HiBD High-Performance Big Data

MVAPICH MPI, PGAS and Hybrid MPI+PGAS Library

HiDL High-Performance Deep Learning

“Hey CAI” - Conversational AI Enabled User Interface for HPC Tools

Pouya Kousha, Arpan Jain, Ayyappa Kolli, Prasanna Sainath, Saisree Miriyala

Hari Subramoni, Aamir Shafi, and Dhableswar K. Panda

The Network Based Computing Laboratory

The Ohio State University

http://mvapich.cse.ohio-state.edu/

Follow us on

https://twitter.com/mvapich

Presentation Outline

- **Introduction and Motivation**
- Problem Statement and Proposed Designs
- Performance Evaluation
- Demo
- Concluding Remarks

HPC Profiling Tools and Broad Challenges

- Analyzing performance bottlenecks for HPC is a complicated but critical task
- New and expert HPC users often have a hard time in understanding the performance of their parallel workloads HPC administrators
  - Learning the tool interfaces
  - Learning tool features and the terminologies
- There is a steep learning curves for utilizing HPC profiling tools!

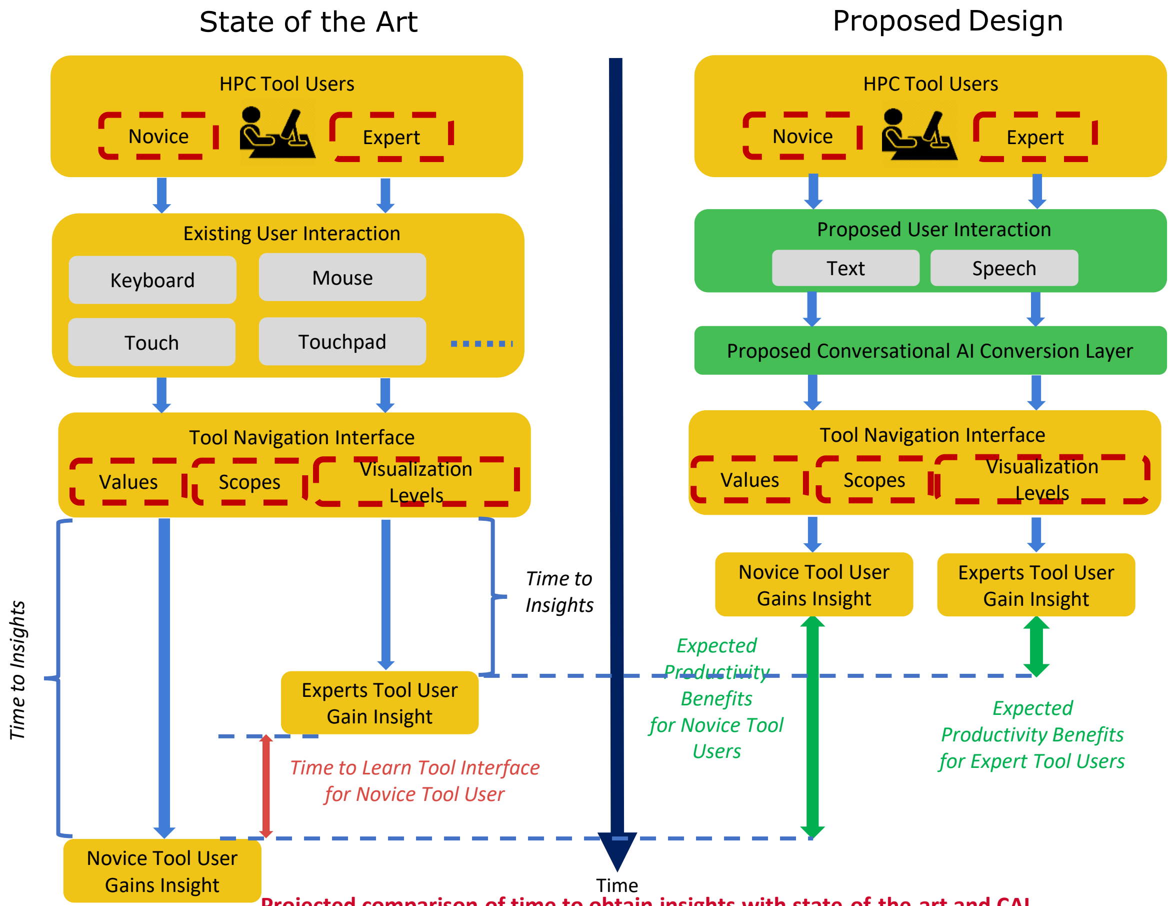
The challenge is to provide intuitive and simple—yet efficient—interfaces to HPC software and hardware resources to eliminate the steep learning curve of HPC tools

Motivation

Courtesy: Paramount Domestic Television

Motivation of Using Conversational User Interface (CAI)

- CAI alleviates the steep learning curve and increases the productivity of expert and novice users
- By using text or speech interface the response time for both users will be lower



Projected comparison of time to obtain insights with state-of-the-art and CAI

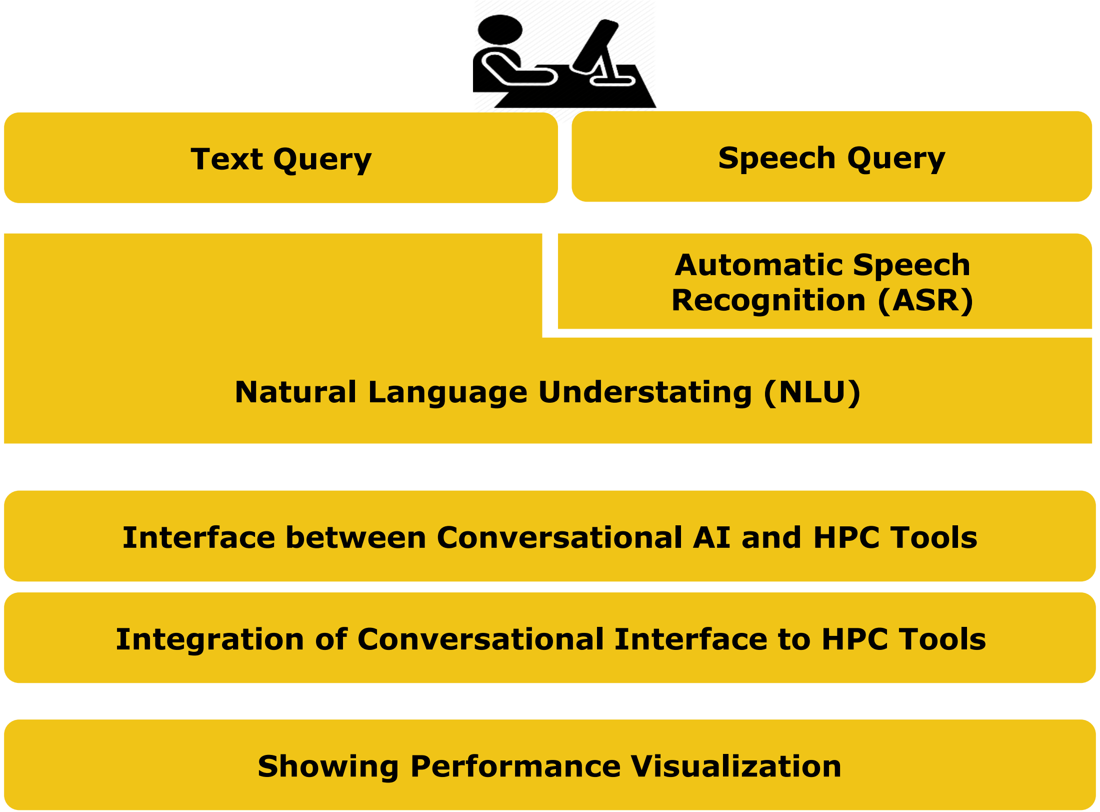
Outline

- Introduction and Motivation
- **Problem Statement and Proposed Designs**
- Performance Evaluation
- Demo
- Concluding Remarks

Problem Statement

- Are there any databases for HPC phrases and terminology?
- Can we use the existing Automatic Speech Recognition(ASR) models for converting speech to text?
- Are there any existing Natural Language Understanding(NLU) model for HPC tools?
- What are the challenges in enabling Conversational AI Interface(CAI) for HPC tools?
- How to integrate CAI to existing HPC tools?

Proposed Framework



Generating HPC Dataset for Speech and Text

We create an HPC dataset for text and speech containing HPC terminology:

i. Generated basic queries and labeled their slots and intents

ii. Developed synonyms for HPC terminologies

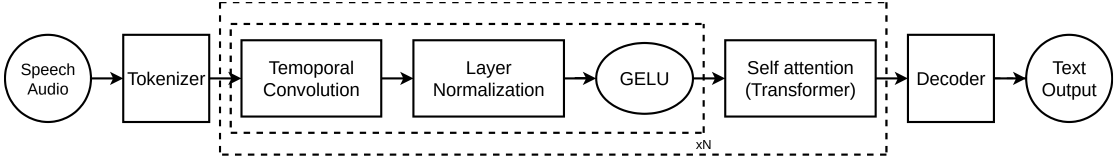
iii. Used the synonyms to generate combinations of queries and labeled their slots and intents in human-supervised manner

- The dataset contains four profiling intents:
  - Node usage (CPU usage, memory usage, etc.)
  - Network usage (Traffic usage, etc.)
  - Process usage (Bytes sent/received for each process, etc.)
  - Statistics (Jobs, active nodes, number of nodes, etc.)

Training Speech and Text Processing Models for HPC

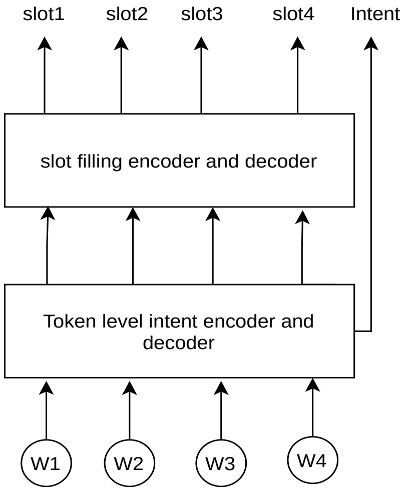
Trained Wav2Vec and Speech2Text models for speech with hyper-parameters tuning on a combination of our in-house HPC-ASR dataset and TIMIT

Wav2Vec architecture



Natural Language Understanding (NLU)

- Trained two transformer-based models
  - *StackPropgation*
  - *JointBert*
- Used HPC-NLU Dataset to identify the task or request in the sentence



JointBert architecture

Integration of Conversational AI to HPC Tools

We selected OSU INAM as the HPC profiling tool to integrate CAI

- Having a URL as input that gives the arguments and visualization selection for our tool to process and handle it

The modifications to our HPC tool is as follows:

1. The tool needs to record the voice and send it to CAI Interface
2. INAM receives and redirect the response URLs to corresponding web pages (No changes)
3. Web UI generator adjusts the values and scopes based on extracted parameters from URL
4. Data Access Object generates the query to retrieve the data from the database (No changes)
5. Pass the data to visualization to plots the visualizations (No changes)

- The changes are minimal if the target HPC tool supports web-UI interface

Outline

- Introduction and Motivation
- Problem Statement and Proposed Designs
- **Performance Evaluation**
- Demo
- Concluding Remarks

Evaluation Platform

Deep Learning Framework: PyTorch is used to define and train DNNs for ASR and NLU

Deep Neural Networks: Speech2Text, Wav2Vec, JointBert, and StackPropagation

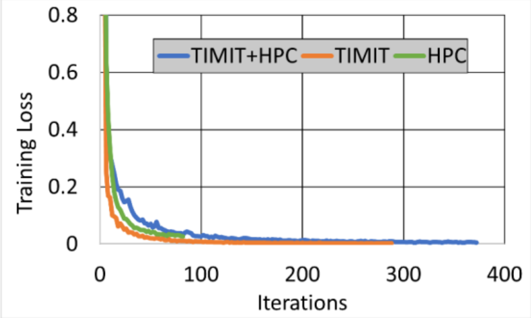
Datasets: LibriSpeech and TIMIT, HPC-ASR Dataset, HPC-NLU Dataset

Architecture	Type	Cores	Speed (GHz)	Label
Broadwell (Server)	CPU	28	2.4	BDW
SkyLake (Server)	CPU	28	2.6	SKX
K80 (Server)	GPU	4992 (Dual socket)	-	K80
V100 (Server)	GPU	CUDA: 5120 Tensor: 640	-	V100
Intel Core i5 8th gen (Surface Pro)	CPU	4	1.8	Client-1
Intel Core i7 11th gen (HP Pavillion)	CPU	4	2.8	Client-2
Intel Core i5 (MacBook Pro)	CPU	4	1.4	Client-3

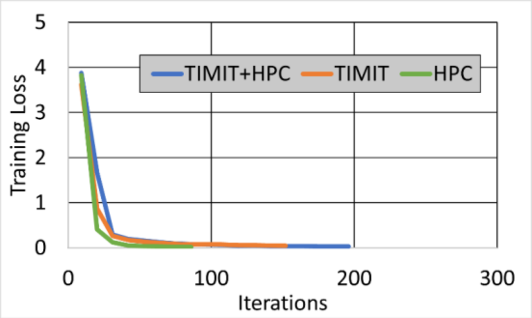
ASR Performance Evaluation

- Improved word error rate for Speech2Text model from 64% to 2.8% for HPC dataset and 27% to 12% for HPC+TIMIT dataset
- Training loss for ASR models fine-tuned on different combinations of datasets
- Speech2text performs slightly better than Wav2Vec and hence we use it as the default ASR model

Train Dataset		Dataset used for Test			
		Speech2Text		Wav2Vec	
HPC	TIMIT	HPC WER	HPC+Timit WER	HPC WER	HPC+Timit WER
✗	✗	64.613	27.53	67.92	27.16
✗	✓	71.15	33.18	77.38	35.43
✓	✗	2.85	21.8	3.24	65.6
✓	✓	2.92	12.18	3.09	14.24



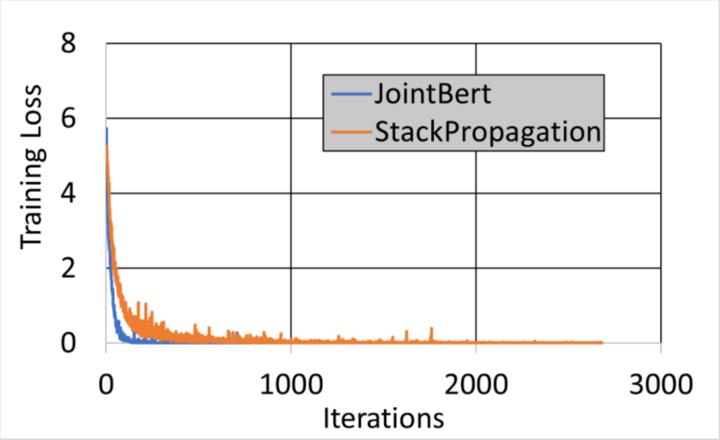
Speech2Text



Wav2Vec

NLU Performance Evaluation

- No pre-trained NLU model is available for HPC profiling tools
- We evaluate the accuracy of predicting intents and filling slots for our trained NLU models versus human-supervised and labeled HPC-NLU dataset
- We choose JointBert as our default model for NLU module as it gives better accuracy for both intents and slots



Training loss of JointBert and StackPropagation models

Model	F1 Score for slots	Intent Accuracy
StackPropagation	0.775	91.79%
JointBert	0.8773	93.36%

ASR+NLU and End-to-end Performance Evaluation

ASR+NLU: ASR and NLU modules are evaluated together as a pipeline to see accuracy of converting speech to intents/slots

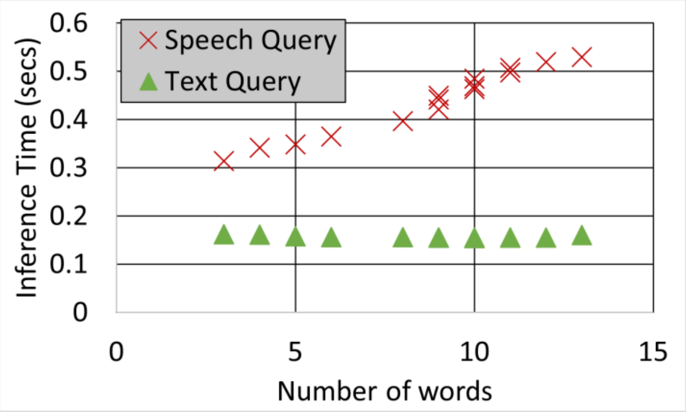
- Using JointBert as NLU model

ASR Model	F1 Score for slots	Intent Accuracy
Speech2Text	0.8295	92.92%
Wav2Vec	0.8349	92.47%

End-to-end Overhead: we evaluate Inference latency evaluation from user speech/text input to generating URL and passing it to the tool

- Since different visualizations vary in rendering time they are not included in the latency

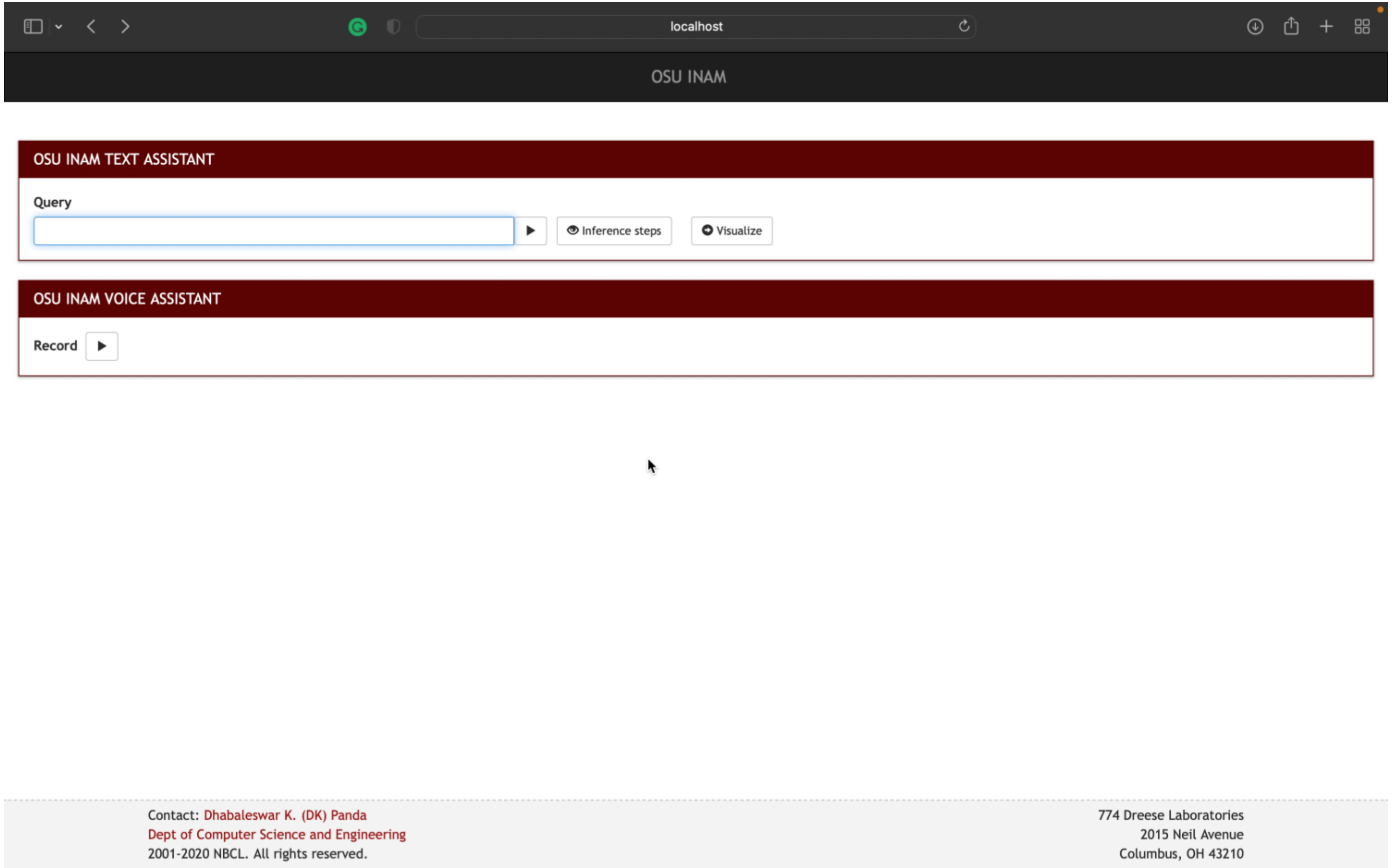
- The inference latency of processing speech increases with an increase in the number of words in the query
- The inference latency of processing text remains the same



Outline

- Introduction and Motivation
- Problem Statement and Proposed Designs
- Performance Evaluation
- **Demo**
- Concluding Remarks

Using CAI - Demo



Outline

- Introduction and Motivation
- Problem Statement and Proposed Designs
- Performance Evaluation
- Demo
- **Concluding Remarks**

Concluding Remarks

- We explored the challenges for designing a conversational (speech/text) interface for HPC tools and
  - Used state-of-the-art AI models for speech and text and adapted it for use in the HPC by retraining them on new HPC datasets we created
  - Demonstrated that CAI delivers higher accuracy
  - Created an interface to convert speech/text data to commands for OSU INAM
  - Showed how how users can utilize the proposed interface to gain insights quicker leading to better productivity
- As future work we plan on releasing various components developed
  - HPC-ASR and HPC-NLU datasets
  - The retrained ASR and NLU models
  - CAI and the enhanced OSU INAM profiling tool with support for CAI
  - Extend CAI to other popular profiling tools
  - Preform user survey

Ref. for more details: Hey CAI- Conversational AI Enabled User Interface for HPC Tools; P. Kousha, A. Jain, A. Kolli, S. Prasanna, S. Miriyala, H. Subramoni, A. Shafi, and DK Panda; ISC'22

Follow us on

https://twitter.com/mvapich

Thank You!

Panda@cse.ohio-state.edu

Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

The High-Performance MPI/PGAS Project
http://mvapich.cse.ohio-state.edu/

The High-Performance Big Data Project
http://hibd.cse.ohio-state.edu/

The High-Performance Deep Learning Project
http://hidl.cse.ohio-state.edu/