Caché: Caching Location-Enhanced Content to Improve User Privacy

Carnegie Mellon University

Shahriyar Amini, Janne Lindqvist, Jason Hong
Jialiu Lin, Eran Toch, Norman Sadeh
Widespread Adoption of Location-Enabled Devices

- 2009: 150M GPS-equipped phones shipped
- 2014: 770M GPS-equipped phones expected to ship (~5x increase!)
- Future: Every mobile device will be location-enabled (GPS or WiFi)

[Berg Insight 2010]
Apps Reveal Private Information

- App reveals:
  - Time of Use
  - User Interest
  - Current Location

- Over time:
  - Mobility Behavior
  - Significant Locations
  - Socioeconomic Status?
Our Approach

- Pre-Fetch content for large regions
- Store content on mobile device
- Determine location using GPS/trusted source
- Respond to queries using stored content
- Periodically update content
Pre-Fetching Insight

• Some location-based content are still useful even when old (time to live)

<table>
<thead>
<tr>
<th>Update Rate</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-Time (STTL)</td>
<td>Traffic flow, parking spots e.g. Loopt, PeopleFinder, Reno, Bustle</td>
</tr>
<tr>
<td>Daily</td>
<td><strong>weather forecasts</strong>, social events, coupons e.g. Dede</td>
</tr>
<tr>
<td>Weekly</td>
<td><strong>movie/theatre schedules</strong>, advertisements, crime rates e.g. Yelp!, GeoNotes, PlaceIts, PlaceMail</td>
</tr>
<tr>
<td>Monthly</td>
<td>restaurant guides, <strong>bus schedules</strong>, geocaches e.g. Wikipedia (geo-tagged pages)</td>
</tr>
<tr>
<td>Yearly</td>
<td><strong>maps</strong>, points of interests, tour guides, store locators e.g. Google Maps, Starbucks, Wal-Mart</td>
</tr>
</tbody>
</table>
Feasibility of Pre-Fetching

• Content doesn’t change too often
  – Average daily amount of change over a 5 month period

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Added %</th>
<th>Removed %</th>
<th>Modified %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weather</td>
<td>25.00</td>
<td>25.00</td>
<td>67.26</td>
</tr>
<tr>
<td>Events</td>
<td>5.28</td>
<td>5.35</td>
<td>11.75</td>
</tr>
<tr>
<td>Yelp POI</td>
<td>0.15</td>
<td>0.06</td>
<td>0.04</td>
</tr>
<tr>
<td>MSN POI</td>
<td>6.69</td>
<td>6.80</td>
<td>1.43</td>
</tr>
<tr>
<td>Bus Schedule</td>
<td>0.00</td>
<td>0.00</td>
<td>0.15</td>
</tr>
<tr>
<td>Map Tiles</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

• Requires <20 MB for Pittsburgh, ~100 MB for NYC
Pre-fetching Content

- Geo-coordinates are continuous
- Content cannot be pre-fetched for every point
- Use a grid to discretize space
Caché Architecture
Application Design

- REST-based content
- Developer defines:
  - Size of cells
  - Content update rate
  - Query string

http://api.yelp.com/v2/search?term=food&ll=#SLL_LAT#, #SLL_LON#
Application Installation

• Regions of interest:
  – 15213
  – Pittsburgh, PA

• Pre-fetch radius
  – 1 km
Content Download

- Pre-fetch only when:
  - Plugged in
  - Connected to WiFi

- Pre-fetch every cell

- Update content at defined update rate
Content Retrieval

- Assume fresh content
- Retrieve content from a single cell
- Content miss results in a live request to LBS
High Content Hit Rate!

• How often will queries be cached?
  – Locaccino: Top 20 people, 460k traces
  – Place naming: 26 people, 118k traces

<table>
<thead>
<tr>
<th>Radius (miles)</th>
<th>Locaccino</th>
<th>Place Naming</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>86%</td>
<td>79%</td>
</tr>
<tr>
<td>10</td>
<td>87%</td>
<td>84%</td>
</tr>
<tr>
<td>15</td>
<td>87%</td>
<td>86%</td>
</tr>
</tbody>
</table>
Caché Android Service

- Android background service for apps
  - Apps modified to make requests to service

<table>
<thead>
<tr>
<th>Application</th>
<th>Original</th>
<th>Added</th>
<th>Removed</th>
</tr>
</thead>
<tbody>
<tr>
<td>mixare</td>
<td>4692</td>
<td>18 (0.4%)</td>
<td>4 (0.1%)</td>
</tr>
<tr>
<td>Panoramio</td>
<td>1268</td>
<td>18 (1.4%)</td>
<td>7 (0.55%)</td>
</tr>
<tr>
<td>Restaurant Request</td>
<td>411</td>
<td>12 (2.9%)</td>
<td>8 (1.9%)</td>
</tr>
</tbody>
</table>

- User specifies home and work locations
- Service only pre-fetches when device is plugged in and connected to WiFi
Limitations

• Doesn’t work for
  – Rapidly changing content (STTL)
  – Apps with client/server interaction (Facebook)
  – Apps with server computation (Navigation)

• Burden falls on the developer
  – Developer has to effectively sweep content

• New regions have to be specified before use
Related Work

• Content Pre-fetching
  – Coda
• Anonymity
  – $k$-anonymity, Spatial and Temporal Cloaking, Privad
• Obfuscation
  – SybilQuery
• Route prediction and caching
  – CacheCloak
Conclusion

*The most private and energy efficient request is the one you don’t make.*

- Tradeoffs: Privacy vs. Utility vs. Cost
- Current solutions present challenges
- Comprehensive privacy solution would rely on several approaches
- Consider development and deployment