```
!pip install fastapi
!pip install uvicorn
!pip install nest-asyncio
!git clone https://github.com/FasterDecoding/Medusa.git
!cd Medusa && pip install -e .
```

```
Collecting tiktoken (from fschat->medusa-llm==1.0)
  Downloading tiktoken-0.8.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.6 kB)
Requirement already satisfied: uvicorn in /usr/local/lib/python3.10/dist-packages (from fschat->medusa-llm==1.0
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-packages (from prompt-toolkit>=3.0.0->
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.10/dist-packages (from rich>=10.
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from rich>=1
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->fscha
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->fschat->
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->fsch
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->fs
Requirement already satisfied: yarl<2.0,>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->fsch
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp
Requirement already satisfied: starlette<0.42.0,>=0.40.0 in /usr/local/lib/python3.10/dist-packages (from fasta
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (from pydantic
Requirement already satisfied: pydantic-core==2.23.4 in /usr/local/lib/python3.10/dist-packages (from pydantic-
Requirement already satisfied: anyio in /usr/local/lib/python3.10/dist-packages (from httpx->fschat->medusa-llm
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from httpx->fschat->medusa-l
Collecting httpcore==1.* (from httpx->fschat->medusa-llm==1.0)
  Downloading httpcore-1.0.6-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from httpx->fschat->medusa-llm=
Requirement already satisfied: sniffio in /usr/local/lib/python3.10/dist-packages (from httpx->fschat->medusa-l
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.10/dist-packages (from httpcore==1.*->
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch->
Collecting wavedrom (from markdown2[all]->fschat->medusa-llm==1.0)
  Downloading wavedrom-2.0.3.post3.tar.gz (137 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 137.7/137.7 kB 6.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting latex2mathml (from markdown2[all]->fschat->medusa-llm==1.0)
  Downloading latex2mathml-3.77.0-py3-none-any.whl.metadata (14 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from reques
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->tr
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.10/dist-packages (from uvicorn->fschat->med
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py>=2.2.
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio->httpx->fs
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from yarl<2.0,>=1.1
Collecting svgwrite (from wavedrom->markdown2[all]->fschat->medusa-llm==1.0)
  Downloading svgwrite-1.4.3-py3-none-any.whl.metadata (8.8 kB)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from wavedrom->markdown2[all]->f
Downloading fschat-0.2.36-py3-none-any.whl (256 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 256.9/256.9 kB 16.4 MB/s eta 0:00:00
Downloading httpx-0.27.2-py3-none-any.whl (76 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.4/76.4 kB 7.4 MB/s eta 0:00:00
Downloading httpcore-1.0.6-py3-none-any.whl (78 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 78.0/78.0 kB 7.4 MB/s eta 0:00:00
Downloading nh3-0.2.18-cp37-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (769 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 769.2/769.2 kB 38.5 MB/s eta 0:00:00
Downloading shortuuid-1.0.13-py3-none-any.whl (10 kB)
Downloading tiktoken-0.8.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 63.1 MB/s eta 0:00:00
Downloading latex2mathml-3.77.0-py3-none-any.whl (73 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 73.7/73.7 kB 7.5 MB/s eta 0:00:00
Downloading markdown2-2.5.1-py2.py3-none-any.whl (48 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 48.4/48.4 kB 4.2 MB/s eta 0:00:00
Downloading svgwrite-1.4.3-py3-none-any.whl (67 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.1/67.1 kB 6.3 MB/s eta 0:00:00
Building wheels for collected packages: medusa-llm, wavedrom
  Building editable for medusa-llm (pyproject.toml) ... done
```

```
import os
os.chdir('Medusa')
!pip install pyngrok transformers accelerate bitsandbytes
```

```
Collecting pyngrok
  Downloading pyngrok-7.2.0-py3-none-any.whl.metadata (7.4 kB)
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (4.44.2)
Requirement already satisfied: accelerate in /usr/local/lib/python3.10/dist-packages (0.34.2)
Collecting bitsandbytes
  Downloading bitsandbytes-0.44.1-py3-none-manylinux_2_24_x86_64.whl.metadata (3.5 kB)
```

```
Requirement already satisfied: PyYAML>=5.1 in /usr/local/lib/python3.10/dist-packages (from pyngrok) (6.0.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from transformers) (3.16.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in /usr/local/lib/python3.10/dist-packages (from tran
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (1.26.4
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from transformers) (24
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers) (
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: safetensors>=0.4.1 in /usr/local/lib/python3.10/dist-packages (from transformers)
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transforme
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.10/dist-packages (from transformers) (4.66.5)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: torch>=1.10.0 in /usr/local/lib/python3.10/dist-packages (from accelerate) (2.5.0+
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub<
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggin
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerat
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->accelerate)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.10.0->acce
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1-
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->transforme
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->tran
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->tran
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.
Downloading pyngrok-7.2.0-py3-none-any.whl (22 kB)
Downloading bitsandbytes-0.44.1-py3-none-manylinux_2_24_x86_64.whl (122.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 122.4/122.4 MB 6.6 MB/s eta 0:00:00
Installing collected packages: pyngrok, bitsandbytes
Successfully installed bitsandbytes-0.44.1 pyngrok-7.2.0
```

```python
import nest_asyncio
nest_asyncio.apply()
```

```python
# need to add 2_ to two files inside model
# look for is_flash_attn_available in modelling_llama_kv.py, modelling_mistral_kv.py
# is_flash_attn_available -> is_flash_attn_2_available to fix the import error.

from fastapi import FastAPI, BackgroundTasks
from pydantic import BaseModel
from typing import List, Optional
import asyncio
from asyncio import Queue
from concurrent.futures import ThreadPoolExecutor
import torch
import numpy as np
from dataclasses import dataclass
import time
from transformers import BitsAndBytesConfig
from medusa.model.medusa_model import MedusaModel
from medusa.model.medusa_choices import mc_sim_7b_63
import os
from pyngrok import ngrok
from huggingface_hub import hf_hub_download
import uvicorn
from queue import Empty
```

```python
# Cell 3 - Define Models and Classes
class GenerationRequest(BaseModel):
    prompt: str
    max_tokens: int = 256
    temperature: float = 0.7
    top_p: float = 0.9
    stream: bool = False

class GenerationResponse(BaseModel):
    text: str
    generation_time: float

@dataclass
class BatchItem:
    def __init__(self, prompt, max_tokens, temperature, top_p, future):
        self.prompt = prompt
        self.max_tokens = max_tokens
        self.temperature = temperature
        self.top_p = top_p
        self.future = future


# Cell 4 - Initialize FastAPI and Global Variables
app = FastAPI(title="Medusa LLM Service")

BATCH_SIZE = 8
BATCH_TIMEOUT = 0.1  # seconds
request_queue = asyncio.Queue()
executor = ThreadPoolExecutor(max_workers=4)


def initialize_models():
    try:
        # Initialize model with Medusa config
        model = MedusaModel.from_pretrained(
            "FasterDecoding/medusa-vicuna-7b-v1.3",
            torch_dtype=torch.float16,
            low_cpu_mem_usage=True,
            device_map="auto"
        )

        tokenizer = model.get_tokenizer()

        if torch.cuda.is_available():
            torch.cuda.empty_cache()

        return model, tokenizer
    except Exception as e:
        print(f"Model initialization error: {e}")
        raise


# Cell 6 - Initialize Model
print("Initializing model... This may take a few minutes...")
model, tokenizer = initialize_models()
print("Model initialized successfully!")
```

```
Initializing model... This may take a few minutes...
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json:  100%                                             143/143  [00:00<00:00,  6.02kB/s]
config.json:  100%                                             566/566  [00:00<00:00,  39.8kB/s]
pytorch_model.bin.index.json:  100%                                        26.8k/26.8k  [00:00<00:00,  1.98MB/s]
Downloading  shards:  100%                                            2/2  [01:36<00:00,  44.82s/it]
pytorch_model-00001-of-00002.bin:  100%                                        9.98G/9.98G  [01:06<00:00,  226MB/s]
pytorch_model-00002-of-00002.bin:  100%                                        3.50G/3.50G  [00:29<00:00,  175MB/s]
tokenizer_config.json:  100%                                          727/727  [00:00<00:00,  55.7kB/s]
tokenizer.model:  100%                                          500k/500k  [00:00<00:00,  16.6MB/s]
special_tokens_map.json:  100%                                          435/435  [00:00<00:00,  25.0kB/s]
You are using the default legacy behaviour of the <class 'transformers.models.llama.tokenization_llama.LlamaTok
You are using the default legacy behaviour of the <class 'transformers.models.llama.tokenization_llama_fast.Lla
Loading  checkpoint  shards:  100%                                          2/2  [00:59<00:00,  27.44s/it]
Some weights of MedusaModelLlama were not initialized from the model checkpoint at lmsys/vicuna-7b-v1.3 and are
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
generation_config.json:  100%                                          132/132  [00:00<00:00,  9.65kB/s]
WARNING:accelerate.big_modeling:Some parameters are on the meta device because they were offloaded to the cpu.
medusa_lm_head.pt:  100%                                          1.48G/1.48G  [00:20<00:00,  33.9MB/s]
/content/Medusa/medusa/model/medusa_model.py:156: FutureWarning: You are using `torch.load` with `weights_only=
  medusa_head_state_dict = torch.load(filename, map_location=model.device)
Model initialized successfully!
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 1.1.weight: copying f
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 2.0.linear.weight: co
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 2.0.linear.bias: copy
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 2.1.weight: copying f
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 3.0.linear.weight: co
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 3.0.linear.bias: copy
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 3.1.weight: copying f
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 4.0.linear.weight: co
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 4.0.linear.bias: copy
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/torch/nn/modules/module.py:2400: UserWarning: for 4.1.weight: copying f
  warnings.warn(
```

```python
# Backend Processing Functions
from queue import Queue, Empty

async def process_batch(batch: List[BatchItem]):
    try:
        start_time = time.time()

        for i, item in enumerate(batch):
            with torch.inference_mode():
                input_ids = torch.as_tensor([tokenizer.encode(item.prompt)]).cuda()

                generations = model.medusa_generate(
                    input_ids,
                    max_steps=item.max_tokens,
                    temperature=item.temperature,
                    medusa_choices=mc_sim_7b_63,
                    top_p=item.top_p
                )

                generated_text = ""
                for output in generations:
                    generated_text = output["text"]

                generation_time = time.time() - start_time

                item.future.set_result({
                    "text": generated_text,
                    "generation_time": generation_time
                })

                if torch.cuda.is_available():
                    torch.cuda.empty_cache()

    except Exception as e:
        for item in batch:
            item.future.set_exception(e)

async def batch_processor():
    print("Batch processor started")
    while True:
        batch = []
        try:
            print("Waiting for first item...")
            # first_item = await asyncio.to_thread(request_queue.get)
            first_item = await request_queue.get()
            print("Got first item")
            batch.append(first_item)

            batch_deadline = time.time() + BATCH_TIMEOUT
            while len(batch) < BATCH_SIZE and time.time() < batch_deadline:
                try:
                    item = request_queue.get_nowait()
                    batch.append(item)
                except asyncio.queues.QueueEmpty:  # Use asyncio.queues.QueueEmpty
                    await asyncio.sleep(0.1)
                    continue

            print(f"Processing batch with {len(batch)} items...")
            await process_batch(batch)

        except Exception as e:
            print(f"Error in batch processing: {e}")
            for item in batch:
                item.future.set_exception(e)


# Cell 8 - FastAPI Endpoints
@app.on_event("startup")
async def startup_event():
    asyncio.create_task(batch_processor())

@app.post("/generate", response_model=GenerationResponse)
async def generate(request: GenerationRequest):
    print(f"Received request with prompt: {request.prompt}")
    future = asyncio.Future()

    item = BatchItem(
        prompt=request.prompt,
        max_tokens=request.max_tokens,
        temperature=request.temperature,
        top_p=request.top_p,
        future=future
    )

    print("Putting item in queue...")
    await request_queue.put(item)
    print("Item placed in queue")
    print("Waiting for result...")
    result = await future
    print(f"Got result: {result}")

    return GenerationResponse(**result)

@app.get("/health")
async def health_check():
    return {"status": "healthy"}
```

```
<ipython-input-10-dd842fd58458>:2: DeprecationWarning:
        on_event is deprecated, use lifespan event handlers instead.

        Read more about it in the
        [FastAPI docs for Lifespan Events](https://fastapi.tiangolo.com/advanced/events/).

   @app.on_event("startup")
```

```
# Cell 9 - Setup ngrok and Start Server
# Note: You need to sign up for ngrok and get an auth token
!ngrok authtoken 2o8EyZCZfWRWEc1nnKDe0ORtAJ4_6sgfuHGZWaek4cuZL7uS1
public_url = ngrok.connect(8000)
print(f"Public URL: {public_url}")

if __name__ == "__main__":
    config = uvicorn.Config(app, port=8000, log_level="info")
    server = uvicorn.Server(config)
    server.run()
```

```
Authtoken saved to configuration file: /root/.config/ngrok/ngrok.yml
INFO:     Started server process [489]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
Public URL: NgrokTunnel: "https://c395-34-82-224-117.ngrok-free.app" -> "http://localhost:8000"
Batch processor started
Waiting for first item...
Received request with prompt: Donald Trump is
Putting item in queue...
Item placed in queue
Waiting for result...
Got first item
Processing batch with 1 items...
Waiting for first item...
Got result: {'text': 'a president who has been widely criticized for his divisive rhet The 2019-2024 Outlook for
INFO:     223.178.85.206:0 - "POST /generate HTTP/1.1" 200 OK
WARNING:pyngrok.process.ngrok:t=2024-10-30T01:47:45+0000 lvl=warn msg="Stopping forwarder" name=http-8000-b9e6a37
INFO:     Shutting down
INFO:     Waiting for application shutdown.
INFO:     Application shutdown complete.
INFO:     Finished server process [489]
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-11-63fa228499fa> in <cell line: 7>()
      8     config = uvicorn.Config(app, port=8000, log_level="info")
      9     server = uvicorn.Server(config)
---> 10     server.run()

                                9 frames

/usr/local/lib/python3.10/dist-packages/uvicorn/server.py in capture_signals(self)
    330         # done LIFO, see https://stackoverflow.com/questions/48434964
    331         for captured_signal in reversed(self._captured_signals):
--> 332             signal.raise_signal(captured_signal)
    333
    334     def handle_exit(self, sig: int, frame: FrameType | None) -> None:

KeyboardInterrupt:
```

```
# Run this in VSCode while above cell running. Colab doesnt allow multiple cells to run simultaneously
import requests

def test_api():
    # test_prompt = "Once upon a time"
    test_prompt = "Donald Trump is"
    url = "https://c395-34-82-224-117.ngrok-free.app/generate" # Copy paste the ngrok URL here / not a neat way to do :

    try:
        response = requests.post(
            url,
            json={
                "prompt": test_prompt,
                "max_tokens": 50,
                "temperature": 0.7,
                "top_p": 0.9
            },
            timeout=300
        )

        # Print response details for debugging
        print(f"Status Code: {response.status_code}")
        print(f"Response Headers: {response.headers}")
        print(f"Raw Response: {response.text}")

        # Check if response is successful
        response.raise_for_status()

        try:
            return response.json()
        except requests.exceptions.JSONDecodeError as e:
            print(f"Failed to decode JSON: {e}")
            print(f"Response content: {response.content}")
            return None

    except requests.exceptions.RequestException as e:
        print(f"Request failed: {e}")
        return None

# Test the API
result = test_api()
if result:
    print("\nParsed JSON Response:", result)
```

Start coding or generate with AI.

Start coding or generate with AI.