Technical Disclosure Commons

Defensive Publications Series

October 02, 2017

PROCESSING MULTIPLE TASKS BY VIRTUAL ASSISTANTS

Google Inc.

PROCESSING MULTIPLE TASKS BY VIRTUAL ASSISTANTS

ABSTRACT

An interactive assistant, referred to herein as “an interactive assistant,” “a virtual assistant,” or simply an “assistant,” is described that processes multiple tasks presented in a single query by a user. The user may, in other words, issue a query requesting that the virtual assistant perform multiple tasks. The virtual assistant may parse and buffer (or, stated differently, cache) each of the tasks of the multiple tasks, and process them appropriately, such as serially or concurrently (which may be referred to as being processed “in parallel”).

DESCRIPTION

Virtual, intelligent, or computational assistants (e.g., also referred to simply “assistants”) execute on counter-top devices, mobile phones, automobiles, and many other type of computing devices. Assistants output useful information, responds to users’ needs, or otherwise performs certain operations to help users complete real-world and/or virtual tasks. Some assistants are configured to record reminders, notes, lists, and other information as persistent records that a user can later retrieve or modify during a subsequent interaction with the assistant.

A virtual assistant is described that processes multiple tasks presented in a single query by a user. The user may, in other words, issue a query requesting that the virtual assistant perform multiple tasks. The virtual assistant may parse and buffer (or, stated differently, cache) each of the tasks of the multiple tasks, and process them appropriately, such as serially or concurrently (which may be referred to as being processed “in parallel”).

In operation, a device may provide an interface with which to interact with the virtual assistant. The device may include a cellular phone (such as a so-called “smart phone”), a tablet computer, a laptop computer, a desktop computer, a workstation, a home audio device (such as a so-called “smart speaker”), a gaming console, a portable gaming console, an audio/visual (AV) receiver, a digital disc player, and the like.

The device may include, in some examples, a display by which to present the interface for interacting with the digital assistant. The device may further include one or more sensors by which to receiver data input by a user of the device, the data specifying interactions with the virtual assistant. The sensors may include capacitive touch sensors integrated with the display to capture text interactions, one or more transducers (e.g., a microphone) for capturing audio interactions, cameras for capturing image and/or video interactions, a global positioning system (GPS) sensor for capturing location-based interactions, a compass for capturing positional-based interactions, etc.

In some examples, a server or dedicated device (which may be separate from the device providing the interface by which to interact with the digital assistant - which itself may be referred to as a “client device”) may provide some or all of the functionality of the digital assistant. The server may receive the data indicative of the interaction (e.g., any combination of the above text, audio, image, video, location-based, or position-based interactions) from the client device. The server may include one or more processors configured to process the data indicative of the interactions in accordance with the digital assistant to generate a result, and provide the result to the client device responsive to the data indicative of the interactions.

The digital assistant may represent a software agent trained through, at least in some examples, machine learning to process data indicative of natural language text and audio, and possibly images and video to perform tasks or services for individuals. The digital assistant may, to provide a few examples, answer queries, provide directions to destinations, schedule meetings and/or appointments, generate electronic mail (e-mail) and/or text messages, initiate a telephone call, and the like.

As a user interacts with the assistant, the assistant may obtain personal information about the user. Examples of personal information include: habits, routines, preferences, notes, lists, contacts, communications, interests, location histories, and other types of user information. After receiving explicit permission from the user, the assistant may store, the personal information at user information data stores and, in the course of providing assistant services, make use of the personal information stored at the user information data stores.

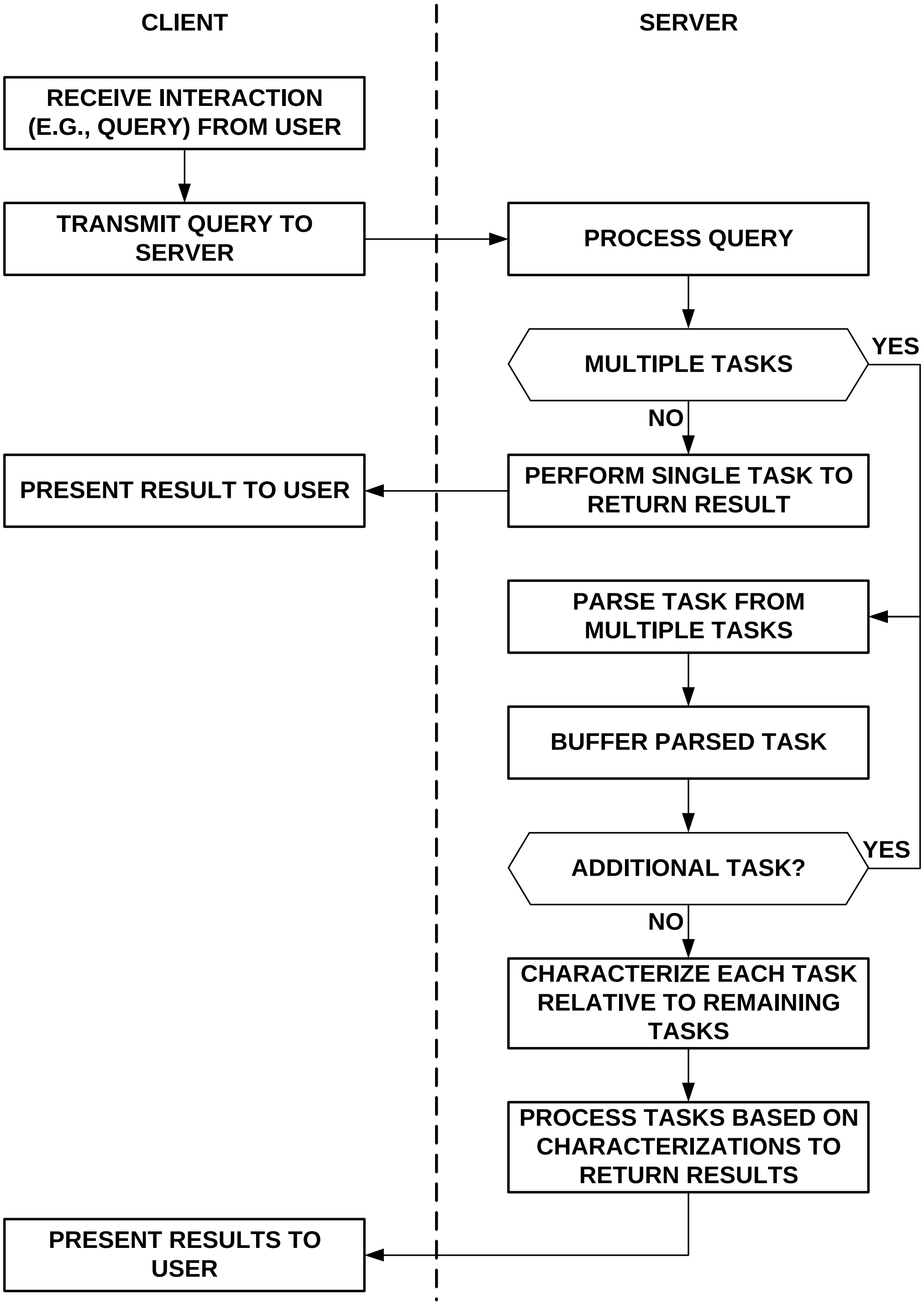
A user may be provided with controls allowing the user to make an election as to both if and when the assistant, the computing device, or the computing systems described herein can collect or make use of supplemental data (e.g., user information or contextual information about a user's social network, social actions or activities, profession, a user's preferences, or a user's current location), and if and when the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed.

For example, a user's identity may be treated so that no personally identifiable information can be determined for the user, or a user's geographic location may be generalized where location information is obtained (such as to a city, ZIP code, or state level), so that a particular location of a user cannot be determined. Thus, the user may have control over what supplemental data is collected about the user, how that supplemental data is used, and what supplemental data is provided to the user.

Furthermore, the external systems, the computing devices, and the assistant treat the information stored at the information stores so that the information is protected, encrypted, or otherwise not susceptible to unauthorized use. The information stored at the information data stores may be stored locally at the computing devices and/or remotely (e.g., in a cloud computing environment provided by the external systems and which is accessible via a network).

In some instances, a user may interface with the device to issue a single query containing multiple tasks to the digital assistant. The digital assistant may parse each task from the multiple tasks, buffering each task until, in some example, the digital assistant is able to perform the task. The digital assistant may characterize each task parsed from the multiple tasks relative to each other. Based on the characterization of each task, the digital assistant may determine whether each of the tasks parsed from the multiple tasks is to be performed serially or concurrently.

To illustrate, consider that a user may issue a query for the digital assistant to “order a small cheese pizza for home delivery, and play my news summary.” The query in this examples includes two tasks, i.e., a first task – order a small cheese pizza for home delivery, and a second task – play the news summary associated with the user issuing the query. The digital assistant may characterize the first task relative to the second task, determining that the first task is unrelated to the second task. The digital assistant may also characterize the second task relative to the first task, determining that the second task is unrelated to the first task. The digital assistant may, based on the characterization of the first task being unrelated to the second task, determine that the first task and the second task may be performed concurrently. The digital assistant may then perform the first task and the second task concurrently.


CLIENT
SERVER
RECEIVE INTERACTION (E.G., QUERY) FROM USER
TRANSMIT QUERY TO SERVER
PROCESS QUERY
MULTIPLE TASKS
YES
NO
PRESENT RESULT TO USER
PERFORM SINGLE TASK TO RETURN RESULT
PARSE TASK FROM MULTIPLE TASKS
BUFFER PARSED TASK
ADDITIONAL TASK?
YES
NO
CHARACTERIZE EACH TASK RELATIVE TO REMAINING TASKS
PROCESS TASKS BASED ON CHARACTERIZATIONS TO RETURN RESULTS
PRESENT RESULTS TO USER

As shown in the above flowchart, the client device may receive an interaction from the user (e.g., in the form of a query). The client device may transmit the query to the server. The server may process the query. When the query includes a single task, the server may perform the single task to return a result, providing the result to the client. The client may present the result to the user.

However, when the query includes multiple tasks, the server may parse a task (such as the first task discussed above) from the multiple tasks. The server may next buffer the parsed task. The server may determine whether there are additional tasks to be parsed from the multiple tasks. Responsive to determining that an additional task is still to be parsed from the multiple tasks, the digital assistant may parse another task (such as the second task discussed above), and buffer the additional task. The digital assistant may iterate in this manner until there is not an additional task to be parsed from the multiple tasks.

Responsive to determining that there is not an additional task to be parsed from the multiple tasks, the digital assistant may characterize each task relative to the remaining tasks. Although described as characterizing each task parsed from the multiple tasks relative to every other task parsed from the multiple tasks, the digital assistant may characterize each task relative to every other task and apply the characterization as if the two tasks are an unordered set. In other words, applying the foregoing to the example of the first task and the second task discussed above, the digital assistant may group the first task and the second task in an unordered set, characterizing the unordered set as being unrelated. The digital assistant applies the characterization of “unrelated” to the first task relative to the second task, and the second task relative to the first task. Processing the tasks in this manner may reduce the number of comparisons during characterization.

To illustrate, consider a serial processing example in which the user issues a query for the digital assistant to “provide directions from my current location to the nearest park, and, after arriving at the park, show a map of the park.” In this example, the digital assistant parses and buffers two tasks, i.e., a third task – provide directions from my current location to the nearest park, and a fourth task – after arriving at the park, show a map of the park. The digital assistant may characterize the third task relative to the fourth task as being related, such that the two tasks cannot be performed concurrently. The digital assistant may process each of the third task and the fourth task based on the characterization, which in this example occurs serially.

The digital assistant may process the third task to return the result (i.e., the direction from the user's current location to the nearest park in the above example). The client device may present the result to the user. The digital assistant may next, after arriving at the nearest park, process the fourth task to return the result (i.e., a map of the nearest park). The client device may present the map of the nearest park to the user.

Although described as presenting the result, the presentation of the result may not necessarily involve actual presentation of some item to the user, but that the server or the client device may perform some operation or task related to the query, such as texting a contact, emailing a contact, opening an application, scheduling a meeting or other event, initiating a phone call, pushing information to an attached wearable device (such as a head mounted display or a so-called smart watch), etc.

Although described as providing the indication to the server, the client device may locally process the indication of the intended option and provide the result to the user. That is, the client device need not interface with the server hosting the digital assistant to return the result to the client. Furthermore, in some examples, the client device may host the digital assistant itself, thereby avoiding network expense, power consumption and the like associated with interfacing with the remotely hosted virtual assistant.