それってLocal LLMじゃダメですか？

―AIエージェントへの適用評価―

LINEヤフー株式会社
岡田 嘉, 大浦 晋

LINEヤフー

自己紹介

岡田 嘉

ネットワークユニット ネットワーク1ディビジョン

2023～ LINEヤフー(旧ヤフー株式会社)
ネットワークエンジニア、ネットワークエンジニア、ネットワークエンジニア

大浦 晋

ネットワークユニット ネットワーク1ディビジョン

2022～ LINEヤフー(旧ヤフー株式会社)
Local LLM 選定と構築を担当、自宅サーバー勢

Agenda

01

Local LLMとCloud LLM

02

Local LLMのはじめ方

03

Local LLMを試してみる

04

まとめ

Agenda

理解する

Local LLMを始めるにあたり、必要な事を知る

動かしてみる

実際にLocal LLMをAIエージェントに組み込んでみる

Local LLM と Cloud LLM

Local LLM

データ主権
データが外部に出ない

コスト構造
初期投資型、使用量に依存しない

制御性
モデル・設定・SLAを完全に制御可能

運用負荷
インフラ管理・メンテナンスが必要

Cloud LLM

即座に利用可能
インフラ構築不要

最新モデル
常に最新の高性能モデルにアクセス

スケーラビリティ
需要に応じて自動スケール

従量課金
使った分だけ支払い（予測困難）

Local LLMのはじめ方

Local LLMのはじめ方

LLM 本体

豊富な選択肢
様々なモデルがインターネット上で公開

商用利用可能
ライセンスが許可されたものも多数存在

実行環境

ハードウェア
コンピューター（GPU 推奨、CPU でも可）

ソフトウェア
推論実行を担うソフトウェアが必要

Local LLMのはじめ方

モデルの選択

用途に応じた性能

- 重視する能力による選択
  - コーディング能力
  - 日本語の正確さ

ライセンスの確認

- 商用利用の可否
- 企業規模による制約
- 社内ポリシーとの整合性

継続的な情報収集

Hugging Face Hub で多数公開されているため定期的に確認

Local LLMのはじめ方

実行環境の選択：ハードウェア

GPU 構成

✓ **推奨**
強力な GPU 搭載で高速・高品質

CPU 構成

✓ **選択肢**
生成速度と引き換えにコスト面で優位

性能のポイント

テキスト生成における重要指標

⚠ **メモリ速度**
メモリの読み書き速度が重視される

CPU 選択時のチェックポイント

- ✓ CPU 単体の性能
- ✓ メモリの帯域幅
- ✓ CPU のメモリチャンネル数

Local LLMのはじめ方

実行環境の選択：ソフトウェア

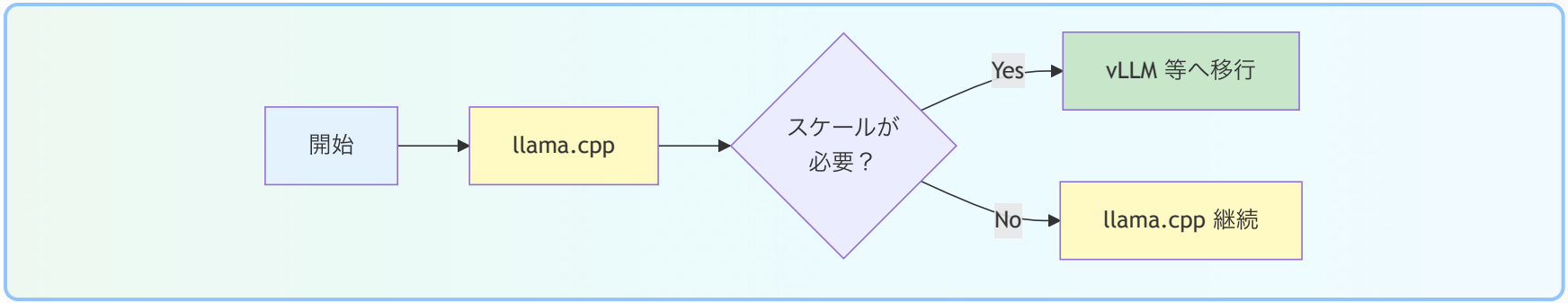
llama.cpp

推奨理由

- C++ のオープンソースで、多彩なハードウェアに対応
- テキスト生成速度も優秀

対応ハードウェア

- CPU only
- 各社GPU



LLMとメモリサイズ

パラメーター数と量子化

⚠ メモリと性能のトレードオフ

基本原則

- メモリ消費 ↗ ⇒ 品質 ↗ & 速度 ↘
- メモリサイズはLocal LLMの最大の制約

gpt-oss の例

- **20B モデル**：16GB 推奨
- **120B モデル**：80GB 推奨

量子化による削減

量子化とは

浮動小数点の精度を落とす事で、パラメーター数を維持したままメモリ消費を削減

120B モデルの圧縮例

- **16bit**：約 250GB
- **4bit**：約 60GB

実践的なモデル選定の考え方

私の選び方

モデルサイズ
まず 7B〜13B クラス

フォーマット
GGUF 形式が提供されている

ライセンス
商用利用可能

言語性能
日本語評価が一定以上

小さく試す

Local LLM を試してみる

Local LLM を試してみる

検証したハードウェア構成

GPU サーバー

NVIDIA A100 80GB × 1

- ✓ 大規模モデル対応
- ✓ 高負荷検証用

CPU サーバー

Xeon Gold 6138 仮想16コア

- ✓ 小〜中規模モデル
- ✓ 低コストでの検証用

Notice

- 仮想マシンでの利用のため、特にCPUサーバーは多少のオーバーヘッドの可能性がある

Local LLM を試してみる

試験パターン

Local LLMを AI Agentに組み込み、ネットワーク機器に対して行った検証

1 パース試験

LLMに装置のコマンド出力を読ませて値を取得

2 単純判断試験

Agentに装置のコマンド出力結果を読ませて値が規定値内か判断

3 単機能 Agent

Agentとして使い、MCPで取得した情報から判断

4 フロー処理 Agent

確認手順を渡し、手順に沿った確認を実行

Local LLM を試してみる

試験パターン

- ４ベンダーのNOSを対象に実施
- Agent試験においてはコマンドはパラメタ部以外を明示的に指定する
- 各ベンダーに対し、５シナリオを目安に実施する
- 出力は、jsonによる構造化出力、通常出力を織り交ぜて試験する
- 試行回数は20種程度×2回を目安に実施
- タイムアウトは1chatでレスポンス90秒、1agentで試行5分

Local LLM を試してみる

検証に用いた LLM

Local LLM

gpt-oss 20b q4_k_m (CPU)

- CPU Only 構成
- 最も安価な構成
- 期待通り動作すれば理想的

gpt-oss 120b (GPU)

- GPU 構成
- やや高コストな構成
- 期待に届けば上々

Cloud LLM

OpenAI o4-mini

- 推論機能が強い
- ラフな依頼でも的確
- 利用料が比較的安価

OpenAI GPT-4.1

- 大量のデータ処理に強い
- プロンプト調整が重要

Local LLMを試してみる

試験1:パース試験

LLM	正答率	実行時間
gpt-oss 20b q4_k_m (CPU)	70%(Timeout:30%)	27.1秒
gpt-oss 120b (GPU)	95%(誤答5%)	2.9秒
OpenAI o4-mini	95%(誤答5%)	5.7秒
OpenAI gpt-4.1 (temp=0.0)	95%(誤答5%)	2.2秒
OpenAI gpt-4.1 (temp=1.0)	95%(誤答5%)	1.9秒

- データのパース精度にはコマンド結果のjson/一般出力に差はなかった
  - ※1 試行において、全モデル解析に失敗したが、このパターンにおいてはjson出力がサポートされていないため比較できず
- コマンドの出力が長くなるとgpt-oss 20b q4_k_m (CPU)はタイムアウト(180sec)する事が多い
- コマンド出力が長くなると実行時間差は加速度的に平がる傾向
- この試験においてTemperatureは結果に差を与えない

Local LLMを試してみる

試験2:単純判断試験

LLM	正答率	実行時間
gpt-oss 20b q4_k_m (CPU)	58%(Timeout:42%)	27.1秒
gpt-oss 120b (GPU)	100%	4.9秒
OpenAI o4-mini	100%	6.9秒
OpenAI gpt-4.1 (temp=0.0)	100%	3.7秒
OpenAI gpt-4.1 (temp=1.0)	100%	4.7秒(最大17秒)

- データのパース精度にはコマンド結果のjson/一般出力に差はなかった
  - ※1 試行において、全モデル解析に失敗したが、このパターンにおいてはjson出力がサポートされていないため比較できず
- コマンドの出力が長くなるとgpt-oss 20b q4_k_m (CPU)はタイムアウトが顕著
- クラウドLLMは時々スパイク的に実行時間が長くなる傾向
- この試験においてTemperatureは結果に差を与えない

Local LLMを試してみる

試験3:単機能Agent

LLM	正答率	実行時間
gpt-oss 20b q4_k_m (CPU)	55%(Timeout:45%)	130.2秒
gpt-oss 120b (GPU)	100%	111.7秒
OpenAI o4-mini	100%	139.3秒
OpenAI gpt-4.1 (temp=0.0)	60%	72.3秒
OpenAI gpt-4.1 (temp=1.0)	60%	71.1秒

- 試行時間へはMCPのレスポンス時間も含む
- gpt-oss 120b (GPU)はAgentとしての利用では指示追従性が高い
- クラウドもモデルによる結果の差が顕著
- プロンプトリファインを行っていない試験ケースにおいて差が顕著
- この試験においてTemperatureは結果に差を与えない

Local LLMを試してみる

試験4:フロー処理Agent

LLM	正答率	実行時間
gpt-oss 20b q4_k_m (CPU)	0%(MaxTimeExe.)	N/A
gpt-oss 120b (GPU)	100%	96.1秒
OpenAI o4-mini	100%	140.3秒
OpenAI gpt-4.1 (temp=0.0)	80%	91.1秒
OpenAI gpt-4.1 (temp=1.0)	80%	91.3秒

- gpt-oss 20b q4_k_m (CPU)は全ての試行で実施されたjiraチケットの読み込みを完了できず失敗
- gpt-oss 120b (GPU)は1度、指示追従への追加指示が入ったが問題なく完了、十分実用できる
- OpenAI GPT-4.1 (temp=1.0)も1度、指示追従への追加指示が入ったが問題なく完了
- Temperatureは現実的な実行結果に差を与えないと考えてよさそう

まとめ

- gpt-oss 20bをCPUでAgentとして使うのはかなり難しい
- gpt-oss 120bをGPUでAgentとして使うのは十分実用（ただしシーケンシャル実行に限る）
- Agentやネットワーク機器のコマンド解釈では、LLMの性能は適度でよい
- プロンプトのLLMによるリファインは結果の向上に顕著な効果があった
- Agentの実行精度に非決定性はあまり関係がなさそう

Local LLMによるAI Agentの実用化結構いけそうかも

LINEヤフー