Using LLMs on “Aristotelis” HPC infrastructure

George Vlahavas
Researcher @ Datalab, AUTh

Outline

- LLMs: Open source vs Proprietary
- How to run LLMs locally with Ollama
- How to run LLMs on Aristotelis HPC infrastructure
- Performance comparisons
- Comparison of different models: from small to large

LLMs: Open source vs Proprietary

- LLMs:
  Open source (local deployment)
  vs. Proprietary (e.g., ChatGPT)
- Pros & Cons
- Requirements for local LLMs

Proprietary LLMs: Pros & Cons

- **Pros**
  - **No need for infrastructure**: provided as a service
  - **Ease of use**: available UIs, APIs
  - **Accuracy**: trained with tons of data
  - **Scalability:** can handle high volumes of data, suitable for enterprise-level applications
- **Cons**
  - **Privacy:** Data (e.g., prompts, text for annotation) provided in proprietary LLM services (e.g., ChatGPT) become available at the company (e.g., OpenAI).
  - **Cost:** paid subscription for access, cost per API request
  - **Dependence on provider:** no access if the provider experiences issues
  - **Usage limits:** rate limits, token limits, monthly limits, etc
  - **Censored models:** will not respond to *any* prompt

Open-source LLMs (local deployment): Pros & Cons

- **Pros**
  - **Privacy:** Information & data stay local (at your system)
  - **Cost:** zero cost for usage
  - **Usage limits:** no limits for usage (apart from local hardware limitations)
  - **Uncensored models:** there are versions of the more popular models that will reply to any prompt
  - **Offline:** once you have downloaded a model, you don't need a network connection
  - **Customizability:** users can modify the models to their specific needs
  - **Enhanced security:** can be audited for security vulnerabilities
- **Cons**
  - **Need for infrastructure:** Infrastructure (typically powerful) is needed to support, in particular, large LLMs
    → …but, see "Aristotelis" HPC infrastructure
  - **Ease of use:** need to deploy & run service
    → … but, availability UIs/APIs availability & today's how-to manual
  - **Accuracy:** typically much smaller models
    → …but, things are progressing quite fast and fine tuning is possible

Requirements for Local LLMs

Model Parameters	RAM	HD Space
2B	4 GB	~1.5 GB
7B	8 GB	~4.5 GB
13B	16 GB	~7.5 GB
33B	32 GB	~20 GB
70B	64 GB	~40 GB

RAM can be VRAM or system RAM. Ollama will offload as many layers as can fit to the GPU and process the remaining with system RAM/CPUs.

How to run LLMs locally with *Ollama*

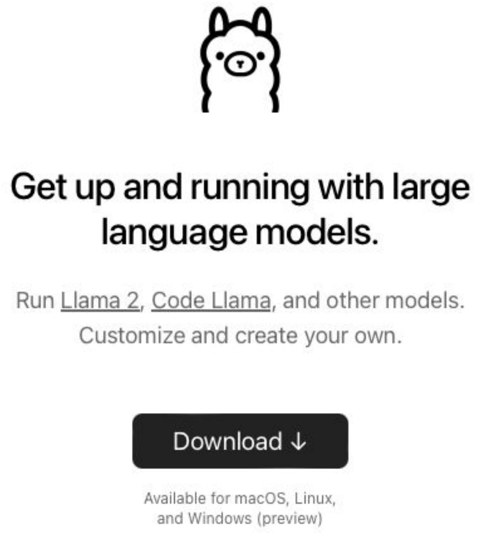
- The Ollama framework
- Installation & usage
- Local hardware limitations

What is Ollama?

- Ollama is a software that allows users to build and run LLMs locally. Its functionality is comparable to Docker, but for LLMs.
- It can be used in many ways: interactive shell, API, Python library…
- It contains a library of ~70 pre-built models that can be easily used in a variety of applications, including ***LLaMA3***, ***WizardLM2***, ***Mistral*** and ***Gemma***
- Will use a GPU if there is one, otherwise will fallback to CPU
- Website: https://ollama.com/
- Github: https://github.com/ollama/ollama

Using Ollama locally

- Download and install according to the instructions at https://ollama.com/download
- There are versions for:
  - Linux
  - macOS
  - Windows (preview)


Get up and running with large language models.

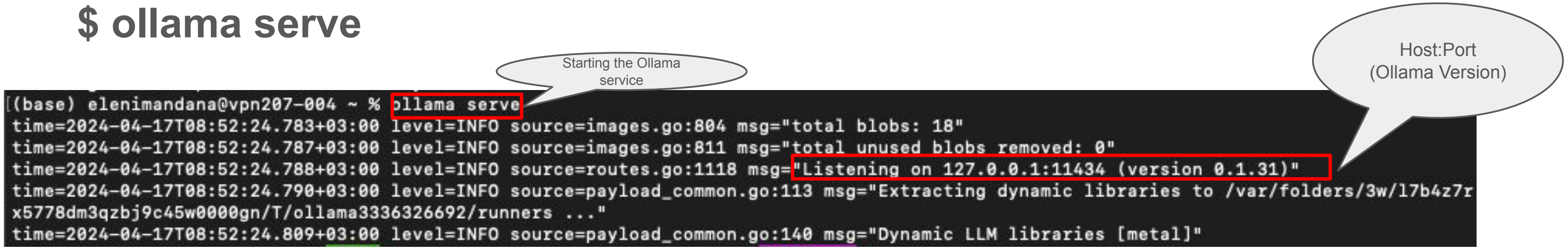
Run Llama 2, Code Llama, and other models. Customize and create your own.

Available for macOS, Linux, and Windows (preview)

Using Ollama locally - starting the service

- The Ollama service provides a locally deployed service that ollama clients can connect to
- To start the Ollama service, you can launch the desktop app (macOS/Windows)
- Or start the service from the command line (all platforms):

$ ollama serve



```
[(base) elenimandana@vpn207-004 ~ % ollama serve
time=2024-04-17T08:52:24.783+03:00 level=INFO source=images.go:804 msg="total blobs: 18"
time=2024-04-17T08:52:24.787+03:00 level=INFO source=images.go:811 msg="total unused blobs removed: 0"
time=2024-04-17T08:52:24.788+03:00 level=INFO source=routes.go:1118 msg="Listening on 127.0.0.1:11434 (version 0.1.31)"
time=2024-04-17T08:52:24.790+03:00 level=INFO source=payload_common.go:113 msg="Extracting dynamic libraries to /var/folders/3w/l7b4z7r
x5778dm3qzbj9c45w0000gn/T/ollama3336326692/runners ..."
time=2024-04-17T08:52:24.809+03:00 level=INFO source=payload_common.go:140 msg="Dynamic LLM libraries [metal]"
```

Using Ollama locally - managing models

- Download a model with:

  $ ollama pull <model_name>

  Example:

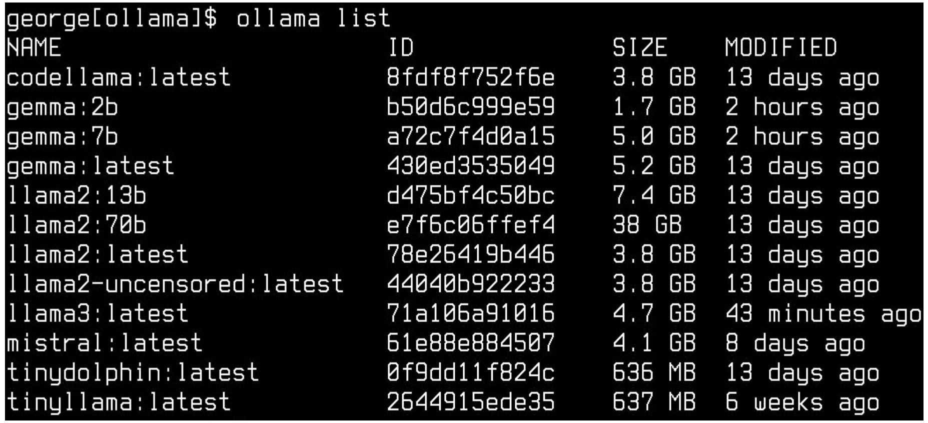
  $ ollama pull gemma:2b

- Get a list of all models you have downloaded:

  $ ollama list

- Remove a model that you no longer want:

  $ ollama rm <model_name>



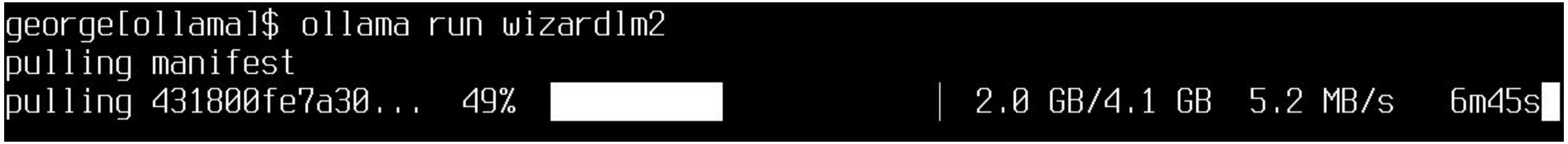
```
george[ollama]$ ollama list
NAME                          ID              SIZE    MODIFIED
codellama:latest              8fdf8f752f6e    3.8 GB  13 days ago
gemma:2b                      b50d6c999e59    1.7 GB  2 hours ago
gemma:7b                      a72c7f4d0a15    5.0 GB  2 hours ago
gemma:latest                  430ed3535049    5.2 GB  13 days ago
llama2:13b                    d475bf4c50bc    7.4 GB  13 days ago
llama2:70b                    e7f6c06ffef4    38 GB   13 days ago
llama2:latest                 78e26419b446    3.8 GB  13 days ago
llama2-uncensored:latest      44040b922233    3.8 GB  13 days ago
llama3:latest                 71a106a91016    4.7 GB  43 minutes ago
mistral:latest                61e88e884507    4.1 GB  8 days ago
tinydolphin:latest            0f9dd11f824c    636 MB  13 days ago
tinyllama:latest              2644915ede35    637 MB  6 weeks ago
```

Using Ollama locally - running a model

- You can interact with a model by running:

  $ ollama run <model_name>

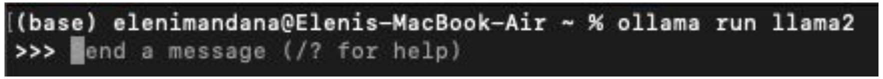
- When using the run command, if a non-local model is selected, it will be downloaded automatically



```
george[ollama]$ ollama run wizardlm2
pulling manifest
pulling 431800fe7a30...   49%                           | 2.0 GB/4.1 GB  5.2 MB/s   6m45s
```

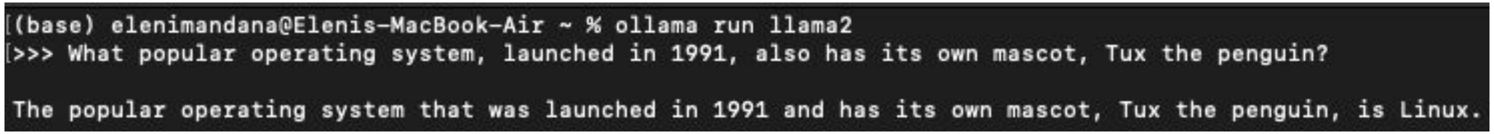
Using Ollama locally - interaction

- After running **$ ollama run <model_name>** you get a message prompt:



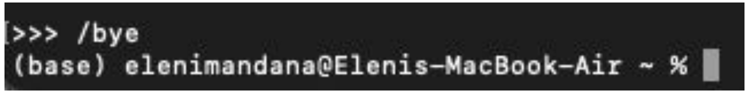
```
[(base) elenimandana@Elenis-MacBook-Air ~ % ollama run llama2
>>> Send a message (/? for help)
```

- You can then type any prompt and get an answer from your desired model



```
[(base) elenimandana@Elenis-MacBook-Air ~ % ollama run llama2
[>>> What popular operating system, launched in 1991, also has its own mascot, Tux the penguin?

 The popular operating system that was launched in 1991 and has its own mascot, Tux the penguin, is Linux.
```

- To exit type: **/bye** (or Ctrl-D)



```
[>>> /bye
(base) elenimandana@Elenis-MacBook-Air ~ %
```

Local Hardware Limitations

- Trying to run local LLMs on:
  - A desktop PC:
    - Intel Core i7-9700K CPU @ 3.60GHz
    - 32 GB RAM
    - NVIDIA GeForce GTX 1050 Ti 4GB
  - A laptop:
    - Intel Core i7-1165G7 @ 2.80GHz
    - 32 GB RAM
    - No GPU

	Desktop		**Laptop**
Model	**Layers in GPU**	**Tokens/s**	**Tokens/s**
tinyllama:1.1b	23/23	77	26
gemma:2b	19/19	36	11
llama3:8b	13/33	5.8	3.8
llama2:13b	10/41	3.3	2.3
qwen:32b	5/65	1.2	1.0
llama3:70b	ERROR	-	?

How to run LLMs on Aristotelis HPC infrastructure

- Ollama Installation
- OnDemand
- Batch jobs
- Interactive session
- REST API
- Jupyter notebook
- Clients and UIs
- Potential Issues

How to run LLMs on Aristotelis HPC infrastructure

- Ollama Installation
- OnDemand
- Batch jobs
- Interactive session
- REST API
- Jupyter notebook
- Clients and UIs
- Potential Issues

Ollama Installation on Aristotelis

- Use preinstalled module (easiest but may sometimes lag a few versions behind)

```
$ module load ollama
```

- Install in user’s home (latest version)

```
$ # install ollama, run these whenever there is a new release available
$ mkdir -p $HOME/ollama/bin
$ curl -L https://ollama.com/download/ollama-linux-amd64 -o \
    $HOME/ollama/bin/ollama
$ chmod +x $HOME/ollama/bin/ollama
$ # to use it, just add it to your $PATH
$ export PATH=$PATH:$HOME/ollama/bin
```

How to run LLMs on Aristotelis HPC infrastructure

OnDemand

- Running OnDemand is just like running locally
- OnDemand nodes offer Nvidia Quadro RTX 6000 6GB GPUs
- Follow the instructions at https://hpc.it.auth.gr/web-portal/ to login to Aristotle Desktop
- Run **ollama serve** in one terminal
- Run **ollama run <model_name>** on another

** Run **module load ollama** on every terminal you open before running ollama commands*

OnDemand - Potential Issue

- If someone else is already running an ollama service on the same node, you'll get an error message
- Solution: specify a different port, try until it works

```
$ ollama serve
Error: listen tcp 127.0.0.1:11434: bind: address already in use
$ OLLAMA_HOST=127.0.0.1:15678
$ ollama serve
```

How to run LLMs on Aristotelis HPC infrastructure

- Ollama Installation
- OnDemand
- Batch jobs
- Interactive session
- REST API
- Jupyter notebook
- Clients and UIs
- Potential Issues

First Steps

- You need to login with **ssh** (https://hpc.it.auth.gr/intro/) to a login node. Run:

  $ ssh <username>@aristotle.it.auth.gr

- You need a VPN if you are not inside AUTH network. VPN instructions here: https://it.auth.gr/manuals/eduvpn/
- You can also use the Web Portal to login (https://hpc.it.auth.gr/web-portal/). No VPN needed.
- Then you need the bash script to submit your job…
- Instructions about submitting ollama batch scripts at https://hpc.it.auth.gr/applications/ollama/

Batch job

Create SLURM submission script and use Ollama through that

```
#!/bin/bash
#SBATCH --job-name=Ollama-batch
#SBATCH --partition=ampere
#SBATCH --time=10:00
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --output=output.log
module load ollama
# Create a temp directory
export TMPDIR=$SCRATCH/ollama_tmp
mkdir -p $TMPDIR
# Choose a random available port for the ollama service
while true; do
   OLLAMA_PORT="`shuf -i 10000-20000 -n 1`"
   ss -lpna | grep -q ":$OLLAMA_PORT " || break
done
export OLLAMA_HOST=127.0.0.1:${OLLAMA_PORT}
# Start Ollama service
ollama serve &> serve_ollama_${SLURM_JOBID}.log &
# Wait until Ollama service has been started
sleep 20
# Run Ollama using llama3 model
ollama run llama3 "How do you schedule a job with slurm?"
# Terminate Ollama service
killall ollama
```

- Loads latest version of ollama module.
- Sets Ollama to use a random available port
- Starts the ollama service in the background and redirects both standard output and standard error to a log file named **serve_ollama_{SLURM_JOBID}.log.**
- Executes the ollama run command using the model llama3. If the model is not already installed in the user's account, the system will first execute ollama pull to download the model. This may add some time to the process
- You may submit multiple prompts
- The job exits as soon as it is completed

SBATCH options

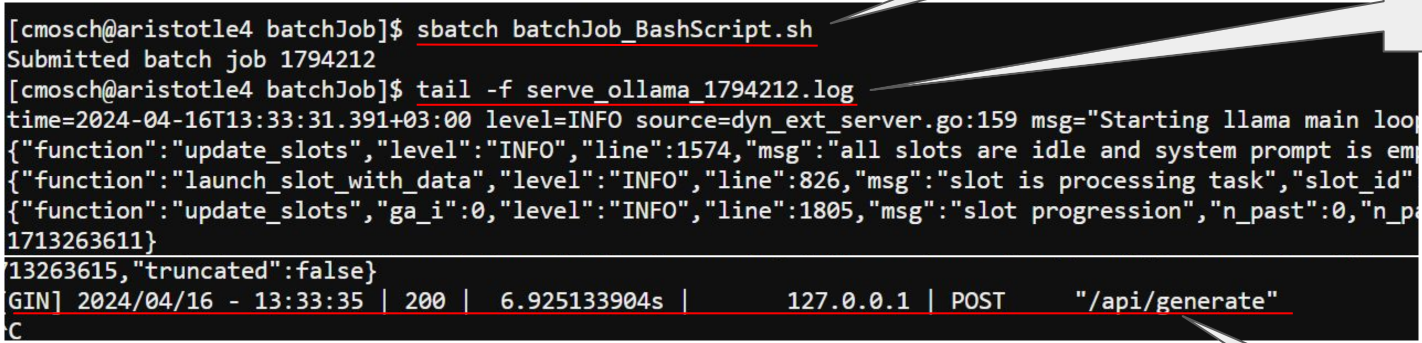
Examples of SBATCH options:

1. --job-name: defines the name of the job to be submitted
2. --partition: specifies the queue in which the job will be submitted
3. --time: determines the maximum time we need to complete the job
4. --output: specifies the path where the output channel (STDOUT) of our batch job will appear

```
### Batch Script
#!/bin/bash

#SBATCH option1=value1
#SBATCH option2=value2
```

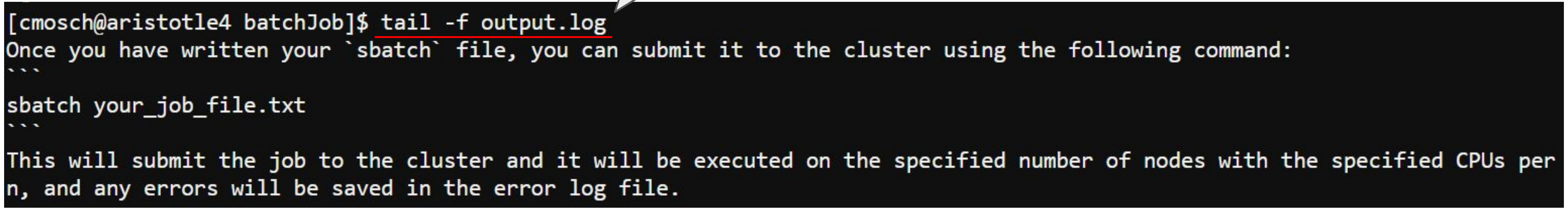
```
#!/bin/bash
#SBATCH --job-name=Ollama-batch
#SBATCH --partition=ampere
#SBATCH --time=10:00
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --output=output.log
```

Batch job - output



```
[cmosch@aristotle4 batchJob]$ sbatch batchJob_BashScript.sh
Submitted batch job 1794212
[cmosch@aristotle4 batchJob]$ tail -f serve_ollama_1794212.log
time=2024-04-16T13:33:31.391+03:00 level=INFO source=dyn_ext_server.go:159 msg="Starting llama main loo
{"function":"update_slots","level":"INFO","line":1574,"msg":"all slots are idle and system prompt is em
{"function":"launch_slot_with_data","level":"INFO","line":826,"msg":"slot is processing task","slot_id"
{"function":"update_slots","ga_i":0,"level":"INFO","line":1805,"msg":"slot progression","n_past":0,"n_p
1713263611}
13263615,"truncated":false}
GIN] 2024/04/16 - 13:33:35 | 200 |  6.925133904s |       127.0.0.1 | POST     "/api/generate"
C
```

Submit a job
Tail the .log file of Ollama service
POST command on Ollama API
Tail the .log file of the submitted job



```
[cmosch@aristotle4 batchJob]$ tail -f output.log
Once you have written your `sbatch` file, you can submit it to the cluster using the following command:
```
sbatch your_job_file.txt
```
This will submit the job to the cluster and it will be executed on the specified number of nodes with the specified CPUs per
n, and any errors will be saved in the error log file.
```

Ollama Python library

Note: The Ollama service needs to be running…

- The Ollama Python library is a convenient toolkit that allows you to directly utilize the available Large Language Models in Ollama within your Python scripts. It can be installed with pip:

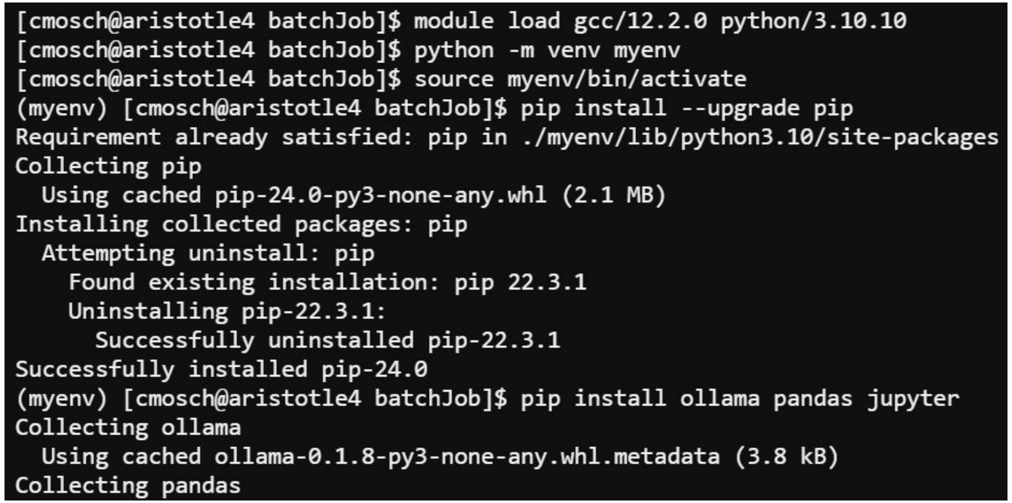
  $ pip install ollama

```python
#!/usr/bin/env python3

import ollama
import os

ollama_host = os.getenv('OLLAMA_HOST')

client = ollama.Client(host=ollama_host)
response = client.chat(model='llama3', messages=[
  {
    'role': 'user',
    'content': 'What popular operating system, launched in 1991, \
                also has its own mascot, Tux the penguin?'
  }
])

print(response['message']['content'])
```

A small example on how to run Llama3 from the Ollama Python library

Batch job with Python script - Python environment

How to create a virtual environment including ollama



```
[cmosch@aristotle4 batchJob]$ module load gcc/12.2.0 python/3.10.10
[cmosch@aristotle4 batchJob]$ python -m venv myenv
[cmosch@aristotle4 batchJob]$ source myenv/bin/activate
(myenv) [cmosch@aristotle4 batchJob]$ pip install --upgrade pip
Requirement already satisfied: pip in ./myenv/lib/python3.10/site-packages
Collecting pip
  Using cached pip-24.0-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.3.1
    Uninstalling pip-22.3.1:
      Successfully uninstalled pip-22.3.1
Successfully installed pip-24.0
(myenv) [cmosch@aristotle4 batchJob]$ pip install ollama pandas jupyter
Collecting ollama
  Using cached ollama-0.1.8-py3-none-any.whl.metadata (3.8 kB)
Collecting pandas
```

- Simplest way → use Python's built-in venv module
- Create and activate the environment myenv
- Useful to upgrade Python's standard package manager (pip) to the latest available version before proceeding with the installation of additional packages
- Install via pip all the desired libraries

More information on https://hpc.it.auth.gr/languages/python/

Batch job with Python script - batch script

Python script and ollama python library

- Starts the ollama service and pipes its output to both the console and a log file named **serve_ollama_${SLURM_JOBID}.log**
- Activates the Python virtual environment where ollama and its dependencies are installed
- Executes a Python script named *test_ollama.py* with unbuffered output and redirects its output to a file named *python_script_output.txt*

```
#!/bin/bash
#SBATCH --job-name=Ollama-batch
#SBATCH --partition=ampere
#SBATCH --time=10:00
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --output=output.log
module load ollama
# Create a temp directory
export TMPDIR=$SCRATCH/ollama_tmp
mkdir -p $TMPDIR
# Choose a random available port for the ollama service
while true; do
   OLLAMA_PORT="`shuf -i 10000-20000 -n 1`"
   ss -lpna | grep -q ":$OLLAMA_PORT " || break
done
export OLLAMA_HOST=127.0.0.1:${OLLAMA_PORT}
# Start Ollama service
ollama serve &> serve_ollama_${SLURM_JOBID}.log &
# Wait until Ollama service has been started
sleep 20
# Activate the virtual environment
source ollama_env/bin/activate
# Run the python script
python -u test_ollama.py > python_script_output.txt
# Terminate Ollama service
killall ollama
```

How to run LLMs on Aristotelis HPC infrastructure

- Ollama Installation
- OnDemand
- Batch jobs
- Interactive session
- REST API
- Jupyter notebook
- Clients and UIs
- Potential Issues

Interactive Session - batch script

You can run the Ollama service on the HPC cluster and the client on a login node!

```bash
#!/bin/bash
#SBATCH --job-name=Ollama-service
#SBATCH --partition=ampere
#SBATCH --time=10:00
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --output=output.log
module load ollama
# Create a temp directory
export TMPDIR=$SCRATCH/ollama_tmp
mkdir -p $TMPDIR
# Choose a random available port for the ollama service
while true; do
   OLLAMA_PORT="`shuf -i 10000-20000 -n 1`"
   ss -lpna | grep -q ":$OLLAMA_PORT " || break
done
# Retrieve the public IP address of the host
IP_ADDR=$(curl ip.me)
# Set the OLLAMA_HOST variable
export OLLAMA_HOST=${IP_ADDR}:${OLLAMA_PORT}
# Send a push notification to my phone (see https://ntfy.sh for more)
curl -d "${SLURM_JOBID}@${OLLAMA_HOST}:${OLLAMA_PORT}." ntfy.sh/ollama123
# Start Ollama service
ollama serve &> serve_ollama_${SLURM_JOBID}.log
```

- Retrieves the public IP address of the host using curl from the ip.me service and sets it as the OLLAMA_HOST environment variable
- Starts the ollama service and pipes its output to both the console and a log file named **serve_ollama_${SLURM_JOBID}.log**
- You don’t run ***ollama run*** in the batch script in this case

Interactive Session

- Submit the job:

```
$ sbatch ollama_service.sh
Submitted batch job 1796446
$ tail -f serve_ollama_1796446.log
time=2024-04-24T08:43:49.320+03:00 level=INFO source=images.go:817 msg="total blobs: 44"
time=2024-04-24T08:43:49.356+03:00 level=INFO source=images.go:824 msg="total unused blobs removed: 0"
time=2024-04-24T08:43:49.360+03:00 level=INFO source=routes.go:1143 msg="Listening on 155.207.96.50:12906 (version 0.1.32)"
```

- Connect to it from the login node:

```
$ module load ollama
load ollama 0.1.32 (PATH)
$ export OLLAMA_HOST=155.207.96.50:12906
$ ollama run llama3
>>> Send a message (/? for help)
```

Make sure you connect to the right IP and port from the login node!

How to run LLMs on Aristotelis HPC infrastructure

- Ollama Installation
- OnDemand
- Batch jobs
- Interactive session
- REST API
- Jupyter notebook
- Clients and UIs
- Potential Issues

REST API

- Remember that ***ollama*** runs a service component. What you get with ***ollama run*** is just a client. The simplest client is a just a REST API
- Generate a response:

```
curl http://155.207.96.50:11434/api/generate -d '{ "model": "llama3", \
  "prompt":"Why is the sky blue?" }'
```

155.207.96.50 is the IP of the HPC node that runs the ollama service in this example and 11434 is the port. Substitute them for the ones you’re using.

- Chat with a model:

```
curl http://155.207.96.50:11434/api/chat -d '{ "model": "llama3", \
  "messages": [{"role": "user", "content": "Why is the sky blue?"}]}'
```

REST API - example output

```
{"model":"llama3","created_at":"2024-04-20T20:24:08.725750389Z","response":"What","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:08.882499464Z","response":" a","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:09.036335811Z","response":" great","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:09.191317995Z","response":" question","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:09.354884382Z","response":"!\n\n","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:09.518166884Z","response":"The","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:09.673687543Z","response":" sky","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:09.827425264Z","response":" appears","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:09.983749935Z","response":" blue","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:10.137785563Z","response":" because","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:10.293128093Z","response":" of","done":false}
{"model":"llama3","created_at":"2024-04-20T20:24:10.449808154Z","response":" a","done":false}
...
```

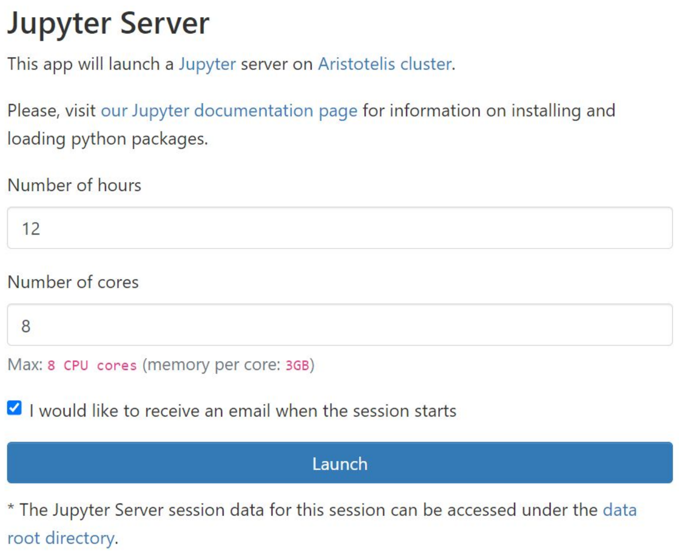
** REST API can be accessed from login nodes. For **ampere** and **ondemand** it can also be accessed from all AUTH network for ports 11434-11443.*

How to run LLMs on Aristotelis HPC infrastructure

- Ollama Installation
- OnDemand
- Batch jobs
- Interactive session
- REST API
- Jupyter notebook
- Clients and UIs
- Potential Issues

Jupyter Notebook (1/3)

- Start ollama service with the same batch script as in the Interactive Session. Be sure to note the host ip!
- Start a Jupyter Server on the cluster (Through the menu on *https://hpc.auth.gr* choose Interactive Apps -> Jupyter)



Jupyter Server

This app will launch a Jupyter server on Aristotelis cluster.

Please, visit our Jupyter documentation page for information on installing and loading python packages.

Number of hours

12

Number of cores

8

Max: `8 CPU cores` (memory per core: `3GB`)

☑ I would like to receive an email when the session starts

Launch

* The Jupyter Server session data for this session can be accessed under the data root directory.

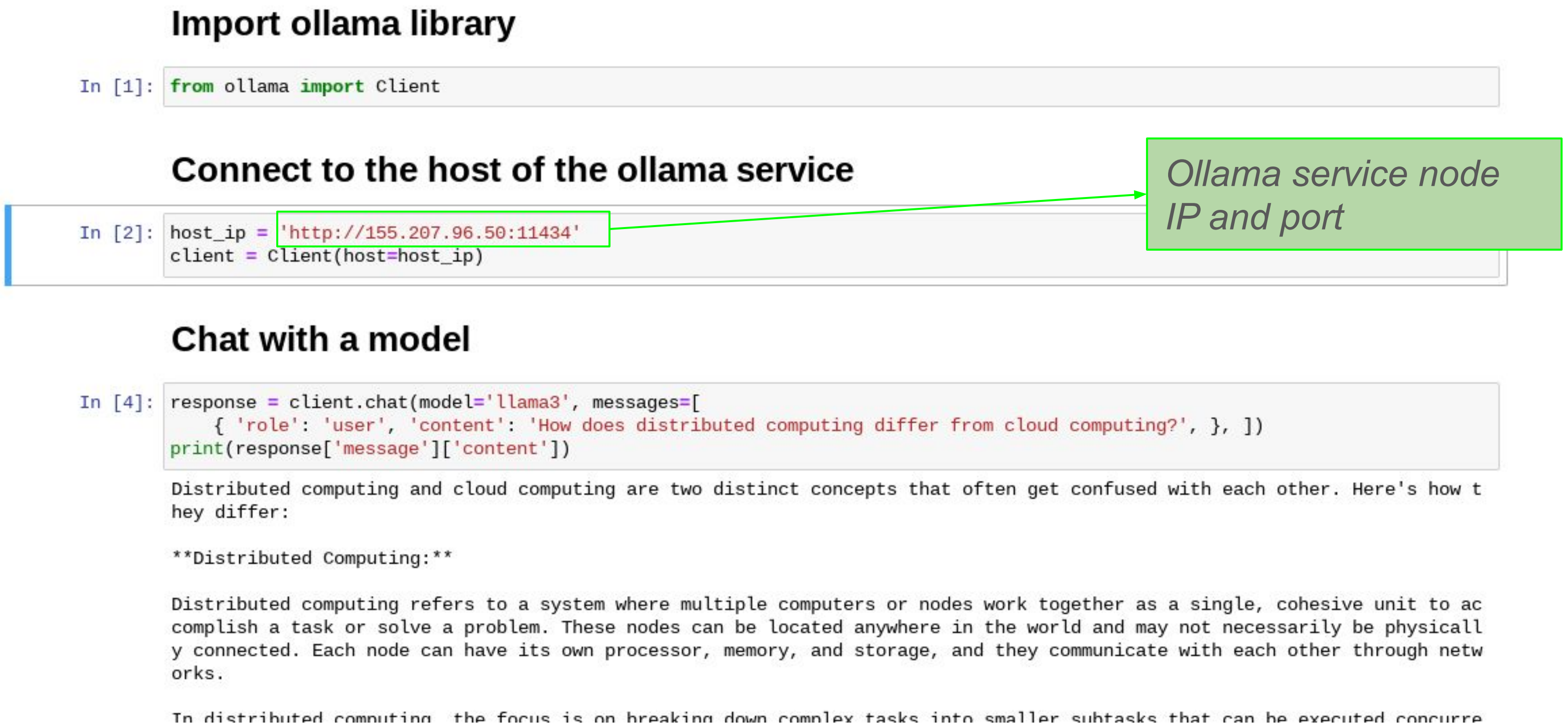
Jupyter Notebook (2/3)

- Read the instructions at https://hpc.it.auth.gr/applications/jupyter/ on how to setup a Python virtual environment
- Make sure you install the ollama library

```
mkdir envs
cd envs
python -m venv ollama_env
source ollama_env/bin/activate
pip install --upgrade pip
pip install jupyter
python -m ipykernel install --user --name my-custom-env --display "Ollama env"
pip install ollama
```

Jupyter Notebook (3/3)

- Create a new notebook using the new ollama environment



Import ollama library

```
In [1]: from ollama import Client
```

Connect to the host of the ollama service

```
In [2]: host_ip = 'http://155.207.96.50:11434'
        client = Client(host=host_ip)
```

Ollama service node IP and port

Chat with a model

```
In [4]: response = client.chat(model='llama3', messages=[
            { 'role': 'user', 'content': 'How does distributed computing differ from cloud computing?', }, ])
        print(response['message']['content'])
```

```
Distributed computing and cloud computing are two distinct concepts that often get confused with each other. Here's how t
hey differ:

**Distributed Computing:**

Distributed computing refers to a system where multiple computers or nodes work together as a single, cohesive unit to ac
complish a task or solve a problem. These nodes can be located anywhere in the world and may not necessarily be physicall
y connected. Each node can have its own processor, memory, and storage, and they communicate with each other through netw
orks.

In distributed computing, the focus is on breaking down complex tasks into smaller subtasks that can be executed concurre
```

How to run LLMs on Aristotelis HPC infrastructure

- OnDemand
- Batch jobs
- Interactive session
- Jupyter notebook
- REST API
- Clients and UIs
- Potential Issues

Clients and UIs

You can build other clients on top of the REST API. There are several different ones available

- Web UI
- Terminal
- Editor
- Mobile
- Plugins
- Libraries

There is a long list at https://github.com/ollama/ollama

Clients and UIs - batch script

You’ll also need to set the **OLLAMA_ORIGINS=”*”** variable in your batch script to allow access from web browsers. It’s a CORS issue.

Note the smaller range of ports.

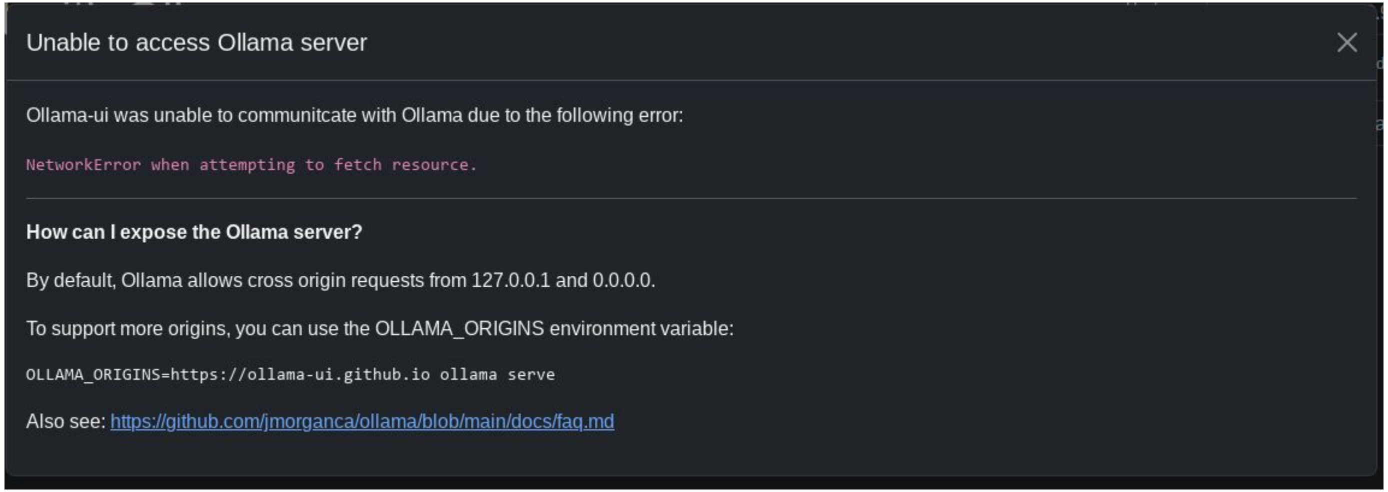
```
#!/bin/bash
#SBATCH --job-name=Ollama-service
#SBATCH --partition=ampere
#SBATCH --time=10:00
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --output=output.log
module load ollama
# Create a temp directory
export TMPDIR=$SCRATCH/ollama_tmp
mkdir -p $TMPDIR
# Choose a random available port for the ollama service
while true; do
   OLLAMA_PORT="`shuf -i 11434-11443 -n 1`"
   ss -lpna | grep -q ":$OLLAMA_PORT " || break
done
# Retrieve the public IP address of the host
IP_ADDR=$(curl ip.me)
# Set the OLLAMA_HOST variable
export OLLAMA_HOST=${IP_ADDR}:${OLLAMA_PORT}
# Allow browsers to connect to the service
export OLLAMA_ORIGINS=”*”
# Send a push notification to my phone (see https://ntfy.sh for more)
curl -d "${SLURM_JOBID}@${OLLAMA_HOST}:${OLLAMA_PORT}." ntfy.sh/ollama123
# Start Ollama service
ollama serve &> serve_ollama_${SLURM_JOBID}.log
```

Clients and UIs - Webapp (1/4)

A simple UI is available as a webapp at https://ollama-ui.github.io/ollama-ui/

- Can be used with Firefox
- Doesn't work with Chrome

You will get this error message when your load the webapp. That's OK, close the message…



Unable to access Ollama server

Ollama-ui was unable to communitcate with Ollama due to the following error:

```
NetworkError when attempting to fetch resource.
```

How can I expose the Ollama server?

By default, Ollama allows cross origin requests from 127.0.0.1 and 0.0.0.0.

To support more origins, you can use the OLLAMA_ORIGINS environment variable:

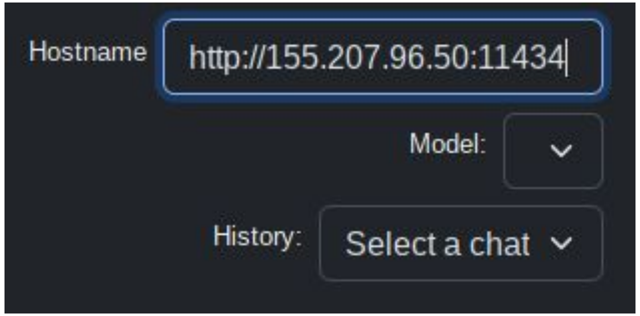
```
OLLAMA_ORIGINS=https://ollama-ui.github.io ollama serve
```

Also see: https://github.com/jmorganca/ollama/blob/main/docs/faq.md

Clients and UIs - Webapp (2/4)

You need to enter the ollama service IP address along with the port in the format

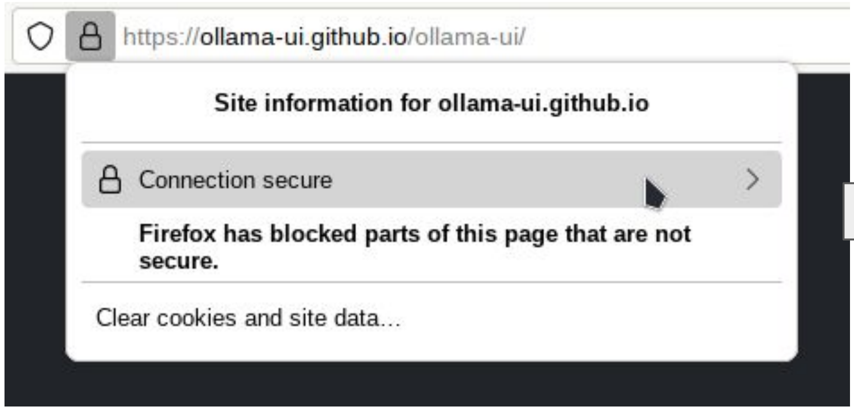
`http://`*`ip_address:port`*



Hostname http://155.207.96.50:11434

Clients and UIs - Webapp (3/4)

Click on the lock icon on the address bar and select to *Disable protection for now* (it's a "Mixed content" issue, loading http content from an https page)



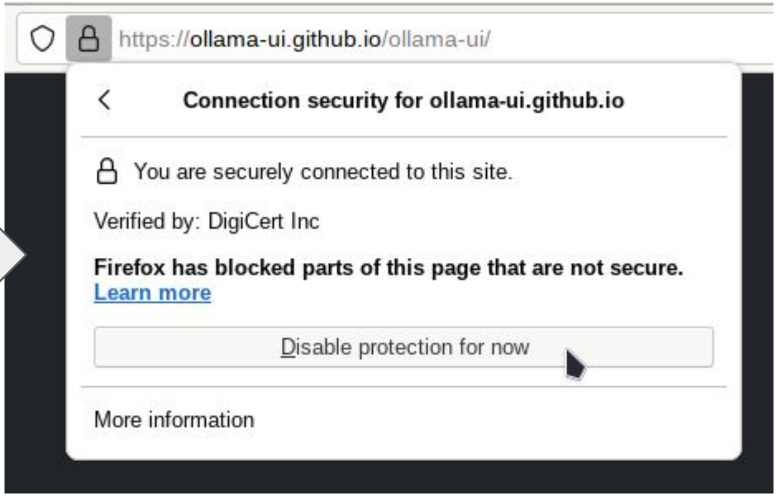
https://ollama-ui.github.io/ollama-ui/

Site information for ollama-ui.github.io

Connection secure

Firefox has blocked parts of this page that are not secure.

Clear cookies and site data…



https://ollama-ui.github.io/ollama-ui/

Connection security for ollama-ui.github.io

You are securely connected to this site.

Verified by: DigiCert Inc

Firefox has blocked parts of this page that are not secure. Learn more

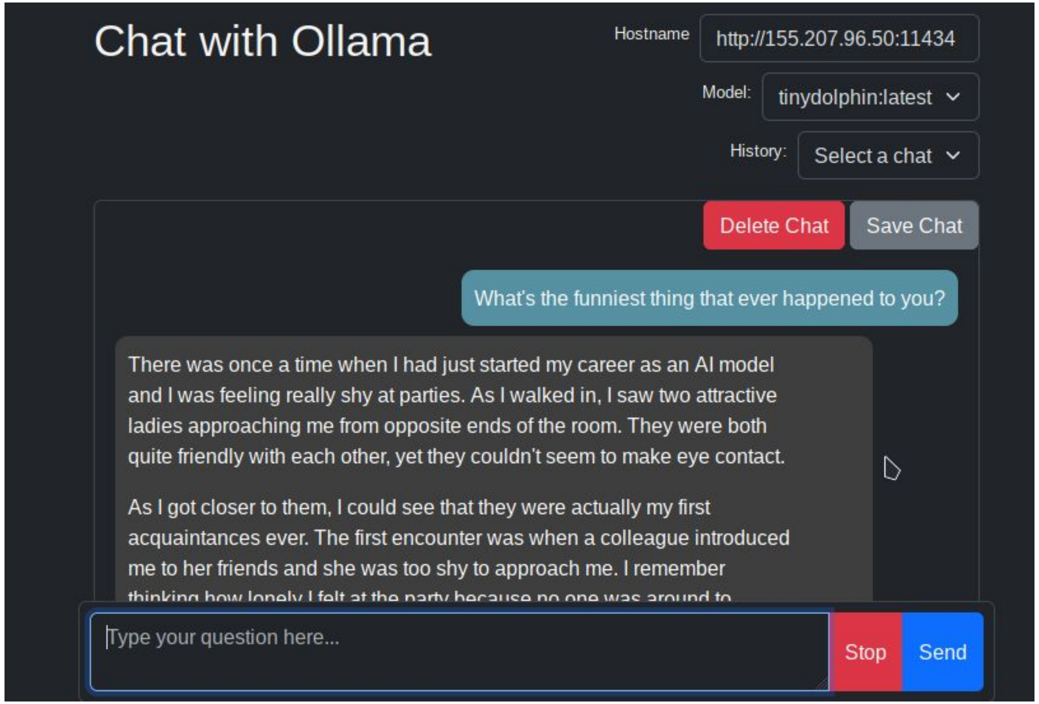
Disable protection for now

More information

Clients and UIs - Webapp (4/4)

And then it works!

- The available models will be shown in the *Model* dropdown and you can select the one you'd like to use.
- You can save any conversations you'd like to keep



Hostname: http://155.207.96.50:11434

Model: tinydolphin:latest

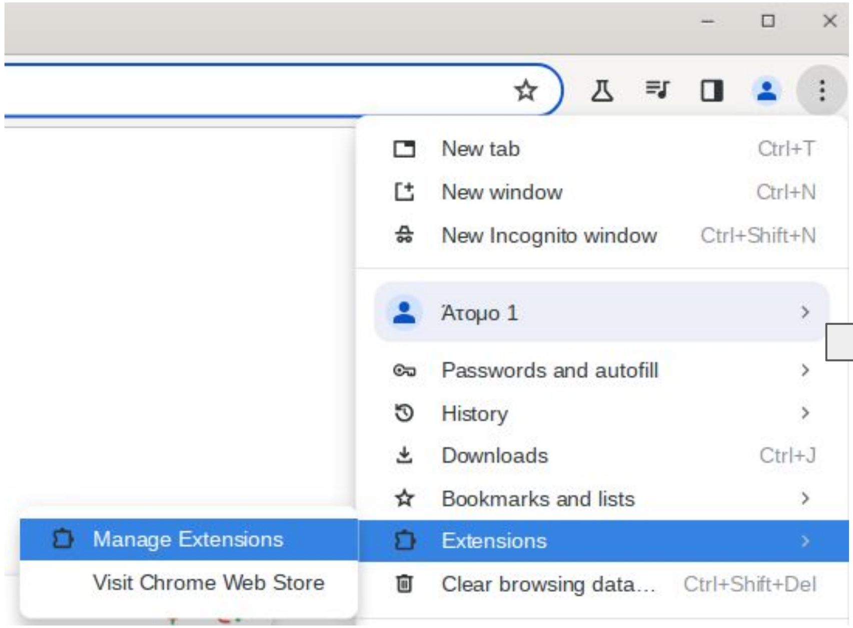
What's the funniest thing that ever happened to you?

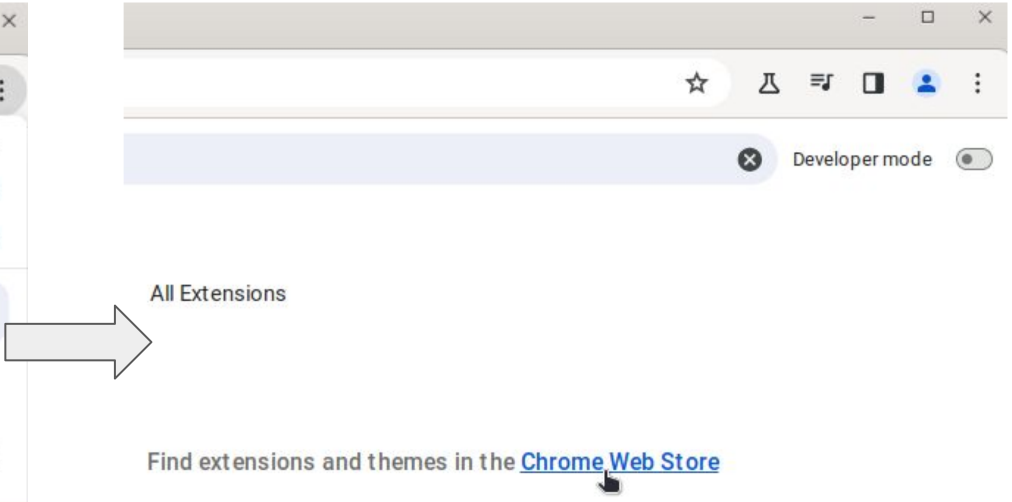
There was once a time when I had just started my career as an AI model and I was feeling really shy at parties. As I walked in, I saw two attractive ladies approaching me from opposite ends of the room. They were both quite friendly with each other, yet they couldn't seem to make eye contact.

As I got closer to them, I could see that they were actually my first acquaintances ever. The first encounter was when a colleague introduced me to her friends and she was too shy to approach me. I remember thinking how lonely I felt at the party because no one was around to

Clients and UIs - Chrome browser plugin (1/7)

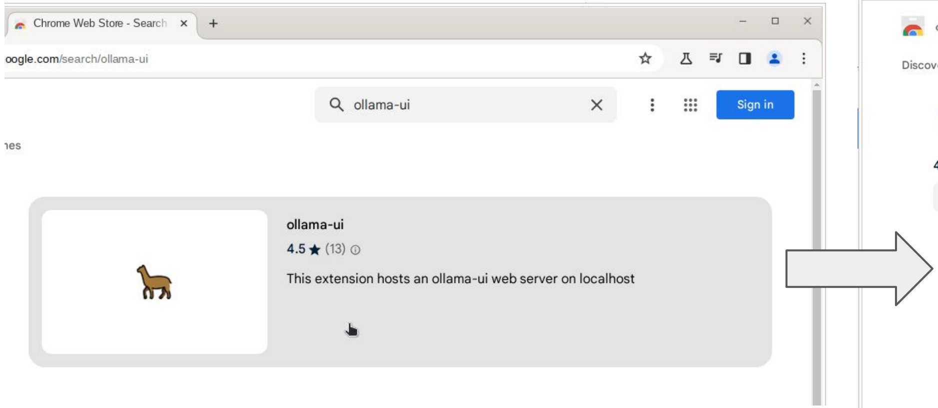
The webapp doesn’t work with Chrome, but there is a plugin for it. Go to the *Chrome Web Store* …



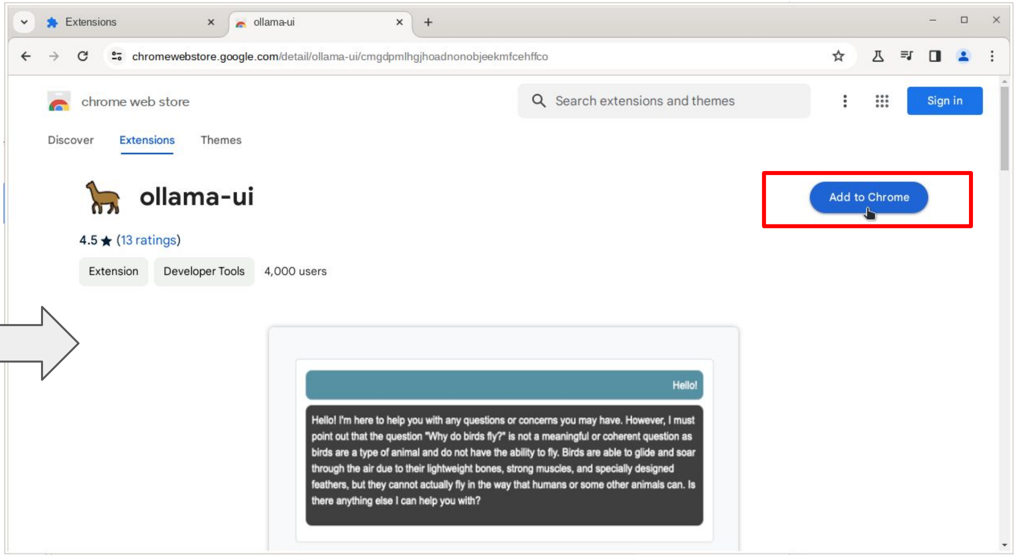


Clients and UIs - Chrome browser plugin (2/7)

Select to add the ***ollama-ui*** extension to Chrome…



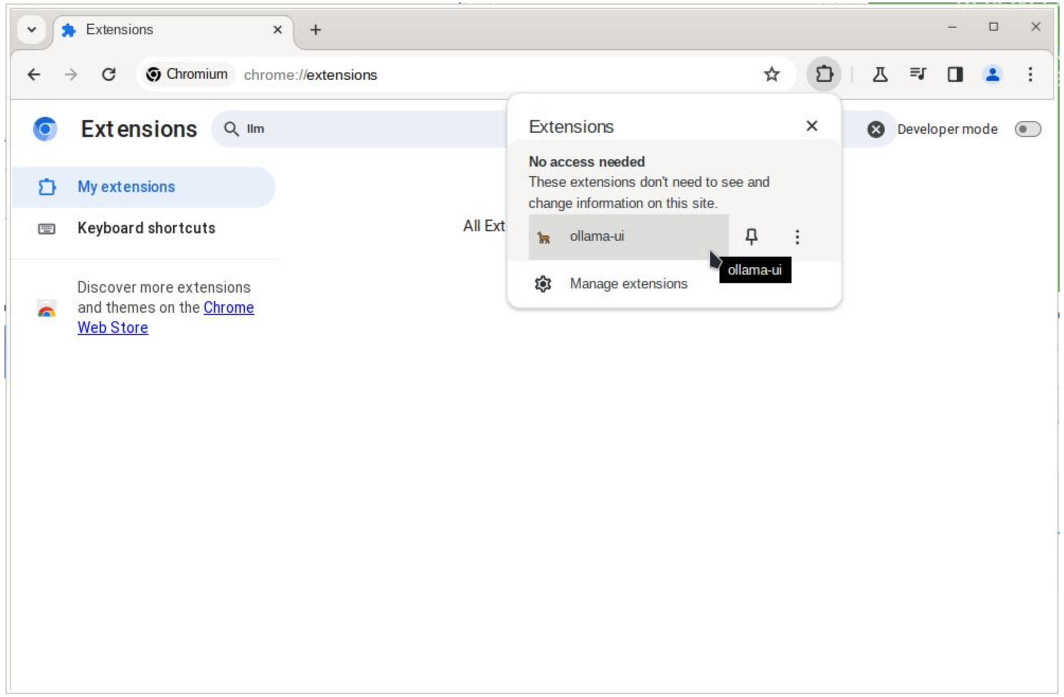
Chrome Web Store - Search
oogle.com/search/ollama-ui
ollama-ui
Sign in
nes
ollama-ui
4.5 ★ (13)
This extension hosts an ollama-ui web server on localhost



Hello! I'm here to help you with any questions or concerns you may have. However, I must point out that the question "Why do birds fly?" is not a meaningful or coherent question as birds are a type of animal and do not have the ability to fly. Birds are able to glide and soar through the air due to their lightweight bones, strong muscles, and specially designed feathers, but they cannot actually fly in the way that humans or some other animals can. Is there anything else I can help you with?

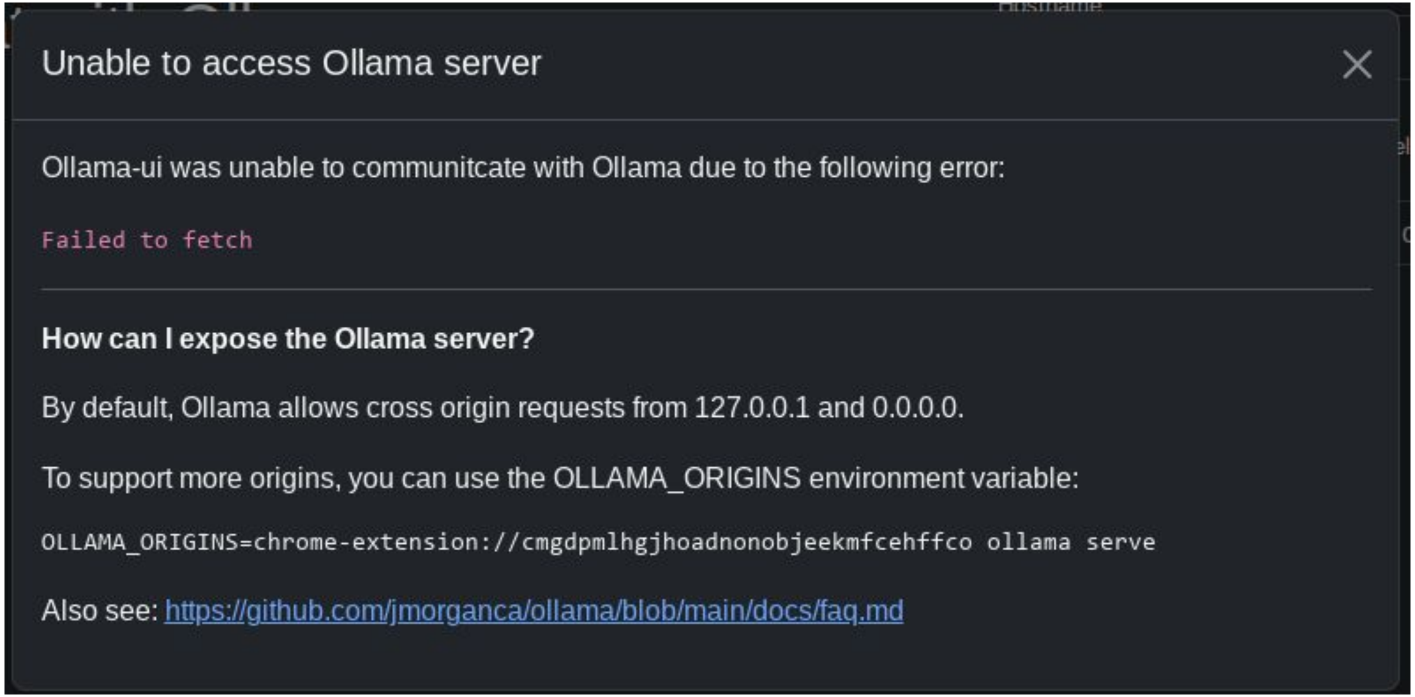
Clients and UIs - Chrome browser plugin (3/7)

And it will be available from the extensions button on the Chrome toolbar. Click on it…



Clients and UIs - Chrome browser plugin (4/7)

Start the extension and you'll get the same error message as with Firefox. That's OK, close the message…



Unable to access Ollama server

Ollama-ui was unable to communitcate with Ollama due to the following error:

```
Failed to fetch
```

How can I expose the Ollama server?

By default, Ollama allows cross origin requests from 127.0.0.1 and 0.0.0.0.

To support more origins, you can use the OLLAMA_ORIGINS environment variable:

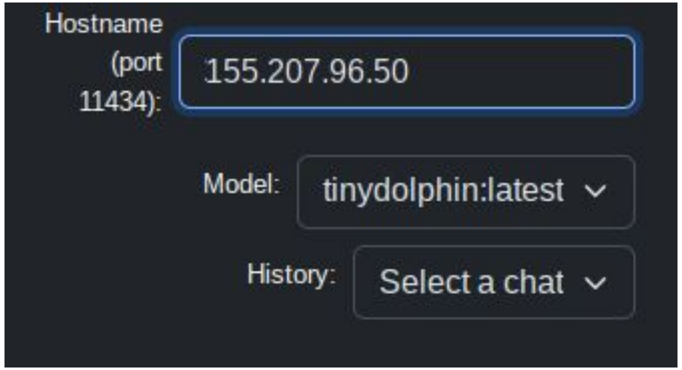
```
OLLAMA_ORIGINS=chrome-extension://cmgdpmlhgjhoadnonobjeekmfcehffco ollama serve
```

Also see: https://github.com/jmorganca/ollama/blob/main/docs/faq.md

Clients and UIs - Chrome browser plugin (5/7)

… and set the hostname to the ***IP address*** of the HPC cluster node the ollama service is running on.

Once you do, the available models will be shown in the *Model* dropdown and you can select the one you'd like to use.



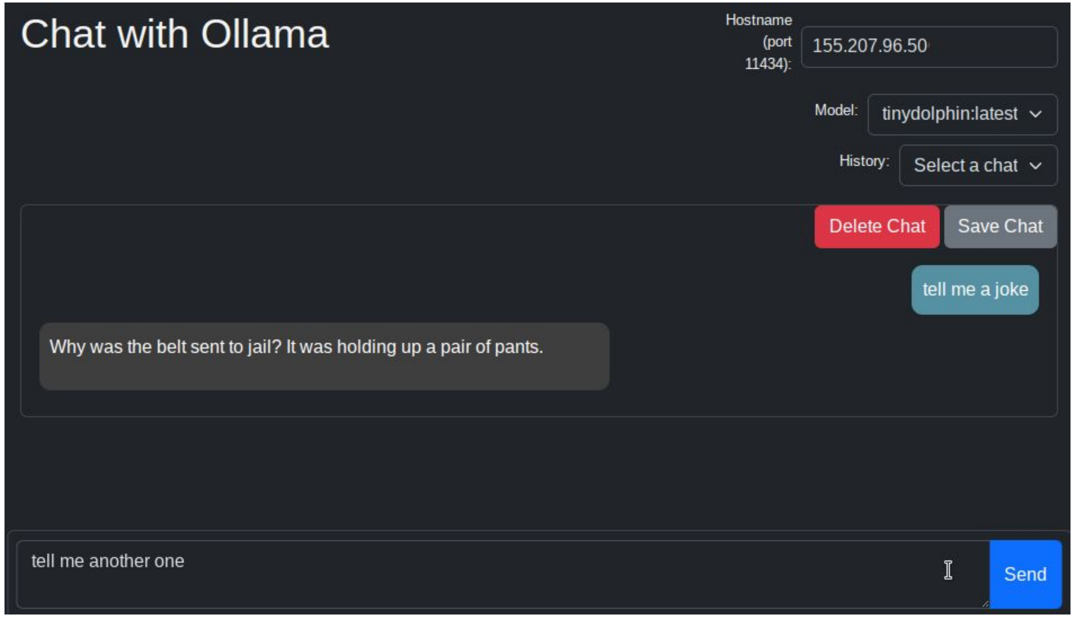
Hostname (port 11434): 155.207.96.50

Model: tinydolphin:latest

History: Select a chat

Clients and UIs - Chrome browser plugin (6/7)

You’re then able to chat with your selected model. You can save any conversations you’d like to keep.



Hostname (port 11434): 155.207.96.50

Model: tinydolphin:latest

History: Select a chat

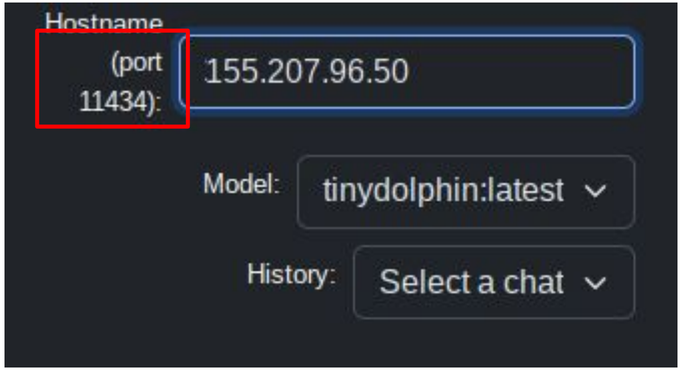
tell me a joke

Why was the belt sent to jail? It was holding up a pair of pants.

tell me another one

Clients and UIs - Chrome browser plugin (7/7)

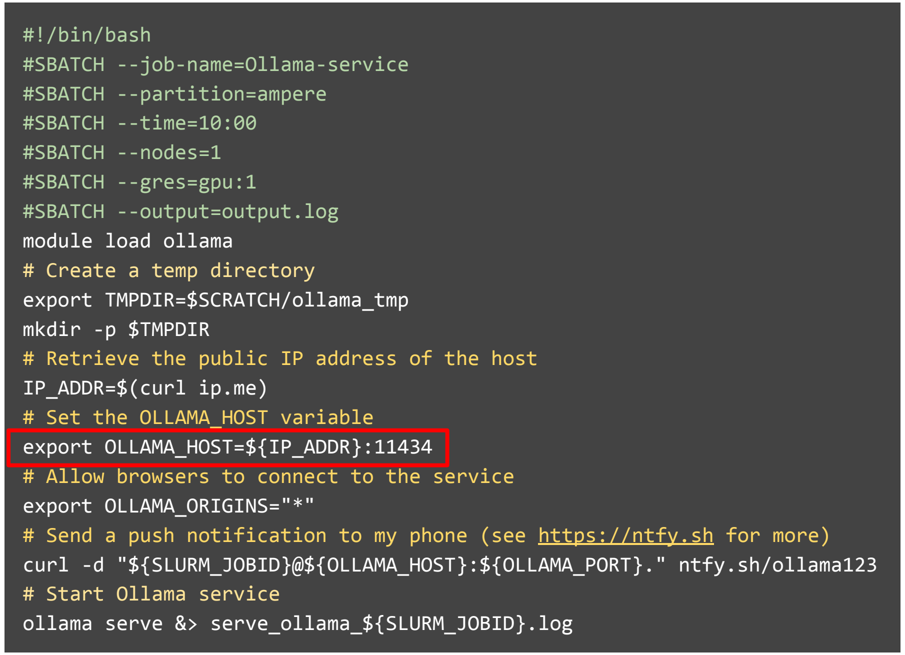
- Issue: the Chrome extension only works with port 11434.
- You'll have to set this port explicitly
- Possibility of port clashing



Hostname (port 11434): 155.207.96.50

Model: tinydolphin:latest

History: Select a chat



```
#!/bin/bash
#SBATCH --job-name=Ollama-service
#SBATCH --partition=ampere
#SBATCH --time=10:00
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --output=output.log
module load ollama
# Create a temp directory
export TMPDIR=$SCRATCH/ollama_tmp
mkdir -p $TMPDIR
# Retrieve the public IP address of the host
IP_ADDR=$(curl ip.me)
# Set the OLLAMA_HOST variable
export OLLAMA_HOST=${IP_ADDR}:11434
# Allow browsers to connect to the service
export OLLAMA_ORIGINS="*"
# Send a push notification to my phone (see https://ntfy.sh for more)
curl -d "${SLURM_JOBID}@${OLLAMA_HOST}:${OLLAMA_PORT}." ntfy.sh/ollama123
# Start Ollama service
ollama serve &> serve_ollama_${SLURM_JOBID}.log
```

How to run LLMs on Aristotelis HPC infrastructure

- OnDemand
- Batch jobs
- Interactive session
- Jupyter notebook
- REST API
- Clients and UIs
- Potential Issues

Potential Issues

- Not enough HD space for big models
  - Default $HOME for users is 20GB
  - Large models can't fit
- Possible solutions:
  - Request more space for your account: https://hpc.it.auth.gr/home-directories/
  - Store the ~/.ollama directory under $SCRATCH and symlink it from there
    - Files get deleted from $SCRATCH after 30 days

```
$ mkdir -p $SCRATCH/ollama-models
$ ln -s $SCRATCH/ollama-models ~/.ollama
```

Performance Comparisons

- Model Performance on Aristotelis

Model performance on Aristotelis

	ampere		ondemand		rome	batch
Model	**Layers in GPU**	**Tokens/s**	**Layers in GPU**	**Tokens/s**	**Tokens/s**	**Tokens/s**
tinyllama:1.1b	23/23	233	23/23	135	8.3	7.3
gemma:2b	19/19	156	19/19	106	6.9	4.3
llama3:8b	33/33	97	33/33	66	5.3	3.0
llama2:13b	41/41	84	22/41	<0.1	3.1	?
qwen:32b	65/65	38	?	?	?	?
llama3:70b	81/81	24.3	?	?	?	?

* *ollama does not work on the* ***gpu*** *cluster for now*

Model performance on Aristotelis - GPU vs CPU

- Running on GPUs is considerably faster!
- 7b and 8b models can run adequately on CPUs. If your local PC is not capable of running them, they can provide an alternative
- On CPUs, you should make sure to run the process **in a single socket**. Otherwise performance will suffer.

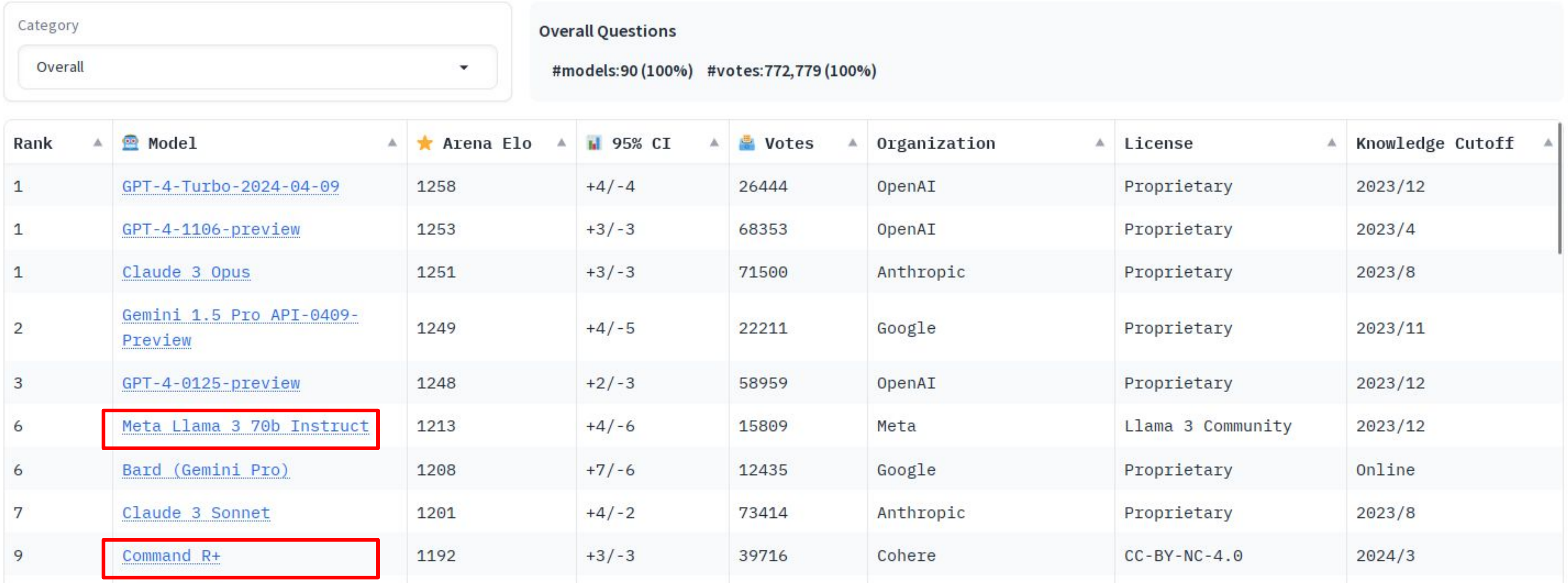
```
#!/bin/bash
#SBATCH --job-name=Ollama-cpu-serve
#SBATCH --partition=rome
#SBATCH --time=20:00
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=64
#SBATCH --extra-node-info=1:64
#SBATCH --output=output-rome.log
```

Comparison of different models: from small to large

- LLM Leaderboard
- Quality of responses example
- Model Overview

LLM Leaderboard (1/2)

LMSYS Chatbot Arena



Category

Overall

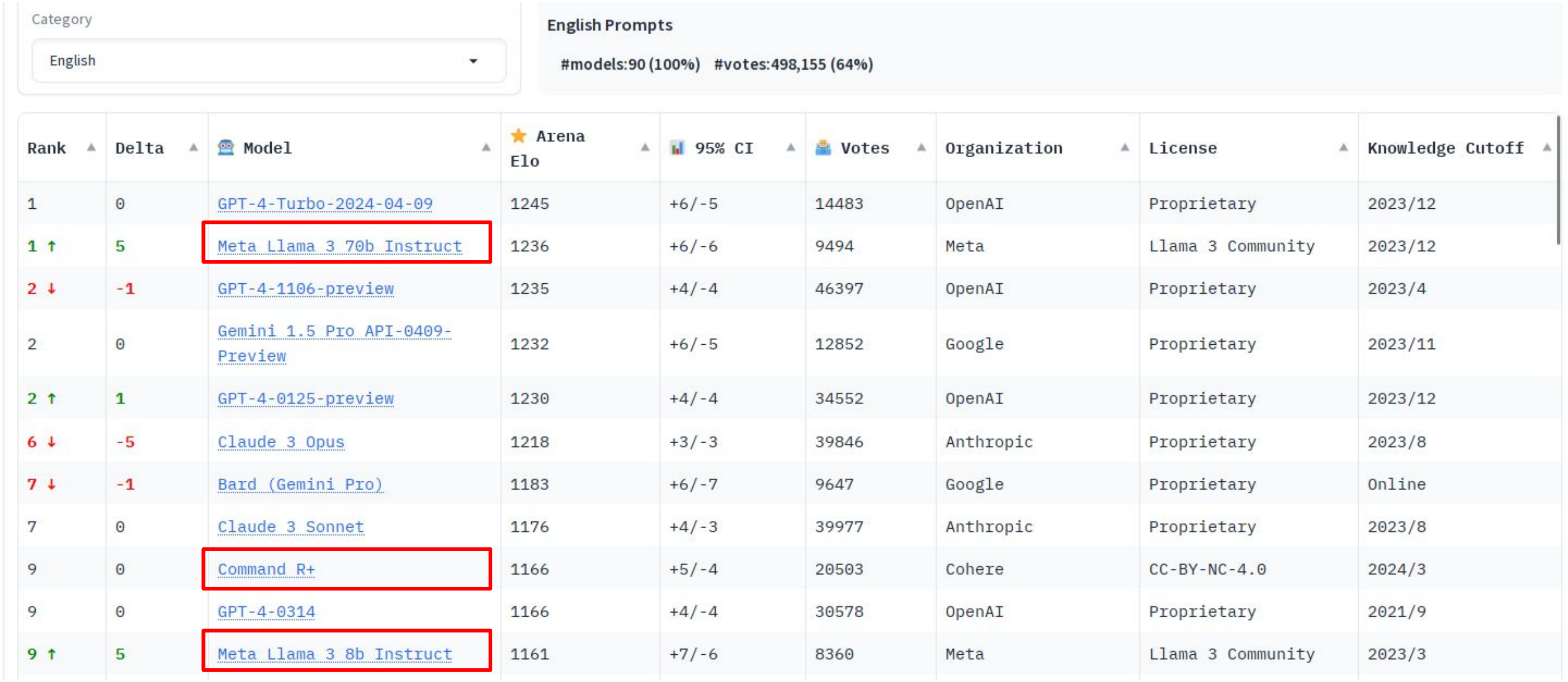
Overall Questions

#models:90 (100%) #votes:772,779 (100%)

Rank	Model	Arena Elo	95% CI	Votes	Organization	License	Knowledge Cutoff
1	GPT-4-Turbo-2024-04-09	1258	+4/-4	26444	OpenAI	Proprietary	2023/12
1	GPT-4-1106-preview	1253	+3/-3	68353	OpenAI	Proprietary	2023/4
1	Claude 3 Opus	1251	+3/-3	71500	Anthropic	Proprietary	2023/8
2	Gemini 1.5 Pro API-0409-Preview	1249	+4/-5	22211	Google	Proprietary	2023/11
3	GPT-4-0125-preview	1248	+2/-3	58959	OpenAI	Proprietary	2023/12
6	Meta Llama 3 70b Instruct	1213	+4/-6	15809	Meta	Llama 3 Community	2023/12
6	Bard (Gemini Pro)	1208	+7/-6	12435	Google	Proprietary	Online
7	Claude 3 Sonnet	1201	+4/-2	73414	Anthropic	Proprietary	2023/8
9	Command R+	1192	+3/-3	39716	Cohere	CC-BY-NC-4.0	2024/3

LLM Leaderboard (2/2)

LMSYS Chatbot Arena



Category

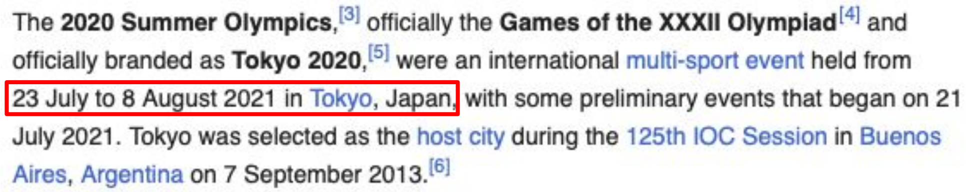
English

English Prompts

#models:90 (100%) #votes:498,155 (64%)

Rank	Delta	Model	Arena Elo	95% CI	Votes	Organization	License	Knowledge Cutoff
1	0	GPT-4-Turbo-2024-04-09	1245	+6/-5	14483	OpenAI	Proprietary	2023/12
1 ↑	5	Meta Llama 3 70b Instruct	1236	+6/-6	9494	Meta	Llama 3 Community	2023/12
2 ↓	-1	GPT-4-1106-preview	1235	+4/-4	46397	OpenAI	Proprietary	2023/4
2	0	Gemini 1.5 Pro API-0409-Preview	1232	+6/-5	12852	Google	Proprietary	2023/11
2 ↑	1	GPT-4-0125-preview	1230	+4/-4	34552	OpenAI	Proprietary	2023/12
6 ↓	-5	Claude 3 Opus	1218	+3/-3	39846	Anthropic	Proprietary	2023/8
7 ↓	-1	Bard (Gemini Pro)	1183	+6/-7	9647	Google	Proprietary	Online
7	0	Claude 3 Sonnet	1176	+4/-3	39977	Anthropic	Proprietary	2023/8
9	0	Command R+	1166	+5/-4	20503	Cohere	CC-BY-NC-4.0	2024/3
9	0	GPT-4-0314	1166	+4/-4	30578	OpenAI	Proprietary	2021/9
9 ↑	5	Meta Llama 3 8b Instruct	1161	+7/-6	8360	Meta	Llama 3 Community	2023/3

Quality of responses (1/4)


The **2020 Summer Olympics**,[3] officially the **Games of the XXXII Olympiad**[4] and officially branded as **Tokyo 2020**,[5] were an international multi-sport event held from 23 July to 8 August 2021 in Tokyo, Japan, with some preliminary events that began on 21 July 2021. Tokyo was selected as the host city during the 125th IOC Session in Buenos Aires, Argentina on 7 September 2013.[6]

Small models

- tinydolphin:1.1b missed the fact that they actually took place in a different year
- gemma:2b knows that it doesn't know! Impressive for an LLM.

tinydolphin:1.1b

```
The 2020 Summer Olympics took place from July 24th
to August 9th in Tokyo, Japan.
```

gemma:2b

```
I cannot access real-time information and cannot
provide specific dates for the 2020 Olympics. For
the most up-to-date information, please check the
official Olympics website or any reputable news
sources.
```

Quality of responses (2/4)

Medium models

- gemma:7b was correct and just on point
- llama2:7b was wrong and so was llama2:13b

llama2:7b

```
The 2020 Summer Olympics, also known as the XXXII
Olympiad, took place from July 24 to August 9,
2020, in Tokyo, Japan.
```

gemma:7b

```
The 2020 Olympics took place from July 23 to
August 8, 2021.
```

llama2:13b

```
The 2020 Summer Olympics, officially known as the
Games of the XXXII Olympiad, took place from July
24 to August 9, 2020, in Tokyo, Japan.
```

Quality of responses (3/4)

- But llama3:8b and llama3:70b both got it perfectly right. Their replies were almost identical and very informative!

llama3:8b

```
There was no "2020 Olympics" that took place. The
Olympic Games were postponed to 2021 due to the
COVID-19 pandemic.

The originally scheduled Tokyo 2020 Summer
Olympics were supposed to take place from July 24
to August 9, 2020. However, on March 24, 2020,
the International Olympic Committee (IOC) and the
Tokyo Organizing Committee announced that
the Games would be postponed to 2021.

The rescheduled Tokyo 2020 Summer Olympics took
place from July 23 to August 8, 2021.
```

llama3:70b

```
There was no "2020 Olympics". The correct name is
the 2020 Summer Olympics, also known as Tokyo
2020.

The 2020 Summer Olympics were originally
scheduled to take place from July 24 to August 9,
2020, in Tokyo, Japan. However, due to the
COVID-19 pandemic, the International Olympic
Committee (IOC) and the Tokyo Organizing
Committee
announced on March 24, 2020, that the Games would
be postponed to 2021.

The rescheduled 2020 Summer Olympics took place
from July 23 to August 8, 2021, in Tokyo, Japan.
```

Quality of responses (4/4)

- Some proprietary LLMs just for the sake of comparison…

Gemini-1.5-pro

```
The 2020 Summer Olympics, officially known as the
Games of the XXXII Olympiad, were held in Tokyo,
Japan, from July 23 to August 8, 2021.
```

Claude-3-Sonnet

```
The 2020 Summer Olympics were originally scheduled to
take place in Tokyo, Japan from July 24 to August 9,
2020. However, due to the COVID-19 pandemic, the
Olympics were postponed by a year. The rescheduled
2020 Tokyo Olympics ultimately took place from July 23
to August 8, 2021, still in Tokyo.

So while originally planned for 2020, the Games of the
XXXII Olympiad went ahead in 2021 after the
unprecedented delay caused by the global health
crisis. It was the first time in Olympic history that
the Games had been postponed rather than canceled.
```

ChatGPT 3.5

```
The 2020 Summer Olympics, despite their name, actually
took place in 2021 due to the postponement caused by
the COVID-19 pandemic. The Games were held in Tokyo,
Japan, from July 23 to August 8, 2021.
```

ChatGPT 4

```
The 2020 Summer Olympics, originally schedules for
2020, were postponed due to the COVID-19 pandemic and
took place from July 23 to August 8, 2021, in Tokyo,
Japan.
```

Model Overview

- Small models (1-2b):
  - Designed to be efficient and compact
  - Useful when accuracy is not the primary concern
  - Can be easily fine-tuned to specific uses
    - Text classification
    - Sentiment analysis
    - Basic chatbots
    - Simple definitions
    - Grammar/syntax corrections
  - Can be used even in underpowered devices
- Medium models (~7-13b)
  - Improved task performance
  - Enhanced contextual understanding
  - Increased ability to generalize
  - Efficient on average modern PCs
  - Very fast on high-end hardware
- Larger Models (70b):
  - State-of-the-art
  - Deep understanding of complex patterns and relationships
  - Capacity for abstraction
  - May rival top proprietary LLMs
  - Need high-end hardware to operate efficiently
- Size matters, but it’s not everything
- Things are progressing at an incredible pace

Thank you!

Questions?