standard Workflow in LangGraph

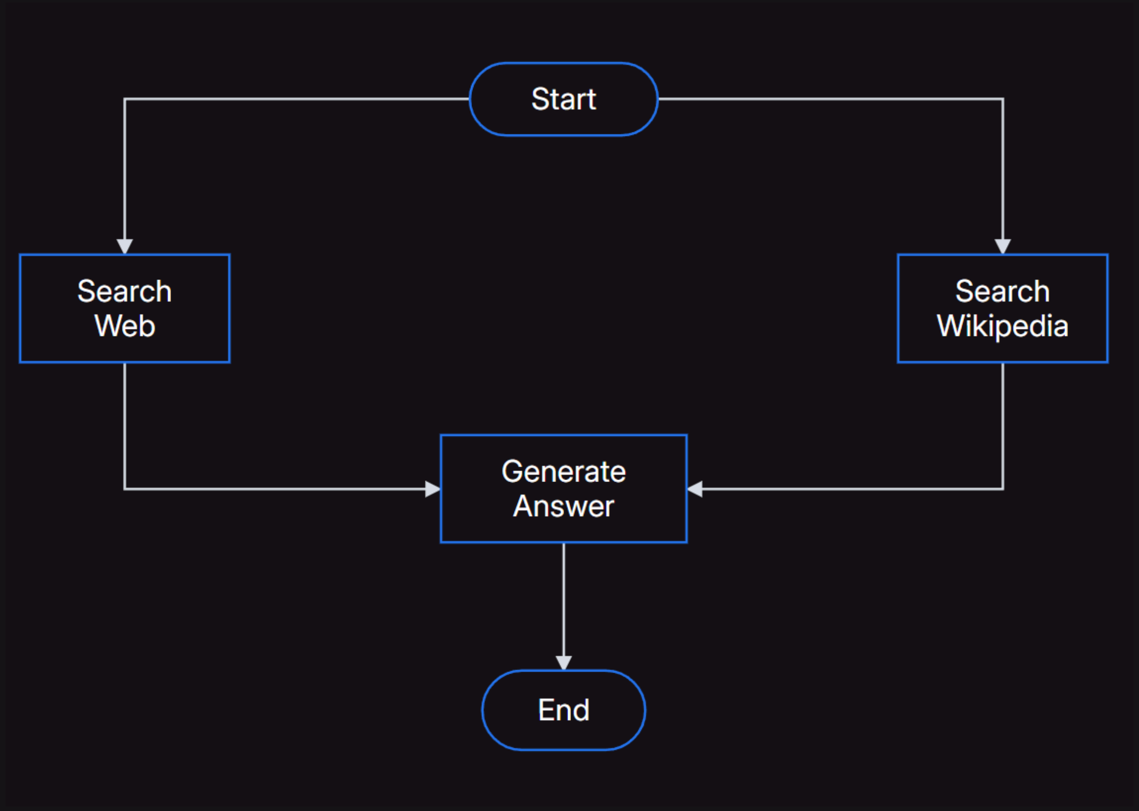
- Defining the Graph Structure
  - Create nodes representing distinct tasks or operations.
  - Use edges to define the flow between nodes, specifying fan-out points where tasks can be executed in parallel
- Utilizing Conditional Edges
  - Employ conditional edges to direct the flow based on specific conditions, enabling dynamic branching and parallel processing (ONLY in case you need routing)
- Aggregating Results
  - After parallel execution, use fan-in mechanisms to combine results from multiple nodes into a subsequent node for further processing or final output.



Analytics Vidhya

Parallel Steps Execution in LangGraph

NOTE: LangGraph will automatically wait till all the tasks in the parallel nodes have been executed before running the aggregation node



Analytics Vidhya

Thanks