# Cache-Control: A Survey in Fast Delivery Across the CPU and the Web

## Background

> *"There are only two hard things in Computer Science: cache invalidation and naming things."*
>
> – Phil Karlton

In the digital age, the paradigm of caching—preparing high-latency data ahead of time, ensuring it can be retrieved instead locally at low latency—is a problem of paramount importance. It is notoriously difficult to do well, yet absolutely essential in scenarios where data access has unacceptably high latency. In the CPU, where uncached memory reads cost precious clock cycles compared to reads from registers or caches, the answer results in a complex architecture of multi-level caches and specialized hardware for predicting memory accesses ahead of time—in some cases even involving hardware-based perceptrons for better branch prediction. On the web, caching has led to a huge economy of Content Delivery Networks (CDNs), where webpages are copied and hosted across the globe, as close as possible to the end-user, to prevent lengthy network calls back to the origin server. This is not to mention the several other levels of web caching found in browsers and on the origin servers themselves. In both cases, the end goal is the same: minimize the distance and time between the user and the requested data.

PROPOSAL

# Proposal

This senior project aims to act as a survey comparing techniques used across the two domains, offering contrasts and making accessible insights from the intersection of the two. The academic literature on CPU caching is extensive, and is integral to improving performance on the hardware level, for all processes performed by a computer. The results of CPU caching are implemented in a proprietary manner by Intel, AMD, and other chip companies alike, often differing even per-CPU as hardware permits. Web caching is similarly documented, though the higher intrinsic latency of the web leads to simpler requirements and implementations for caches. CDNs like CloudFlare use proprietary, private eviction policies for their global distributed caches (often the object of much frustration on the behalf of the web developer, mind you), whereas browser caches are often publicly documented and implemented, as seen in engines like Chromium and Webkit.

# Deliverables

The project will result in the following deliverables:

1. A survey paper comparing findings across the web and the CPU as described above. This will include a small suite of tests comparing results of caching at different levels on the web: browser, CDN, and origin server, described alongside their CPU-level analogies.
2. A supplemental website outlining the findings and comparisons with additional tools for web developers in debugging HTTP headers for proper multi-level cache control—one of the most integral yet hard-to-conceptualize issues facing websites today.
3. If time permits, a rudimentary implementation of a CDN cache with branch prediction for the web—scanning documents for internal anchors and preparing the cache with them will be provided.

# Schedule

The proposed schedule for my project is below.

| | |
|---|---|
| Week 1 | Collect project research in CPU caches. Collect reference material and conduct interviews on SoTA CPU cache architectures from Yale professors. |
| Week 2 | Collect project research in web caches—CDNs, server-side caches, and browser caches (Chromium, WebKit). Outline similarities and differences with CPU caches. |
| Week 3 | Compose initial findings into basic paper structure. Continue collecting references, research, and interviews. |

| | |
|---|---|
| Week 4 | Set up basic testing environment for web cache timings. Run basic timing tests for different levels of web caching. |
| Week 5 | Compile reports together into data section of paper. |
| Week 6 | Add finishing touches to paper, collect further references and readings. |
| Week 7 | Create supplemental website compiling results of paper. |
| Week 8 | Build small Cache-Control header testing tool bundled into deployment of the website. Add additional hyperlinks for further reading on the website. |
| Week 9 | Final QA of paper and website. Receive feedback from professors and iterate for final product. |
| Week 10 | Design and print poster compiling results. |

The goal of this project is to ultimately provide context to web developers on why caching is important and supplement the traditional web development process with better tooling for debugging caches—hence the website and header testing tool.