Analytics Vidhya

Invoking vs. Streaming in LangGraph

Dipanjan Sarkar

Head of Community & Principal AI Scientist at Analytics Vidhya

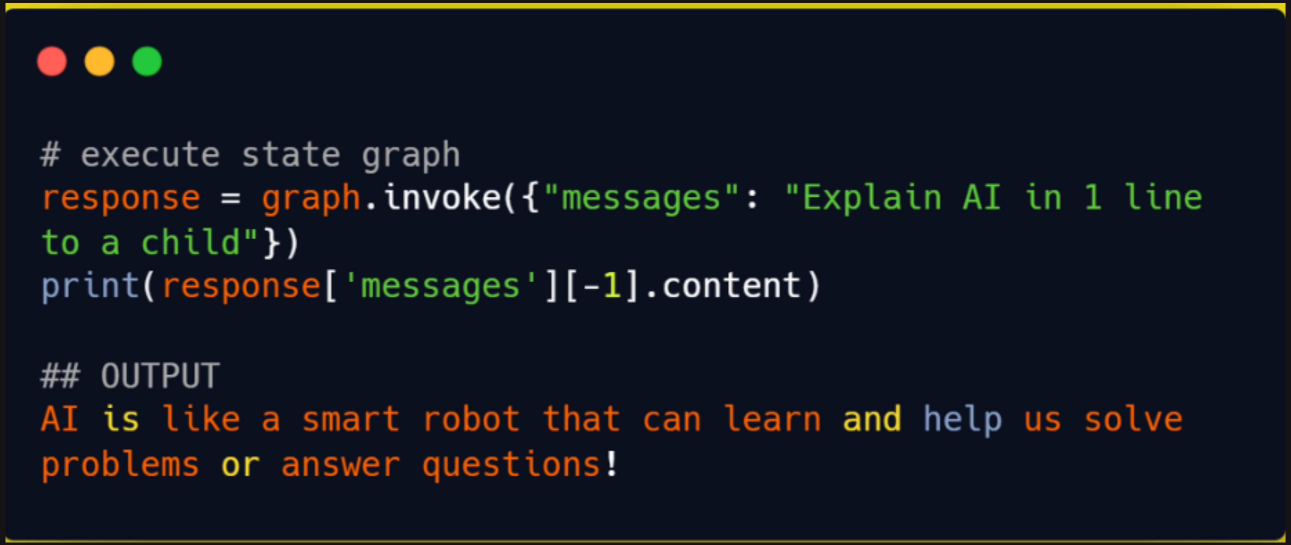
Google Developer Expert - ML & Cloud Champion Innovator

Published Author

Invoking in LangGraph

Invoking

- The invoke method executes the entire graph and returns the final state upon completion.
- This approach is straightforward but may lead to latency, especially with long-running processes, as users must wait until the entire execution finishes to receive any output.

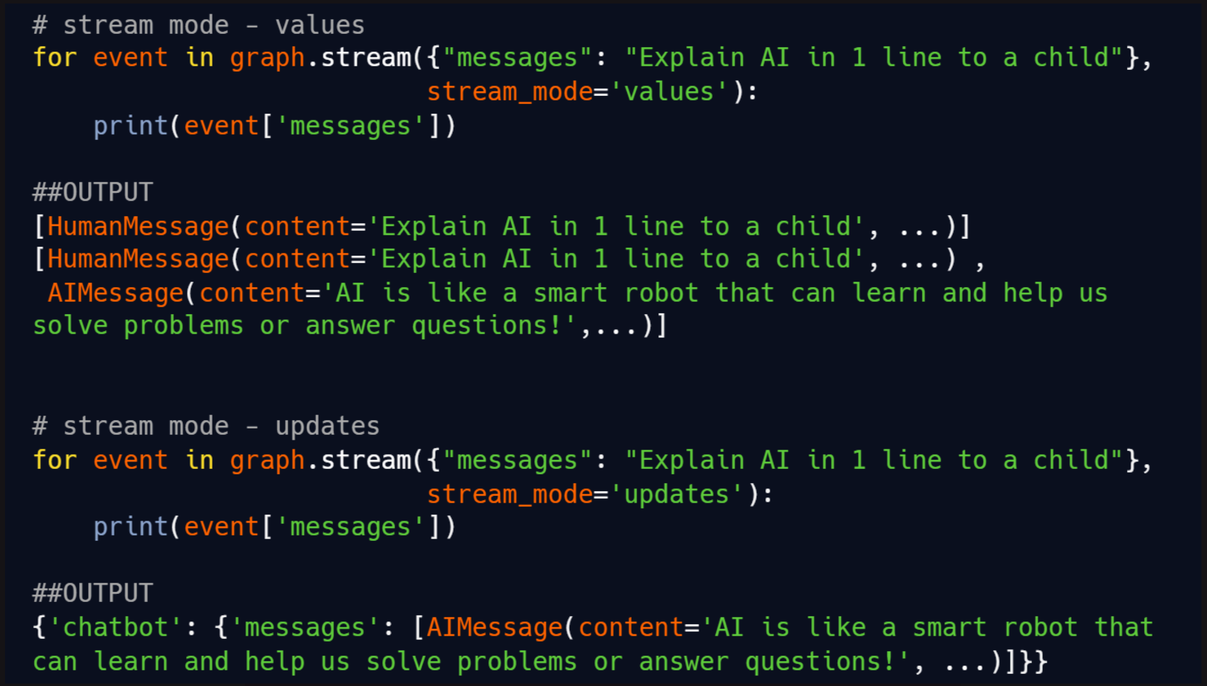


```
# execute state graph
response = graph.invoke({"messages": "Explain AI in 1 line
to a child"})
print(response['messages'][-1].content)

## OUTPUT
AI is like a smart robot that can learn and help us solve
problems or answer questions!
```

Streaming in LangGraph

- The stream method allows for real-time data emission during graph execution, providing intermediate outputs as they become available.

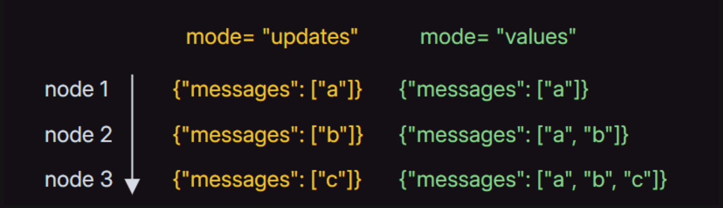


```
# stream mode - values
for event in graph.stream({"messages": "Explain AI in 1 line to a child"},
                          stream_mode='values'):
    print(event['messages'])

##OUTPUT
[HumanMessage(content='Explain AI in 1 line to a child', ...)]
[HumanMessage(content='Explain AI in 1 line to a child', ...) ,
 AIMessage(content='AI is like a smart robot that can learn and help us
solve problems or answer questions!',...)]


# stream mode - updates
for event in graph.stream({"messages": "Explain AI in 1 line to a child"},
                          stream_mode='updates'):
    print(event['messages'])

##OUTPUT
{'chatbot': {'messages': [AIMessage(content='AI is like a smart robot that
can learn and help us solve problems or answer questions!', ...)]}}
```

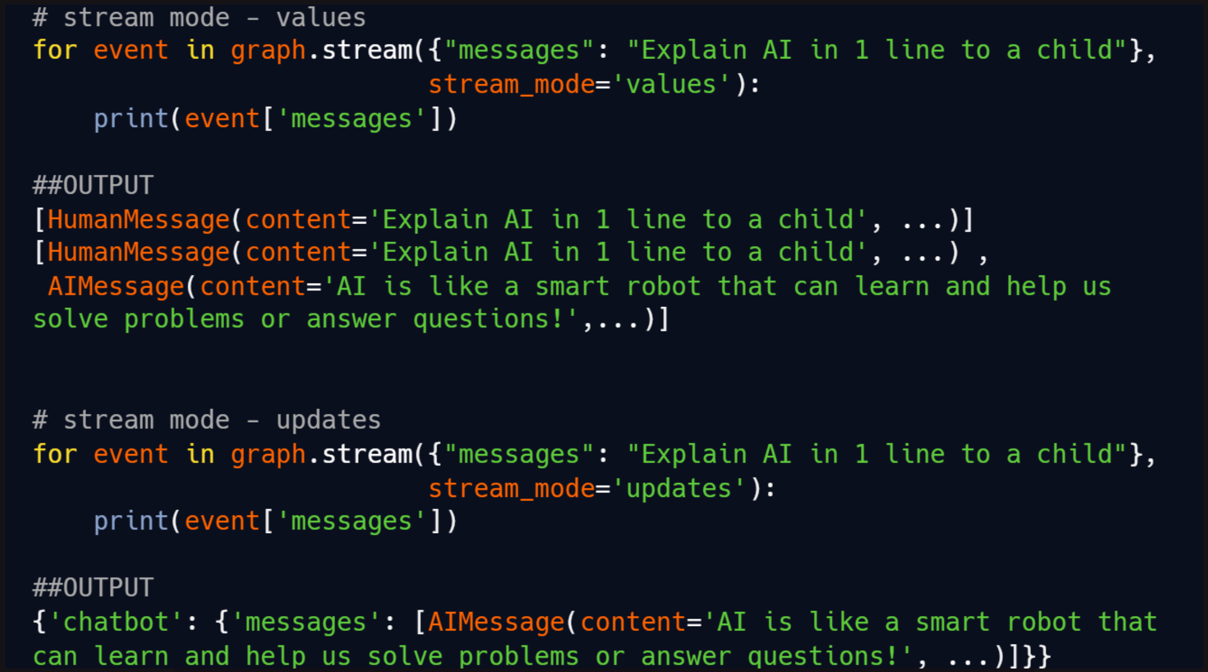


	mode= "updates"	mode= "values"
node 1	{"messages": ["a"]}	{"messages": ["a"]}
node 2	{"messages": ["b"]}	{"messages": ["a", "b"]}
node 3	{"messages": ["c"]}	{"messages": ["a", "b", "c"]}

Streaming in LangGraph

LangGraph supports multiple streaming modes to cater to different needs:

- Values
  - Streams the full state of the graph after each node execution.
- Updates
  - Streams only the updates to the state after each node execution.
- Messages
  - Streams LLM tokens and metadata for the graph node where the LLM is invoked.
- Custom
  - Streams custom data from inside your graph nodes.

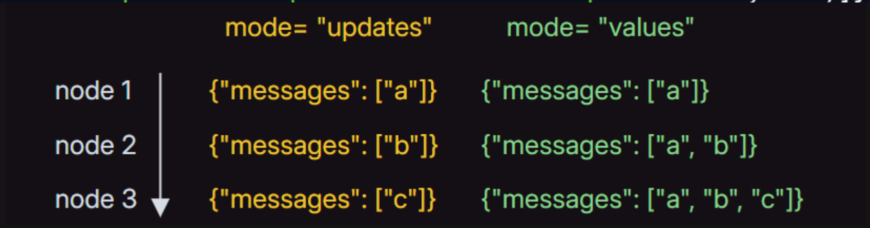


```
# stream mode - values
for event in graph.stream({"messages": "Explain AI in 1 line to a child"},
                          stream_mode='values'):
    print(event['messages'])

##OUTPUT
[HumanMessage(content='Explain AI in 1 line to a child', ...)]
[HumanMessage(content='Explain AI in 1 line to a child', ...) ,
 AIMessage(content='AI is like a smart robot that can learn and help us
solve problems or answer questions!',...)]


# stream mode - updates
for event in graph.stream({"messages": "Explain AI in 1 line to a child"},
                          stream_mode='updates'):
    print(event['messages'])

##OUTPUT
{'chatbot': {'messages': [AIMessage(content='AI is like a smart robot that
can learn and help us solve problems or answer questions!', ...)]}}
```

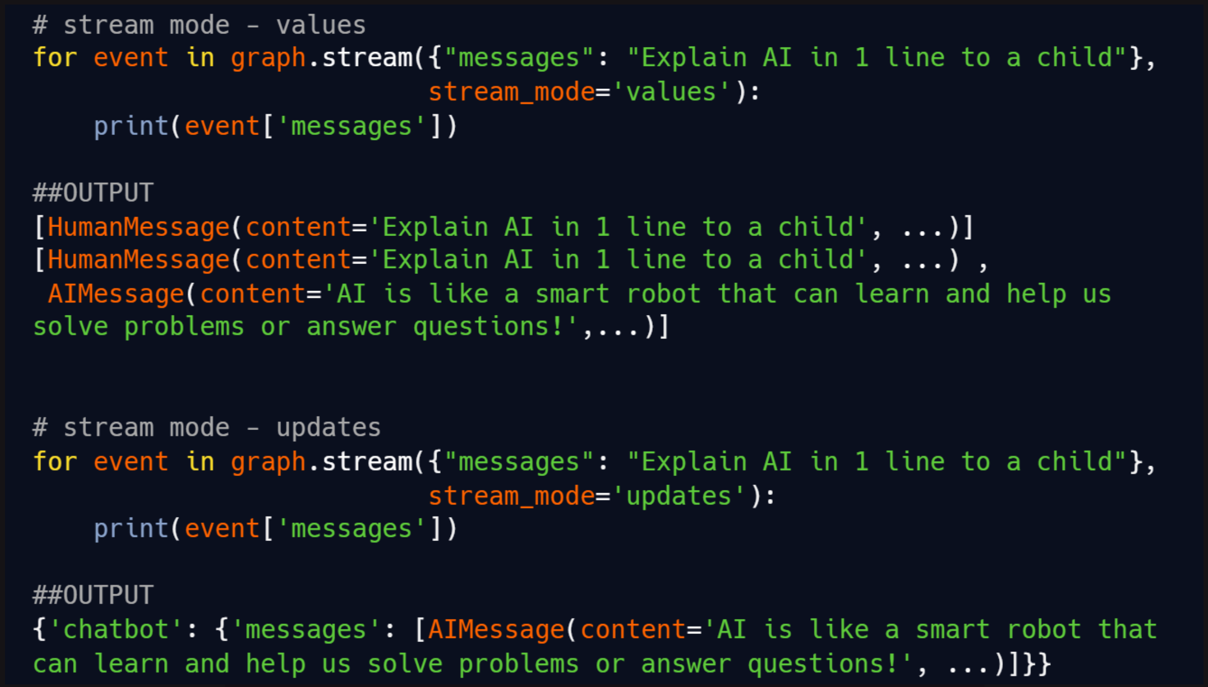


	mode= "updates"	mode= "values"
node 1	{"messages": ["a"]}	{"messages": ["a"]}
node 2	{"messages": ["b"]}	{"messages": ["a", "b"]}
node 3	{"messages": ["c"]}	{"messages": ["a", "b", "c"]}

Analytics Vidhya

Streaming in LangGraph

- By leveraging streaming, applications can provide users with immediate feedback, enhancing responsiveness and overall experience.
- This is particularly beneficial for long-running tasks, as users receive incremental updates rather than waiting for the entire process to complete.

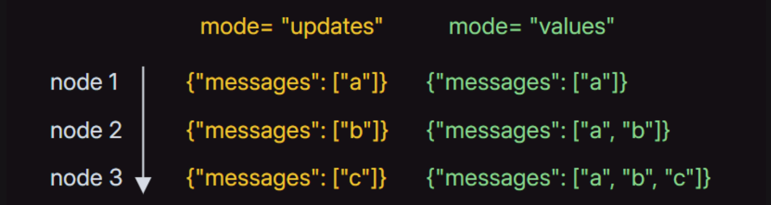


```
# stream mode - values
for event in graph.stream({"messages": "Explain AI in 1 line to a child"},
                          stream_mode='values'):
    print(event['messages'])

##OUTPUT
[HumanMessage(content='Explain AI in 1 line to a child', ...)]
[HumanMessage(content='Explain AI in 1 line to a child', ...) ,
 AIMessage(content='AI is like a smart robot that can learn and help us
solve problems or answer questions!',...)]


# stream mode - updates
for event in graph.stream({"messages": "Explain AI in 1 line to a child"},
                          stream_mode='updates'):
    print(event['messages'])

##OUTPUT
{'chatbot': {'messages': [AIMessage(content='AI is like a smart robot that
can learn and help us solve problems or answer questions!', ...)]}}
```



	mode= "updates"	mode= "values"
node 1	{"messages": ["a"]}	{"messages": ["a"]}
node 2	{"messages": ["b"]}	{"messages": ["a", "b"]}
node 3	{"messages": ["c"]}	{"messages": ["a", "b", "c"]}

Analytics Vidhya

Thanks