Technical Disclosure Commons

Defensive Publications Series

December 19, 2018

COGNITIVE CLOUD COLLABORATION

M. David Hanes

Sebastian Jeuk

Gonzalo Salgueiro

Chuck Byers

COGNITIVE CLOUD COLLABORATION

AUTHORS:
M. David Hanes
Sebastian Jeuk
Gonzalo Salgueiro
Chuck Byers

ABSTRACT

Techniques are described for using a cognitive learning instance to learn a user's preferences across a wide range of cloud collaboration services and provide a customized user experience. This customized user experience is then leveraged on the back-end for optimizing cloud resources.

DETAILED DESCRIPTION

Cloud collaboration services allow users to connect via chat, voice, and video, while also sharing documents and resources. At the same time, most of these services are contained in separate clouds (usually Security as a Service (SaaS)) that do not communicate. User experiences are therefore quite different and there is no coordination of user state and intent.

What is needed is a way to provide a consistent experience to the user while always ensuring that resources in the various collaboration clouds are readily available for user tasks and needs.

A cognitive learning instance may be employed that would understand a user's preference for communication, roles (e.g., work identity, personal identity, etc.), special functions needed (e.g., hearing/vision/speech impaired), location (e.g., at work, home, driving, etc.), relationships between callers (e.g., family members, work colleagues, etc.), and so on. In one example, users may control how much is shared. A Cognitive Cloud Collaboration System entity may build a profile and abstract the communication method to a degree allowing users to communicate in a much easier manner. Also, the cognitive model may evolve its user experience over time as the user changes behavior patterns or registers different preferences. Figure 1 below illustrates a high level overview.

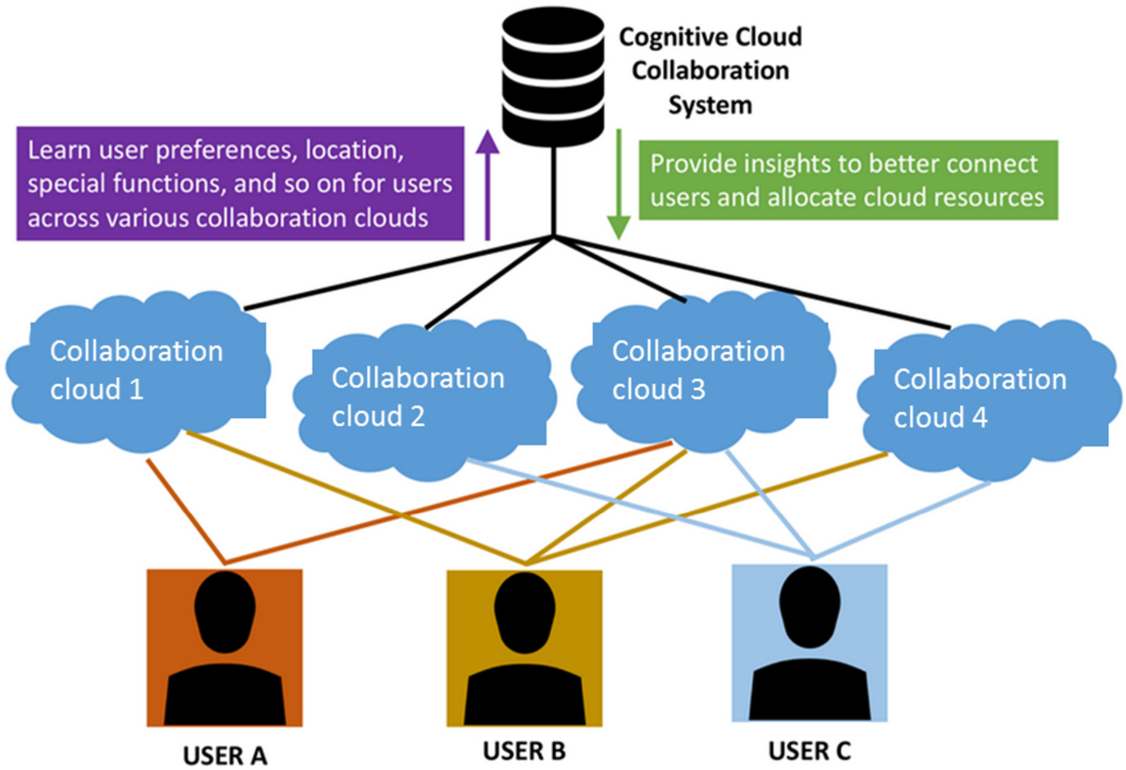

Figure 1

The learning phase of the cognitive model may use several possible strategies. One is a dialogue / questionnaire with the user about collaboration preferences. This may be easily implemented as a web form or a chatbot (e.g., using digital assistants or as a virtual member of a collaboration session). The user may be asked about many preference options, such as the preferred method of collaboration at different times of the day, days of the week, locations, or activities. The system may also learn which of the many collaboration systems the user prefers for various tasks, file formats to be used, security settings, etc. It may allow for free-form rules (e.g., "my husband hates it when I get interrupted during dinner"; "never show me desktop images while I am driving, but put them on my car's dashboard display if I am a passenger"; "never enable my camera before 8AM or after 8PM"; "you always knowing where I am and what I am doing really creeps me out"; etc.). The ML system may map the user preferences onto actionable rules for the collaboration systems to use. If anything is ambiguous, the ML system may ask clarifying questions or send the user a quick email, text message, or social media query.

This process is useful for initial setup of the system, but would not be efficient to evolve the user's preferences or integrate new collaboration capabilities over time. That problem may be solved by monitoring the user's behaviors, and integrating them into a preference database. In one example, a user's preferred audio connection is a cell phone, but he/she never answers a cell phone call while in an open office building, and instead immediately goes to a privacy room and calls back in. Here, instead of calling the cell phone, the system may learn to send a text message to the cell phone that says it found a free audio privacy room nearby and will call that room in thirty seconds. If the user does not answer until forty-five seconds have elapsed, that timing may be adjusted by the ML algorithm for the next time.

The collaboration systems may also be equipped with an ever-present avatar that was connected to the preference engine. A trigger phase "Hey [NAME]", for instance (either typed into a text chat window or said into a microphone), may activate the interface, disconnect the user's audio/video from the collaboration stream temporarily, and enable a dialogue with the chatbot to adjust system settings. Once the user is done with the dialogue, the user may say "Thanks [NAME]" and the system may return to the collaboration session with the new settings taking immediate effect. Machine learning techniques may be essential to maintaining the efficiency of the user interface with the settings engine.

On the cloud side, this same cognitive entity would learn a user's needs and usage patterns so that resources across multiple collaboration clouds could be allocated effectively and efficiently. Figure 2 below illustrates how this idea can be used for Cloud planning and optimization.

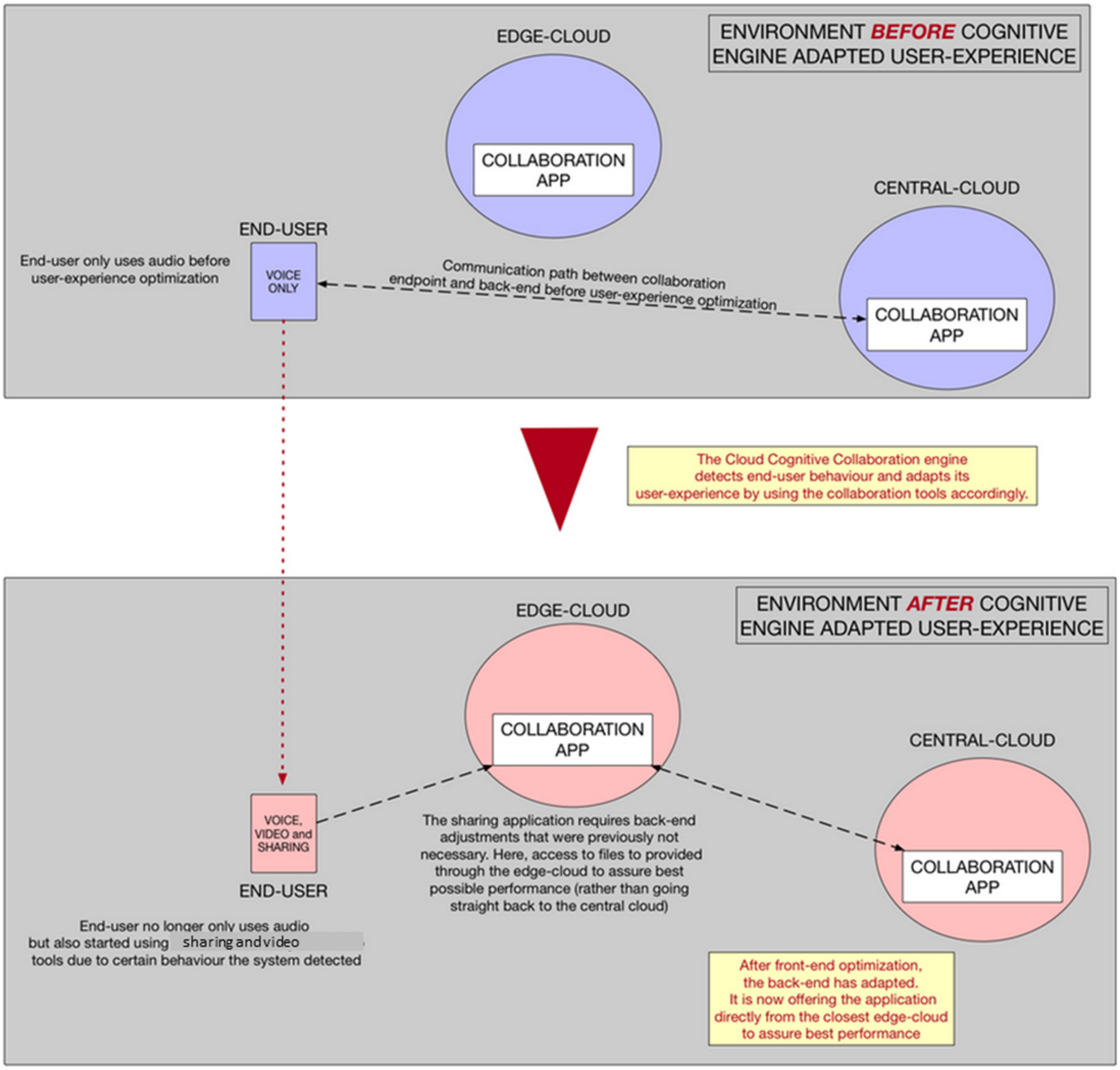

Figure 2

Figure 2 illustrates an embodiment that enables cloud optimization. This embodiment not only enhances the user's collaboration experience to offer the most optimal configuration from a front-end perspective but also dynamically adjust the back-end to reflect the front-end changes. The front-end and user experience changes that are driven through the cognitive cloud collaboration engine adjusts the way a user experiences collaboration tools. However, these changes not only require adjustments directly visible to the user, but they also often require updates in the back-end to provide the most optimal performance based on the front-end adjustments over time and situation.

Provided herein is an integrated and directly linked optimization process for the back-end that reflects changes done on the front-end to enhance the user-experience (e.g., sending a text rather than calling, knowing a user will not answer a "Call Me" operation, etc.). Back-end optimizations include, but are not limited to, the location of the application a tenant is leveraging, the optimized Software-Defined networking in a Wide Area Network (SD-WAN) connectivity to the right collaboration tools, tenant specific Quality of Service (QoS) in a SD-WAN owned by the collaboration service provider, and other performance oriented optimizations.

The optimizations in the back-end are direct results of the front-end changes to enhance the user experience. They are driven through the user-interaction with the system and the behavior learned over time. Each user, having different behaviors and hence having a different user experience, requires a different back-end optimization. Regulations may be put in place by the service owner to assure continuity and stability of the overall system. The service provider may assure the right balance between optimal performance as well as a stable and consistent network. These regulations can be defined through tenant SLAs, whereby a golden SLA tenant has the most flexibility and hence performance optimization operations are done frequently. A silver or bronze SLA tenant, while still getting performance optimization operations, have restrictions that helps introduce overall network stability, balancing out more dynamic/changing tenants.

This system may tightly integrate various collaboration systems into a holistic "does everything" system. Further, it may initially learn and dynamically adjust user preferences using several techniques to keep the system efficient and unobtrusive, and use the cognitive cloud to share these preferences across all authorized collaboration systems. On the back end, this system provides insights for optimizing the collaboration services across multiple clouds based on the desired front-end user experience.

In summary, techniques are described for using a cognitive learning instance to learn a user's preferences across a wide range of cloud collaboration services and provide a customized user experience. This customized user experience is then leveraged on the back-end for optimizing cloud resources.