

UNICORNS 2.0

Disaster Cloud

The cloud that helps us help others

Edvinas Jaras, Bartosz Wasylów, Konrad Stojanowski

12/1/2013

Introducing an application that helps endangered areas to forecast natural disasters and helps them cope with the disaster after it occurs.

MOTIVATION

Recently we were all witnesses to the dramatic incident in Riga (Latvia) where a shopping mall collapsed killing 54 people and leaving even more hurt. However, the mobile phones proved to be very useful in finding the victims who were trapped under debris. We believe we can make mobile phones even more effective in providing help to people endangered by both natural and man-caused disasters. What is more, we can try to lessen the number of victims in such cases.

Our motivation for this project is that in case of a natural disaster lurking above many a soul, even if an appropriate alarm is raised on TV or radio, it does not reach everyone as not everyone has either of them turned on at all times. In some places endangered by the natural hazards the local governments invest in sirens which are to sound in case of any emergency. Unfortunately, those do not reach everyone either. However, almost everyone has a mobile phone now and such a general notifying system can be more effective and can help propagate the information to wider range of people that could be affected by the disaster. We will concentrate mainly on the disasters caused by natural factors as they can be forecasted easier than those man-caused.

INTRODUCTION

Coping with disasters, on the top level, is divided into three main parts:

- 1) Prevention phase, which covers detecting the catastrophe and notifying the people involved;
- 2) Rescue phase, which follows unsuccessful prevention phase, covering rescuing people at stake and immediate reaction including creating shelters and managing volunteers to distribute needed resources.
- 3) The last phase is recovery after the cataclysm, in which we want to provide areas with needed communication and possibly information about funding.

Prevention: Prevention is the most important part, considering dealing with disasters. Many times they happen suddenly and consume sometimes even millions of beings. The abruptness of this kind of events often deprives us of the possibility of saving the victims, but in the age of electronic devices being able to carry millions of operations, we can outdo the catastrophe, allowing endangered people to leave the area before the disaster strikes. Such an undertaking requires terabytes of data and enormous processing power, which often aren't accessible to meteorological companies and offices, although are available by using modern cloud solutions.

Introducing a mobile phone application of such importance brings a chance, though, that the product won't be useful, due to too many users without the knowledge how to make full use of it. That is why, in the prevention phase, we should introduce officials, such as policemen, attendants or mayors. We should also explain the notion of this application to the masses and encourage people to install it and to get used to it. The more phones participate in the cloud system, the faster and more accurate it becomes, allowing more cases when lives at stake are rescued. In the age of phones being used mainly to entertain us, it is also important to explain and profoundly anchor the idea, that phones can also serve to save lives.

After processing terabytes of information, detected anomalies in data (which possibly signify the disaster) are being validated and the users are notified by mobile phone alerts. Low risk and high risk, depending on the probability of data and users' input.

If the evacuation had not succeeded or had not been finished, the officials ought to release an official order to evacuate the area and to supply the area with resources needed. Facebook statuses or tweets would be incorporated into the final data processing, and would be spread around the area, increasing social awareness.

First volunteers can register via the Disaster Cloud to provide the victims with appropriate help.

The application would prompt the people to enter the battery-saving mode as there can be energy cuts, and the phones should serve for the whole evacuation or rescue process.



Rescue: After the disaster strikes and if there are still people in the area of impact, rescue teams should be formed and sent to save the endangered. The application allows the rescue teams to be registered and managed at a higher level, bringing help to more people.

The strike of a disaster or high probability of one, followed by a lack of service, triggers survival mode, with an option to turn it off, when one is not in danger.



The survival mode switches all other applications off to prolong the battery life for as long as needed, sending only periodical messages. This would allow the server to access their location and provide it to rescue teams in proximity. In survival mode all other connections are turned off, but if triggered by a rescuer's phone nearby, it starts ringing.

Despite all of the phones in the impact area getting the survival mode message, people who have not been affected by the disaster itself can switch it off, automatically informing the server that they do not need any help from the rescuers.

Users who do not need any help can register as a volunteers, learn about coping with current situation, send a report or a request, contact officials etc.

A volunteer can be assigned to rescue teams or shelters, according to their abilities, where they can help the injured and the people without any place to stay or without any food.

The rescue teams, the safe places, and resources needed to make them useful (both human and non-human) are being managed by trained officials, who alongside the server, can make optimal use of the available resources.

Volunteers, who are assigned to the appropriate rescue teams, wait while connected to the server for it to assign them areas to be searched.

When in the approximate vicinity of a victim, rescue team members can generate an impulse, which makes the victim's phone ring revealing its position, allowing them to save the trapped owner.

Offline mode: During a disaster, the mobile network can be damaged, causing the most of the application functions to be useless. Regardless, the phones without any service, switch to the

survival mode, allowing the impulse sent by the rescuers to trigger the ringing, hence enabling them to find the trapped more easily.

Recovery: After the disaster, the affected area would still need time to recover from the damage. According to statistical data, the world’s attention to the disaster area lasts for less than a month, while a full recovery process lasts for nearly a year. That is why the application would provide the areas with needed funding by attracting willing donors and as well as legal help, which is very important to people who lost their lifetime savings.

Intelligent resourcing and resource management tools allow communities to report their needs what leads to them being covered faster, all thanks to mobile network. When given appropriate tools, local people can restore the area in shorter time, due to their inner motivation.

In the end, the families can find each other faster, thanks to the improved communication.

CLIENT SIDE

The client will gather information regarding temperature, humidity, pressure, magnetic field (and other sensory data available with almost every Android phone) from the user’s environment periodically. The data collected will be uploaded to the cloud immediately or daily, depending on user’s preferences.



The user may agree to share unused computational resources of his mobile phone with the cloud, so that some of the calculations will be performed on the device. That will enhance the cloud’s performance and give a more precise weather forecast to the end users (and also to the user who shares his resources).

The client also receives information from the cloud about possible natural hazards around the user. The information may have different levels of “severity”, e.g. it can be “temporary notice” or “urgent alarm”. Different natural hazards are more probable or often in some parts of the world (e.g. Pacific

Ocean for typhoons) or at given part of the year (e.g. typhoon season). In such a case the client should be informed upon the possibility of the hazard occurrence. If the information gathered by the cloud allows it to raise an urgent alarm, e.g. a volcano explosion, typhoon, tsunami, flood or other serious hazard in the user's neighbourhood, the phone should get the information immediately and sound a very loud alarm (regardless of the "mute" or "vibration" mode of the phone).



The client allows also the user to raise alarms on his own. If a person is a witness of any disaster, either natural or man-caused, he can report it using the app. Firstly, if the disaster has natural causes and the data collected from other sources can support this (e.g. the user raised a typhoon alert and the weather conditions are such that it can happen), the entitled administrator (or the cloud itself – after exceeding some level of danger) can send the information to all other phones in that area. The man-caused disasters can be handled similarly. The person can raise an alarm about e.g. a plane crash, a building collapse or other disaster. To support the claim, the person may add a photo or a video recording which can give more information on the severity of the danger.



That can be obviously abused by people who want to create social unrest; therefore all the user input should be carefully scrutinized. Moreover the users who try to deceive the system should be punished severely, for example be banned from any future user input.

SERVER-SIDE (CLOUD) COMPUTATION

Server-side computation happens in several steps, namely: collection, filtering, analysis and response. This happens over a static time period, and as such clients receive periodic updates with results regarding weather, as well as warnings about any possible disaster threats. Since mostly cloud computing is used, servers do not need (very) high computational power themselves, though they do require high bandwidth to effectively interact with the cloud system. They would also need access to a database where they could store and look-up needed information. Detailed descriptions of all steps are provided below.

Collection: Server collects and accumulates data from clients as well as meteorological stations over a set time period. The stations provide sensitive data in specific areas, while clients provide a bigger amount of data spread over a wider area. Client data, due to its size and location (usually highly populated areas) can provide a good description of current conditions in zones of importance. Station data, as a lot of the stations are placed in non-populated areas (seas, oceans, mountain ranges, etc.) can provide data which can be used to forecast incoming changes, and can be also used to issue warnings about possible disaster scenarios.

Filtering: A lot of client data can be distorted and irrelevant to the statistic. The client can be indoors, or the mobile device can be in a place where its readings would be distorted by external means. There is also the possibility of a device malfunctioning. We need to filter out such results, and we can achieve that by several means:

1) *Noise-cancelling:* An algorithm similar to the ones used for noise-cancelling in image and audio correction. Such a method will filter out reports that differ in their measurements from the surrounding readings by more than some specified amount (more about constants in the computation part).

2) *Statistical analysis:* We can apply mathematical and physical formulae to compute the mean, deviation and uncertainty values in these results. Results with high uncertainty values should have a low weighting later on in the computation part. Results which deviate from the mean by an amount bigger than some constant multiplied by our standard deviation should be investigated further (check if the client is isolated or in-doors) and apply the steps from the method described above.

The best result would be obtained from combining the above mentioned methods and possibly with the addition of other techniques as well. User made reports from banned members (reputation calculations discussed in next step) should be excluded. All of the filtering is done using cloud resources.

Note: After filtering, non-standard results should be saved separately in case of an anomaly or a sudden onset of a disaster, since extremely high winds or very high humidity can be the result of a tornado or a flood/typhoon which appeared spontaneously. Recognizing the limitations of our

system, this should be kept some time, in case this particular client was directly affected by the disaster and such data may be used later in the search and rescue process.

Computation: During the computation stage, several values are computed using the cloud's processing power:

1) *Current conditions:* Using the data collected from clients' sensors, a map of current conditions can be created and then analyzed using meteorological techniques (based on differences in pressure, humidity and temperature fluctuations) to check if any warnings should be issued. The weighted average should be computed in areas with a high density of input. The weights are determined using the uncertainty (innate uncertainty of sensors and propagated uncertainties in further calculations).

2) *User created warnings:* Since every client can send a disaster report, this should be verified in a second round of filtering now that the average conditions have been computed. If the user's report is unsupported by any factors, like the probability of the event occurring in the area (tsunami in the Sahara desert would have quite a low weight), other reports or weather conditions, reported by surrounding sources of data (clients or stations), then it may be dismissed, and the user will receive a decrease in his reputation. Users submitting truthful reports will have their reputation increased. If the report comes from a trusted user, then it may be considered even if it's unlikely. Data to support such reports can be gathered from social media like Facebook, Twitter, Google+, and the like, using keywords or special tags. If the report is not dismissed it will have a warning rating associated with it based on the proximity (calculated client-side based on the distance between them and issuer of the report) and it's likeliness.

3) *Constants:* Using machine learning algorithms and user input, the server can modify its constants to better assign weights and warning levels, also the ones used for filtering can be changed. This should be implemented so the system can adapt and be better prepared for real-world disasters.

After it has been computed, it goes onto the response stage.

Response: The server would then send the warnings to the users, as well as any other requested weather data. If the warnings exceed some critical level, the server would have the devices sound an alert and devices on the cloud would retransmit the signal further, extending the network to increase awareness of a potential risk. Server would also send warning about disasters which have a high probability of occurring in that area, so that the client would be up to date whenever he switches locations.

SUMMARY

The processing power and vast amount of data, enabled by a cloud system comprised of millions of mobile devices, will be able to produce a really accurate weather and climate report, which may be used to inform the population of imminent catastrophes. To summarize, such software coupled with the necessary hardware, would not only accelerate rescue missions and make them safer, but also make a big contribution to preventing casualties by raising alarms and awareness.

LINKS USED

<http://www.organicmarketingservices.com/wp-content/uploads/2013/05/cloudcomputing.jpg>

<http://www.bbc.co.uk/news/world-europe-25069947>