Eywa: Crowdsourced and cloudsourced omniscience

Murat Demirbas Yavuz Selim Yilmaz Muhammed Fatih Bulut

Ubiquitous Computing Lab
Department of Computer Science and Engineering
University at Buffalo, SUNY, Buffalo, NY, 14260
{ *demirbas* | *yavuzsel* | *mbulut* }*@buffalo.edu*

***Abstract*—Here we present our ubiquitous computing vision, Eywa. Eywa is an open publish-subscribe system that employs crowdsourcing for tasking and social networks & machine learning for identifying relevance. We argue that crowdsourcing (and the social networks and machine learning that enable it) should be a first class citizen in ubiquitous computing. We also observe that cloud computing is a natural platform to host such future ubiquitous computing systems. We discuss about some applications enabled by Eywa, and focus on our *CuratedLiving* application (which emphasizes "less choice more relevance" approach) as a case study.**

***Keywords*-Crowdsourced collaboration; Smartphones; Internet of things; Social networks; Cloud computing**

I. Introduction

Although Weiser outlined his ubiquitous computing vision in 1991 and called for computers to come outside and become embedded in the fabric of our physical world [1], the technology trends went the opposite way, and a lot of physical services became virtualized and moved into the computer (and the cloud) in the next couple of decades. Since then we saw the rise of e-mail, e-commerce services, and even e-friendships in social networks (Facebook, Twitter). However, behind the scenes, we also saw a growth in the physical part as well. With the advances in MEMS technology, it has become feasible to produce various types of sensors inexpensively, in very small form factor, and in low-power usage. What started with wireless sensor networks research for instrumenting the world [2], found market adoption lately in the form of Arduino devices [3], Maker movement [4], and Internet of Things [5]. More recently smartphones saw widespread adoption[1] and brought mini computers with instant access to the cloud to our fingertips. Smartphone localization then paved the way to location-based services and location-aware services (smartphone-based cab hailing [6], suggest-nearby services by search engines [7], and line wait-time forecasting [8] [2]), which are tell signs of the pendulum swinging back to physical, this time empowered with a very strong cyber part.

While there has been good progress and wide availability of the devices (smartphones, tablets, sensors) to fulfill the ubiquitous computing vision, the-state-of-the-art in software and integration is lagging far behind. Consider DARPA's 2009 network grand challenge on the occasion of the 40th anniversary of the Internet. The challenge was to accurately find 10 weather balloons deployed in arbitrary locations of the U.S. within a day [9]. There was an award of $40,000 for the team that would first report the locations of the 10 balloons accurately, and this challenge was solved within 9 hours. The winning team employed social networks and a multilevel incentive structure, but had to prepare, campaign, and publicize aggressively for an entire month before the challenge day. This points to a big gap between the potential of the smartphones and the reality of the smartphone software today. Why are the existing apps so limited and person-centric? Why can we not have an app that is able to solve similar active collaboration and coordination problems automatically?

II. Eywa vision

We argue that the reason for this gap is the lack of an infrastructure to task/utilize these devices for collaboration and coordination. In the absence of such an infrastructure, the state-of-the-art today is for each device to connect to Internet to download/upload data and accomplish an individual task that does not require collaboration and coordination. In contrast, providing an infrastructure for publish/subscribe and tasking of these devices would enable any device to utilize the data published by several devices in a region, as well as to task several devices in a region to acquire the needed data, if the data is not already being published to the infrastructure.

[1]There are more than 5 billion cellphones, with 20% of these being smartphones, in the world. The smartphone ratio in the US is 50% and rapidly increasing.

[2]In our earlier work, LineKing [8], we have used the sensors on the smartphones to detect the wait times at coffee shops, and leveraged on the computational power provided by cloud to provide future wait time estimations with less than 3 mins mean absolute error.

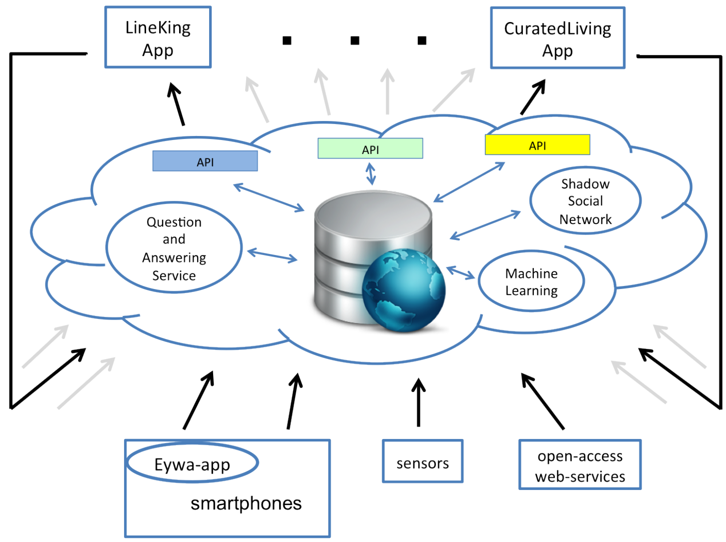

Figure 1. Eywa ecosystem

In order to task/utilize ubiquitous devices for collaboration, we propose a ubiquitous computing middleware, *Eywa*, which provides an open publish-subscribe infrastructure for smartphones, and paves the way for crowdsourced sensing and collaboration applications. Eywa differs from previous work as its emphasis is on active and focused crowdsourcing and it serves as an enabler for users/devices to task other devices automatically to collect the needed data. Eywa employs both social networks and location-induced spatiotemporal social networks for tasking, and leverages on a multiple-choice question-answering service for enabling more effective crowdsourcing. Furthermore Eywa also provides energy-efficiency, resource-pooling, privacy-pooling, and network-effect benefits to the Eywa-affiliated apps.

Next we discuss about the components of Eywa. Figure 1 illustrates an overview of the Eywa ecosystem. [3]

A. Sensing and aggregation component

Sensing. Eywa can get input from many types of devices/people through many different channels: parking lots [11], public transportation schedules [12], organization event calendars [13], Internet of Things devices, street webcams, open-access web services such as Open Street Map/traffic, Open Data, and Open Government projects. Most of Eywa's data will be collected from smartphones using Eywa-enabled/affiliated applications, which may include GPS data, wireless access point (WAP) signal data, Bluetooth connection information, and many different modality sensory streams (accelerometer, magnetometer, microphone, ambient audio, ambient lighting, gyroscope, temperature, proximity). Eywa's reach on the smartphone also extends to social network information and status updates (Twitter, Facebook, Foursquare) and event calendars [13].

For implementing the sensing component at the smartphones, a potential solution is to leverage on an Eywa-app installed on the user's smartphone. The first time a user downloads an Eywa-affiliated app (such as LineKing or CuratedLiving), the user will be prompted to install the Eywa-app as a prerequisite to get this app working. Later when the user downloads other Eywa-affiliated apps, Eywa need not be installed again, as it is already installed. The Eywa-app serves as a broker to upload smartphone sensor data to the Eywa-backend, upon the request of the Eywa-backend. The Eywa-app sends periodic heartbeat (location & battery level) messages to the Eywa-backend as the backend needs to track of the phones in order to task them for data collection when needed. The Eywa-affiliated apps on the smartphone also request information from the Eywa-app, which serves as a proxy and communicates to the Eywa-backend as suitable. The Eywa-app provides several advantages including energy-efficiency, resource-pooling, and privacy-pooling. The Eywa-app, of course, uses energy on the phone for sensing/collecting data, but instead of each app doing this separately in a wasteful manner, the Eywa-app allows pooling resources together and enables one sensor reading being reused for several Eywa-affiliated apps.

Aggregation. Eywa will provide a flexible/extensible archi-

[3]The name for this middleware is inspired by the Avatar movie directed by James Cameron. In the fictional universe of Avatar [10], Eywa is the name of the guiding force and deity of the planet Pandora and the Na'vi people. The Na'vi believe that Eywa acts to keep the ecosystem of Pandora in perfect equilibrium. All living things on Pandora connect to Eywa through a system of neuro-conductive antennae; this is why Na'vi can mount their direhorse or mountain banshee steeds and ride them immediately without going through the necessary steps required to domesticate such wild animals. The Na'vi can upload or download memories from Eywa using their queues and it can even be used for mind transfers in certain cases.

tecture to allow mashing-up and combining of data collected from devices, and develop an open publish-subscribe service for the Eywa-backend to enable Eywa-affiliated apps to integrate the data in unanticipated novel ways. For example consider the example of streamed GPS locations to Eywa from smartphones of participant truck drivers. The truck company may access/use this information through Eywa to check on the progress made by the drivers and send them warning messages if there are concerns. The highway authority may use this information to monitor the utilization congestion levels on the roads, and may also use this to send weather/road related warnings. The highway service plazas may also want to use this information to plan their food/accommodation services more accurately. Finally unanticipated applications may also spring. This information may be used for estimating the fresh vegetable/food deliveries to the city and to schedule shopping of health-conscious/dieting users.

The information flow in the Eywa ecosystem is bidirectional. In the Eywa ecosystem, the Eywa-affiliated apps receive data from Eywa, use the data for their purposes, and also publish their resulting data to Eywa in a relevant way to provide back value. This symbiosis relation provides embedded collaboration and fuels the growth of the Eywa ecosystem thanks to the network-effect. In order to keep track of how much value an app (or a user) is getting from the Eywa ecosystem, versus how much value the app (or a user) is giving back to the Eywa ecosystem, Eywa aggregation framework can perform *app-level accounting* (or *user-level accounting*). Eywa-affiliated apps will be added and shared through the Google Play app market as they are developed, however the Eywa middleware may choose to provide limited service or eventually refuse service to an Eywa-affiliated app, if the Eywa-accounting component registers big gaps between the value-drawn versus value-provided for the app.

B. Computing and Machine learning component

Eywa-backend will be hosted on the cloud to provide scalability, high-availability, high-bandwidth to enable many devices to upload data and query Eywa concurrently. Eywa deploys machine-learning and a question-answering service over a cloud computing platform to achieve relevance. Scalable cloud computing architectures and software for machine learning are now mature and widely used. Eywa employs Big Data analytics (using MapReduce [14], Giraph [15], [16], etc.) to process, interpret, and learn from the huge input data streams it aggregates. Eywa can employ well-established/standard machine learning techniques to help learn basic patterns (seasonality, correlation) in the data collected and fit models on these data in order to interpolate for missing data and also for detecting anomalies (to determine change/events) in incoming data [17], [18]. For finding appropriate people for a task, Eywa can leverage on the question-answering service described below as this may prove to be a hard task for the machine learning component.

[3]Network effect is the effect that one user of a good or service has on the value of that product to other people. When network effect is present, the value of a product or service is dependent (often grows quadratically) on the number of others using it. This applies to Eywa because as the Eywa-affiliated apps increase and Eywa users increase in number, Eywa's overhead will decrease quickly and benefits will increase quickly.

C. Social networking and Crowdsourcing component

Crowdsourcing refers to distributing a query to several users in order to exploit the wisdom-of-crowds effect. Examples of crowdsourcing may be a participant-powered weather/rain-radar (with better precision and ground-truth than meteorological weather radars) [19], friendsourced suggestions (restaurants, music, places, experience, topics, other friends) [20], expertsourced advice, and crowdsourced location-based queries. Eywa relies primarily on smartphones as the devices for crowdsourced tasking. Smartphones provide several advantages for crowdsourcing. They cover large areas due to their mobility. They are personal and administrated by their users, and provide the potential of interacting with the phone user for tasks requiring human intelligence and intervention, such as taking a picture of a requested location, answering a question. Eywa leverages on social networks to enable crowdsourced tasking and collaboration applications. Social networks are important for tasking and collaboration applications because a person can ask for favors from her friends and social network more easily than complete strangers. Dually, one may enjoy volunteering his time more for friends/acquaintances than complete strangers. Furthermore, suggestions sourced from friends and acquaintances have more relevance.

Eywa maintains a shadow social network at the Eywa-backend, which is mirrored from Facebook and Twitter by means of the Eywa-app (authorized by the user on her phone). Eywa-app uses the public social network APIs (e.g. Facebook Open Graph API) to collect the relevant social network information for the users, and shares this with the Eywa-backend. This shadow social network also maintains a device-user association, as Eywa will use this to task appropriate smartphones from a user's social network to start sensing and publishing information to Eywa when needed. Since this is a shadow social network, the Eywa related sensing/tasking information will not litter the main social network timeline of the user.

Instead of all Eywa-affiliated apps getting separate access to the social networks of the users, our shadow social network model provides a privacy-pooling opportunity to better protect the privacy of the users. Eywa will also play a role in personal data privacy on the smartphones. Today 26% of Android apps in Google Play can access personal data, such as contacts and email, and 42% have access to GPS location data –in many cases, whether they need it or not. By using the Eywa middleware, it can be possible to regulate the app access to personal data through the Eywa APIs. Eywa does not share any unauthorized personal information with third parties, including Eywa-affiliated apps. For apps that require sensor readings, Eywa only provides what is required as anonymized and collated information when possible.

In addition to social networks of devices, Eywa also considers location-induced networking of devices. Although the device users may not be acquainted and in each others social circles, it may happen that these users share the same spaces regularly. This concept is also known as *familiar strangers*. For example, two students taking the same class in a big auditorium, or people working in the same building but for different companies, or people living in the same neighborhood and use the same shopping malls may fit into this category. These people have relevant information to share with each other, but not the appropriate social connections/channel to share this information. Eywa fills in this gap, by also taking into account the location-induced networking of devices (we call this as spatiotemporal device networks) as a relevant component. Eywa captures the spatiotemporal device networks using WiFi-direct [21] technology.

Incentive-based tasking. Eywa employs a virtual currency model to maintain an incentive mechanism. In this user-level incentive model, every user needs to pay in order to get a sustained-level of service/value from Eywa, and every user gets rewarded when they publish/provide value to Eywa. Eywa's incentive mechanism can be honed to maintain the quality of queries/answers and to eliminate spam queries (by making them harder/costly to initiate). Consider the DARPA network challenge example [9]; in the Eywa ecosystem, such an experiment will be easy if the user possesses and agrees to spend her virtual currency on this. This will not be a big deal for a user that provides back meaningful results and value to Eywa, since she will earn virtual currency back as these results get used. The Eywa incentive mechanism will aim to force the users to be profitable to each other, and maintain a healthy ecosystem based on symbiotic mutualism. To this end, Eywa can employ techniques to detect the market price for certain tasks. By getting the accounting information of how much a user draws value from Eywa and provides value back to the Eywa ecosystem, Eywa can calculate the price to be charged or coins to be rewarded for tasks.

Question-answering service. Eywa employs a multiple-choice question-answering service to improve the crowdsourcing/tasking, which may otherwise require significant machine learning to determine which people/devices would be most suitable for a given task. By crowdsourcing even this tasking process, Eywa avoids significant complexity compared to using a fully machine learning based solution. For example, when looking for experts or the most suitable people to task for a job, Eywa may first employ light machine learning and identify several candidates. Then the question-answering service may ask a user the following multiple choice question: "Among your friends <a, b, c, and d> that I found as suitable for this submitted task/topic, which one could be the best to perform/answer?". The multiple-choice question-answering component can also help reduce the machine learning requirements of the CuratedLiving app (which we describe in Section 3) by narrowing the many choices encountered, and providing the intelligence needed for customizing the CuratedLiving app.

The insight behind the multiple-choice question-answering service is that, for crowdsourcing, asking multiple choice questions is more productive than asking open-ended questions. Also, providing multiple choice questions is often feasible. The original asker can provide the multiple options when the question is about deciding among some choices (hotels to stay, product to buy, music to listen). It is acceptable to expect the asker to write in multiple choices for her question. And, when multiple choice options cannot be provided by the asker, it is possible to automate the process of adding multiple choices to an open space question [22].

As a first step to build a multiple-choice question-answering component, we recently started working on a crowdsourced system to play "Who wants to be a millionaire" (WWTBAM) television quiz-show live. Our work was inspired by IBM Watson's success at Jeopardy and aims to utilize the crowd (not a supercomputer!) to answer WWTBAM questions faster and more accurately than any contestant. To this end, we developed an Android and iOS app and the backend software to enable users to play "Who wants to be a millionaire" live. We target a Turkish audience due to the high popularity of the show there. When the show is on air in Turkey, this app makes a notification sound to alert users to pick up their phones and start playing. Our two students type the questions as they appear in the show, and the app users enter their answers using their phones. The users enjoy the game-play and the participation. By pressing a button in the app, they have a chance to brag on Facebook and Twitter about the hard questions they answered, and after the game they can see their ranking among the other users.

The app has been live for 15 weeks as of this writing, and has already been downloaded by 147,000 people in this relatively short time. The data we collect suggests that it is feasible to build the superplayer by aggregating answers from many players. By just going with the most selected answer, we can answer 88% of the questions correctly, however these are mostly entry-level questions. After question 6, the most selected answers' success rates plunge to 60%, 50%, and 30% quickly with each consecutive question. We are investigating how we can improve the success rates of these later/tougher questions. As one approach, our app is asking the players to enter a confidence level ("certain", "guessing", and "no idea") when they answer a question. To keep this confidence level assessment accurate, we award/deduce virtual gold coins to/from the user by factoring with respect to this confidence level. Our initial tests show that by taking the confidence levels into account, we can increase the success rates for the tougher questions, for which the most voted option has low success rates. Another thing we will try is to use the answer history of the users to learn about their expertise and also weigh this in when aggregating answers.

III. Curated living service as a case study

The *CuratedLiving* service builds on top of Eywa, and aims to deliver less complex more relevant experiences to its users. The user tells CuratedLiving what type of day/week she wants to have and CuratedLiving provides a select number of paths (carefully curated choices) to arrange these goals. For example, the user may say "Over the weekend, I need to buy some dresses, do groceries (the grocery list is automated/learned), and also want to see a movie, meet with 2-3 friends to chat". The CuratedLiving then suggests a couple plans, which may also be refined along the way opportunistically, for realizing these goals. This is a sophisticated endeavor and builds on Eywa as the underlying service. CuratedLiving also requires cloud computing hosted machine learning to learn the preferences and the mood of the user and provide customized plans based on these. For example, initially, CuratedLiving chooses a direct and low-congested road (based on the traffic reports posted to Eywa), however, taking into account the mood of the user (sensed via her smartphone), the radio show or music the user is enjoying at the car (again sensed via the smartphone), CuratedLiving may instead opt for a longer but more comfortable or scenic drive. Based on the best available parking spots (again learned from Eywa), CuratedLiving may determine where to enter the mall, and this in turn may affect the plans as to which activity to take on first. At some point in the day, based on the availability of friends nearby (whose mobile positions are learned from Eywa), CuratedLiving may arrange a joint lunch. Alternatively based on the interests of the friends, this is converted to an art gallery visit followed by snack at a cafe (whose line-wait times are input to Eywa via LineKing [8]). Finally, based on customized/bargained deals, CuratedLiving may guide the user to buy the dresses and groceries.

CuratedLiving application uses Eywa via the Eywa APIs. These APIs enable CuratedLiving to record subscriptions to Eywa events/notification based on <keyword, location, time> as well as semantic clues. In real-world there is always some uncertainty so the Eywa notifications also include associated certainty levels. The APIs also include ways to modify the granularity and the fidelity level of information to be obtained from the publishing sources. For example, if the user trusts a friend more on some topic, the CuratedLiving service may instruct Eywa to prioritize data collection or tasking regarding that friend.

We envision that CuratedLiving will be able to improve itself and customize itself to serve better to the habits/peculiarities of its users by processing the data collected by the service, including the feedback/ratings of users about the experiences. The service can process these data and cluster users into groups and suggest a user specific experiences that users similar to her rated as most satisfying/pleasing. Enabling technologies for the CuratedLiving application are already in place. Siri provides a nice natural language/audio interface to the application. Google self-driving cars enable extra level of automation in the mobility of the user to provide a seamless curated living experience. Social networks and smartphones are other enabling technologies that have established themselves deeply and found widespread adoption.

A. Discussion about CuratedLiving

Curated living has always been around in some form. The culture/society we are raised in shapes/curates our lives. We also curate our lives indirectly by reading books (self-help books, cooking books, travel books) or talking to friends. CuratedLiving provides a more direct/efficient way of curating our lives. But, of course, making something fast and efficient is not always necessarily a worthy goal. An inefficient life curation process may have its advantages; it makes us more conscious and aware of the choices we need to make, and this strengthens our sense of self. As we make mistakes and regret them, we develop our tastes, likes and dislikes, and our ambition in life. *An unexamined life is not worth living* as Socrates declared.

We welcome self-driving cars because driving is a rot, and our greatest aspirations are not to be *chauffeurs*. But could it be the case that by giving away the rot of curating our lives, we are giving away too much? How much is too much? To answer these questions it is important to involve social sciences into this discussion. In any case, the CuratedLiving service will be tunable to provide a spectrum that gives only nominal guidance up to giving full service. Even in the full service mode, CuratedLiving will sprinkle some (hard-computed) perfectly reasonable deviations from the beaten track to make our day and life richer. Only by providing these opportunities to explore, the choices to make mistakes, it would be possible for the users to develop their tastes, whims, and sense of self.

IV. The next 100 ubiquitous computing applications

In addition to the CuratedLiving application that we presented as a case study, Eywa enables several novel applications. We list some here. In the CuratedLearning app, the objective is to empower users to learn in a manner customized for their learning preferences and acquire useful study habits in a gradual manner. The Audubon society national bird counting app aims to enable the bird watching hobbyists to collaborate on precise and accurate surveying of the bird species [23]. Several social collaboration apps may also build on Eywa, including pick-up soccer games, arranged ride-sharing, and support groups for exercising, weight-watching, and recovering addicts. Other apps built over Eywa may include: smart health apps, crowdsourced surveillance (Amber alert for missing children, homeland security), ad-hoc self-improvement clubs where everyone teaches something to the group, socializing (meeting interesting people in nearby crowds), forming opportunistic flashcrowds to make a difference (art, cleaning campaigns), and augmented reality gaming. Another application domain of Eywa could be question-answering in the enterprise. Enterprises already indicate significant interest in social networks for question answering within the enterprise. Combined with microfinancing/microtransactions, the Eywa-affiliated applications such as CuratedBusiness/Office, CuratedLiving, CuratedLearning, CuratedTravel, etc. may even give people the chance to provide service related to their hobbies/passions and make a living. This grassroots participation economy can provide a segue into a more productive and creative society.

Eywa can also be very useful for extreme-events and disaster recovery contexts. For example in the wake of a superstorm, Eywa would be useful for discovering scarce resources such as water, gas, and medical/cleaning supplies. Eywa can also be useful for organizing volunteering teams, as it can know what services volunteers might provide based on their abilities, capabilities, and context; and then distribute help based on the availability of volunteers and priority (elderly, disabled, and children get help first).

Recently, as part of a multiple faculty team at SUNY Buffalo, we started the PhoneLab [24] project to enable next-generation collaborative smartphone experiments. PhoneLab is a large-scale smartphone testbed that provides an order of magnitude more participants than typical smartphone experiments. As the first phase of Phonelab project, our team already distributed 200 phones to participants at UB. Phonelab provides access to both kernel layer and app layer and overcomes the limitations of App Store or Google Play. We plan to leverage on PhoneLab, as an information outlet and smartphone testbed, to build a first prototype of Eywa for enabling campus-wide ubiquitous collaboration and coordination applications.

References

[1] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[2] http://research.microsoft.com/en-us/projects/senseweb.

[3] http://www.arduino.cc.

[4] http://p2pfoundation.net/Maker_Movement.

[5] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787 – 2805, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128610001568

[6] https://www.uber.com.

[7] http://googleblog.blogspot.com/2010/02/refine-your-searches-by-location.html.

[8] M. Bulut, Y. Yilmaz, M. Demirbas, H. Ferhatosmanoglu, and N. Ferhatosmanoglu, "Lineking: Crowdsourced line wait-time estimation using smartphones," in *MobiCASE: Fourth International Conference on Mobile Computing, Applications and Services*, 2012.

[9] http://archive.darpa.mil/networkchallenge/.

[10] "Avatar wiki," http://james-camerons-avatar.wikia.com/wiki/Eywa.

[11] R. Sudhaakar, A. Sanzgiri, M. Demirbas, and C. Qiao, "A plant-and-play wireless sensor network system for gate monitoring," *The 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, 2009.

[12] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, ser. MobiSys '12. New York, NY, USA: ACM, 2012, pp. 379–392. [Online]. Available: http://doi.acm.org/10.1145/2307636.2307671

[13] http://ubicomp.cse.buffalo.edu/ubupdates.

[14] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492

[15] "Apache giraph framework," http://incubator.apache.org/giraph/.

[16] G. Malewicz, M. H. Austern, A. J. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski, "Pregel: a system for large-scale graph processing," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, ser. SIGMOD '10. New York, NY, USA: ACM, 2010, pp. 135–146. [Online]. Available: http://doi.acm.org/10.1145/1807167.1807184

[17] http://ubicomp.cse.buffalo.edu/citypulse.

[18] C. G. Akcora, M. A. Bayir, M. Demirbas, and H. Ferhatosmanoglu, "Identifying breakpoints in public opinion," *Workshop on Social Media Analytics*, 2010.

[19] M. Demirbas, M. A. Bayir, C. G. Akcora, Y. Yilmaz, and H. Ferhatosmanoglu, "Crowd-sourced sensing and collaboration using twitter," *11th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2010.

[20] M. F. Bulut, Y. S. Yilmaz, and M. Demirbas, "Crowdsourcing location-based queries," in *PerCom Workshops*, 2011, pp. 513–518.

[21] http://www.wi-fi.org/discover-and-learn/wi-fi-direct.

[22] C. Lin, Mausam, and D. Weld, "Crowdsourcing control: Moving beyond multiple choice," *Human Computation Workshop (HCOMP)*, 2012.

[23] http://birds.audubon.org/christmas-bird-count.

[24] R. Baldawa, M. Benedict, M. F. Bulut, G. Challen, M. Demirbas, J. Inamdar, T. Ki, S. Y. Ko, T. Kosar, L. Mandvekar, A. Sathyaraja, C. Qiao, and S. Zawicki, "Phonelab: A large-scale participatory smartphone testbed," in *NSDI 9th USENIX Symposium on Networked Systems Design and Implementation*, ser. NSDI '12, 2012.