Cristina Racoviță

Bogdan Bîndilă

GREEN LM INFERENCE

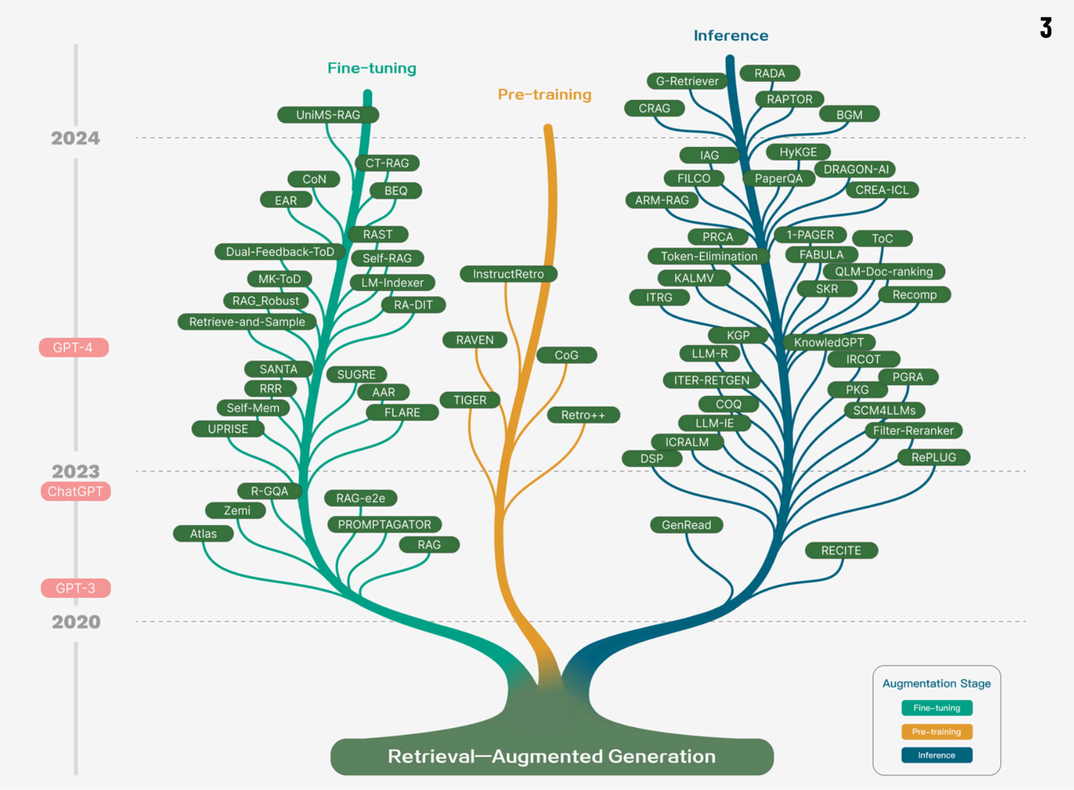
IMPACT OF RAG

GROUP 2

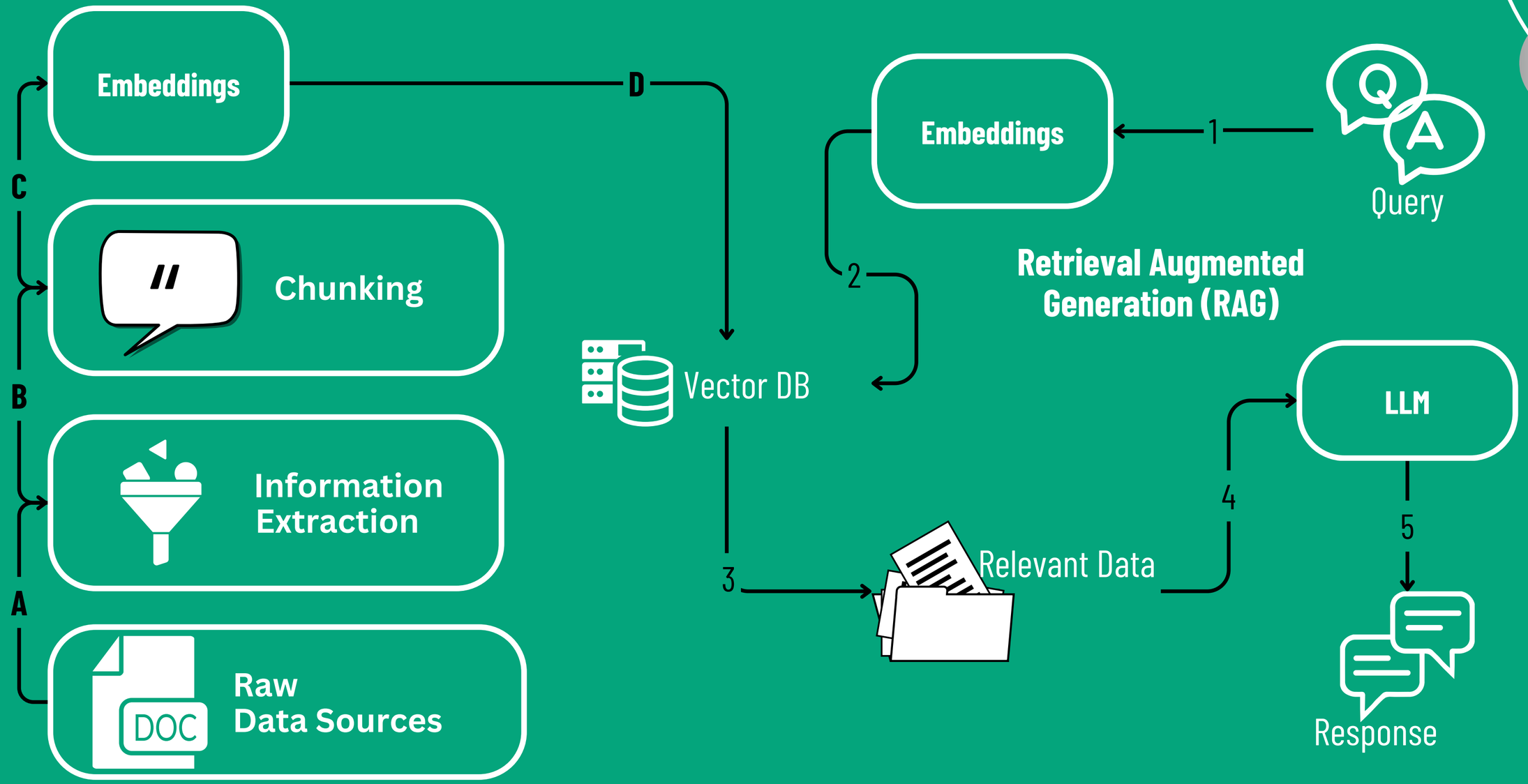
Shifting focus towards inference and RAG

August 2024 ~ **200 million active users**[1]
10% one query per day ~ **1500 MWh**
training GPT-3 = 1287 MWh[2]

Rise of inference-based use cases:
2023 was the year of **RAG**[2]



1. https://www.reuters.com/technology/artificial-intelligence/openai-says-chatgpts-weekly-users-have-grown-200-million-2024-08-29/
2. https://www.moodys.com/web/en/us/insights/resources/the-rise-of-ai-agents.pdf
3. Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL] https://arxiv.org/abs/2312.10997



The mechanics behind RAG

Where’s the gap?

LLMs are **impractical** for academic researchers and businesses with limited resources [4]

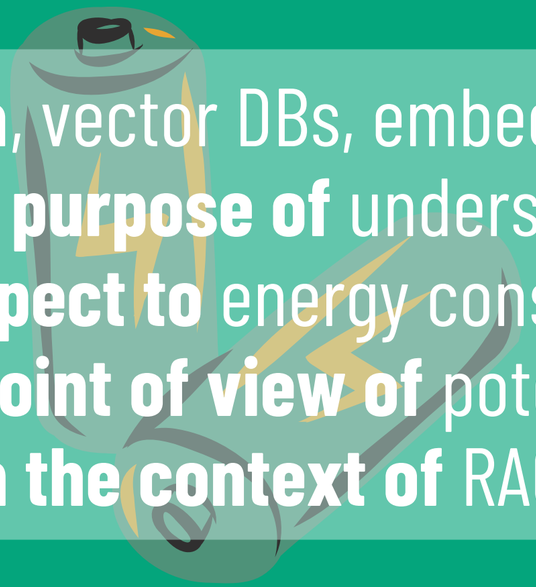
No emphasis on energy consumption during RAG

Studies on the power consumption of LLMs do not consider the use case of RAG,[5, 6] except for very limited work where the components are not varied[7]

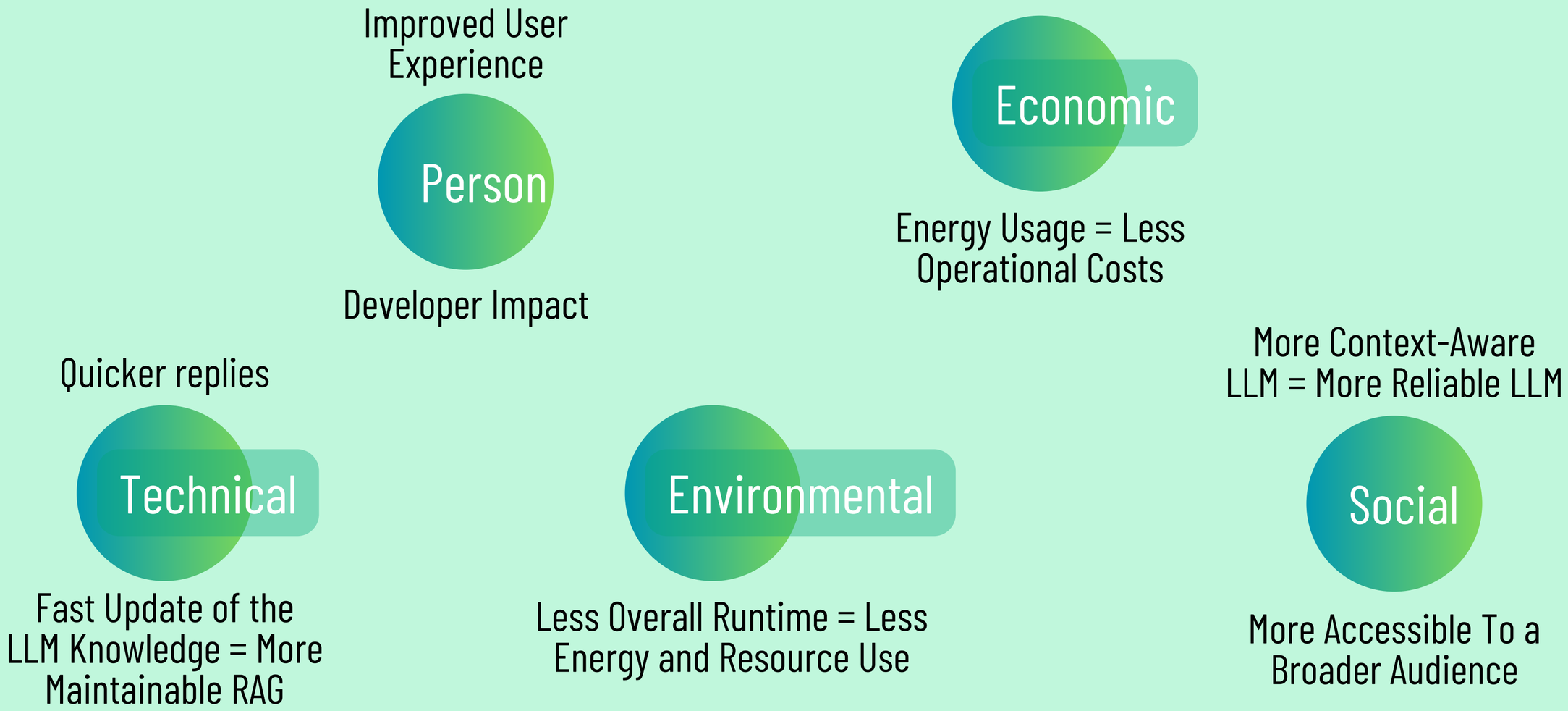
None of the benchmarks for embedding models[8, 9, 10] or vector databases[11] consider resource usage at all

4. Chen, Lihu, and Gaël Varoquaux. "What is the Role of Small Models in the LLM Era: A Survey." arXiv preprint arXiv:2409.06857 (2024)
5. Jovan Stojkovic, Esha Choukse, Chaojie Zhang, Inigo Goiri, and Josep Torrellas. 2024. Towards Greener LLMs: Bringing Energy-Efficiency to the Forefront of LLM Inference. arXiv:2403.20306 [cs.AI] https://arxiv.org/abs/2403.20306
6. Erik Johannes Husom, Arda Goknil, Lwin Khin Shar, and Sagar Sen. 2024. The Price of Prompting: Profiling Energy Use in Large Language Models Inference. arXiv preprint arXiv:2407.16893 (2024)
7. Mauricio Fadel Argerich and Marta Patiño-Martínez. 2024. Measuring and Improving the Energy Efficiency of Large Language Models Inference. IEEE Access 12 (2024), 80194–80207. https://doi.org/10.1109/ACCESS.2024.3409745
8. Martin Aumüller, Erik Bernhardsson, and Alexander John Faithfull. 2018. ANNBenchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms. CoRR abs/1807.05614 (2018). arXiv:1807.05614 http://arxiv.org/abs/1807. 05614
9. https://qdrant.tech/benchmarks/
10. https://redis.io/blog/benchmarking-results-for-vector-databases/
11. Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. arXiv:2210.07316 [cs.CL] https://arxiv. org/abs/2210.07316

Analyze the impact of data, vector DBs, embedding and language models
For the purpose of understanding
With respect to energy consumption
From the point of view of potential users
In the context of RAG.



Impacts


Improved User Experience
Person
Developer Impact
Economic
Energy Usage = Less Operational Costs
Quicker replies
Technical
Fast Update of the LLM Knowledge = More Maintainable RAG
Environmental
Less Overall Runtime = Less Energy and Resource Use
More Context-Aware LLM = More Reliable LLM
Social
More Accessible To a Broader Audience

Methodology

Datasets

Selected from the MTEB[11] and include queries

Name	Domain	Size	Entries	Queries
nfcorpus [N]	Medical Information	5.97 MB	3956	3240
arguana [A]	Medical Information	9.69 MB	10080	1410
cqadupstack-webmasters [C]	Web Development	13.1 MB	17911	506

Language Models

gemma-2-2b-it[12]

2.61B parameters

context size: 8k tokens

Llama-3.2-1B-Instruct[13]

1.24B parameters

context size: 128k tokens

4-bit quantization

12. Team, Gemma, et al. "Gemma 2: Improving open language models at a practical size." arXiv preprint arXiv:2408.00118 (2024).
13. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/

Methodology

Embedding Models

1. have the same underlying architecture[14]
2. be among the most performant in their category

Embedding model	Embedding size	Parameters
gte-large	1024	335M
gte-base	768	109M
gte-small	384	33.4M

Vector Databases

Picked the most used client-server vector DBs according to GitHub stars

Weaviate

14. Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. CoRR abs/2308.03281 (2023). https://doi.org/10.48550/ARXIV.2308. 03281 arXiv:2308.03281

Methodology

Metrics

CPU Cores Energy (J)

GPU Core NVVDD Output Power (J)[15]

RAM Energy (J)

duration (s)

Measured with HWiNFO at 10Hz

Hardware

Laptop

Intel i7-7700HQ 4c/8t 2.80GHz

Samsung 960 Evo SSD

16 GB RAM

NVidia GTX 1050Ti Mobile 4 GB

Software

Windows 10 Pro 22H2

vector DBs run in Docker

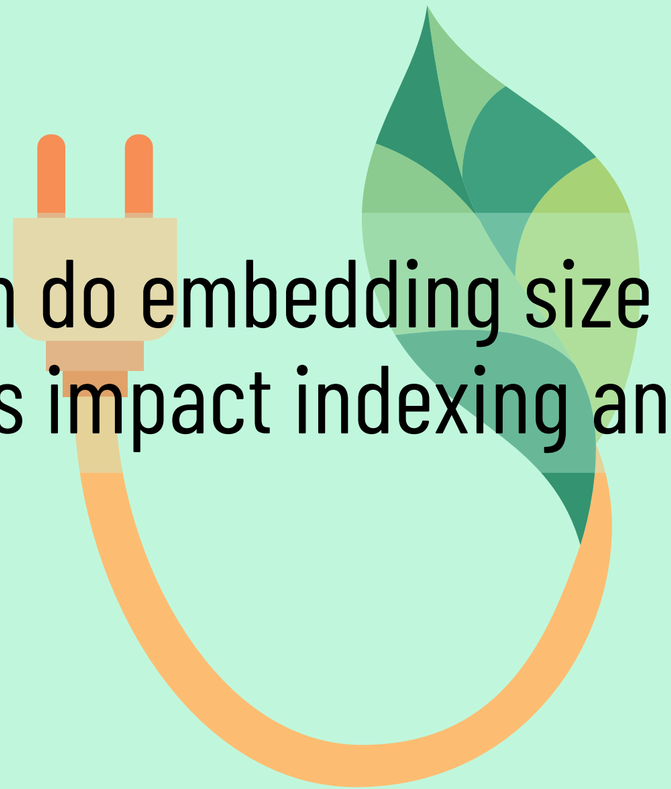
LMs run using Ollama

Python programming language

Bash scripts to automate experiments

Execute experiments for RQ1, RQ2 and RQ3 10 times

15. https://forums.developer.nvidia.com/t/nvidia-387-12-breaks-power-reading-in-nvidia-smi/53946/3



RQ1: How much do embedding size and number of embeddings impact indexing and retrieval?

vector DB: Milvus

Workload 1: Indexing
each dataset

model	CPU Cores Energy [J]	DRAM Energy [J]	duration [s]
gte-large	1243.29 ± 711.74	56.82 ± 32.85	74.87 ± 42.94
gte-medium	1202.28 ± 699.04	54.71 ± 31.88	72.2 ± 41.79
gte-small	1193.89 ± 693.78	54.78 ± 32.18	71.68 ± 41.24

Kruskal-Wallis test: no statistical difference

dataset	CPU Cores Energy [J]	DRAM Energy [J]	duration [s]
A	1057.17 ± 64.33	48.3 ± 4.42	63.46 ± 1.93
C	2121.37 ± 87.0	97.05 ± 5.66	127.5 ± 3.11
N	460.93 ± 49.97	20.96 ± 2.9	27.78 ± 1.28

Workload 2: Querying
top 10 entries for 500 queries

model	CPU Cores Energy [J]	DRAM Energy [J]	duration [s]
gte-large	145.6 ± 13.44	5.8 ± 0.48	11.75 ± 1.32
gte-medium	138.51 ± 16.31	5.66 ± 0.9	10.61 ± 1.28
gte-small	112.85 ± 11.96	4.59 ± 0.37	9.02 ± 1.02

dataset	CPU Cores Energy [J]	DRAM Energy [J]	duration [s]
A	128.1 ± 20.63	5.35 ± 1.05	9.6 ± 1.03
C	123.26 ± 16.61	5.09 ± 0.67	9.68 ± 1.15
N	145.6 ± 14.71	5.61 ± 0.62	12.1 ± 1.33

Results reported as mean ± std

The embedding size does not influence the indexing.

RQ2: To what extent can the choice of the vector DB diminish resource usage?

cosine similarity
embedding size: 758
dataset: C (17911 rows)

Indexing

database	CPU Cores Energy [J]	DRAM Energy [J]	duration [s]
milvus	2142.85 ± 115.0	100.7 ± 6.63	132.04 ± 2.16
qdrant	2333.64 ± 30.18	106.59 ± 3.2	351.55 ± 18.21
weaviate	199.29 ± 4.74	14.0 ± 0.43	10.99 ± 0.19

Results reported with 95% CI

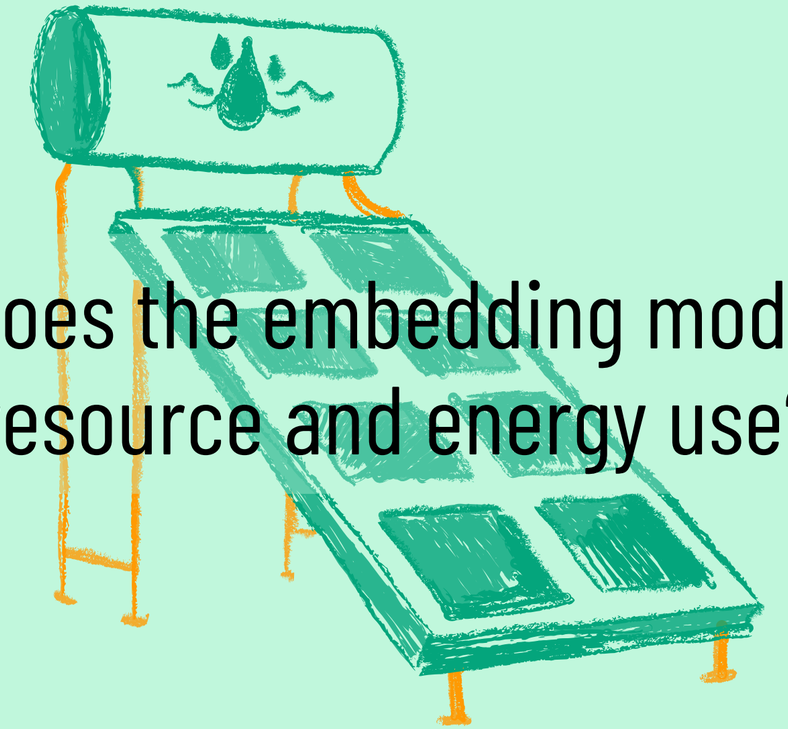
Querying

top 10 entries for 500 queries

database	CPU Cores Energy [J]	DRAM Energy [J]	duration [s]
milvus	120.4 ± 3.3	4.8 ± 0.19	9.93 ± 0.11
qdrant	125.71 ± 11.09	7.08 ± 0.8	8.91 ± 0.18
weaviate	25.02 ± 0.47	1.24 ± 0.04	2.14 ± 0.02

Kruskal-Wallis test: no statistical difference between Milvus and Qdrant during querying

Weaviate is much more efficient for indexing (up to 31.98x) and querying (up to 4,64x) when having one client.



RQ3: How does the embedding model influence resource and energy use?

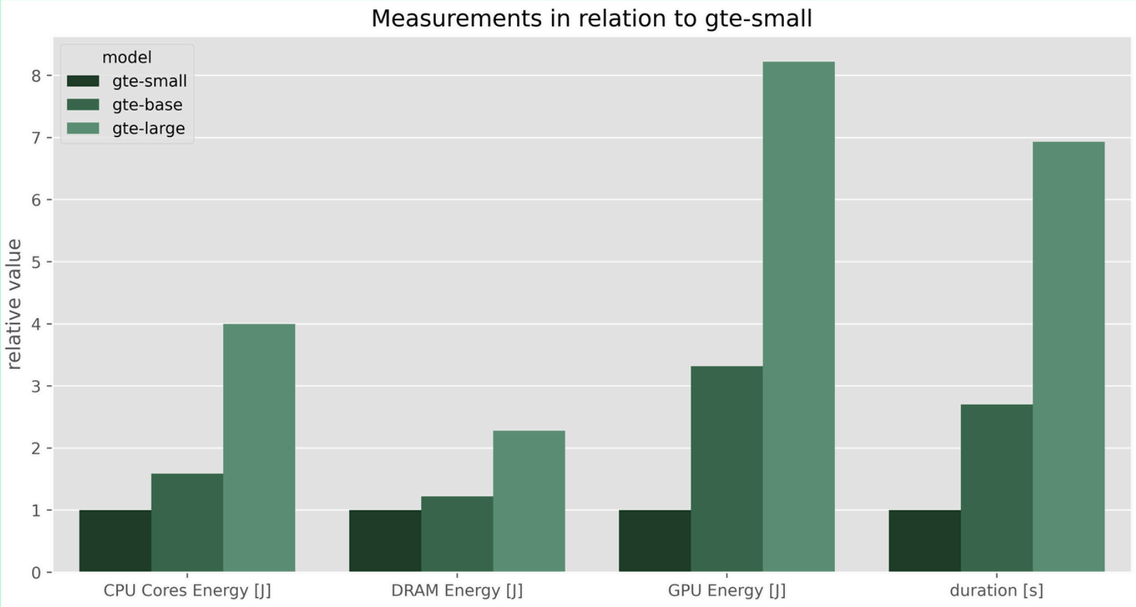
model	CPU Cores Energy [J]	DRAM Energy [J]	GPU Energy [J]	duration [s]
gte-base	314.81 ± 5.35	19.31 ± 0.4	7894.51 ± 25.4	242.31 ± 0.37
gte-large	792.99 ± 122.57	36.02 ± 5.18	19563.35 ± 24.57	621.28 ± 0.37
gte-small	198.31 ± 2.63	15.8 ± 0.29	2379.56 ± 13.25	89.62 ± 0.09

Results reported with 95% CI

dataset: N (3956)

batch size: 1

Wilcoxon signed-rank test rejects the null hypothesis



model	rank	retrieval score
gte-large	50	52.22
gte-base	54	51.14
gte-small	76	49.46

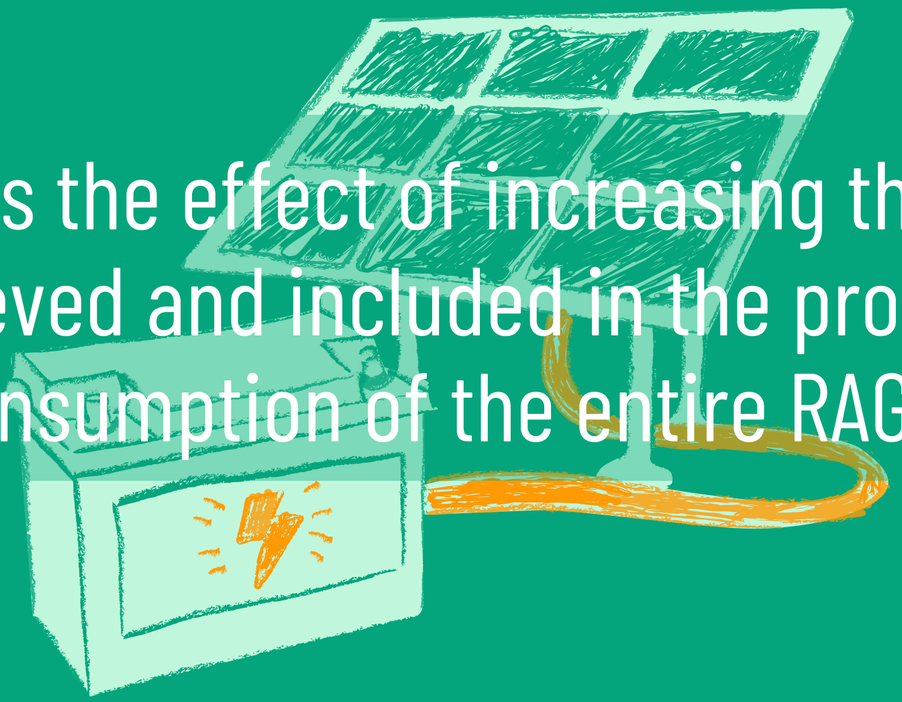
total models: 448

retrieval score range: 62.65 - 7.94[16]

Larger embedding models consume a lot of resources with marginal benefits.

16. https://huggingface.co/spaces/mteb/leaderboard

RQ4: What is the effect of increasing the number of texts retrieved and included in the prompt on the energy consumption of the entire RAG process?



Answering a question with RAG

# retrieved docs	CPU Cores Energy [J]	DRAM Energy [J]	GPU Energy [J]	duration [s]	# tokens prompt
1	82.7 ± 10.55	9.03 ± 1.13	140.04 ± 16.39	10.32 ± 1.25	362.94 ± 21.99
2	83.37 ± 9.01	9.65 ± 1.05	163.44 ± 16.24	11.75 ± 1.2	681.55 ± 32.49
3	97.3 ± 10.42	11.22 ± 1.21	211.69 ± 19.6	14.38 ± 1.45	1018.26 ± 42.67
4	117.87 ± 11.65	13.41 ± 1.34	270.3 ± 21.69	17.89 ± 1.62	1353.32 ± 49.68
5	134.94 ± 12.54	15.21 ± 1.43	307.32 ± 23.05	20.35 ± 1.73	1690.76 ± 56.45

Wilcoxon signed-rank test rejects the null hypothesis

Results reported with 95% CI

Total consumed resources increase linearly with the number of included documents in the prompt.

The number of generated tokens affects the energy consumption the most.

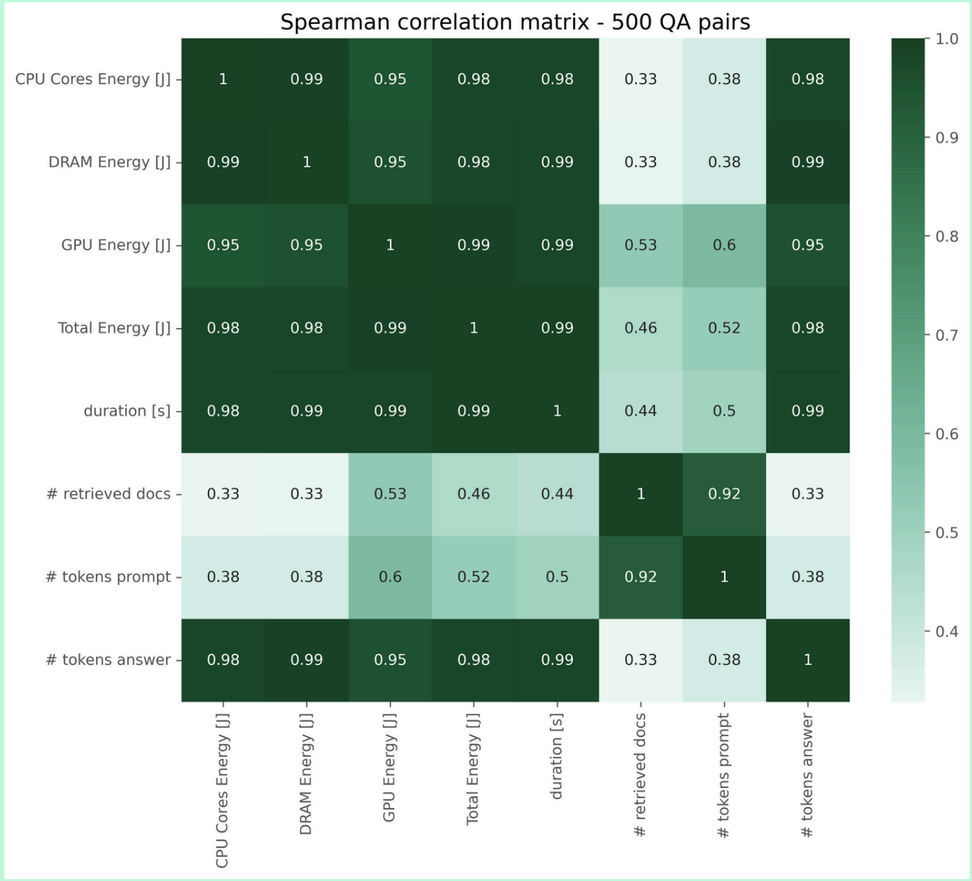
dataset: N (3956)

vector DB: Qdrant

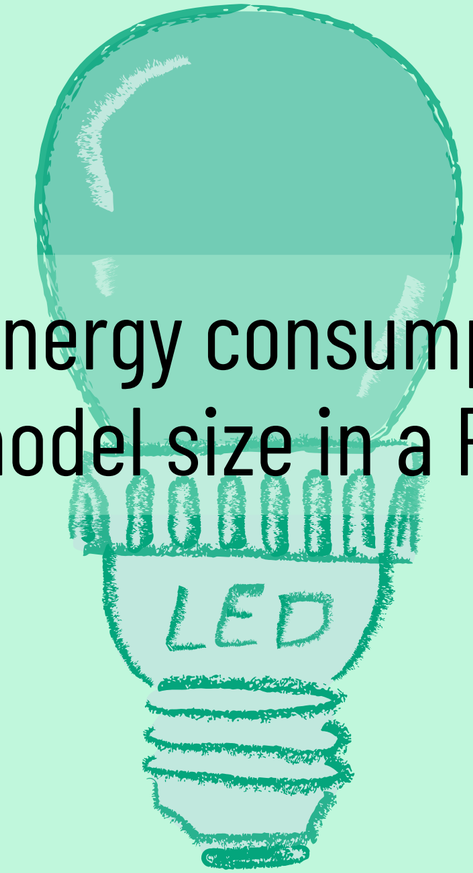
embedding model: gte-base

number of questions: 100

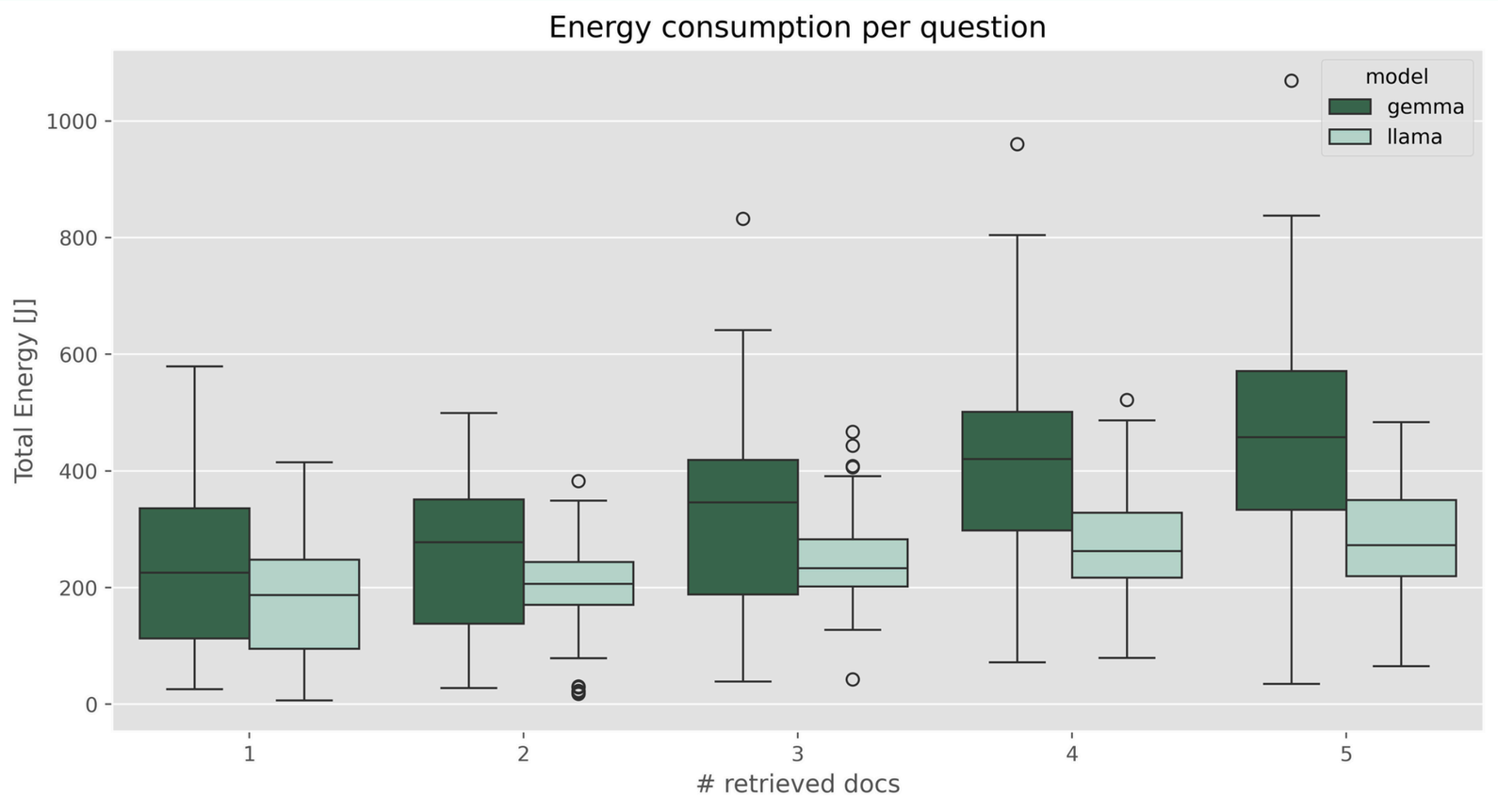
language model: Gemma 2 2B



RQ5: How does energy consumption change with language model size in a RAG system?



same setup as for RQ4



metric	Gemma / Llama
total energy	1.9
duration	2.56

Threats to Validity

Internal Validity

Windows as operating system

Only one active user

External Validity

Only one hardware and software environment for our experiments

Limited settings for our experiments

Test only up to 3 variations per RAG component

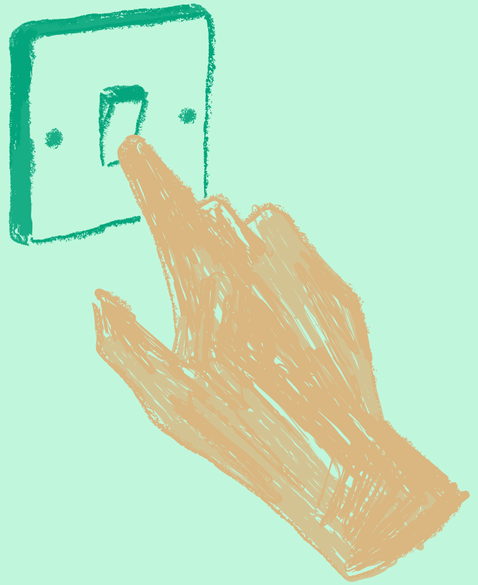
Construct Validity

All our measurements depend on HWiNFO accuracy

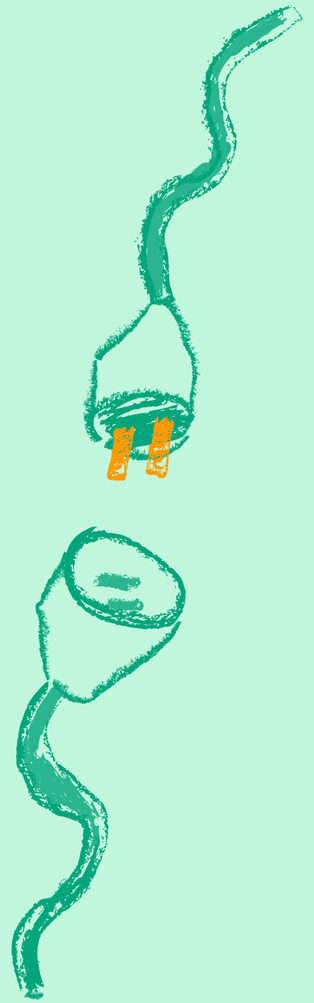
Only CPU, GPU*, and RAM Energy is measured

Conclusion Validity

Low statistical power due to the small sample sizes



Mitigation Strategies



Strict protocol for running the experiments:

- restarting the laptop before doing each experiment and no Wi-fi connection
- keeping the laptop plugged in and using "best performance" battery profile
- minimizing the HWiNFO tool before starting the measurements
- monitoring the GPU's memory usage to ensure we do not exceed its limits
- loading the language and embedding models beforehand
- setting a fixed seed for random processes to ensure repeatability
- unified settings for the vector DBs

Conclusion validity:

- subtract idle energy consumption and account for the Docker overhead
- choosing each statistical test based on the data properties
- repeat experiments 10 times and report the CI

Conclusion

Smart choices can be made about the components of a RAG system, as some significantly impact energy consumption.

Future work

Perform combinatorial experiments with all components of an RAG system

Increase the variety of tested components (e.g. larger datasets)

Record measurements with a higher sampling rate

Extend the framework to other operating systems

Run the experiments on multiple machines

Thank You

Questions?