INL/CON-23-72098-Revision-0

Integrating Artificial Intelligence into Science Gateways

July 2023

Brandon Samuel Biggs Jr

Changing the World's Energy Future

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

INL/CON-23-72098-Revision-0

Integrating Artificial Intelligence into Science Gateways

Brandon Samuel Biggs Jr

July 2023

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Integrating Artificial Intelligence into Science Gateways

BRANDON BIGGS, Idaho National Laboratory, USA

Science gateways are altering the manner in which people interact with high performance computing (HPC) by providing a web browser based interface to advanced computing platforms. In particular, science gateways lower the barrier to using HPC by simplifying the process of submitting workloads to such systems and by offloading the efforts required to use HPC to the maintainers of the system. While science gateways decrease the time-to-science that comes with using such advanced systems, progress can still be made in improving the user's experience. In this paper we explore two strategies for integrating artificial intelligence tools commonly found in non-HPC service workflows: voice activated assistants and chatbots.

Since August 2021, the HPC group at Idaho National Laboratory answers an average of 581 support tickets per month of which a large percentage could be addressed via these two strategies.

This work defines the key capabilities that an HPC voice activated assistant and chatbot would need to address for a userbase consisting of largely non-expert users as well as a design for integration into the Open OnDemand science gateway.

Additional Key Words and Phrases: machine learning, artificial intelligence, science gateways

ACM Reference Format:
Brandon Biggs. 2023. Integrating Artificial Intelligence into Science Gateways. In . ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Science gateways are web based portals that provide access to scientific tools and resources. These tools and resources can include virtual desktops, integrated development environments, applications that require graphical user interfaces, and other programs that have been simplified for users by the administrators of the science gateway. Science gateways are typically utilized by scientists and researchers and are intended to reduce the technical barrier, or friction, often required to access such resources. While the friction to high performance computing (HPC) workloads has been reduced by science gateways [11] there is still work that can be done. In this work, we present a novel strategy at further reducing this friction by using user-supporting AI tools to provide additional avenues for users to receive assistance. In order for us to test this strategy, this work presents a methodology for integrating two AI systems. These systems consist of an speech driven HPC job submission AI and a large language model (LLM) powered conversation chatbot.

Since August 2021, INL HPC has had an average of 581 tickets per month. 295 or 50.7% of those are account related support tickets. This includes account and group management, password reset questions and assistance, and other issues related to logging in. Additionally, in that same time period, over 400 tickets have been submitted with questions pertaining to job submissions. Integrating AI into a science gateway aims to be beneficial as it allows for a personalized and interactive user experience that has not previously been possible. The LLM powered conversation chatbot can provide customized educational resources specific to the organization while the speech

PEARC '23, July 23–27, 2022, Portland, OR

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

Support Ticket	Desired Chatbot Response
I can't connect to the cluster machine by using 'ssh cluster1' or 'ssh cluster2'. Actually, this is the first time to try it. Could you let me know whom I have to contact to for this issue ?	You can contact our support staff by sending an email to...
I can't log in. I may need to reset my password. How can I do that?	You can reset your password by visiting our website...
Can you install Python 3.10 and Tensorflow 2.9 on the system?	This is outside of my current scope. A member of our support staff will contact you soon.
I am trying to build a code after I have pulled a remote branch. I am getting the error: ... Building wheel for package (setup.py) ... done OSError: Errno 16Device or resource busy	This is outside of my current scope. A member of our support staff will contact you soon.

Table 1. The desired outcome of the chatbot would be to assist in directing users to official documentation, web pages, or just as importantly, know when not to try to solve the issue.

driven job submission AI can guide the user through a job submission or answer other basic questions via natural language. It supports various language not previously seen by conventional rule based models. In addition, AI systems can free up time from employees to focus on other tasks. Table 1 shows several examples of where an AI trained on HPC could be beneficial. While some support requests may be easier to answer, many requests would still require human intervention. Determining when human intervention is needed requires additional research efforts and is outside the scope of this work.

This work is structured as follows: in Section 2, a brief background of AI with science gateways is provided. In Section 3 the strategy for integrating two AI systems is outlined. In Section 4 we provide suggestions for further AI implementations in science gateways and conclude.

2 RELATED WORK

There exists two modalities of science gateways: science gateways for accessing data and science gateways for accessing compute resources. Many science gateways contain overlap and may provide AI assisted tools to analyze data or the frameworks for researchers to develop their own AI models. The methods presented in this paper are based on AI integration with science gateways that interact directly with computational resources. More specifically this work uses Open OnDemand [6–8] to integrate these strategies. The integration methods outlined here are not specific to Open OnDemand and could be applied to either modality.

As the models used to create natural language AI have advanced, integrating them into applications is becoming more common. LLMs are evolving rapidly and are being integrated with many applications such as search engines while many language and image AI services have become available on consumer devices. Additionally, previous work has integrated AI into HPC profiling tools [9]. Because of the varying implementations of AI, integration of these models into science gateways as a means to assist the user in performing science presents an exciting opportunity.

3 AI INTEGRATION STRATEGIES

Two AI systems were integrated into a science gateway. The first system is a job submission conversational virtual assistant (Figure 1). This HPC personal assistant assists the user in learning

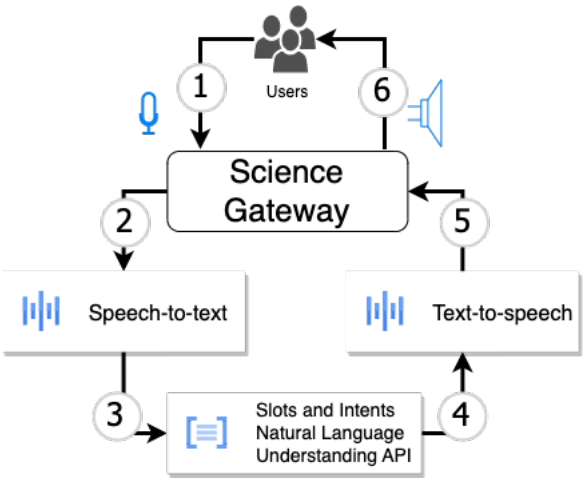

Fig. 1. The architecture for creating a speech-to-text and text-to-speech conversational AI that can interact with a science gateway.

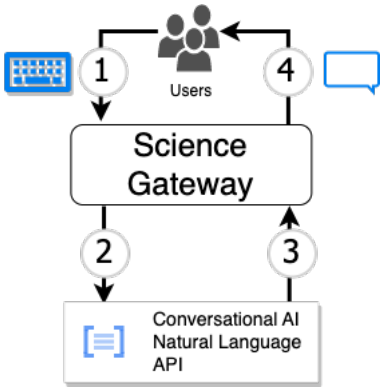

Fig. 2. The architecture for creating a speech-to-text and text-to-speech conversational AI that can interact with a science gateway.

more about the HPC systems and helps submit jobs by asking each piece of information that is needed to submit a resource request. The second system is an LLM based conversational AI chatbot that has been trained on scientific material (Figure 2). This chatbot provides an opportunity for users to ask the science gateway questions that may be relevant to the tasks they are trying to solve. Additionally, the infrastructure for the chatbot allows for a human-in-the-loop intervention if the user needs assistance that the LLM generated responses can not provide. The following two subsections provide the implementation details for the AI integration strategies.

3.1 Using A Conversational AI Voice Assistant to Request HPC Resources

Figure 1 demonstrates the architecture of an HPC voice assistant resource request system. This system was created to act similarly to other voice assistants. However, several differences exist between this implementation of a voice assistant and popular every day voice assistants such as Amazon's Alexa [2] or Google's voice assistant [1]. First, this voice assistant would be substantially more narrow in scope. Rather than trying to be a voice assistant that can help in many facets of life, this voice assistant targets HPC users. Second, for an initial implementation of the system, each model used is openly accessible and can be self hosted and self trained. This alleviates concerns of intellectual property being shared outside of the organization as well as provides opportunity for additional training on domain data that improves the model for an organization's specific needs. However if desired, cloud models could also be used. Third, custom voice assistants can be tightly integrated with HPC systems in addition to science gateways, including schedulers, monitoring applications, or scientific codes with additional programming efforts.

Several AI models are needed to make this a fully functioning voice assistant system. The first model needed is a speech-to-text model to transcribe a user's voice. The transcription API used in this implementation was a local instance of NVIDIA NeMo's xlarge [10] model from HuggingFace [5] to do speech-to-text (STT) transcription. This model did not require any additional training and could be used "out-of-the-box". Additional training could be performed to increase the model's performance on HPC specific jargon. Audio files that are sent along with the predicted text can

be stored for training allowing future models to be finely tuned for HPC words and phrases thus improving the model in the future.

The second model needed was a natural language understanding (NLU) slots and intents model. Slots are the entities that need to be defined to perform proper actions. An example of this would be cluster name or job duration. The NLU model uses intents to understand what the user was requesting as each user may use different language or technical jargon when requesting resources. For the implementation of an NLU model, the open source Rasa framework [3] was utilized. This framework allows for a model to be created that understands the specific values that need to be set based on the user's intention. This NLU model also enables a back and forth dialog with the user to gather all of the required information. The Rasa model accepts the text transcription of the user's request, along with a conversation ID to track the conversation in the future. The model parses the transcribed text, adds it to a conversation tracker, and then returns a text response of how it interpreted the text transcription based on the conversation at that point. This model did require training on an HPC NLU data set. To do that training, a basic HPC NLU dataset was created as part of our initial implementation of a system. Once the data set was in place, the default Rasa training pipeline was used. This HPC NLU data set along with the Rasa settings will be open sourced as part of this work.

The third and final model needed for the voice assistant is a text-to-speech (TTS) system. This system uses the CoquiTTS model[4] to generate audio from a given piece of text. This model is not required, as the transcription from the NLU model could be displayed to the user as text, however the TTS system adds functionality in situations where displayed text may not be useful or possible.

A further breakdown on each step of the model implementation is described in the following steps:

(1) A user authenticates with the science gateway and navigates to the specific application that has been integrated with AI. The user will then be prompted to grant the web page permissions to access the computer's microphone.
(2) The user clicks and holds a button that is displayed on the web page. The web page records audio from the user's computer as they verbally request HPC resources while the button is held.
(3) The recorded audio resource request is sent to a server API via AJAX, an asynchronous client side web technology.
(4) Once the audio file is transcribed, the text is sent through the NLU model to be parsed into the intention of the user and any slots that were defined in the request are set.
(5) After the NLU model interprets the request, a text response is returned and converted to an audio file by the TTS API.
(6) The newly generated audio file is sent from the TTS API back to the science gateway.
(7) Once the TTS audio file is received by the science gateway, it is then played to the user. This cycle repeats until the user has provided all necessary information to the NLU model and the user's resource request can be submitted via the science gateway as if they had manually entered all of the request information.

3.2 Scientific Conversation Chatbot

Figure 2 demonstrates the architecture of the LLM driven scientific chatbot. This chatbot was intended to reduce friction by answering questions, provide insight into scientific and technical issues, while assisting HPC support staff.

Unlike the voice assistant implementation, the conversation chatbot only used a single LLM. In an attempt to narrow the scope of queries and improve the responses of the AI for scientific

questions and conversation, the 6.7 billion parameter version of the scientific LLM from Meta AI, Galactica [14] was used. This LLM was trained using scientific knowledge such as textbooks and papers by the team at Meta AI and then the weights were released for anyone to download. This LLM can be replaced by any other domain specific large language model such as BioMedLM (formerly PubMed GPT) [15], a more general models such as the conversation models provided by OpenAI, or even multi-language models such as BLOOM [12]. Very recently released AI efforts with instruction-following based LLMs such as Stanford's Alpaca [13] would provide even better support especially once trained on HPC specific instructions. Additionally, different APIs could be created providing access to multiple models or workflows.

Further implementation details are described in the following steps:

(1) A user authenticates with the science gateway and navigates to the specific application that has been integrated with AI.
(2) A user interacts with the chat box by clicking a chat button on the bottom right corner of the science gateway page. Once this button is clicked, a chat window appears and provides a prompt for the user to type their message. Once a message has been typed out, the user can send the message.
(3) The user's message is sent to a conversational API via AJAX.
(4) The user's message is parsed and responded to via the LLM and sent back to the science gateway.
(5) Once the newly generated message is received by the science gateway, it is displayed to the user in the same chatbox. The user can then choose to continue the cycle by sending another message or closing the chat window and ending the conversation.

4 CONCLUSION

Integrating user-supporting AI tools into science gateways presents the opportunity to transform the way HPC users access tools, resources, and data. With HPC technology rapidly changing, it has become more difficult for new users to keep up with the latest frameworks, tools, and research. For our organization, this is supported by the large number of tickets we receive related to questions about these emerging technologies. By incorporating language model AI technologies, science gateways can continue to assist users by reducing the barrier to HPC while shifting that time that was previously spent on learning how to use the systems to more productive efforts.

In this paper, we have presented two different methods in which artificial intelligence can be integrated into science gateways via text or speech. In future work, we hope to expand the information that the NLU and chatbot models have access to as well as increasing the extent of the slots and intents data set. This would include a broader range of topics and workflows to improve advanced questions. Additionally, we want to explore implementing a human-in-the-loop feedback mechanism to improve each of the the models utilized and monitor the state of LLMs to utilize their strengths while potentially retraining the models with more specific domain information. This would include the exploration of when an LLM should respond to the user and when it should not.

The integration of user-supporting language model AI technologies into science gateways has the potential to better users' experiences, and further decrease the friction involved with HPC platforms. By incorporating more accessible technologies via natural language and speech, the provided science gateway interfaces can become more intuitive and user friendly, allowing more people to use the scientific resources provided by institutions. The integration of these growing fields and technologies into science gateways represents an opportunity to improve the usability while decreasing the friction and increasing the impact of these platforms.

ACKNOWLEDGMENTS

This research made use of the resources of the High Performance Computing Center at Idaho National Laboratory, which is supported by the Office of Nuclear Energy of the U.S. Department of Energy and the Nuclear Science User Facilities under Contract No. DE-AC07-05ID14517.

The author would like to especially thank Matthew Anderson for helpful conversations in developing this work.

REFERENCES

[1] Alphabet. [n. d.]. Google Assistant. https://assistant.google.com

[2] Amazon. [n. d.]. Amazon Echo Alexa Devices. https://www.amazon.com/Amazon-Echo-And-Alexa-Devices/b?ie=UTF8&node=9818047011

[3] Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open Source Language Understanding and Dialogue Management. https://doi.org/10.48550/ARXIV.1712.05181

[4] Gölge Eren and The Coqui TTS Team. 2021. *Coqui TTS.* https://doi.org/10.5281/zenodo.7584954 If you want to cite , feel free to use this (but only if you loved it).

[5] Thomas Wolf et al. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.* Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

[6] Dave Hudak, Doug Johnson, Alan Chalker, Jeremy Nicklas, Eric Franz, Trey Dockendorf, and Brian L. McMichael. 2018. Open OnDemand: A web-based client portal for HPC centers. *Journal of Open Source Software* 3, 25 (2018), 622. https://doi.org/10.21105/joss.00622

[7] David E. Hudak, Thomas Bitterman, Patricia Carey, Douglas Johnson, Eric Franz, Shaun Brady, and Piyush Diwan. 2013. OSC OnDemand: A Web Platform Integrating Access to HPC Systems, Web and VNC Applications. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery* (San Diego, California, USA) *(XSEDE '13).* Association for Computing Machinery, New York, NY, USA, Article 49, 6 pages. https://doi.org/10.1145/2484762.2484780

[8] David E. Hudak, Douglas Johnson, Jeremy Nicklas, Eric Franz, Brian McMichael, and Basil Gohar. 2016. Open OnDemand: Transforming Computational Science Through Omnidisciplinary Software Cyberinfrastructure. In *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale* (Miami, USA) *(XSEDE16).* Association for Computing Machinery, New York, NY, USA, Article 43, 7 pages. https://doi.org/10.1145/2949550.2949644

[9] Pouya Kousha, Arpan Jain, Ayyappa Kolli, Saisree Miriyala, Prasanna Sainath, Hari Subramoni, Aamir Shafi, and Dhableswar K Panda. 2022. "Hey CAI"-C onversational AI Enabled User I nterface for HPC Tools. In *High Performance Computing: 37th International Conference, ISC High Performance 2022, Hamburg, Germany, May 29–June 2, 2022, Proceedings.* Springer, 87–108.

[10] Oleksii Kuchaiev, Jason Li, Huyen Nguyen, Oleksii Hrinchuk, Ryan Leary, Boris Ginsburg, Samuel Kriman, Stanislav Beliaev, Vitaly Lavrukhin, Jack Cook, et al. 2019. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577* (2019).

[11] Bradlee Rothwell, Matthew Sgambati, Garrick Evans, Brandon Biggs, and Matthew Anderson. 2022. Quantifying the Impact of Advanced Web Platforms on High Performance Computing Usage. In *Practice and Experience in Advanced Research Computing* (Boston, MA, USA) *(PEARC '22).* Association for Computing Machinery, New York, NY, USA, Article 20, 8 pages. https://doi.org/10.1145/3491418.3530758

[12] Teven Le Scao et al. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. https://doi.org/10.48550/ARXIV.2211.05100

[13] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

[14] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085* (2022).

[15] Laura et al. Weidinger. 2021. Ethical and social risks of harm from Language Models. https://doi.org/10.48550/ARXIV.2112.04359