TÀI LIỆU HƯỚNG DẪN · LARGE MODEL INFERENCE & MEMORY OPTIMIZATION

CẨM NANG CÀI ĐẶT & SỬ DỤNGAIRLLM BY LYOGAVIN

Chạy Mô Hình 70B Trên GPU 4GB VRAM · Kỹ Thuật Layer-by-Layer Inference · Tối Ưu Bộ Nhớ Đỉnh Cao

⭐ GitHubRepository: https://github.com/lyogavin

vin/airllm(~33.800+Stars)👑Tácgiả/Tổchức:GavinLi(

Iyogavin) Giấyphép: Apache-2.0 **Côngnghệthenchốt:** Lay

er-wiseDiskStreaming, Block-by-blockExecution, Zer

- MemorySpikes **Mô hình hỗ trợ:** Llama 370B, Qwen72B, Mixtr

al8x7B, DeepSeek-V2, Falcon40B **Phản ứng tối thiểu:** 1 Ca

rd đồ họa phổ thông 4GB VRAM (GTX 1650, RTX 3050...) và ổ cứn

```
g
S
S
D
```

1. AIRLLM LÀ GÌ & PHÉP MÀU NÀO GIÚP CHẠY 70B TRÊN 4GB VRAM?

Theo cách tính thông thường, để nạp một mô hình 70 tỷ tham số (70B) ở định dạng 16-bit vào bộ nhớ, bạn cần ít nhất 140GB VRAM — tương đương 2 chiếc card đồ họa máy chủ NVIDIA A100 có giá hàng chục nghìn USD. Ngay cả khi lượng tử hóa (quantization) xuống 4-bit, bạn vẫn cần tối thiểu 35GB-40GB VRAM.

AirLLM do kỹ sư Gavin Li phát minh đã phá bỏ hoàn toàn định kiến này bằng kỹ thuật phân lớp suy luận (Layer-by-Layer Inference).

- **Nguyên lý truyền luồng theo lớp (Streaming Layers):** Mô hình Transformer thực chất gồm 80 lớp xếp chồng lên nhau. AirLLM chỉ nạp ĐÚNG 1 lớp vào VRAM tại một thời điểm, thực hiện xong phép tính thì giải phóng ngay và nạp lớp tiếp theo từ SSD vào.
- **Giữ nguyên 100% chất lượng mô hình gốc:** Không bắt buộc phải nén lượng tử hóa làm giảm độ thông minh; bạn có thể chạy mô hình ở độ chính xác đầy đủ FP16.
- **Chạy được trên cả GPU phổ thông hoặc MacBook cũ:** Card màn hình chỉ cần 4GB VRAM là đủ chỗ trống cho 1 lớp đơn lẻ tính toán.

2. YÊU CẦU HỆ THỐNG

- **GPU:** Bất kỳ card NVIDIA nào có tối thiểu 4GB VRAM hỗ trợ CUDA (GTX 1060, RTX 2060, RTX 3060...).
- **Ổ cứng (Bắt buộc):** Ổ SSD NVMe tốc độ đọc cao (khuyến nghị 3000MB/s+) và trống từ 150GB dung lượng để lưu weights mô hình.
- **RAM máy tính:** Tối thiểu 16GB RAM hệ thống.
- **Python:** Python 3.10+ cùng PyTorch phiên bản hỗ trợ CUDA.

3. HƯỚNG DẪN CÀI ĐẶT & CHẠY THỬ

Cài đặt gói AirLLM qua pip

```
p
i
p
i
n
s
t
a
l
l
a
i
```

```
r
l
l
m
```

Đoạn code Python suy luận mô hình 70B (test_airllm.py)

```
f
r
o
m
a
i
r
l
l
m
i
m
p
o
r
t
A
u
t
o
M
o
d
e
l

#
K
h
ở
i
t
ạ
o
m
ô
h
ì
n
h
L
l
a
```

```
ma-370B(tự động stream từ ổ pv vào 4GB VRAM) model=AutoMo
```

```
del.from_pretrained("meta-llama/Meta-Llama-3-70B-I
```

```
nstruct")

input_text=["Hãy giải thích chức năng ngôn ngữ của
```

```
của lực cháp dẫn theo thuyết tương đối rộng:"]

input_tokens
```

```
=
m
o
d
e
l
.
t
o
k
e
n
i
z
e
r
(
i
n
p
u
t
_
t
e
x
t
,

r
e
t
u
r
n
_
t
e
n
s
o
r
s
=
"
p
t
"
,
```

```
padding=True, truncation=True, max_length=128)

#B
```

```
ắtđầusuylluậngeneration_output=model.generate(
    inpu
```

```
t_tokens["input_ids"].cuda(), max_new_tokens=50, u
```

```
se_cache=True, return_dict_in_generate=True)

outpu
```

```
t=model.tokenizer.decode(generation_output.sequenc
```

```
es[0])print(output)
```

4. LƯU Ý VỀ TỐC ĐỘ & ĐÁNH ĐỔI HIỆU NĂNG

> **ĐÁNH ĐỔI GIỮA BỘ NHỚ VÀ TỐC ĐỘ**
> Vì p

hải liên tục đọc ghi các lớp mô hình từ ổ cứng SSD vào GPU, tốc

độ sinh token sẽ phụ thuộc trực tiếp vào tốc độ đọc của ứng

SSD (thường đạt khoảng 1–3 token/giây trên SSD NVMe). Phư

ơ
n
g
p
h
á
p
n
à
y
c
ự
c
k
ỳ
h
o
à
n
h
ả
o
c
h
o
v
i
ệ
c
c
h
ạ
y
b
a
t
c
h
j
o
b
,
t
ổ
n
g
h
ợ

ptàiliệu,xửlýnềnhoặcthửnghiệmnghiênc ứumàkhôngphả

> ibỏrahàngtrămtriệuđồngmuacardchuyêndụng.

Chúc bạn trải nghiệm sức mạnh của những siêu mô hình 70B ngay trên cỗ máy tính cá nhân cùng AirLLM!

Nội dung kỹ thuật tổng hợp từ tài liệu chính thức của repo https://github.com/lyogavin/airllm