How much can we micro-cache Web pages?

Xiao (Sophia) Wang, Arvind Krishnamurthy, and David Wetherall

University of Washington

How often do you use Web apps v.s. native apps?

How often do you use Web apps v.s. native apps?

Because Web apps are slow compared to native apps

Why are Web apps slow compared to native apps?

	Web apps
Updates	
Layout compilation	
Code compilation	

Why are Web apps slow compared to native apps?

	Web apps
Updates	Just in time
Layout compilation	
Code compilation	

Update from the server for every request which blocks rendering.

Why are Web apps slow compared to native apps?

	Web apps
Updates	Just in time
Layout compilation	Just in time
Code compilation	

Compute layout (HTML + CSS) which blocks rendering.

Why are Web apps slow compared to native apps?

	Web apps
Updates	Just in time
Layout compilation	Just in time
Code compilation	Just in time

Compile and run JavaScript which blocks rendering.

Why are Web apps slow compared to native apps?

	Web apps	Native apps
Updates	Just in time	
Layout compilation	Just in time	
Code compilation	Just in time	

Why are Web apps slow compared to native apps?

	Web apps	Native apps
Updates	Just in time	Explicit; offline
Layout compilation	Just in time	
Code compilation	Just in time	

Update from the server explicitly/offline, which does **NOT** block rendering.

Why are Web apps slow compared to native apps?

	Web apps	Native apps
Updates	Just in time	Explicit; offline
Layout compilation	Just in time	Beforehand
Code compilation	Just in time	

Compute layout beforehand, which does **NOT** block rendering.

Why are Web apps slow compared to native apps?

	Web apps	Native apps
Updates	Just in time	Explicit; offline
Layout compilation	Just in time	Beforehand
Code compilation	Just in time	Beforehand

Compile code beforehand, which does **NOT** block rendering.

Can we make Web apps behave like native apps?

	Web apps	Native apps
Updates	Offline; BG	Explicit; offline
Layout compilation	Beforehand	Beforehand
Code compilation	Beforehand	Beforehand

How can we achieve this behavior?

How can Web apps behave like native apps?

- **Distinguish between data/layout/code**

  → Cache rendered layout & compiled JavaScript

- **Micro-caching**: cache at a finer granularity than object-based caching

  → Fetch only differences from the network, in the background

How can we implement the techniques?

- Distinguish between data/layout/code
  - Modify Web pages to make this distinction
  - Modify browser to cache computed layout and compiled JavaScript at fine granularity
- Micro-caching
  - Modify Web pages: store page fragments in localStorage; assemble fragments using JavaScript

Web page (before)

Index.html (ephemeral cache)

```
<html>
 <head>
  <style> … </style>
  <script> … </script>
 </head>
 <body>
  <div> … </div>
  <table> … </table>
 </body>
</html>
```

Web page (after)

Index.html (long-term cache)

```
<html>
  <script>
    if (!hasPageInLocalStorage)
      fetchRemotely();
    loadPageFromLocalStorage();
  </script>
</html>
```

Infrequently happens

Web page contents stored in localStorage; HTML provides an entry point

We examine the effectiveness

How much can we micro-cache Web pages?

- Identify differences between two versions of a Web page
- Infer whether a difference belongs to data, layout, or code

Methodology

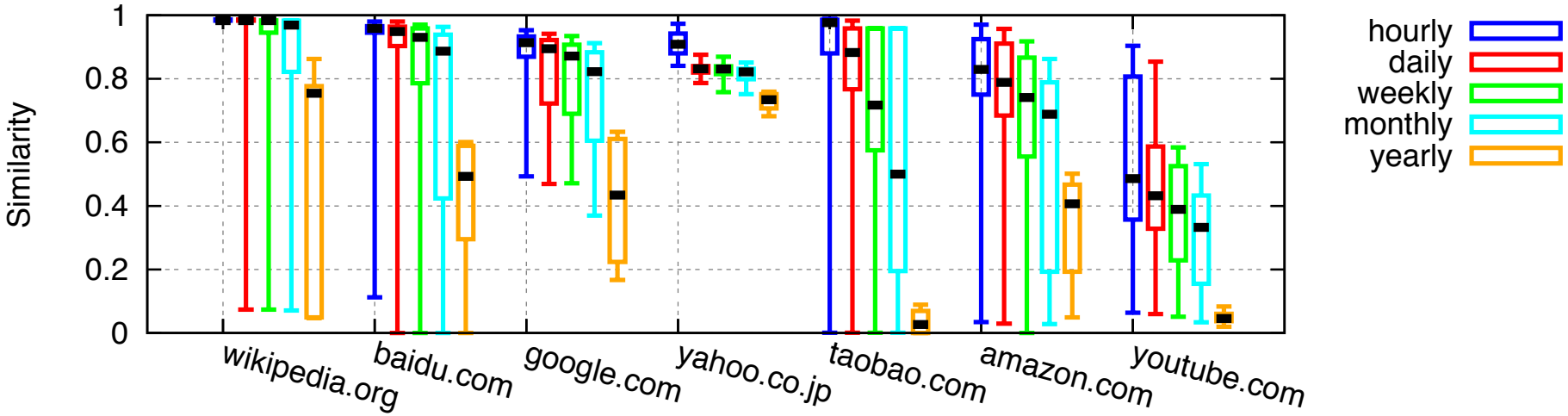
- Identify differences between two versions of a Web page
  - Diff-like analysis
- Infer whether a difference belongs to data, layout, or code
  - Context inference

Datasets

- 7 top desktop pages over two years, fetched per hour
  - Content rich/scarce
  - Geographically distributed
  - 5 categories
- Their mobile counterparts over a month

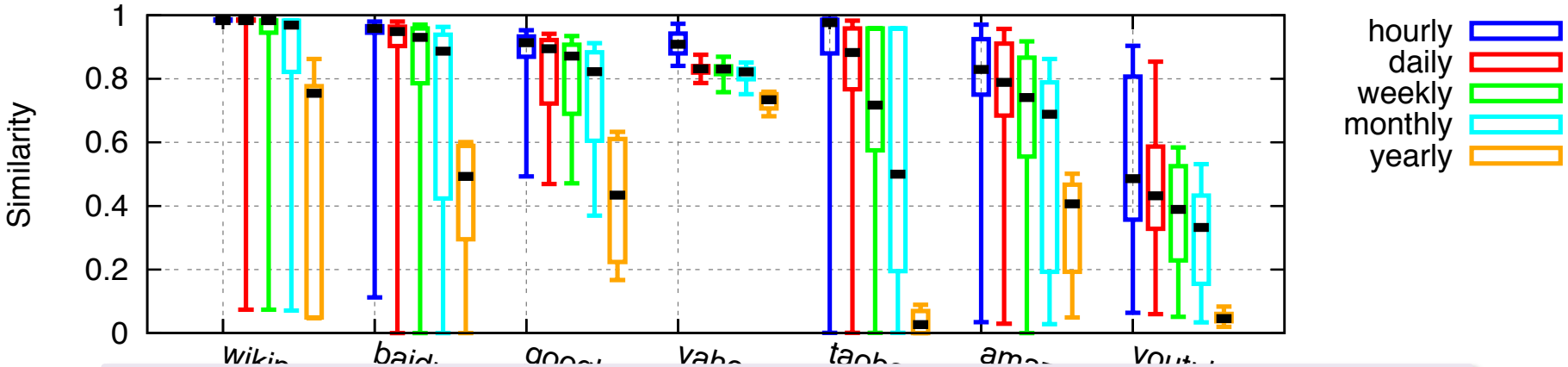
How much/often is a Web page being updated?

- Similarity: % matched lines (lower bound)



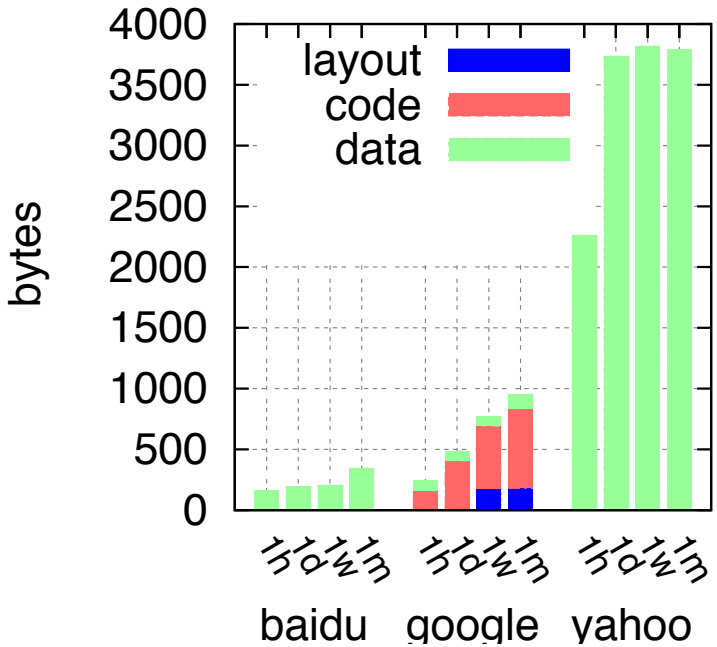
How much/often is a Web page being updated?

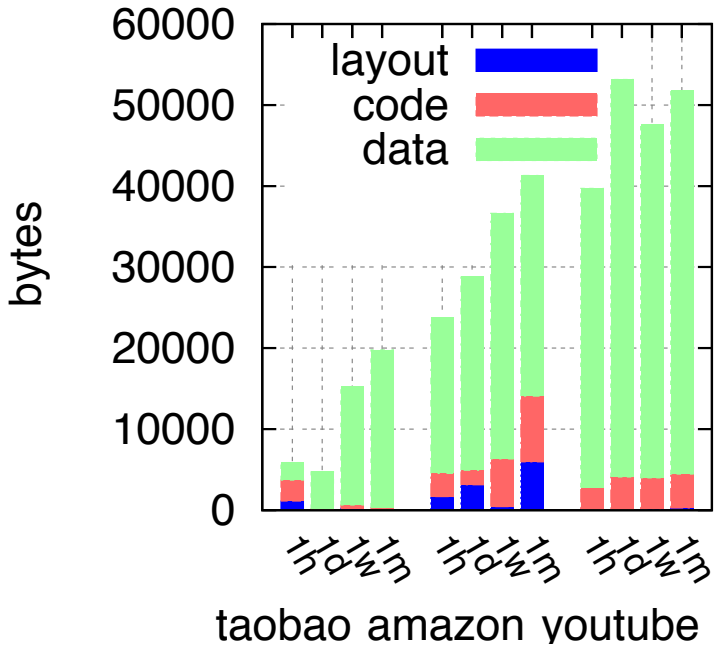
- Similarity: % matched lines (lower bound)



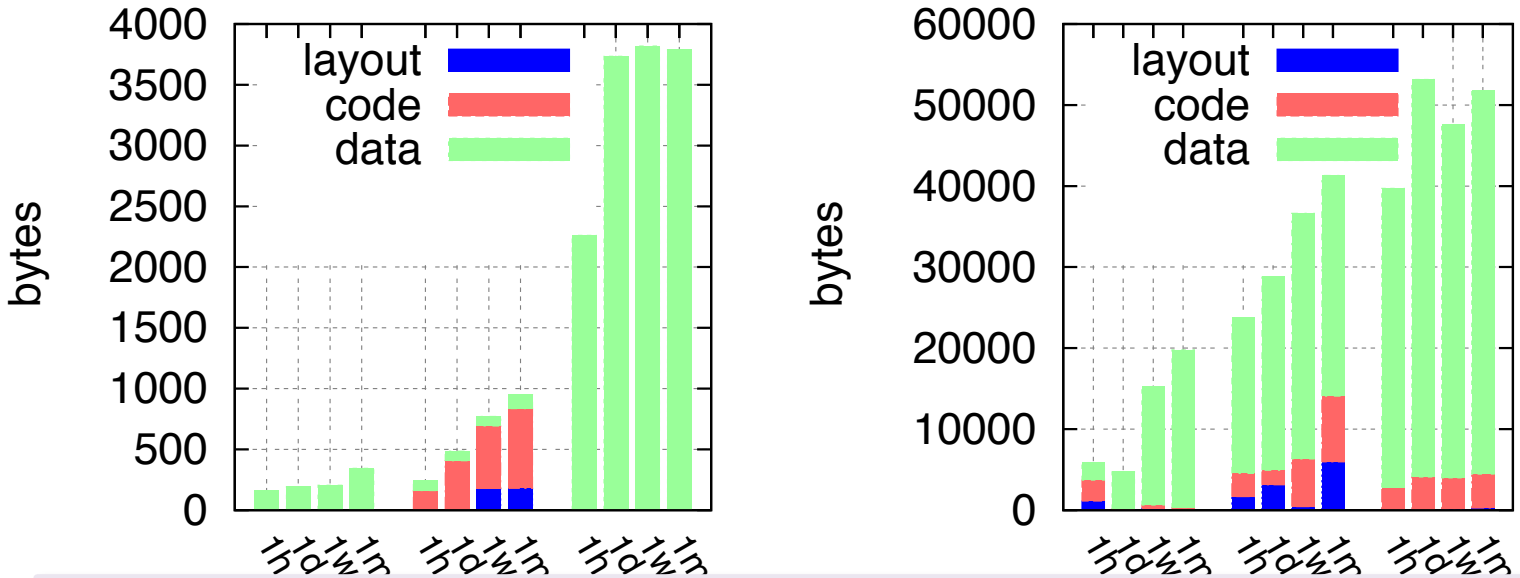
Less than 20% (10%) of content-scarce pages changed after a month (day).

What are the updates?





What are the updates?



Most updates are made to data. Layout and code are highly micro-cacheable.

Conclusions

- In order to improve page load times, we propose to separately cache data/layout/code at a fine granularity inside browsers
- Our measurements show that layout and code are highly micro-cacheable. Micro-caching significantly reduces traffic.
- Our future work will focus on the implementation, which will allow us to measure the improved page load times.