

A Case for CACE

Suman Nath

Microsoft Research

Recent years have seen two significant trends in the computing landscape: an increasing availability of sensors-integrated smartphones and the Cloud to which phones can be continuously connected to. This provides an opportunity for a novel class of applications, which we call *Context-Aware Cloud-Edge* (CACE) applications. Such applications distribute over many smart *edge-devices* (e.g., smartphones) and the Cloud, and react based on the operating conditions of users, their social states, and the surrounding environment. A canonical example is a *friend-finder* app on a phone (e.g., the *Loopt* app on iPhone, Android, and Windows Phone) that notifies a user whenever any of her friends are near her current location. Other examples include *connected cars*¹ that provides location-aware telematics and data analytics over a large number of cars connected to the Cloud, location-aware coupon services (e.g., *GeoQpons* on Android and iPhone) that notify a user when she is close to a business offering coupons that she might like, mobile multiplayer games (e.g., *iMobsters* on Android) that monitor and react on players' mutual interactions and status, and in-car dashboard apps that provide real-time route and gas station recommendations.

There are two main ingredients in these applications: smart edge-devices and the Cloud, both of which have become ubiquitous by recent technological developments. We believe that many more such applications will appear in coming years. Therefore, it makes sense to build a common platform that captures the common ingredients and allows an application developer to quickly prototype a new application. Building such a platform, however, requires addressing several challenges.

Reliable Context Inference. Physical context is at the core of CACE applications. Some recent works have shown how to efficiently use sensors on smart phones to infer various context attributes such as a user's location, transportation mode, social states, etc. However, existing solutions apply only to a small set of context attributes and many of the solutions are not robust enough to be used as simple plug-and-play. Providing robust and simple plug-and-play solutions for a rich collection of context attributes can enable many new useful applications.

Program Specification. Writing a Cloud-Edge application is itself oftentimes non-trivial. The usual practice is to write them in standard procedural or object-oriented languages such as Objective C, Java, or C#. This puts a high burden on application developers, as they often need to implement a significant part of the core application logic (e.g., filtering and correlating data feeds from various devices), which is often distributed in nature, from the scratch. Further, both the data source and the application logic usually have a significant temporal component. Specifying and processing distributed temporal logic can be hard even for experienced developers. A simple declarative programming framework can significantly simplify this task.

¹<http://thinkd2c.wordpress.com/2011/04/07/microsoft-and-toyota-connected-car-in-the-cloud/>

Efficient execution. It is challenging to tune a Cloud-Edge application to make efficient use of resources such as energy and communication bandwidth of edge-devices. Currently, application developers implement various ad hoc resource optimization techniques such as duty cycling, off-loading expensive computation the Cloud, compressing data during transmission, etc. However, implementing such optimizations is often nontrivial. The situation aggravates when an application needs to compare or correlate data from multiple edge devices. For such applications, resource optimization involves deciding what computation to push, and to which edge device. Such computation placement decisions require solving optimization problems involving various factors such as the network topology, rates of the data streams, data upload and download costs, pairs of streams to correlate, etc. Moreover, since these parameters can change over time, the decision needs to be dynamically updated. The complexity of implementing such optimizations often outweighs the cost of developing the core functionality of an application.

Privacy. Context-aware applications require users to release their contexts, which raises serious privacy concerns. Recent works have proposed solutions based on suppressing contexts or releasing modified (e.g., generalized) contexts. However, in many scenarios, these techniques do not necessarily prevent an adversary from inferring sensitive contexts. Suppression itself can leak information. Released nonsensitive contexts can reveal information about sensitive context to an adversary who knows temporal correlation between sensitive and nonsensitive contexts. A formal framework to guarantee privacy against such strong adversary is necessary for widespread adoption of context-aware applications.

At Microsoft Research, we have been working on various solutions to address these challenges [1, 2, 3, 5, 4, 6]. For details, please check <http://research.microsoft.com/~sumann>.

References

- [1] Hossein Ahmadi, Nam Pham, Raghu Ganti, Tarek Abdelzaher, Suman Nath, and Jiawei Han. Privacy-aware regression modeling of participatory sensing data. In *ACM SenSys*, 2010.
- [2] Badrish Chandramouli, Joris Claessens, Suman Nath, Ivo Santos, and Wenchao Zhou. RACE: Real-time applications over cloud-edge. In *ACM SIGMOD*, 2012.
- [3] Michaela Goetz, Suman Nath, and Johannes Gehrke. MaskIt: Privately releasing user context streams for personalized mobile applications. In *ACM SIGMOD*, 2012.
- [4] Alexandra Meliou, Wolfgang Gatterbauer, Suman Nath, and Dan Suciu. Tracing data errors with view-conditioned causality. In *ACM SIGMOD*, 2011.
- [5] Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *ACM SIGMOD*, 2010.
- [6] Arsalan Tavakoli, Aman Kansal, and Suman Nath. On-line sensing task optimization for shared sensors. In *ACM/IEEE IPSN*, 2010.