Going Viral: Flash Crowds in an Open CDN

IMC 2011 (Short Paper)

Patrick Wendell, U.C. Berkeley
Michael J. Freedman, Princeton University
What is a Flash Crowd?

• “Slashdot Effect”, “Going Viral”

• Exponential surge in request rate (precisely defined in paper)
Key Questions

• What are primary drivers of flash crowds?

• How effective is cache cooperation during crowds against CDNs?

• How quickly do we need to provision resources to meet crowd traffic?
CoralCDN

• Network of ~300 distributed caching proxies
CoralCDN

• Network of ~300 distributed caching proxies

1. Local cache
2. Peer cache
3. Origin fetch
The Data

• Complete CoralCDN trace over 4 years

• 33 Billion HTTP requests

• Per-request logging
  – <Time, URL, client IP, proxy IP, content cached?, …>
Finding Crowds

Source Data
33 Billion HTTP Requests

Crowd Detection
3,553 Crowds

Pruning Misuse
2,501 Crowds
Crowd Sources
### Common Referrers

<table>
<thead>
<tr>
<th>Referrer</th>
<th># Crowds</th>
</tr>
</thead>
<tbody>
<tr>
<td>digg.com</td>
<td>123</td>
</tr>
<tr>
<td>reddit.com</td>
<td>20</td>
</tr>
<tr>
<td>stumbleupon.com</td>
<td>15</td>
</tr>
<tr>
<td>google.com</td>
<td>11</td>
</tr>
<tr>
<td>facebook.com</td>
<td>10</td>
</tr>
<tr>
<td>dugmirror.com</td>
<td>8</td>
</tr>
<tr>
<td>duggback.com</td>
<td>4</td>
</tr>
<tr>
<td>twitter.com</td>
<td>4</td>
</tr>
</tbody>
</table>
## Common Referrers

<table>
<thead>
<tr>
<th>Referrer</th>
<th># Crowds</th>
</tr>
</thead>
<tbody>
<tr>
<td>digg.com</td>
<td>123</td>
</tr>
<tr>
<td>reddit.com</td>
<td>20</td>
</tr>
<tr>
<td>stumbleupon.com</td>
<td>15</td>
</tr>
<tr>
<td>google.com</td>
<td>11</td>
</tr>
<tr>
<td>facebook.com</td>
<td>10</td>
</tr>
<tr>
<td>dugmirror.com</td>
<td>8</td>
</tr>
<tr>
<td>duggback.com</td>
<td>4</td>
</tr>
<tr>
<td>twitter.com</td>
<td>4</td>
</tr>
</tbody>
</table>
## Common Referrers

<table>
<thead>
<tr>
<th>Referrer</th>
<th># Crowds</th>
</tr>
</thead>
<tbody>
<tr>
<td>digg.com</td>
<td>123</td>
</tr>
<tr>
<td>reddit.com</td>
<td>20</td>
</tr>
<tr>
<td>stumbleupon.com</td>
<td>15</td>
</tr>
<tr>
<td>google.com</td>
<td>11</td>
</tr>
<tr>
<td>facebook.com</td>
<td>10</td>
</tr>
<tr>
<td>dugmirror.com</td>
<td>8</td>
</tr>
<tr>
<td>duggback.com</td>
<td>4</td>
</tr>
<tr>
<td>twitter.com</td>
<td>4</td>
</tr>
</tbody>
</table>
# Common Referrers

<table>
<thead>
<tr>
<th>Referrer</th>
<th># Crowds</th>
</tr>
</thead>
<tbody>
<tr>
<td>digg.com</td>
<td>123</td>
</tr>
<tr>
<td>reddit.com</td>
<td>20</td>
</tr>
<tr>
<td>stumbleupon.com</td>
<td>15</td>
</tr>
<tr>
<td>google.com</td>
<td>11</td>
</tr>
<tr>
<td>facebook.com</td>
<td>10</td>
</tr>
<tr>
<td>dugmirror.com</td>
<td>8</td>
</tr>
<tr>
<td>duggback.com</td>
<td>4</td>
</tr>
<tr>
<td>twitter.com</td>
<td>4</td>
</tr>
</tbody>
</table>
CDN Caching Strategies
Cooperation in Caching

Greedy Caching

Fully Cooperative Caching
Benefits of Cooperation?

• Depends how clients distribute over proxies

• Depends how many objects a crowd contains
Clients Use Many Proxies

- Clients globally distributed, even during crowds
- Most caches participate in most crowds

![Graph showing the distribution of clients and proxies involving in crowds with a note: Very few large, concentrated crowds]
Crowds Contain Many Objects

![Bar Chart]

- [0,10): 348 URLs
- [10,100): 708 URLs
- [100,1000): 766 URLs
- [1000,10000): 548 URLs
- 10,000+: 131 URLs
Benefits from Cooperation

56% of crowds: some improvement

40% of crowds: major improvement

Absolute Hit Rate Improvement
Provisioning Resources For Crowds
Examples of Resource Provisioning

• CDN: static content
  – Expand cache set for particular domain
  – $\Omega$(Seconds)

• Cloud Computing Platform: dynamic service
  – Spin up new VM instances
  – $\Omega$(Minutes)

• If you squint, these are similar problems
# Required Resource Spin-up Time

<table>
<thead>
<tr>
<th>Spin-up</th>
<th>% Crowds Underprovisioned</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 Minutes</td>
<td>75%</td>
</tr>
<tr>
<td>1 Minute</td>
<td>50%</td>
</tr>
<tr>
<td>10 Seconds</td>
<td></td>
</tr>
</tbody>
</table>

1-2 Minutes on EC2
Conclusions

• What are primary drivers of flash crowds?
  – Aggregators and portals, but also social/search

• How effective is cache cooperation during crowds against CDNs?
  – Large benefit for 40% of crowds

• How fast do we need to provision resources during crowds?
  – Likely require sub-minute responsiveness
Questions?

cs.berkeley.edu/~pwendell
Extra Slides / Charts
Actual Spin-up Times on EC2

![Cumulative Fraction of Spin-Ups vs. VM Spin-Up Time](chart.png)
How Fast is Fast?

Fraction of Crowds > Increase

Largest Minute-over-Minute Increase (req/min)

- 60 min
- 30 min
- 10 min
- 5 min
- 2 min
- 1 min
Origin Hits Saved by Cooperation

Fraction of Requests Served from Cache vs. Unique URLs in Crowd (Normalized)

- Cooperative Caching
- Non-Cooperative Caching

(Cooperative - Non-Cooperative)
Bursty Redirection

![Bar chart showing bursty redirection for bostonherald.com](chart.png)
Detecting Crowds

1. Rapid surge in request rate
   \[ r_{i+1} > 2r_i \quad \text{for several } i \]

2. High rate of traffic relative to inferred capacity
   \[ r_{\text{max}} > r_{\text{avg}} \times 20 \]
Crowd Mitigation/Insurance

Content Mostly Static

- Level 3 Communications
- Limelight Networks
- Akamai

Caching CDNs

Content Mostly Dynamic

- Amazon Web Services
- Rackspace Hosting
- Windows Azure

Scalable Storage and Computation