Multi-Threaded Tasks

John D. Lanza
Foley & Lardner LLP

Follow this and additional works at: http://www.tdcommons.org/dpubs_series

Recommended Citation
http://www.tdcommons.org/dpubs_series/570

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.
MULTI-THREADED TASKS

ABSTRACT

Systems and methods described herein allow for optimizing handling of sequence dependent tasks in voice activated computer environment. A data processing system can receive a first audio input signal and a second input audio signal from a user device associated with a corresponding user. The data processing system can determine, for each of the first and second input audio signals, a respective user request indicative of a service or online action requested by the user. The data processing system can generate a first action data structure and a second action data structure based on the user requests identified from the first and second input audio signals. The data processing system can determine one or more sequence dependency parameters associated with the first and second action data structures. The data processing system can then construct a multi-thread action data structure according to the sequence dependency parameter, and perform operation associated therewith according to an order defined by the multi-thread action data structure.

DETAILED DESCRIPTION

In a voice activated computer environment, a virtual personal assistant can be configured to engage in audio conversations with users via their client devices. Users can make audio requests for a variety of services including cab rides, plane tickets, streaming services, restaurants, movies, electronic shopping, etc. A data processing system can process input audio signals received from client devices to determine and handle corresponding user requests. Handling user requests can include generating action data structures based on the determined
user requests, and providing the generated action data structures to service providers. The service providers can process received action data structures and respond, e.g., via the data processing system, to the user’s requests indicated in the corresponding received action data structures.

The data processing system can serve thousands of users per unit time (e.g., per few minutes). In each conversation with the virtual personal assistant, the corresponding user can make multiple requests. Each user request can be indicative of a corresponding service. The data processing system needs to keep track of and handle all the requests made by each user. Some user requests (or user requested services) may involve multiple conversation segments between the virtual personal assistant and the respective users. Such conversations segments may involve getting back to the user, after receiving a request, to ask for more information, user selection among multiple service options, or user confirmation/approval. As such, the data processing system may need to maintain a processing state for each user request (or each user requested service) indicative of a stage at which the processing of that user request is.

Systems and methods of the present disclosure allow the data processing system to improve the efficiency and effectiveness in handling users requests and data communications, over one or more computer networks by, between the data processing system and client devices or computing devices of service providers. The data processing system can generate based on received input audio signals corresponding action data structures for transmission to service provider computing devices. The data processing system can determine some sequence dependency between user requests or corresponding action data structures associated with any given user. The sequence dependency can indicate an order of operation of actions indicated by the action data structures or their corresponding data transmissions. Based on the determined
sequence dependency, the data processing system can merge or combine multiple action data transmissions of a given user into a multi-thread of actions.

The multi-thread can include sequence dependent operations of multiple action data structures associated with a given user. The data processing system can provide the multi-thread or portions thereof to multiple service provider computing devices to accomplish actions indicated by the multi-thread. The data processing system can also provide the multi-thread or portions thereof to at least one content provider computing device to obtain content items associated with the multi-thread. The service provider computing devices and the content provider computing devices can communicate, via or bypassing the data processing system, with the user client device to render audio output to accomplish the actions indicated by the thread, or to render content items at the user client device. The consolidation of multiple action data structures into a multi-thread based on identified sequence dependency allows for efficient and reliable handling of user requests. In particular, such consolidation can allow for proper ordered execution of user requests. Also the consolidation allows for reduced processing power, memory utilization, and communication bandwidth. Requirements of the data processing system are reduced, relative to individual processing of multiple action data structures without such consolidation and sequence dependency identification.

FIG. 1 is a flowchart depicting an example method 100 for optimizing the processing of sequence-dependent user requests received via a voice-based virtual personal assistant in a voice activated computer environment. The method 100 can include, at step 105, the data processing system receiving a first audio input signal and a second input audio signal from a user device associated with a corresponding user. The user can initiate a conversation with a voice-based virtual personal assistant via the corresponding client device and, for example, request “OK, I
would like to go to dinner tonight” in the first input audio signal. The second input audio signal can be received subsequent to the first input audio signal and can include “OK, I would like to catch a movie later.” The client device of the user can receive the first and second audio signals via a corresponding microphone and forward the audio signals as they are received to the data processing system.

At step 110, the data processing can determine, for each of the first and second input audio signals, a respective user request indicative of a service or online action requested by the user. The data processing system can machine-translate the first and second input audio signals to a corresponding text signals. A natural language processor (NLP) component of the data processing system can parse each generated text signal to identify one or more respective keywords, and identify a user request based on the identified keywords. For example, with respect to the first input audio signal, the NLP component can parse the corresponding text signal and identify the keywords “go to,” “dinner,” and “tonight.” Based on these identified keywords, the NLP component can deduce that in the first input audio signal, the user asking for a restaurant to have dinner tonight. Since no location is specified by the user, the NLP component can conclude or assume that the user is seeking a restaurant within a certain distance from his location. The NLP component may access a database or a data structure to identify any user preferences with regard to restaurants.

The NLP component can also parse the text signal corresponding to the second input audio signal and identify the keywords “catch,” “movie,” and “later.” The NLP component can deduce that the second input audio signal indicates that the user is interested to watch a movie after the dinner in the restaurant. In particular, the keyword “later” here can be interpreted by the NLP component to indicate that the movie is to follow the dinner.
The data processing system may further predict other potential user requests, for example, based on records of past conversations (or past user requests) or based on other information that may be specific to a given service or a given service provider. For example, based on past user requests, the data processing system may determine that the user will most likely ask for a ride to the restaurant and/or back home. The data processing system may send an audio signal to the client device requesting whether the user needs a ride. The user may respond with another audio signal, which the data processing system can process to determine whether the user is requesting a ride service. The data processing system may determine that the user usually posts on a social media platform about user’s daily events. The data processing system may confirm with the user regarding social media posting. The data processing system may determine that a user requested service may require logging in to a user account or creating a new account for the user with the corresponding service provider.

At step 115, the data processing system can generate a first action data structure and a second action data structure based on the user requests identified from the first and second input audio signals. The first action data structure can include information related to the first user request, such a search query for restaurants within a geographical region, information for reservation in a specific restaurant, or instructions to retrieve one or more menus from websites associated with one or more restaurants. The second action data structure can include information related to the second user request, such a search query for movies playing within some theaters, a search query related to a specific movie, or instructions to purchase a movie ticket from a website associated with one or more movie theaters. The data processing may create action data structures corresponding to additional user requests or actions predicted or identified by the data processing system. For example, the data processing system can create
action data structures corresponding to logging in to (or creating) a user account, requesting a cab ride, or posting on social media.

At step 120, the data processing system can determine one or more sequence dependency parameters associated with the first and second action data structures. The sequence dependency parameter(s) can be indicative of an order for executing operations associated with the generated action data structures. The data processing system can analyze the generated action data structure to determine the sequence dependency parameter(s). For example, the second action data structure may include a timing parameter greater than a timing parameter of the first action data structure, indicating that operations or actions associated with the second data structure are to occur after executing operations of the first action data structure. Also, the data processing system may give more weight in terms of timing priority to an action data structure associated with a user request (e.g., dinner) that was made prior to other user requests (e.g., movie). An action data structure associated with a ride service may include an indication that the destination address is equal to an address associated with another action data structure (e.g., restaurant address). As such, the data processing may conclude that operations associated with the “dinner” need to proceed operations associated with reserving the ride. The data processing system may determine that the action data structure associated with social media posting indicates that the social media posting is to be performed while the user is in the restaurant. Accordingly, the data processing may deduce that operations associated with the social media posting are to be subsequent to operations associated with the “dinner” and the “ride” to the restaurant. In the case where an operation associated with a given action data structure requires logging in or creating an account, the data processing system may schedule such operation subsequent to the operation of logging in or creating the user account, which may be associated with another action
data structure. Based on such analysis, the data processing system can determine an order of performing operations or tasks associated with separate action data structures. In some instances, the data processing system may send audio requests to the user to ask for or confirm the order of a given operation.

At step 125, the data processing system can construct a multi-thread action data structure according to the sequence dependency parameter. The multi-thread action data structure can include multiple elements indicative of a list of operations or tasks associated with the generated action data structures (e.g., the first and second action data structures and any other action data structures associated with any additional predicted user requests). For example, the operations can include transmission of various pieces of data to different service providers. The multi-thread action data structure can be constructed as a linked list, a tree, a table, or some other type of data structure. Each element in the multi-thread action data structure can include an address of a computing device of a corresponding service provider and data to be transmitted to that service provider. The data processing system can be configured to execute elements of the multi-thread action data structure according to the defined order. For example, the data processing system may not transmit data associated with a given element until data associated with a previous element was successfully transmitted and a confirmation of a service was received from the corresponding service provider. For example, there may be no point of scheduling the ride of a restaurant reservation for the dinner is not confirmed yet.

Each element of the generated multi-thread can include a respective status parameter indicating the status of operation associated with that element. The data processing system can update the status parameters responsive to execution of corresponding operations or tasks. An API of the data processing system can use such status parameters to detect completion of an
action, so that to the data processing system 105 can then transmit data for a different action to a
service provider computing device. The data processing system may identify a deadlock
condition associated with one or more actions or operations of the generated multi-thread action
data structure. In such case, the data processing system may revert back to user for additional
information. The generated multi-thread action data structure may be deleted once all operations
or tasks associated therewith are completely and successfully performed.
Receive a first audio input signal and a second input audio signal from a user device associated with a corresponding user

Determine user requests corresponding to first and second input audio signals

Generate a first action data structure and a second action data structure based on the user requests identified from the first and second input audio signals

Determine one or more sequence dependency parameters associated with the first and second action data structures

Construct a multi-thread action data structures according to the sequence dependency parameter(s)

FIG. 1