Jawa: Web Archival in the Era of JavaScript

Ayush Goel, Jingyuan Zhu, Ravi Netravali,
and Harsha V. Madhyastha

Objective: High fidelity and low cost web archive

Ephemeral web and link rot ➡ Web Archives

RESEARCH ARTICLE

Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content

Shawn M. Jones[1☯*], Herbert Van de Sompel[1☯], Harihar Shankar[1☯], Martin Klein[1☯], Richard Tobin[2‡], Claire Grover[2‡]

DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD
DASH.HARVARD.EDU

HARVARD LIBRARY Office for Scholarly Communication

The Paper of Record Meets an Ephemeral Web: An Examination of Linkrot and Content Drift within The New York Times

UKWA UK WEB ARCHIVE

LIBRARY OF CONGRESS

Problems with Web Archives

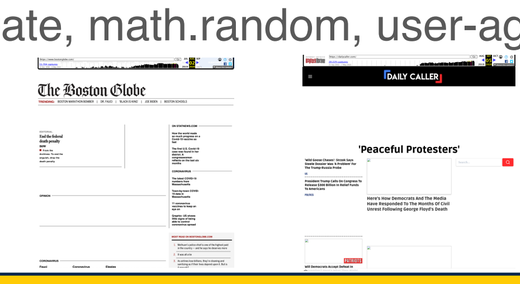
Limited page snapshots

2010-2020: **4x** increase in **JavaScript**

Per page snapshot expensive

Poor Page Fidelity

JavaScript non-determinism: Date, math.random, user-agent



Improve fidelity: Fix sources of JavaScript variation

Randomness (date, math.random, performance)

Insight	Solution
1-1 mapping between diverged URLs	*Use server-side matching techniques*

a.com/b.js?ts=5467 ↕ a.com/b.js?ts=8967

Client characteristics

Insight	Solution
Key contributor to URL divergence.	*Enforce same values across loads*

Crawling device ↕ Replay device

APIs:
- User-agent
- Screen size
- Operating system
- Geolocation
- ….

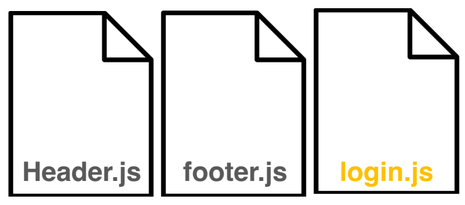
Reduce storage overhead: Remove non-functional code

No back-end origin server

Remove code that interacts with back-end server

Insights

1. Compartmentalized into files



2. Hosted on third-party domains

Solution

Filter list to identify URLs of non-functional scripts

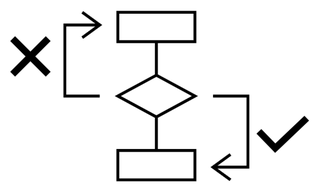
Rules:
- "cdn.onesignal.com"
- "cdn.pushly.com"
- "cdn.parsley.com"
- …

Reduce storage overhead: Remove unused code

Remove (unused) alternate code flows

Insight: Absent sources of non-determinism:
- Server-side state
- Client APIs return values
- Client-side state

Solution: Dynamically identify used code (during page load)



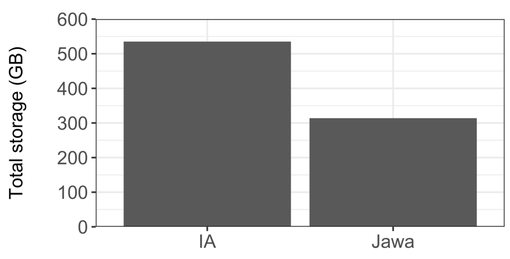
Preserve code for event handlers

Insight	Solution
Order of execution, and inputs to events do not impact code coverage	*Trigger each event once with default inputs* + *Track executed lines of code*

Results

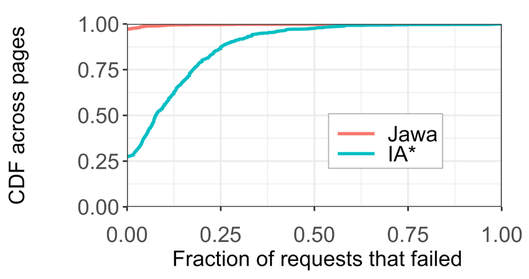
Storage

41% reduction in total storage overhead



Fidelity

Eliminate almost all failed network fetches



Crawling throughput

Improved throughput by 31%