Siskin: Leveraging the Browser to Share Web Content in Disconnected Environments

Sam Sudar$^{1,2}$, Matt Welsh$^1$, Richard Anderson$^2$

1 Google, Inc
2 University of Washington
Disclaimer

- PhD work from Google internship
- github.com/srsudar/SemCache
There is a huge opportunity for the web to improve education in low-resource settings.

Connectivity in these schools is often poor and intermittent.
Offline Educational Resources
Offline educational resources – crucial for African schools in the years to come

Assessing the use of technology and Khan Academy to improve educational outcomes

Digital Library Appropriation in the Context of Sub-Saharan Countries: the Case of eGranary Digital Library Implementation

OERs taking Schools from Resource Poor to Resource Rich

From customized learning programs to offline apps, how children are learning inside and outside the classrooms
Not Perfect

- Can be expensive
  - RACHEL-Pi: $169
  - External USB eGranary: $2,000 + shipping
  - eGranary server: $3,700 + shipping
- Standalone device
  - Updates aren’t built into the equation
  - sysadmin without UI
  - If it breaks you need special training to fix it
  - Single point of failure
Assumptions

- Set of devices on a local WiFi network
- Intermittent or slow connection to the web

Goals

- Allow users to save local copies of webpages
- Discover web content from peers as a distributed cache
- No sysadmin required
- Build as a browser extension so there is no additional hardware
Research Question

Can we leverage web and browser technology to create a sysadmin-free, configuration-free, decentralized web cache to enable offline access to the web in a seamless fashion?
Browser-based approach

- No standalone device
- One-click install model
- Updates automatically (when available)
- No configuration
- No sysadmin
- Built for existing infrastructure
- Decentralized
Normal Web Model
Conventional OER Solutions
Siskin without Connectivity
Making Siskin Work

- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer
Making Siskin Work

- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer
Saving Pages

- MHTML
- Page as single file
- Good, not perfect
Making Siskin Work

- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer
Naively, how do we find peers?
Multicast DNS
- Network-local DNS

DNS-Based Service Discovery
- Use DNS as a database

Zero-conf: ‘it just works’
Review: Regular IP

172.28.7.80 Goes to **one** device
Review: Multicast IP 224.0.0.*

224.0.0.* Goes to all devices
Review: DNS

A? www.google.com

A 216.58.216.164

Response comes from remote server
Multicast DNS (mDNS)

A? www.google.com

A 216.58.216.164

Response comes from device on local network
DNS-Based Service Discovery (DNS-SD)

Tyrion

172.28.7.90:8888
DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

PTR

Arya

Jon

Tyrion
DNS-Based Service Discovery (DNS-SD)

PTR | Arya.siskin.tcp
---|-----------------
PTR | Tyrion.siskin.tcp

172.28.7.90:8888
DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

<table>
<thead>
<tr>
<th>PTR</th>
<th>Arya.siskin.tcp</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR</td>
<td>Tyrion.siskin.tcp</td>
</tr>
</tbody>
</table>

SRV?
Tyrion.siskin.tcp
DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

<table>
<thead>
<tr>
<th>PTR</th>
<th>Arya.siskin.tcp</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR</td>
<td>Tyrion.siskin.tcp</td>
</tr>
<tr>
<td>SRV</td>
<td>tyrion.local 8888</td>
</tr>
</tbody>
</table>

Arya
Jon
Tyrion
DNS-Based Service Discovery (DNS-SD)

PTR Arya.siskin.tcp
PTR Tyrion.siskin.tcp
SRV tyrion.local 8888

172.28.7.90:8888
DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

<table>
<thead>
<tr>
<th>PTR</th>
<th>Arya.siskin.tcp</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR</td>
<td>Tyrion.siskin.tcp</td>
</tr>
<tr>
<td>SRV</td>
<td>tyrion.local 8888</td>
</tr>
</tbody>
</table>

Arya
Jon
Tyrion
DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

<table>
<thead>
<tr>
<th>PTR</th>
<th>Arya.siskin.tcp</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR</td>
<td>Tyrion.siskin.tcp</td>
</tr>
<tr>
<td>SRV</td>
<td>tyrion.local 8888</td>
</tr>
<tr>
<td>A</td>
<td>172.28.7.90</td>
</tr>
</tbody>
</table>
DNS-Based Service Discovery (DNS-SD)

172.28.7.90:8888

<table>
<thead>
<tr>
<th>PTR</th>
<th>Arya.siskin.tcp</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTR</td>
<td>Tyrion.siskin.tcp</td>
</tr>
<tr>
<td>SRV</td>
<td>tyrion.local 8888</td>
</tr>
<tr>
<td>A</td>
<td>172.28.7.90</td>
</tr>
</tbody>
</table>
Making Siskin Work

- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer
Siskin without Connectivity
Cat - Wikipedia
https://en.wikipedia.org/wiki/Cat

Cached Versions Available
Open original link Go
View local copy Open
Betazoid Get
Cancel
List Content

khanacademy.org/math
nytimes.com/story
wikipedia.org/Cat
Annotate Links: naively

khanacademy.org/math1
nytimes.com/story1

wikipedia.org/Cat

...
Hard on bandwidth

1000 pages * 51 characters = 51 kB
10 peers * (51 * 9) = 4.59 MB
40 peers * (51 * 39) = 79.59 MB
Bloom Filter

- Is this page available?
- Oh, probably. 99.9%

- Is this page available?
- Definitely not.

https://en.wikipedia.org/wiki/Bloom_filter
Better on bandwidth

1,000 pages, 0.001 FP = 1.8 kB
10 peers * (1.8 * 9) = 162 kB
40 peers * (1.8 * 39) = 2.8 MB

80 MB to 3 MB with 40 peers
Making Siskin Work

- Save pages
- Peer discovery
- Distributed cache directory
- Pull data from a peer
Conventional browser-based file sharing
WebRTC
WebRTC
WebRTC

WebRTC?

WebRTC.

172.28.7.90:8888
Transfer Speed

![Graph showing transfer speed comparison between WebRTC and WebRTC with Signaling for different file sizes. The y-axis represents milliseconds, and the x-axis represents file size in kilobytes. The graph shows significantly higher transfer times for WebRTC with Signaling compared to WebRTC for larger file sizes.]
What did we learn?

- Browser as infrastructure
  - One-click installation
  - Distributed cache
- Augment the natural browsing experience
  - Browse naturally, stay local when possible
- Static snapshots are imperfect
  - Place for improvement
  - Progressive Web Apps potential way forward
Thank you!
Questions?

github.com/srsudar/SemCache