LangGraph

Qu'est-ce que LangGraph ?

LangGraph est un outil puissant pour créer des applications avec des modèles (LLMs) qui nécessitent des interactions complexes et des processus itératifs. Il étend la bibliothèque LangChain en ajoutant des capacités de calcul cyclique, permettant de coordonner plusieurs chaînes (ou acteurs) à travers différentes étapes de manière cyclique.

LangGraph expliqué

LangGraph améliore LangChain en introduisant la capacité de créer des calculs cycliques. Tandis que LangChain vous permet de définir des chaînes linéaires de calcul à l'aide de Graphes Acycliques Dirigés (DAGs),

Un **Graphe Acyclique Dirigé (DAG)** est un type de graphe utilisé en informatique et en mathématiques. Voici une explication simple :

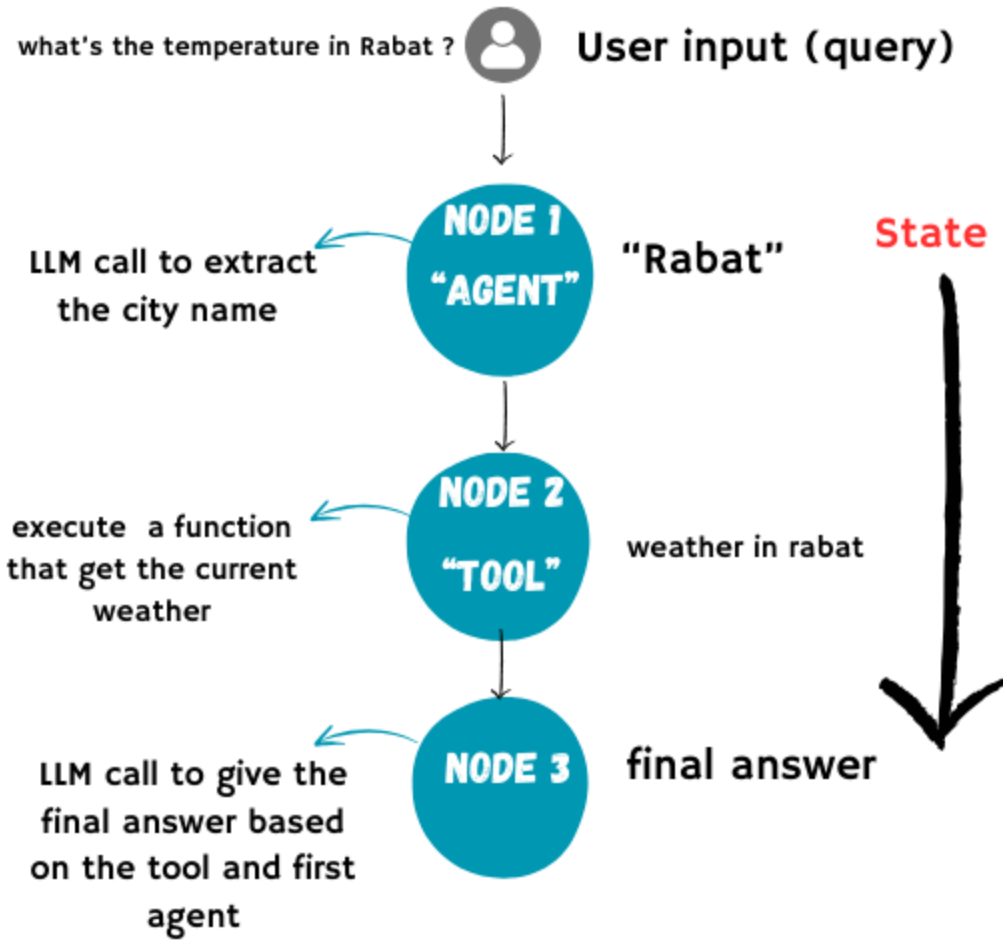
- **Dirigé :** Chaque connexion (ou edge) entre les nodes (ou sommets) a une direction, comme une rue à sens unique. Cela montre dans quel sens vous pouvez aller d'un node à un autre.
- **Acyclique :** Il n'a pas de cycles. Cela signifie que si vous commencez à un node et suivez les directions, vous ne pouvez jamais revenir au même node. Il n'y a pas moyen de rester bloqué dans une boucle.

LangGraph ajoute la capacité d'incorporer des cycles. Cela signifie que vous pouvez concevoir des comportements plus complexes, semblables à des agents, où vous appelez un LLM en boucle, lui demandant de décider quelle action entreprendre en fonction de l'état actuel.

En termes plus simples, LangGraph nous permet de gérer des applications où des décisions sont prises à chaque étape, et ces décisions peuvent influencer le flux du processus de manière dynamique. ..Cela rend possible la création de systèmes interactifs et adaptatifs qui gèrent des séquences de tâches plus complexes.

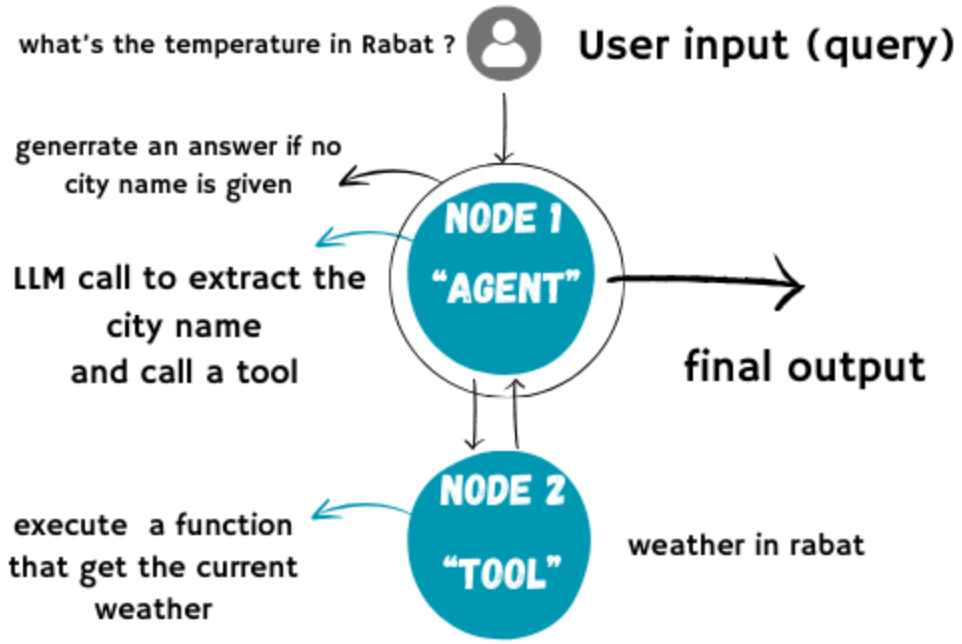
Concepts Clés :

- **Stateful Graph :** LangGraph utilise un stateful graph où chaque partie du processus (ou node) a un rôle spécifique, et l'état global du graph est mis à jour au fur et à mesure qu'il s'exécute.
- **Nodes :** Pensez aux nodes comme aux étapes individuelles de votre processus. Chaque node effectue une fonction spécifique, telle que la gestion des entrées, la prise de décisions ou l'interaction avec d'autres services.
- **Edges :** Les edges sont les connexions entre les nodes qui dictent le flux du processus. Avec LangGraph, vous pouvez avoir des edges conditionnels(if-else), ce qui signifie que le graph peut décider quel node exécuter ensuite en fonction de l'état actuel.



```
from langgraph.graph import Graph

workflow = Graph()


workflow.add_node("agent", function_1)
workflow.add_node("tool", function_2)
workflow.add_node("node 3", function_3)

workflow.add_edge('agent', 'tool')
workflow.add_edge('tool', 'responder')

workflow.set_entry_point("agent")
workflow.set_finish_point("node 3")

app = workflow.compile()
```

conditional egdes

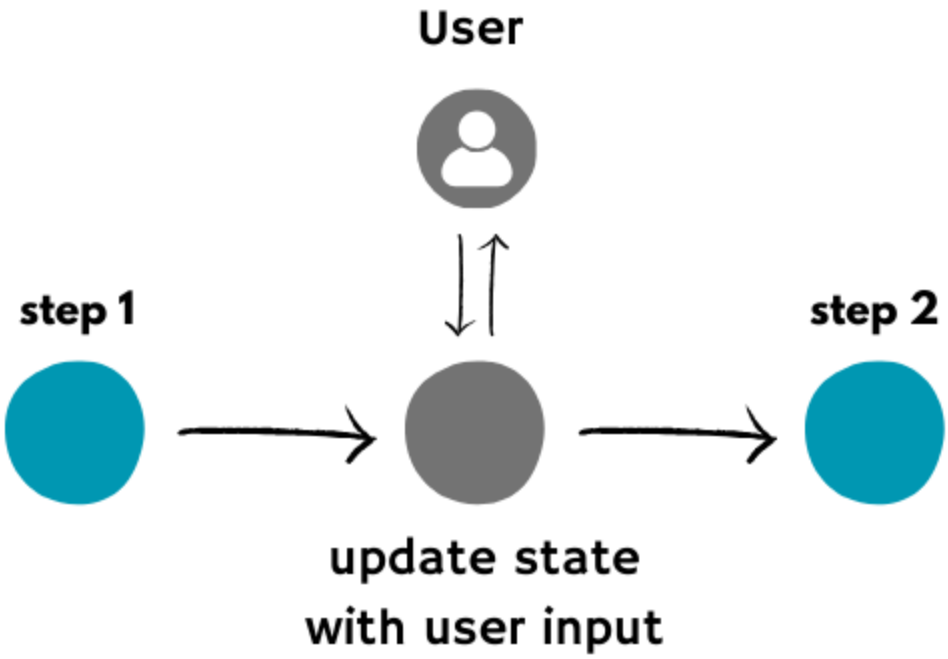


▼ USECASES

Interactions Human-in-the-loop (HIL)

Les interactions Human-in-the-loop (HIL) sont essentielles pour les systèmes agentiques. Attendre une entrée humaine est un modèle d'interaction HIL courant, permettant à l'agent de poser des questions de clarification à l'utilisateur et d'attendre une réponse avant de continuer.

Nous pouvons mettre en œuvre cela dans LangGraph en utilisant un point d'arrêt (breakpoint) : les points d'arrêt permettent d'arrêter l'exécution du graph à une étape spécifique. À ce point d'arrêt, nous pouvons attendre l'entrée humaine. Une fois que nous avons reçu une réponse de l'utilisateur, nous pouvons l'ajouter à l'état du graph et poursuivre l'exécution.



▼ Langchain vs LangGraph

Voici un tableau comparatif entre LangChain et LangGraph :

Caractéristique	LangChain	LangGraph
Architecture	Basée sur des chaînes linéaires	Basée sur des graphes dirigés
Gestion de l'état	Limité à des chaînes simples	Gestion automatique de l'état à travers plusieurs interactions
Modélisation	Modélisation linéaire des processus	Modélisation des processus sous forme de nœuds et d'arêtes
Complexité des applications	Adapté aux applications simples	Adapté aux applications complexes et multi-acteurs
Persistance des données	Non incluse	Incluse, permettant de sauvegarder et restaurer les états
Support des cycles	Non pris en charge	Prise en charge des processus cycliques
Contrôlabilité	Moins contrôlable	Priorité à la contrôlabilité, offrant un contrôle précis sur l'exécution
Coordination des agents	Coordination basique	Coordination avancée des agents, assurant l'ordre correct d'exécution
Utilisation	Interface standard pour interagir avec les modèles et autres composants	Cadre pour définir, coordonner et exécuter plusieurs agents LLM