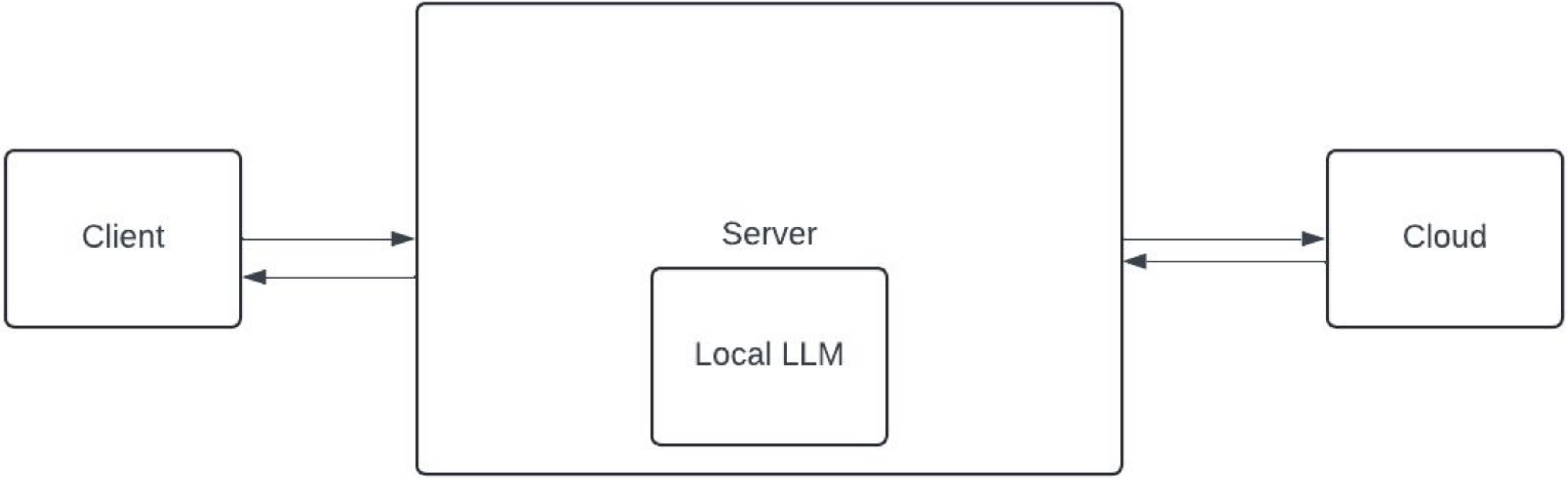
DocuQuest: Optimizing Large Language Model Inference with a Hybrid Cloud-Edge System for Document Summarization

Motivation

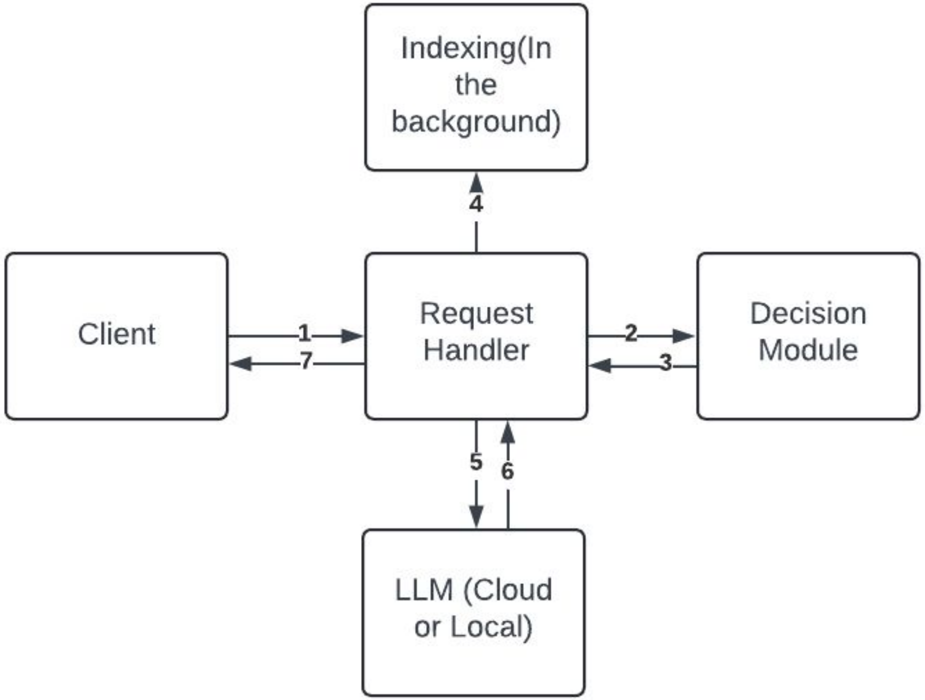
- Large Language Models (LLMs) excel in tasks like summarization and question-answering but are resource-intensive and challenging to deploy due to high latency and memory usage.

- A hybrid system combining local and cloud LLMs offers a balanced solution by dynamically assigning tasks based on complexity, optimizing latency, memory utilization, and performance.

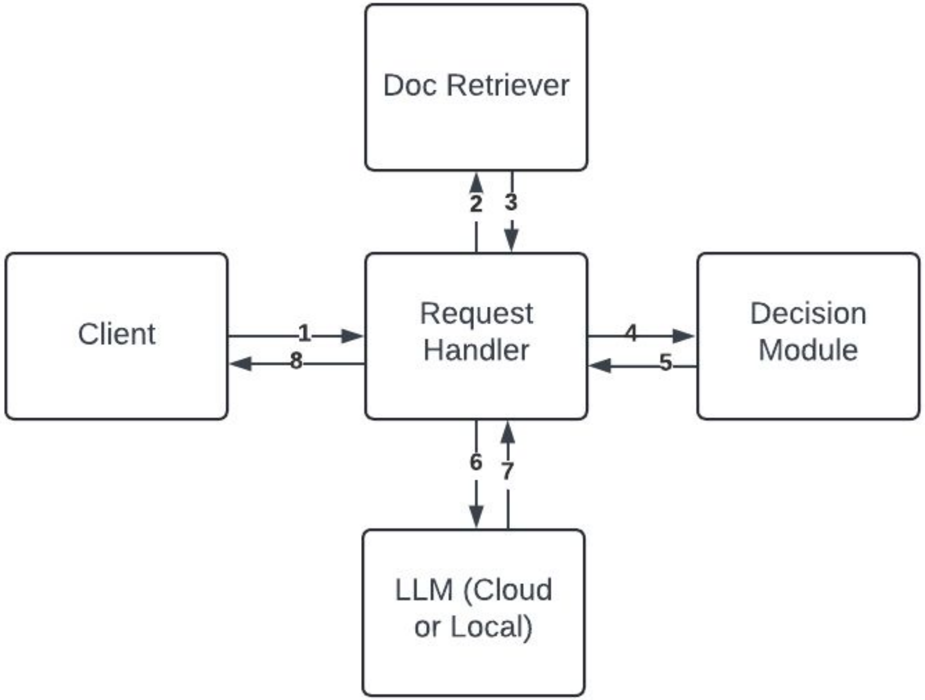
System Design



Low Level Design (/summarize)



Low Level Design (/ask)



Decision Module

- **Objective:** Efficiently allocate tasks to either the local or cloud LLM based on complexity to optimize resource usage

- **System Health Evaluation**
  - Check CPU Usage and Available memory
  - If not sufficient, routes to cloud
- **Document Complexity Prediction**
  - Calculates document complexity based on metrics like type-token ratio, average sentence length, Flesch-Kincaid score
  - A trained classifier predicts document complexity based on this score
  - If low, use local, else use cloud

Tech Stack

- **Frontend:** Streamlit, providing a user-friendly interface for uploading documents and asking questions.
- **Backend:** FastAPI for efficient task handling and routing between local and cloud models, with additional support from MLX (for mac), and LangChain.
- **Models:**
  - **Local LLM:** LLaMA 3.2 (1B parameters).
  - **Cloud LLM:** LLaMA 3.1 (8B parameters).

Experiments and Results

Evaluation Metrics:

- Datasets: XSum, Arxiv, Gov-Report
- Latency: Time taken for task completion.
- Quality: Measured using ROUGE scores and BERTScore F1 for summarization tasks.

Experiments and Results

Metric	Local	Cloud
Time taken(s)	7.721500	**2.349000**
rougeL	**0.137550**	0.116912
BERTScore F1	0.855441	**0.858433**

Mean values for Local and Cloud-based summarization for the X-Sum dataset

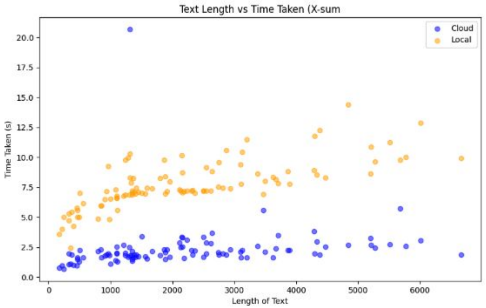
Metric	Local	Cloud
time taken(s)	34.276100	**4.770200**
rougeL	**0.193011**	0.182547
BERTScore F1	0.827140	**0.835522**

Mean values for Local and Cloud summarization for the ARXIV dataset

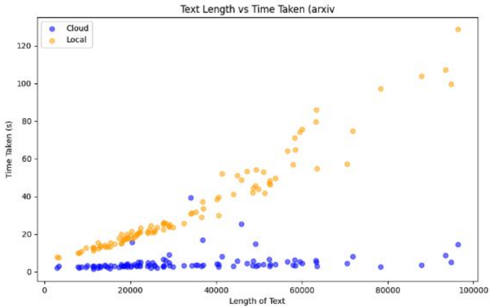
Metric	Local	Cloud
time taken (s)	31.98	**4.599000**
rougeL	0.122849	**0.165567**
BERTScore F1	0.833495	**0.855592**

Mean values for Local and Cloud summarization for the GOV-REPORT dataset

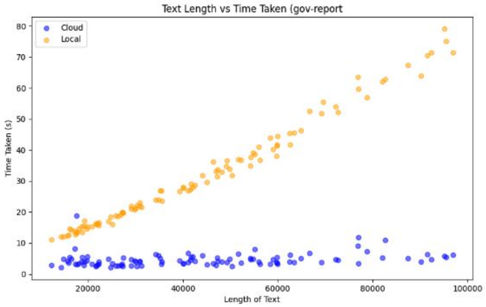
Experiments and Results



(a) X-Sum



(b) Arxiv



(c) Gov-report

Experiments and Results

- Local LLMs are suitable for handling simpler tasks with significantly lower latency.
- Cloud LLMs excel in complex tasks, providing superior quality at the cost of higher latency.

Conclusion

- Developed a hybrid system that dynamically allocates tasks to local or cloud-based LLMs based on complexity
- Demonstrated the feasibility of deploying LLMs on edge devices for simpler tasks, reducing latency and resource consumption

Weaknesses and Future Work

- **Speculative Decoding:** Implemented as a stretch goal but not analyzed due to resource limitations, such as storage capacity (256GB SSD) and free-tier memory constraints on cloud platforms.
- **Mac-Specific Limitations:** The implementation is macOS-dependent, reducing portability and applicability across other platforms.
- **Hardware Metrics Analysis:** GPU and RAM usage analysis was restricted due to macOS incompatibility with tools like `pynvml`.
- **Decision Module Accuracy:** Limited testing under varying system loads and with diverse documents, leaving its robustness and accuracy unverified.

Thank You