A Private Framework for Distributed Computation

Edward Tremel⁠*, Ken Birman⁠*, Robert Kleinberg⁠*, and Márk Jelasity†

There is a growing class of distributed systems applications in which data stored on client platforms must be aggregated or analyzed without revealing private information to the operator. Systems such as the smart power grid, control systems for energy-efficient buildings, and traffic analysis in large cities all depend on the analysis of data supplied by measurement devices, yet the clients being tracked are unwilling to reveal such measurement data directly to the system owner, who might be “curious” about private client information. These systems thus may elicit public opposition despite their useful features because of a perceived privacy risk.

There are ways to upload sensitive data to an aggregator without compromising privacy, but existing options have limitations. One possibility is to keep the data encrypted with keys known only to the clients, but this requires expensive homomorphic encryption if the aggregator is to compute directly on it. Another is to employ a mechanism to de-correlate client identifiers from their data, as Chen et al. do in [4], but this imposes restrictions on the kind of aggregation that can be done. Instead, it would be beneficial to execute needed computation directly on the client platforms, so that the system operator or analyst only sees aggregate results. This approach would provide a better alternative to central aggregation provided it is privacy-preserving, robust, and efficient.

A data aggregation system based on client-side computation suggests a purely peer-to-peer architecture [10], which many systems have used to avoid centralized control [8, 7, 11]. However, peer-to-peer systems have problems of their own, even if we set privacy concerns aside. By eschewing centralization entirely, they can no longer take advantage of the powerful management tools developed for today’s cloud computing model. In traditional peer-to-peer systems, clients are isolated network hosts rather than devices within a single administrative domain, and often have difficulty maintaining connections to each other through firewalls and address translation barriers. Determining the membership of a peer-to-peer network is a surprisingly difficult problem, since there is no one entity that knows the identities of all the clients, and changes in membership may not be detected and propagated in a timely fashion [1]. Without a centralized service to assign and manage node identities, Sybil attacks [6] are always possible, so a peer-to-peer system is extremely vulnerable to a few malicious peers becoming a majority of the apparent nodes in the system. Even choosing peers fairly becomes difficult, because peers usually do not store the entire membership list locally, and it is fairly easy for malicious peers to poison local mem-

---

* Cornell University
† University of Szeged, Hungary
Acknowledgements

This work was supported, in part, by a grant from the NSF Smart Grids program.

References


