日日是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2024年4月17日水曜日

Llama.cppとOllamaを使ってCommand R+をローカルのMacbookで動かしAPEXアプリから呼び出す

巷で話題のCohere（Cohere For AI）の**Command R+**を手元のMacbookで動かして、同じく手元のMacbookで動かしているAPEXアプリからアクセスしてみました。

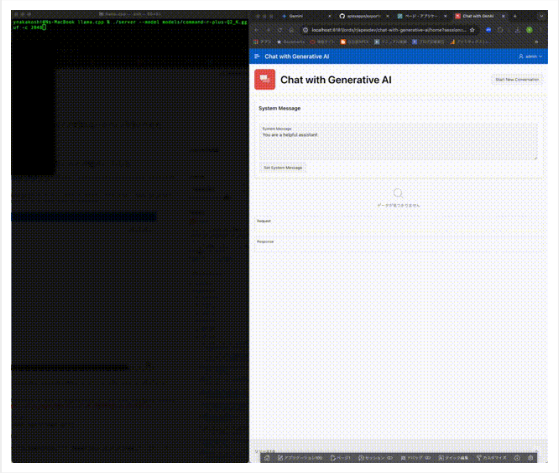
動かしたMacbookのスペックは少し古いですが、**M1 Max (64GB)**です。

Apple MシリーズのマシンでLlama.cppを使ってCommand R+を動かしてみようと思い立ったきっかけは、npakaさんの以下の記事です。

Llama.cpp で Command R+ を試す
https://note.com/npaka/n/n9136a2ebc7f9

npakaさんの記事ではM3 Max (128GB)とのことです。実行するマシンがそれよりスペックが低いので、4ビット量子化のモデルの代わりに2ビット量子化のモデル（**Q2_K - command-r-plus-Q2_K.gguf**）を使っています。

以下のように動作しました。



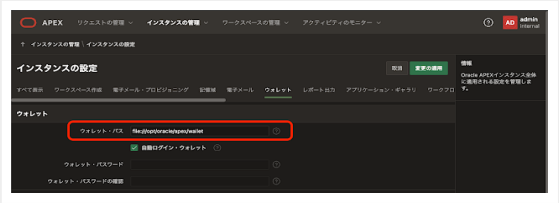
APEXの環境はColimaを使って実行しています。

Oracle Database 23c FreeのコンテナだけでOracle APEXを実行する
https://apexugj.blogspot.com/2024/03/oracle-apex-in-single-23c-free-container.html
データベース単体のコンテナへのOracle APEXのインストールを自動化する
https://apexugj.blogspot.com/2024/03/automate-installation-of-apex-on-database-free-container.html

API呼び出しができるように、データベースにACEを追加しています。

```
begin
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host => '*',
    ace => xs$ace_type(
       privilege_list => xs$name_list('connect'),
       principal_name => 'APEX_230200',
       principal_type => xs_acl.ptype_db));
  commit;
end;
/
```

また、**管理サービス**の**インスタンスの設定**に含まれる**ウォレット・パス**を設定しています。ウォレット・パスに設定したディレクトリ以下に**ewallet.p12**、**cwallet.sso**を配置しています。



APIサーバーは以下のコマンドで実行しています。

Llama.cppをコンパイルしたディレクトリより、

./server --model models/command-r-plus-Q2_K.gguf -c 2048

Llama.cppの代わりにOllamaも使ってみました。Ollamaは以下で実行します。

ollama run command-r-plus:104b-q2_K

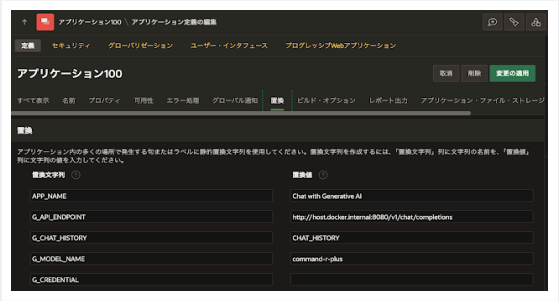
以下の記事で作成したAPEXアプリケーションを使っています。

OpenAIのChat Completions APIを呼び出すAPEXアプリを作成する
https://apexugj.blogspot.com/2024/04/chat-with-generative-ai-sample-app-0.html

同じアプリケーションで、OpenAI、Llama.cppそれとOllamaを呼び出せるように、パラメータを**置換文字列**として設定するように変更しています。

ローカルのMacでLlama.cpp＋Command R+（Q2_K）を動かして、APEXアプリから呼び出す際の設定です。

G_API_ENDPOINTとして**http://host.docker.internal:8080/v1/chat/completions**、
G_MODEL_NAMEとして**command-r-plus**（おそらくLlama.cppではモデル名は見ていない）を指定しています。



OpenAIのAPIを呼び出したときの設定です。

G_API_ENDPOINTは**https://api.openai.com/v1/chat/completions**、**G_MODEL_NAME**として**gpt-3.5-turbo**を指定しています。OpenAIのAPIを呼び出すにはAPIキーの指定が必要なので、**G_CREDENTIAL**として**Web資格証明**の**OPENAI_API_KEY**を指定します。**Web資格証明**はあらかじめ作成しておきます。

ローカルのMacでOllama＋command-r-plus:104b-q2_Kを動かしたときの設定です。

G_API_ENDPOINTは**http://host.docker.internal:11434/v1/chat/completions**、**G_MODEL_NAME**として**command-r-plus:104b-q2_K**を指定しています。

G_API_ENDPOINTとして**http://host.docker.intrernal:11434/api/chat**を指定すると、レスポンスの形式が少々変わります。

今回の作業は以上です。

MLXでも試してみたのですが、そちらの方はCommand R+を動かすにはリソースが足りず動きませんでした。Llama.cppではかろうじて動く感じです。

完

Yuji N. 時刻: 17:55

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

詳細プロフィールを表示