A Scalable and Decentralized Publish/Subscribe-based Weather Alert System

Violetta Vylegzhanina, David Brett and Aniruddha Gokhale

Sponsor: NSF CAREER REU

1. MOTIVATION

Limitations observed in an existing severe weather warning system at Vanderbilt University

- The centralized nature of the system adversely impacts scalability
- Alerts are sent to all registered users => false
- Real-time notifications
- Decentralization
- Scalability

2. PURPOSE

Overcoming the limitations in the existing system by investigating the following dimensions:

- Decentralization
- Scalability
- Ease-of-use
- Real-time notifications
- Elimination of false alarms (future work)

3. CLOUD-BASED ARCHITECTURE FOR ANDROID PLATFORM & APPLICATION LIFECYCLE STEPS

A user views the message on app’s main screen if app is running

4. DESIGN CHOICES FOR CLOUD-BASED NOTIFICATION SERVICE

Challenge:

- A capability needed to handle real time notifications to mobile devices – a publish/subscribe capability
- Reduce the need to reinvent the wheel

Solution:

- Google Cloud Messaging (GCM), which makes interfacing with Android devices easier

5. DESIGN CHOICES FOR LOCATION SERVICE

<table>
<thead>
<tr>
<th>Location Provider</th>
<th>Battery Consumption</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS Provider</td>
<td>Consumes more battery power</td>
<td>Provides the most accurate location</td>
</tr>
<tr>
<td>Network Provider</td>
<td>Consumes less battery power</td>
<td>Provides less accurate location</td>
</tr>
</tbody>
</table>

Our solution uses **GPS provider:**

- It gives more accurate results
- For hazardous weather, people outdoors must be warned => GPS is the right strategy to leverage
- Disable location updates when not needed to conserve battery

Our solution uses **Android Location APIs:**

- They enable apps to utilize a device’s location
- They shield developers from the underlying technology
- They help to minimize power consumption

6. DESIGN CHOICES FOR CLOUD-BASED APP SERVER

Google Cloud Messaging (GCM) cannot understand the semantics of a weather alert or know whom to notify => designed a cloud-based server using Amazon Web Services that can track all the registered devices and current alerts.

- Apache HTTP Server
- MySQL
- PHP

Design Considerations:

- Apache HTTP Server
- MySQL
- PHP

- Design Considerations:
  - Open source & cross-platform
  - Interfaces easily with MySQL and Apache HTTP

Functions:

- Handling requests from devices
- Database connections
- Database queries
- Data manipulations
- Sending push notification requests to GCM server

7. ANDROID APP USER INTERFACE

A user clicks the message on devices’ respective App servers

8. ONGOING & FUTURE WORK

- Eliminating False Alarms:
  - Allow a user to define a radius around a possible hazardous weather location to be notified if a user is located within the radius
  - Understand the dynamics of the event (e.g., tornadoses may change direction)

Study of Design Usage and Applicability:

- Build instrumentation platforms into the app to facilitate the tracking of usage and to gain powerful insights into:
  - Possible design improvements
  - Design portability
  - Adaptation of the app to serve other societal needs

9. LESSONS LEARNED

- Free the app from querying a server for content
- Patterns of reuse:
  - E.g., using Wi-Fi provider instead of GPS provider for an earthquake system to better alert people who are inside so they can get outside

Notification alternatives:

- Urban Airship
- Xtify
- Parse

Dealing with accidental complexities:

- Too much hand-written glue code => need for generative mechanisms (e.g., model-based design)

Integration challenges and dealing with evolution:

- Maintenance challenges
- A threat of vendor lock-in

Supporting the app from app store:

- Requires large-scale experiments to study design scalability