ALCF Webinar series
December 3rd, 2025

DEPLOYING AI INFERENCE SERVICES AT ALCF

BENOIT CÔTÉ
Data Services
Software Developer
ALCF
bcote@anl.gov

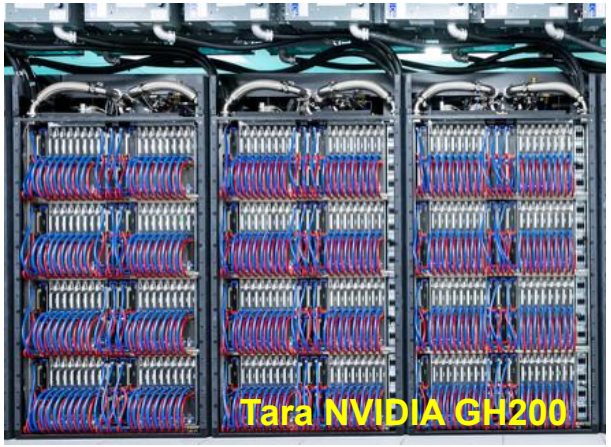
ADITYA TANIKANTI
Data Services
Architect
ALCF
atanikanti@anl.gov

TOM URAM
Data Services and
Workflows Team Lead
ALCF
turam@anl.gov

VENKATRAM VISHWANATH
AI / ML
Team Lead
ALCF
venkat@anl.gov

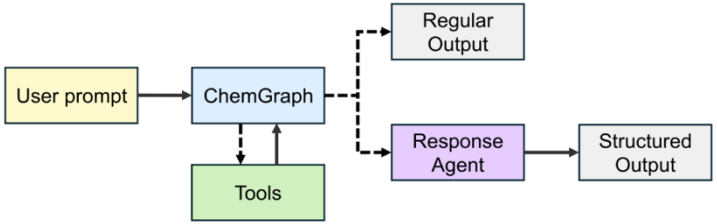
U.S. DEPARTMENT of ENERGY
Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.
Argonne NATIONAL LABORATORY

ALCF IS DEPLOYING DIVERSE INFERENCE SYSTEMS FOR SCIENCE



SOME INFERENCE SERVICE USE-CASES

ChemGraph – Thang et al. https://www.arxiv.org/pdf/2506.06363



Workflow and Cheminformatics

- LangGraph
- ASE
- RDKit
- PubChemPy

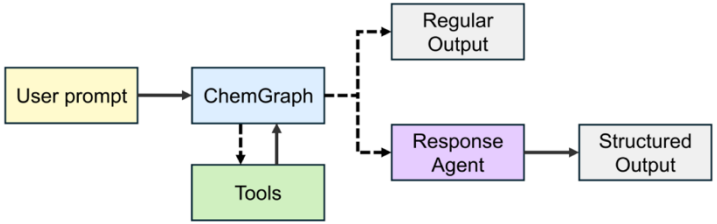
Simulation Backends

Semi-empirical	Ab initio	ML Potentials
- xTB	- NWChem	- MACE
- EMT	- ORCA	- UMA

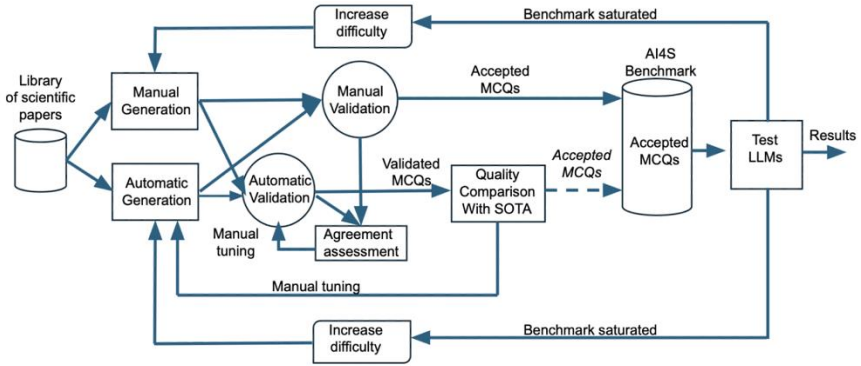
Largest usage is currently for generating synthetic data for science

SOME INFERENCE SERVICE USE-CASES

ChemGraph – Thang et al. https://www.arxiv.org/pdf/2506.06363



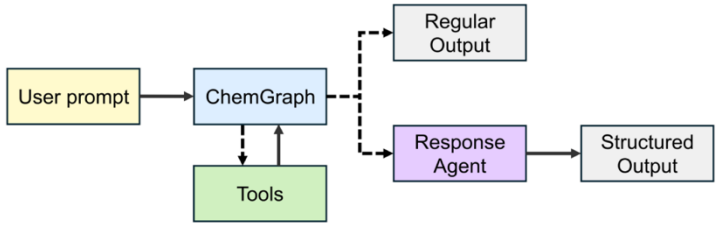
AuroraGPT-EAIRA – Capello et al. https://arxiv.org/pdf/2502.20309



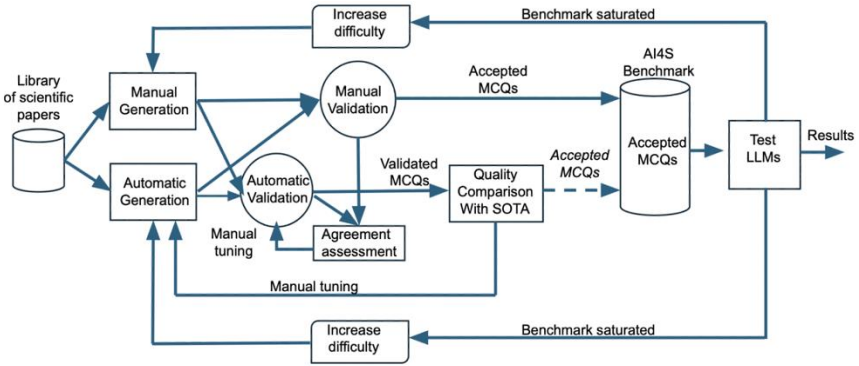
Largest usage is currently for generating synthetic data for science

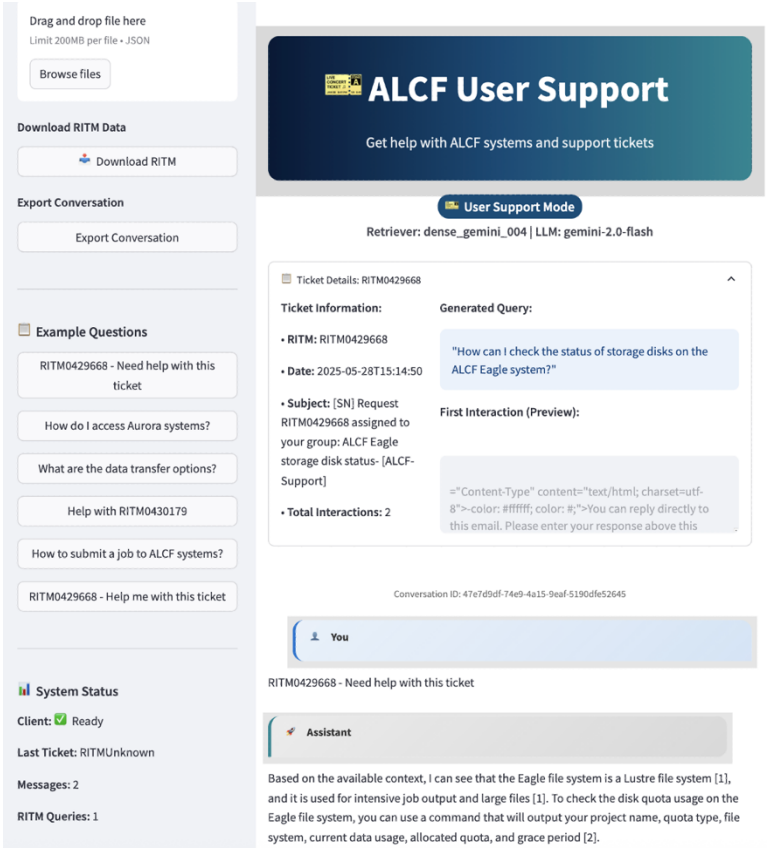
SOME INFERENCE SERVICE USE-CASES

ChemGraph – Thang et al. https://www.arxiv.org/pdf/2506.06363



AuroraGPT-EAIRA – Capello et al. https://arxiv.org/pdf/2502.20309

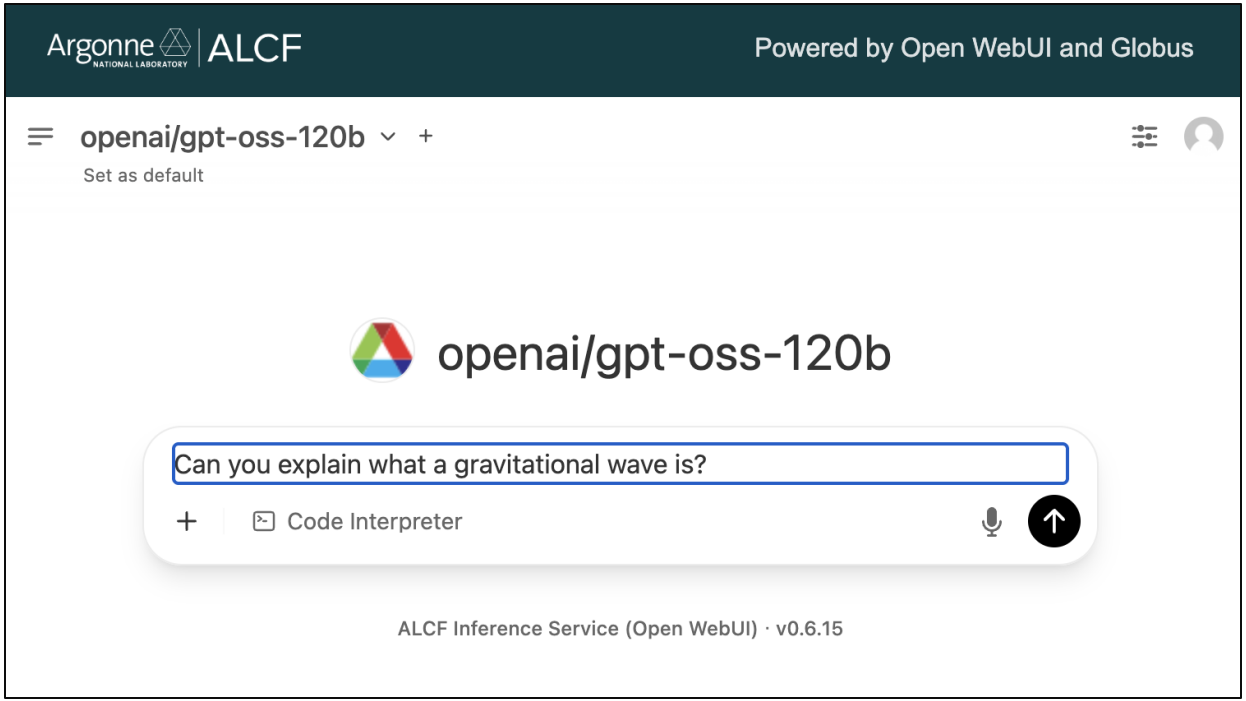




Largest usage is currently for generating synthetic data for science

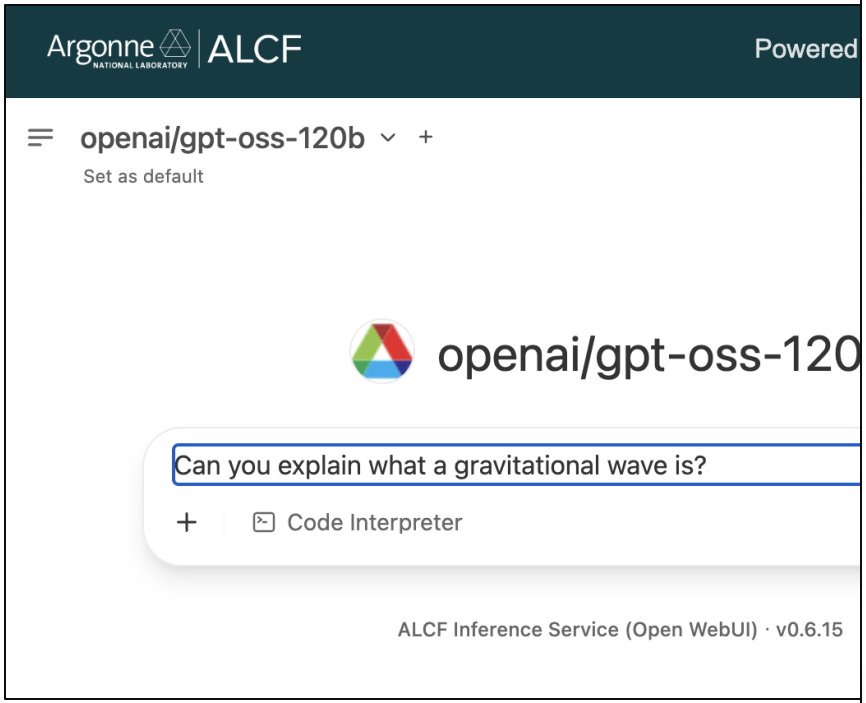
INTERFACING WITH THE INFERENCE SERVICE

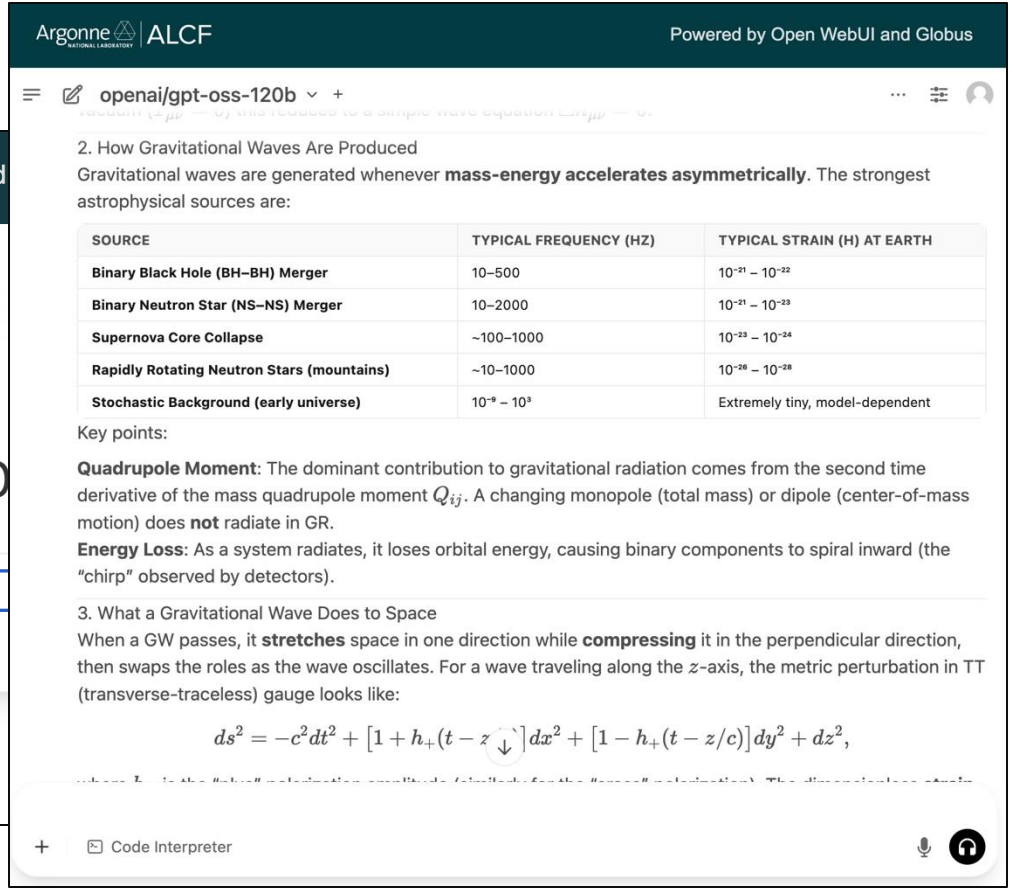
Open WebUI interface
https://inference.alcf.anl.gov



INTERFACING WITH THE INFERENCE SERVICE

Open WebUI interface
https://inference.alcf.anl.gov





INTERFACING WITH THE INFERENCE SERVICE

OpenAI API (including batch)
https://docs.alcf.anl.gov/services/inference-endpoints

cURL | Python (OpenAI SDK)

```
#!/bin/bash

# Get your access token
access_token=$(python inference_auth_token.py get_access_token)

curl -X POST "https://inference-api.alcf.anl.gov/resource_server/sophia/vllm/v1/chat/completions" \
     -H "Authorization: Bearer ${access_token}" \
     -H "Content-Type: application/json" \
     -d '{
            "model": "meta-llama/Meta-Llama-3.1-8B-Instruct",
            "messages":[{"role": "user", "content": "Explain quantum computing in simple terms."}]
         }'
```

API Usage Examples ¶

Querying Endpoint Status

Querying Endpoint Status ›

Chat Completions

Chat Completions ›

INTERFACING WITH THE INFERENCE SERVICE

OpenAI API (including batch)

https://docs.alcf.anl.gov/services/inference-endpoints

cURL | Python (OpenAI SDK)

```
#!/bin/bash

# Get your access token
access_token=$(python inference_auth_token.py get_access_token)

curl -X POST "https://inference-api.alcf.anl.gov/resource_server/sophia
     -H "Authorization: Bearer ${access_token}" \
     -H "Content-Type: application/json" \
     -d '{
           "model": "meta-llama/Meta-Llama-3.1-8B-Instruct",
           "messages":[{"role": "user", "content": "Explain quantum co
         }'
```

API Usage Examples ¶

Querying Endpoint Status

Querying Endpoint Status

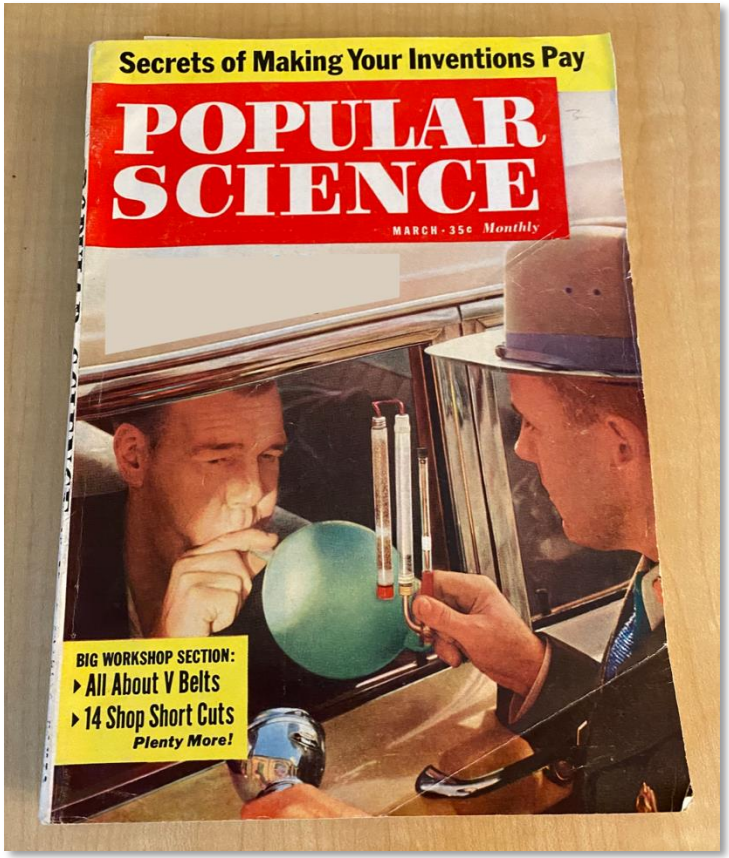
Chat Completions

Chat Completions


```
{
    "id": "chatcmpl-68de443dde8b46659b4c34
    "object": "chat.completion",
    "created": 1755114580,
    "model": "meta-llama/Meta-Llama-3.1-8B
    "choices": [
        {
            "index": 0,
            "message": {
                "role": "assistant",
                "content": "Quantum computing is a new way of processing information that's
 different from the way regular computers work. Here's a simplified explanation:\n\n**Regul
ar Computers:**\n\nRegular computers use \"bits\" to store and process information. Bits ar
e like light switches that can be either ON (1) or OFF (0). When you combine these bits, yo
u get numbers, letters, and other data.\n\n**Quantum Computers:**\n\nQuantum computers use
\"qubits\" (quantum bits) to store and process information. Qubits are special because they
 can be both ON and OFF at the same time, which is called a \"superposition.\" This means a
 qubit can process multiple possibilities simultaneously, making it much faster than regula
r computers for certain tasks.\n\n**Another Key Concept: Entanglement**\n\nQubits can also
be \"entangled,\" which means that when something happens to one qubit, it instantly affect
s the other qubits, no matter how far apart they are. This allows quantum computers to perf
orm calculations on multiple qubits simultaneously, making them incredibly powerful.\n\n**H
ow Quantum Computing Works:**\n\nImagine you have a combination lock with 10 numbers. A reg
ular computer would try each number one by one, taking a long time to find the correct comb
ination. A quantum computer, on the other hand, can try all 10 numbers simultaneously, than
ks to the power of qubits and entanglement. This makes quantum computing incredibly fast fo
r certain tasks, such as:\n\n1. **Cryptography:** Breaking complex codes and encryption met
hods.\n2. **Optimization:** Finding the best solution for complex problems, like logistics
and supply chain management.\n3. **Simulation:** Simulating complex systems, like weather p
atterns and molecular interactions.\n\n**Challenges and Limitations:**\n\nQuantum computing
 is still a developing field, and there are many challenges to overcome, such as:\n\n1. **E
rror correction:** Qubits are prone to errors, which can affect the accuracy of calculation
s.\n2. **Scalability:** Currently, quantum computers are small and can only perform a limit
ed number of calculations.\n3. **Noise:** Quantum computers are sensitive to external noise
, which can disrupt calculations.\n\n**Conclusion:**\n\nQuantum computing is a revolutionar
y technology that has the potential to solve complex problems that are currently unsolvable
 or take too long to solve with regular computers. While it's still in its early stages, re
searchers and companies are working to overcome the challenges and limitations, and we can
expect to see significant advancements in the coming years.",
```

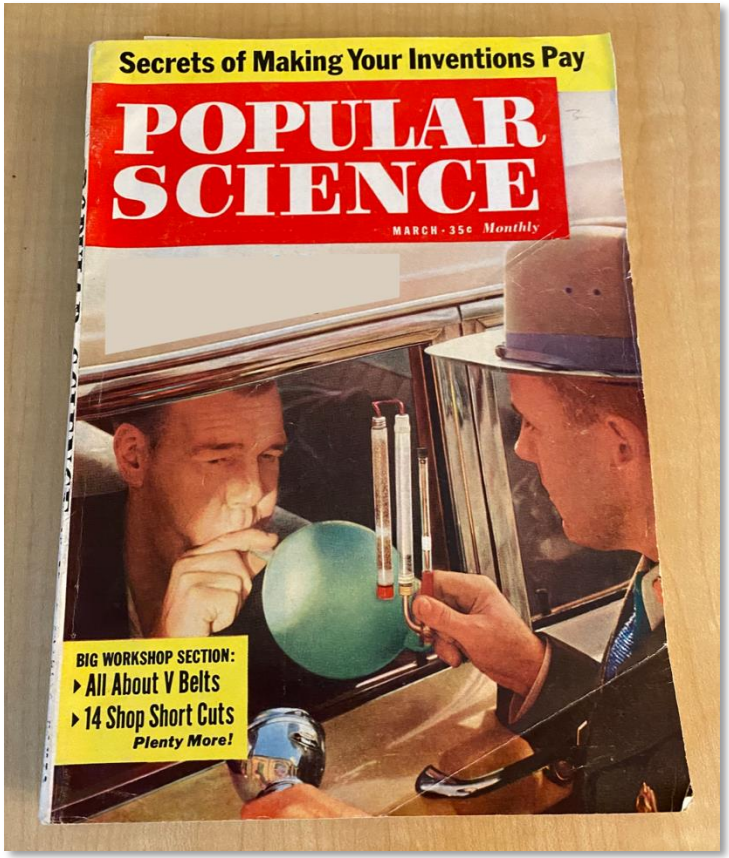
```
"usage": {
    "prompt_tokens": 43,
    "total_tokens": 436,
    "completion_tokens": 393,
    "prompt_tokens_details": null
},
"prompt_logprobs": null,
"kv_transfer_params": null,
"response_time": 3.179178237915039,
"throughput_tokens_per_second": 137.14235798428732
```

BLAST FROM THE PAST



March 1961

BLAST FROM THE PAST



By Martin Mann

The month in science

Now hear this, IBM 704. Everybody talks to machines. It is well

Conversational ability is one clue to machines' intelligence. The British mathematician Turing said (in 1938, before the first digital computer had been built) that a real thinking machine could carry on a conversation with a man in another room—and the man

CONTINUED 23

The month in science continued

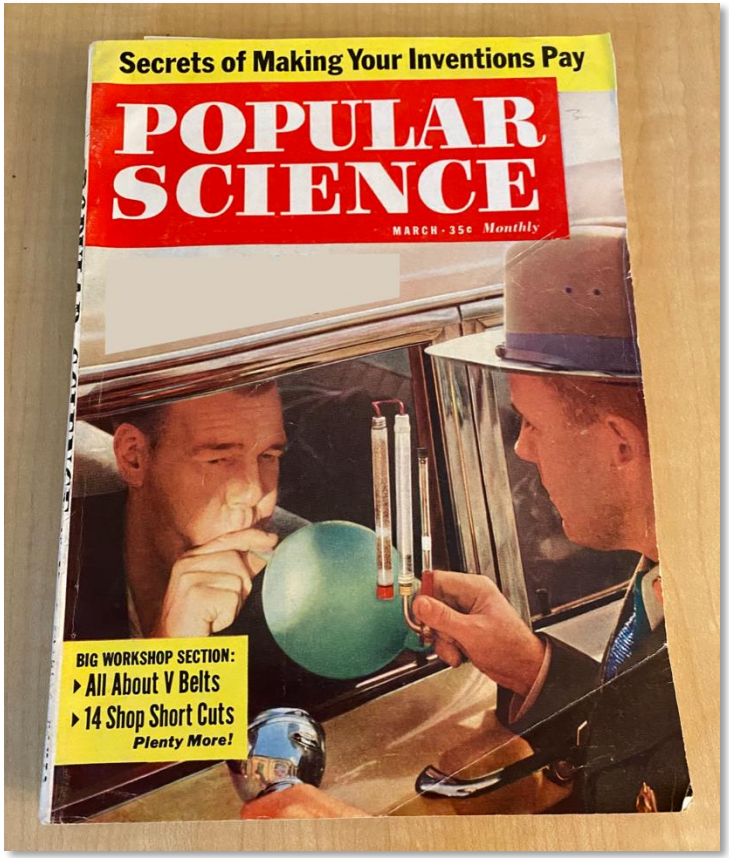
couldn't tell if he were talking to a machine or another man. Some new computers almost pass this test. Here's an excerpt from a man-machine chat about the weather:

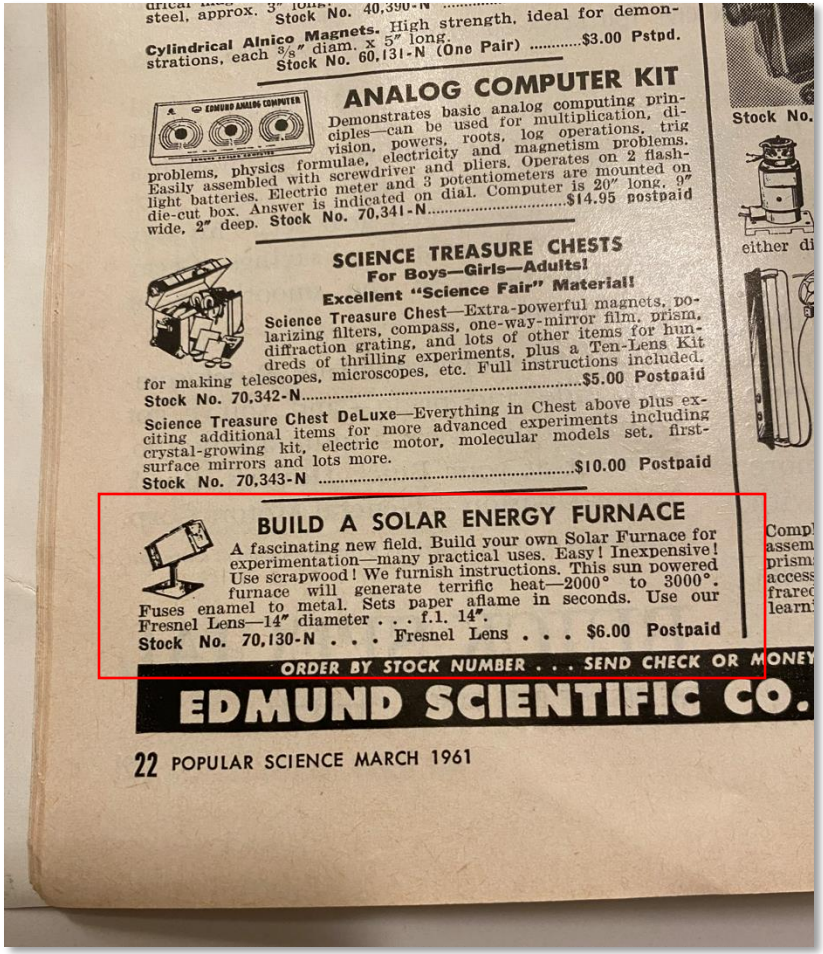
Man: In hot weather one needs at least one bath a day.
Computer: Yes, I was just out and it was sweltering.
Man: When Christmas comes we will have some colder weather.
Computer: Cold? Yes, it's usually pretty frosty during December.

The robots haven't quite taken over yet, though. Robert Sanford points out that no computer dialect includes the word "think." The machine wouldn't know what to do.

March 1961

BLAST FROM THE PAST



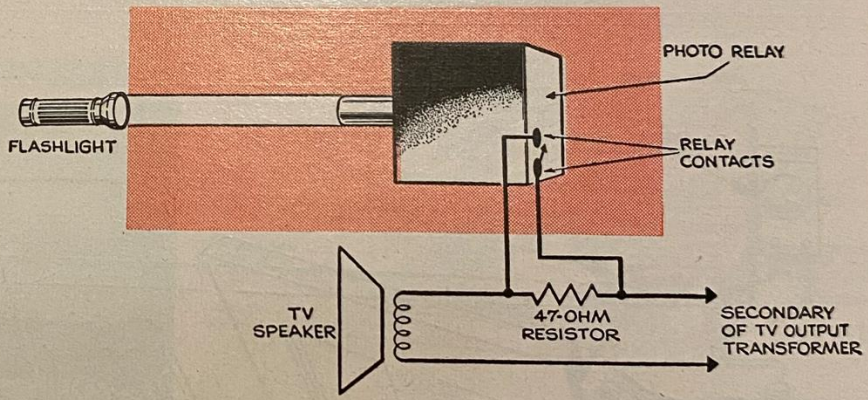


March 1961

BLAST FROM THE PAST

TV-COMMERCIAL KILLER

MOUNT A MAILING TUBE over the relay aperture to exclude room light. Aim it toward your viewing chair. Adjust the sensitivity control until a flashlight aimed straight at it trips the relay. Connect a 47-ohm resistor in series with the speaker voice coil in the TV, and connect the relay as shown in the diagram. Now you can mute noisy commercials with the beam of the flashlight.



PRESENTATION OVERVIEW

- **ALCF Inference Service Architecture**
  - Overview of the system components
  - Authentication and authorization
  - Orchestration and configuration
  - Available models

- **Capabilities and Features**
  - Latency, scaling, federated endpoints
  - Monitoring, production-ready with containers

- **Usage and Examples**
  - How to use the API (Python, cURL)
  - How to use the web interface

PRESENTATION OVERVIEW

- **ALCF Inference Service Architecture**
  - Overview of the system components
  - Authentication and authorization
  - Orchestration and configuration
  - Available models

- **Capabilities and Features**
  - Latency, scaling, federated endpoints
  - Monitoring, production-ready with containers

- **Usage and Examples**
  - How to use the API (Python, cURL)
  - How to use the web interface

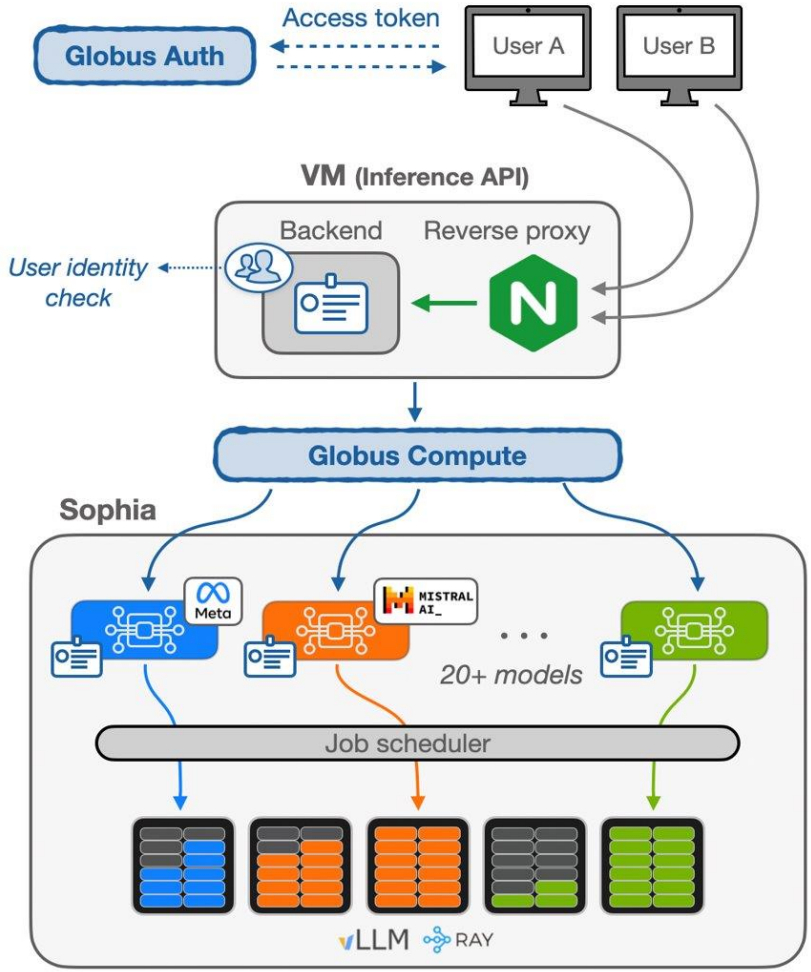
SYSTEM OVERVIEW

- Inference Service leverages ALCF computing resources to serve requests to a growing set of models
- Authentication via Globus Auth and orchestration using Globus Compute
- Combination of “in-memory/active” models and on-demand schedulable models
- Usage metrics are curated to understand and improve the service

...

Minerva (NVIDIA B200)

Metis (SambaNova SN40L)



https://dl.acm.org/doi/10.1145/3731599.3767346

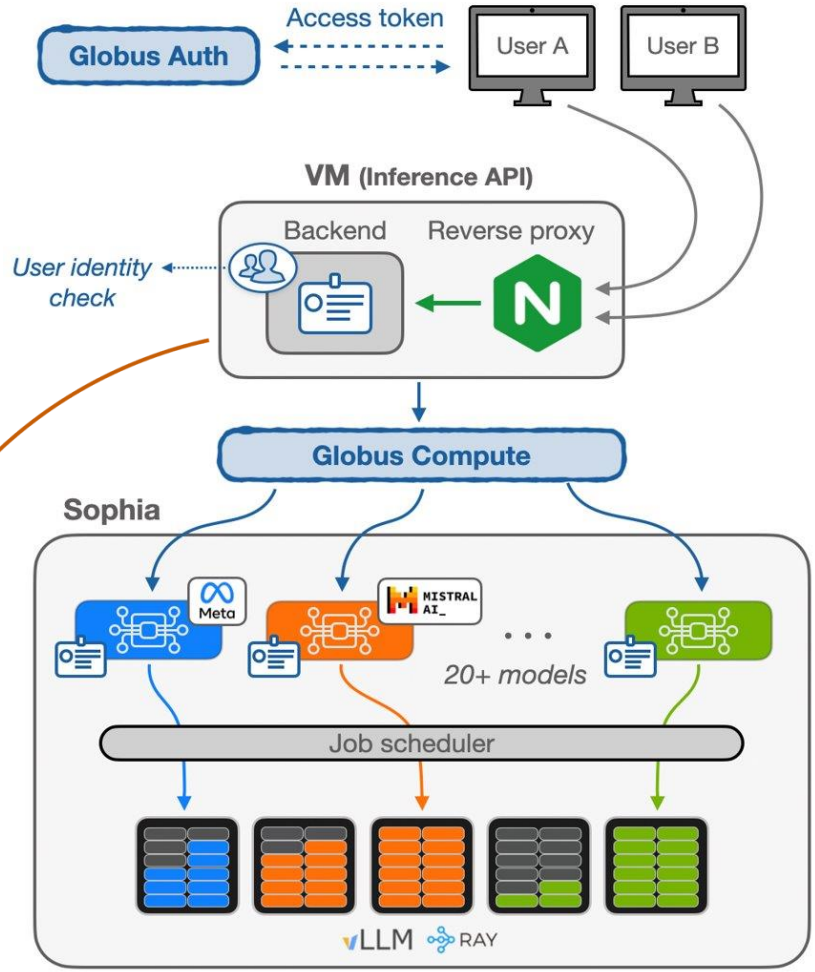
SYSTEM OVERVIEW

- Inference Service leverages ALCF computing resources to serve requests to a growing set of models
- Authentication via Globus Auth and orchestration using Globus Compute
- Combination of “in-memory/active” models and on-demand schedulable models
- Usage metrics are curated to understand and improve the service

...

Minerva (NVIDIA B200)

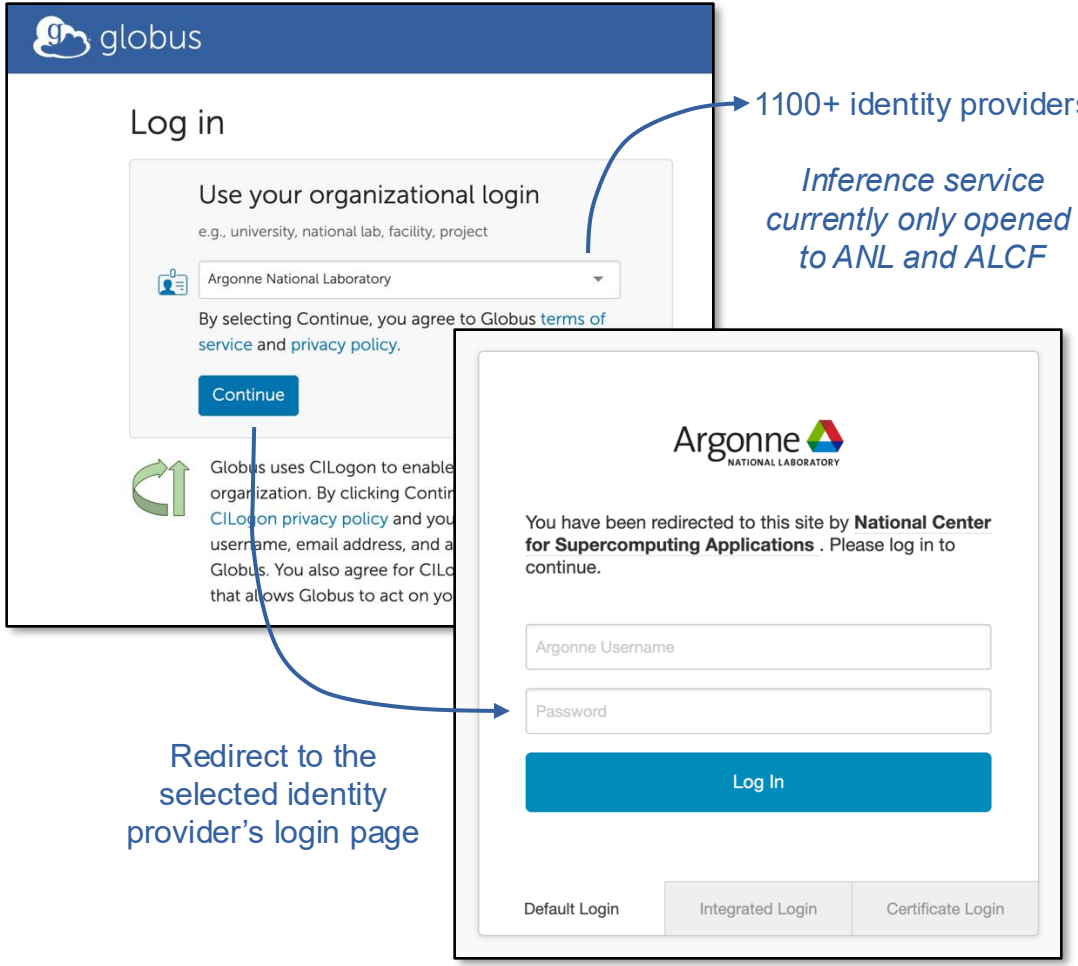
Metis (SambaNova SN40L)



https://dl.acm.org/doi/10.1145/3731599.3767346

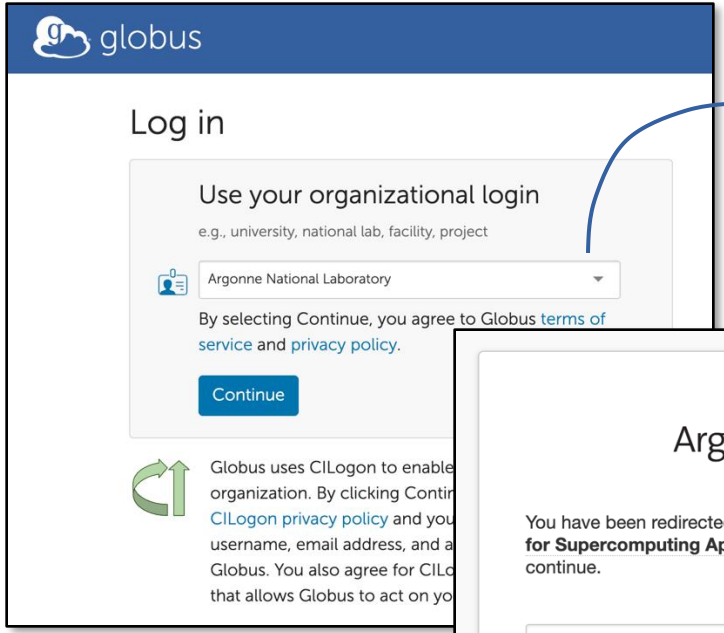
GLOBUS AUTH

- Authentication and authorization platform (OAuth2/OpenID compliant)
- Federated identity provider integrating with different institutions worldwide



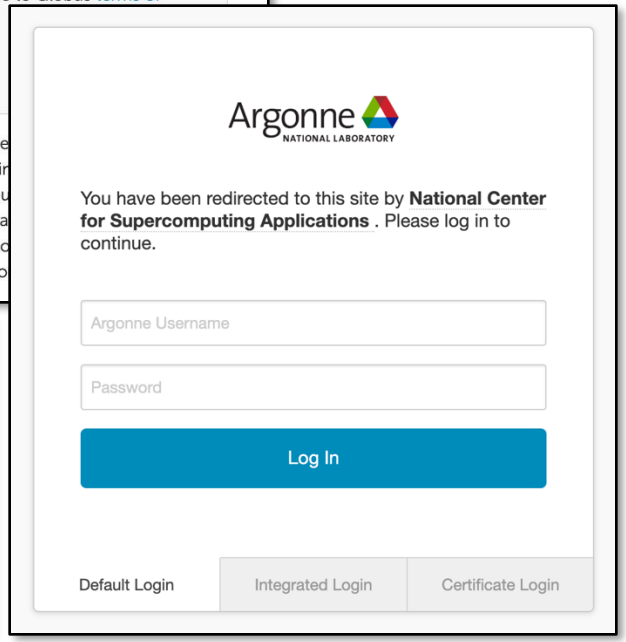
GLOBUS AUTH

- Authentication and authorization platform (OAuth2/OpenID compliant)
- Federated identity provider integrating with different institutions worldwide
- From a user's perspective:
  - Globus Auth generates a token
  - The token is passed to our inference service as an API key



1100+ identity providers

Inference service currently only opened to ANL and ALCF

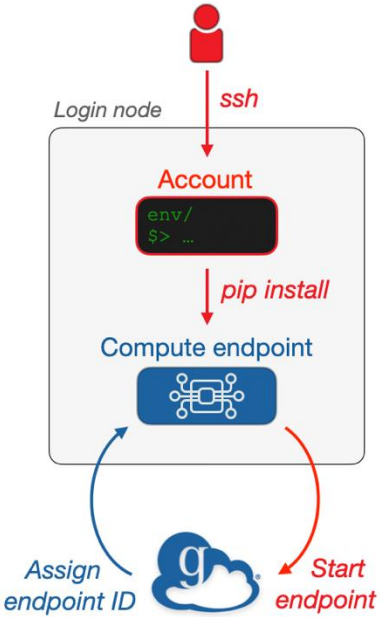



```
client = OpenAI(
    api_key=access_token,
    base_url="https://inference-api.alcf.anl.gov/resource_server/sophia/vllm/v1"
)

response = client.chat.completions.create(
    model="meta-llama/Meta-Llama-3.1-8B-Instruct",
    messages=[{"role": "user", "content": "What are the symptoms of diabetes?"}]
)
```

GLOBUS COMPUTE

Globus Compute can trigger remote analysis on HPC systems from anywhere.
Endpoints deployed on login nodes submit jobs to the scheduler to execute Python functions.

Install endpoint



Register function

```python
# Create Globus Compute client
from globus_compute_sdk import Client
gcc = Client()
```

The function can do whatever you want, including writing data on the filesystem or call more complex codes.

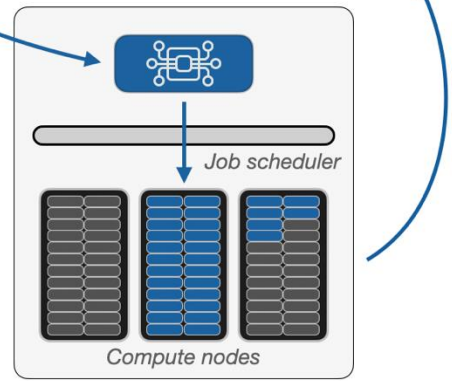
```python
# Define your analysis function
def my_analysis(arguments):

    # Import necessary modules
    import numpy as np
    import scipy

    # Do some analysis using local codes
    # ...

    # Return the computation results
    return ...
```

```python
# Register your function
function_id = gcc.register_function(my_analysis)
```

Run analysis

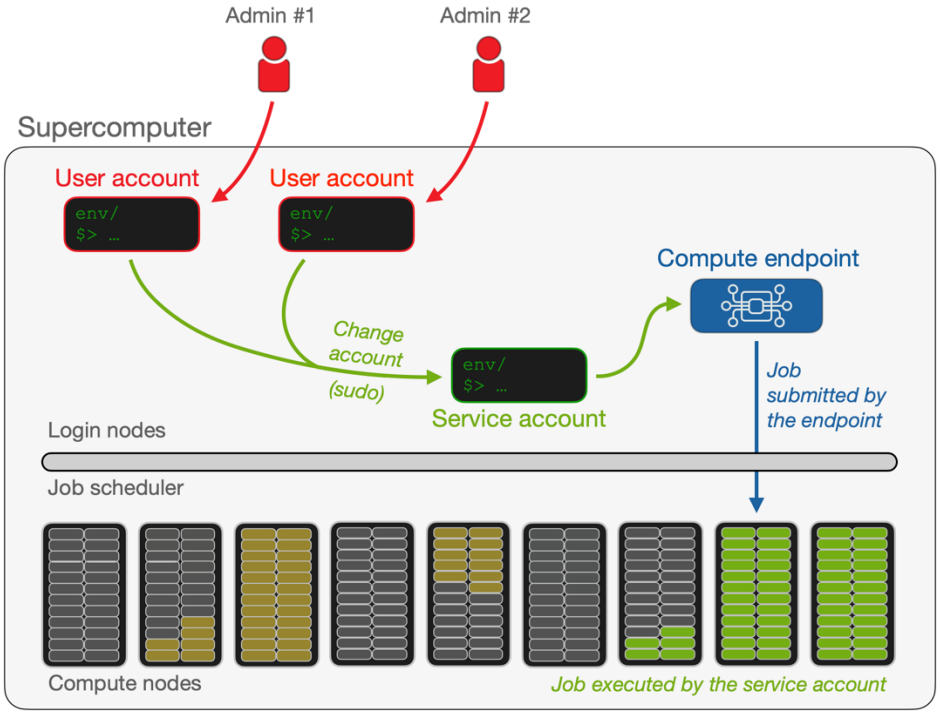
```python
# Submit a function to an endpoint
task_id = gcc.run(
    "my_arguments"
    endpoint_id=endpoint_id,
    function_id=function_id)
```

```python
# Recover results
results = gcc.get_result(task_id)
```



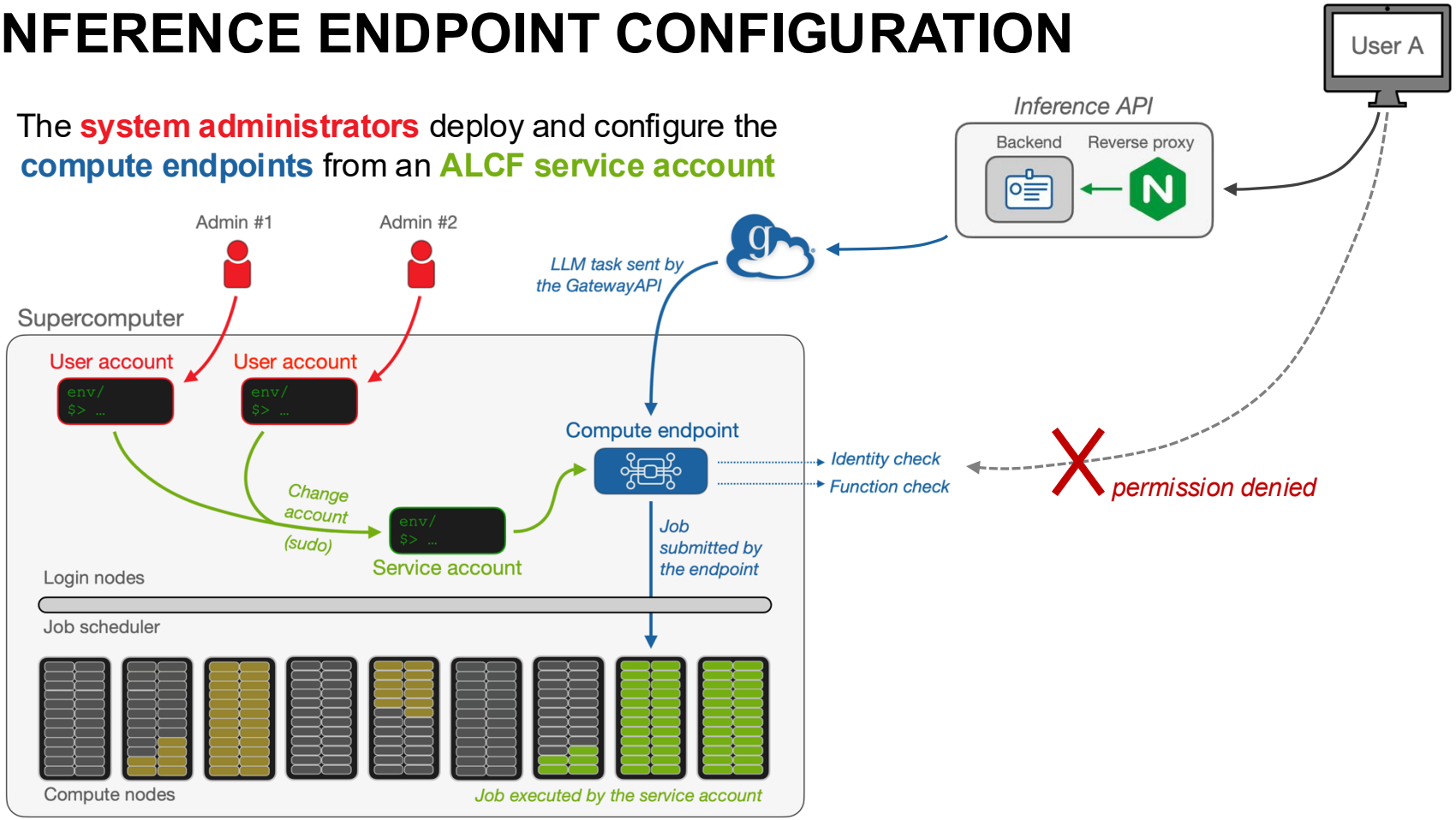
INFERENCE ENDPOINT CONFIGURATION

The **system administrators** deploy and configure the **compute endpoints** from an **ALCF service account**



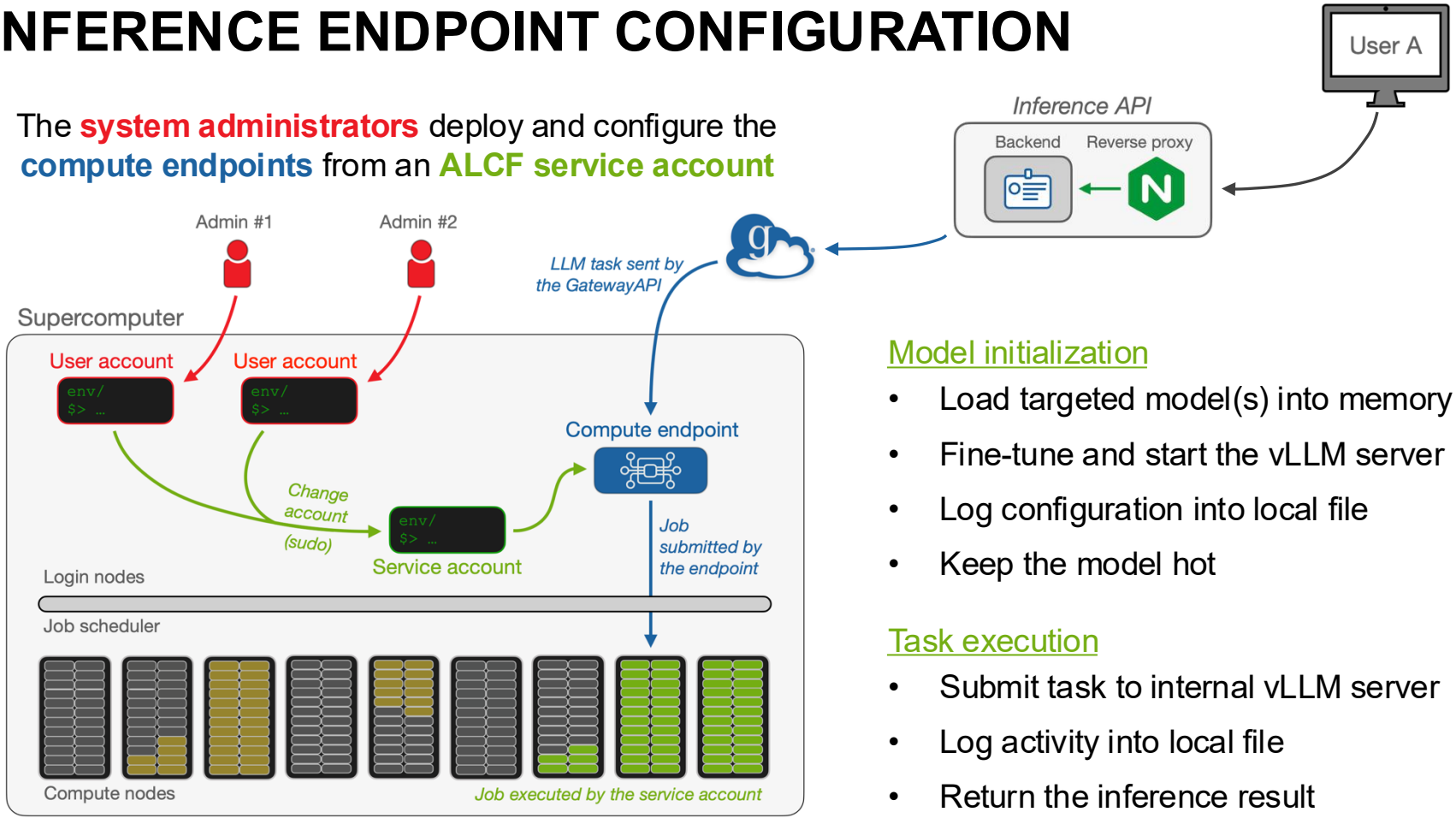
INFERENCE ENDPOINT CONFIGURATION

The **system administrators** deploy and configure the **compute endpoints** from an **ALCF service account**



INFERENCE ENDPOINT CONFIGURATION

The **system administrators** deploy and configure the **compute endpoints** from an **ALCF service account**



Model initialization

- Load targeted model(s) into memory
- Fine-tune and start the vLLM server
- Log configuration into local file
- Keep the model hot

Task execution

- Submit task to internal vLLM server
- Log activity into local file
- Return the inference result

AVAILABLE MODELS ON SOPHIA

B - Batch enabled
T - Tool calling enabled
R - Reasoning enabled
H - Always hot

Family	Models
Meta Llama	Meta-Llama-3.1-70B-Instruct^{BTH}, Meta-Llama-3.1-8B-Instruct^{BTH}, Meta-Llama-3.1-405B-Instruct^{BT}, Llama-3.3-70B-Instruct^{BT}, Llama-4-Scout-17B-16E-Instruct^{BTH}, Llama-4-Maverick-17B-128E-Instruct^{T}
OpenAI	gpt-oss-20b^{BRTH}, gpt-oss-120b^{BRTH}
Mistral	Mistral-Large-Instruct-2407, Mixtral-8x22B-Instruct-v0.1
Aurora GPT	AuroraGPT-IT-v4-0125^{B}, AuroraGPT-Tulu3-SFT-0125^{B}, AuroraGPT-DPO-UFB-0225^{B}, AuroraGPT-7B-OI^{B}
Other Models	Allenai/Llama-3.1-Tulu-3-405B, google/gemma-3-27b-it^{BTH}, mgoin/Nemotron-4-340B-Instruct-hf
Embedding	Salesforce/SFR-Embedding-Mistral, mistralai/Mistral-7B-Instruct-v0.3-embed

https://docs.alcf.anl.gov/services/inference-endpoints/

METIS – SN40L INFERENCE CLUSTER

SambaNova SN40L

- 2x SN40L nodes each with 16x SN40 RDU (32 SN40L Accelerators)
- 1.5TB per RDU – 48TB in aggregate
- Highly optimized for inference

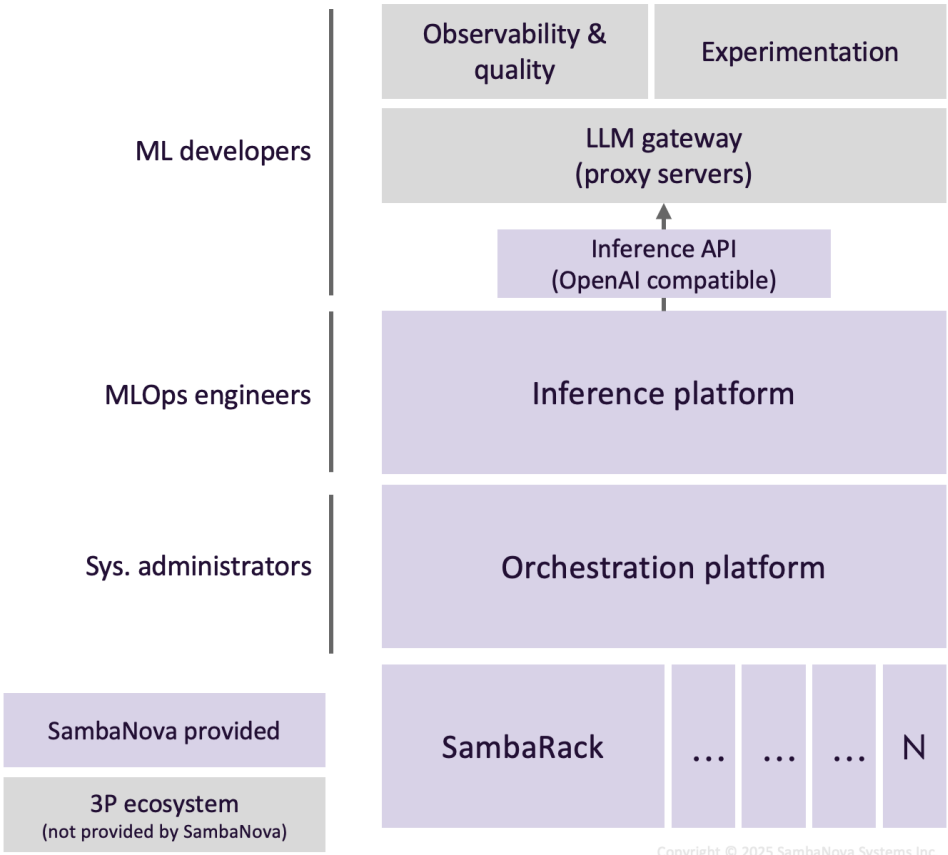
sambanova

SN40L: Accelerating AI

Reconfigurable Dataflow Unit (RDU)
Native multi-tenancy support with fast model switching
Ideal for production inference, multi-tenancy, agentic workflows

- 3-tier Dataflow Memory
- 520 MB On-Chip SRAM Memory → Very fast memory for high speed inference with caching
- 64 GB High Bandwidth Memory → Switch between models in as little as 2 milliseconds
- 1.5 TB High Capacity DDR Memory → Hold large number of models in memory

SambaStack Software Layers & User Personas



OpenAI compatible APIs allows integration with LLMOps platforms (e.g. LiteLLM) for service such as key management, RBAC, rate limits, playgrounds, chargeback, federated clustering…

OpenAI compatible inference API easy adoption, pluggable into upstream tools, frameworks & apps

Features and optimizations for efficient serving of inference requests from a cluster of nodes.
Exposes end-user facing functionality via APIs

K8s based deployment of services to serve inference requests over a cluster of nodes.

SambaRack SN40L-16 nodes new or software upgrade path for installed SambaNova Suite

AVAILABLE MODELS ON METIS

B - Batch enabled
T - Tool calling enabled
R - Reasoning enabled
H - Always hot

Chat Language Models
gpt-oss-120b-131072[RH]
Llama-4-Maverick-17B-128E-Instruct[H]

While Metis currently host less models than Sophia, our Gateway API has a direct connection to its models (i.e. does not rely on Globus Compute). This significantly reduces latency and improves the user experience on our web chat interface.

https://docs.alcf.anl.gov/services/inference-endpoints/

PRESENTATION OVERVIEW

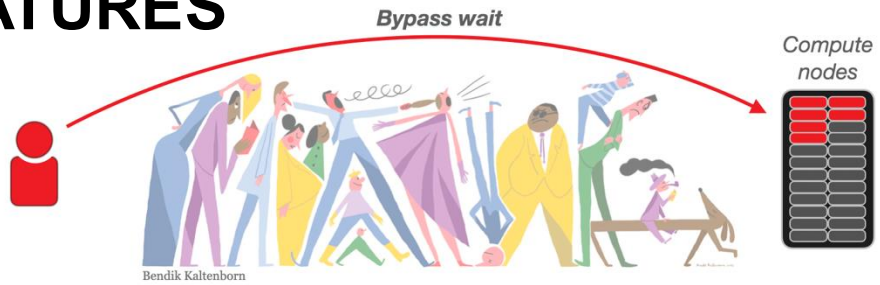
- **ALCF Inference Service Architecture**
  - Overview of the system components
  - Authentication and authorization
  - Orchestration and configuration
  - Available models

- **Capabilities and Features**
  - Latency, scaling, federated endpoints
  - Monitoring, production-ready with containers

- **Usage and Examples**
  - How to use the API (Python, cURL)
  - How to use the web interface

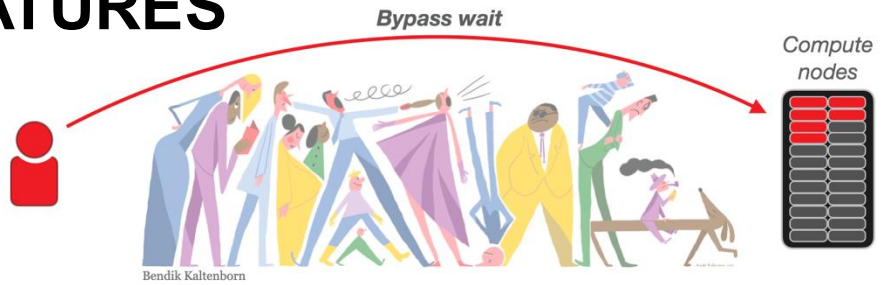
KEY CAPABILITIES AND FEATURES

- **Dedicated Compute Resources**: Selected LLMs persistently served on dedicated nodes. This bypasses HPC queues and “cold starts”.

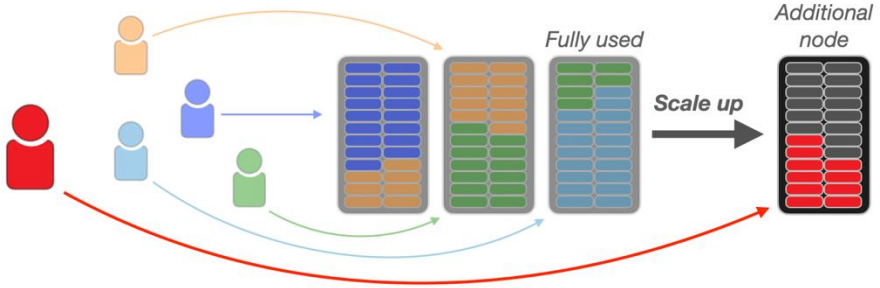


KEY CAPABILITIES AND FEATURES

- **Dedicated Compute Resources**: Selected LLMs persistently served on dedicated nodes. This bypasses HPC queues and “cold starts”.

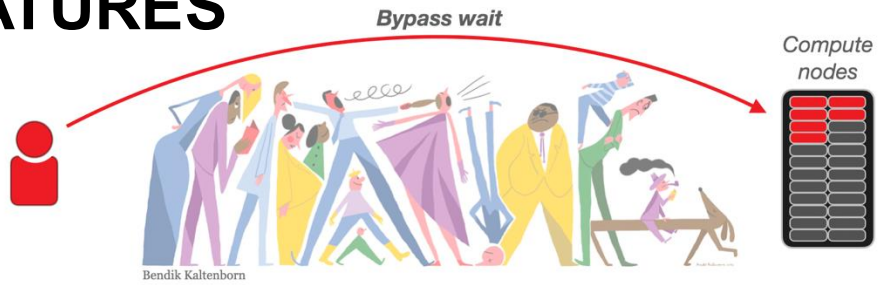


- **Auto-Scaling and Hot Nodes**: New nodes can dynamically be acquired to accommodate higher traffic. Cold models can be dynamically be loaded and kept hot for 24 hours.

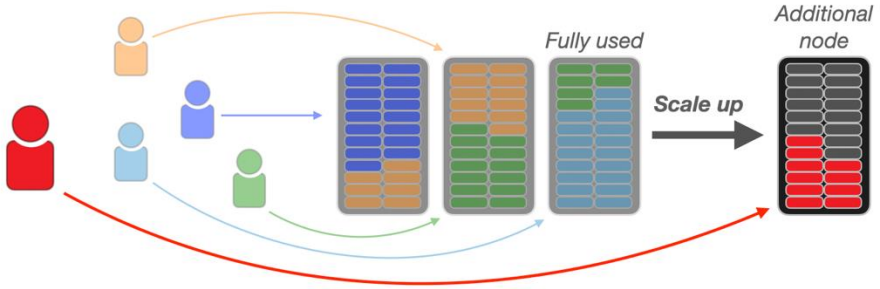


KEY CAPABILITIES AND FEATURES

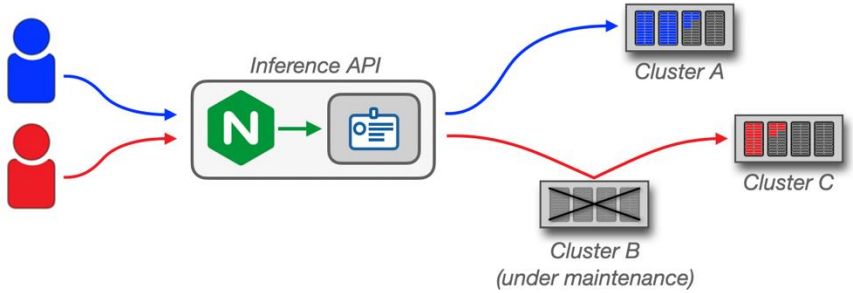
- **Dedicated Compute Resources**: Selected LLMs persistently served on dedicated nodes. This bypasses HPC queues and “cold starts”.



- **Auto-Scaling and Hot Nodes**: New nodes can dynamically be acquired to accommodate higher traffic. Cold models can be dynamically be loaded and kept hot for 24 hours.

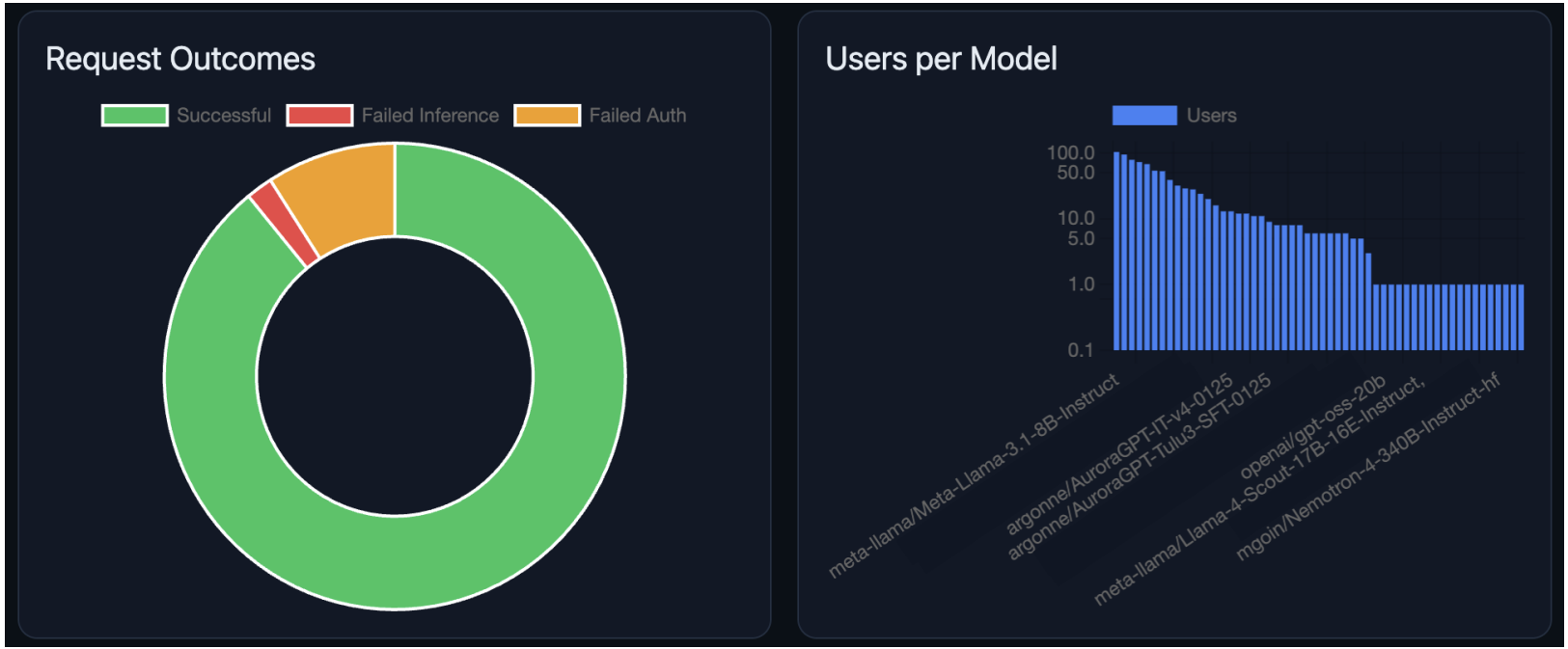


- **Multi-Backend Integration**: Our API can seamlessly route requests to diverse remote hardware, including SambaNova SN40 and Sophia inference clusters.



KEY CAPABILITIES AND FEATURES

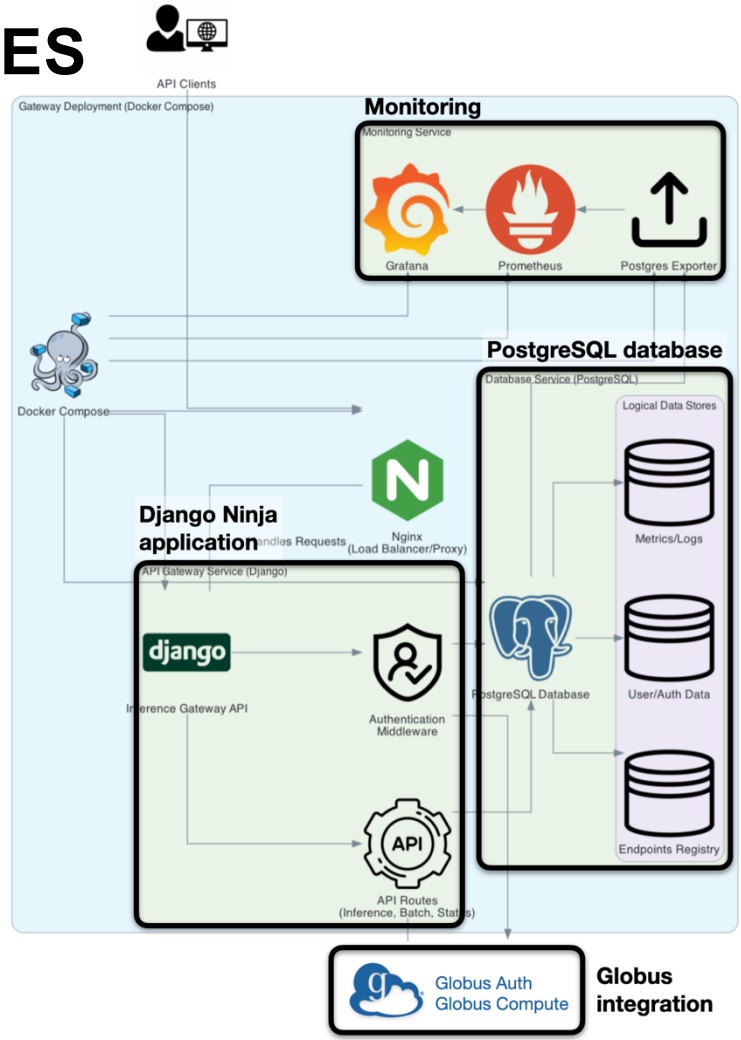
- **Dashboard Monitoring**: A dashboard is available to system administrators and provides various metrics such as recent activities, number of requests and users, token throughput, and latency.
- **Current Status**: ~220 users, ~10 million requests, over 11 billion tokens generated



KEY CAPABILITIES AND FEATURES

- **Production Ready:** Container deployment
  - Django application
  - Gunicorn/Uvicorn + Nginx
  - PostgreSQL
  - Redis cache management
  - Comprehensive logging
  - Globus integration
- **Benchmarking**: Built-in load testing tools
  - Request throughput
  - Token throughput
  - Latency

See our paper: https://dl.acm.org/doi/10.1145/3731599.3767346



BENCHMARK EXAMPLES

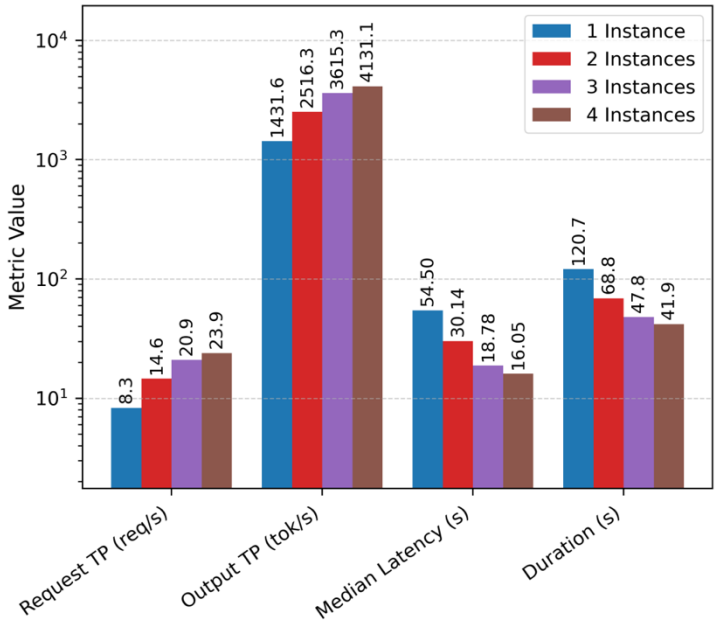


Figure 4: Auto-scaling performance: Single vs. two, three, and four instances for Llama 3.3 70B on Sophia's A100 GPUs under maximum load.

https://dl.acm.org/doi/10.1145/3731599.3767346

BENCHMARK EXAMPLES

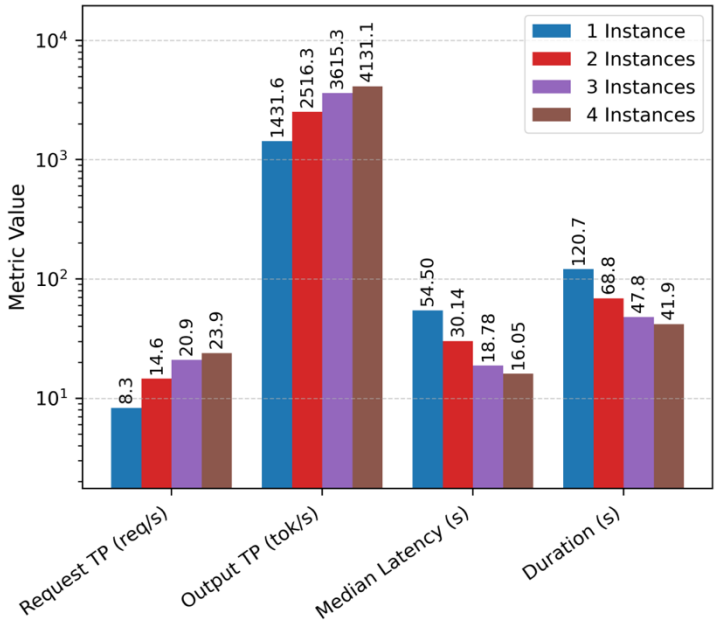


Figure 4: Auto-scaling performance: Single vs. two, three, and four instances for Llama 3.3 70B on Sophia's A100 GPUs under maximum load.

Table 1: WebUI benchmark results per model. Conc = concurrency; TP = throughput tokens; Req = requests.

Model	Conc.	60 s	
		TP/s	Req./s
Llama-3.1-8B	50	690.68	4.97
	100	738.33	5.25
	300	1103.70	7.90
	500	1672.15	12.08
	700	2119.50	14.68
Gemma-27B	50	297.97	2.70
	100	906.62	5.42
	300	1469.53	8.67
	500	1849.67	10.95
	700	2651.40	15.57
Llama-3.3-70B	50	217.38	1.63
	100	785.83	5.88
	300	1061.93	7.92
	500	1646.53	12.30
	700	2134.10	15.67

https://dl.acm.org/doi/10.1145/3731599.3767346

PRESENTATION OVERVIEW

- **ALCF Inference Service Architecture**
  - Overview of the system components
  - Authentication and authorization
  - Orchestration and configuration
  - Available models

- **Capabilities and Features**
  - Latency, scaling, federated endpoints
  - Monitoring, production-ready with containers

- **Usage and Examples**
  - How to use the API (Python, cURL)
  - How to use the web interface

FUTURE WORK

- Tool calling integration
- Working towards centralized logs and monitoring
- Host Gateway API on Kubernetes
- Implement model failure resiliency
- Integrate with local scheduler API to query job status and delete jobs
- Improve federated routing to offload requests to multiple clusters
- More dedicated resources (including Tara and Minerva from NVIDIA)
- Implement Globus-Flow base batch system for users outside of ALCF

LARGE COLLABORATION

Ryan Chard, Nick Saint, Tom Uram, Thang Pham, Murat Keceli, Rajeev Thakur, Ken Raffenetti, Le Chen, Yanfei Guo, Krishna Chetty, Murali Emani, Khalid Hossain, Nathan Nichols, Rachana Ananthakrishnan, Anthony Avarca, Bill Allcock, Tommie Jackson, Ian Foster, Mike Papka, Rick Stevens, ALCF and CELS Operations, Globus Labs, and many more.

Joint collaboration with Globus

CONCLUSION

ALCF Inference Endpoints democratize LLM access for scientific research by:

- Providing seamless access to cutting-edge models
- Supporting diverse use cases: chat, vision, embeddings, agents, and batch processing
- Integrating with multiple HPC backends for optimal performance
- Auto scaling based on request workload

Web Interface: https://inference.alcf.anl.gov/

Documentation (usage examples): https://docs.alcf.anl.gov/services/inference-endpoints/

Contact: support@alcf.anl.gov

See our paper: https://dl.acm.org/doi/10.1145/3731599.3767346