CS787: Advanced Algorithms	**Scribe:** Shashank Rajput
Lecture 13: Caching	**Date:** 03/12/2019

13.1 Recap

13.1.1 Caching

Goal: Minimize number of cache misses in a cache of size k.

Theorem 13.1.1 *Deterministic setting:*

1. *No deterministic eviction policy can achieve competitive ratio* $< k$.
2. *1-bit LRU achieves c.r.* $= k$

Theorem 13.1.2 *Randomized setting:*

1. *No randomized policy can achieve c.r.* $< \log k$. *(Proved in last lecture)*
2. *Marker, a.k.a. randomized 1-bit LRU achieves c.r.* $= \mathcal{O}(\log k)$ *(Will be proved in this lecture)*

13.1.2 Marker

The following is the algorithm for Marker:

- Initialize bit (mark) for every page in cache to 0.
- When new page arrives
  - If in cache, set bit = 1.
  - If not in cache, evict uniformly at random an unmarked page and set bit = 1.
  - If all pages marked, reset all bits = 0. *This is start of a new "phase".*

13.1.3 Offline optimal algorithm

The offline optimal algorithm evicts the farthest in future page (FIF).

13.1.4 Analysis of LRU

OPT evicts atleast one page per phase as shown below:

$$\sigma = \underbrace{[2,\overbrace{3,1,\ldots]}^{\text{OPT miss}}}_{\text{Phase 1}}\underbrace{\overbrace{[4,}^{}\overbrace{2,3,\ldots]}^{\text{OPT miss}}}_{\text{Phase 2}}\underbrace{\overbrace{[1,}^{}2,3,\ldots]}_{\text{Phase 3}}\ldots$$

In the shifted phases, OPT evicts atleast one page.
$\implies$ OPT evicts atleast $\rho - 1$ pages in total in ρ phases.
LRU evicts at most k pages per phase $\implies$ it evicts atmost ρk pages in ρ phases.

13.2 Analysis of Marker

Let m_i = # of pages accessed in phase i that are not in cache at the start of this phase.
$\implies$ # of distinct pages accessed in phases $i-1$ and $i = k+m_i$. This is because there are k distinct page accesses in phase $i-1$ and m_i new page accesses in phase i.
$\implies$ $n_{i-1} + n_i \geq m_i$ where n_i = # of misses for OPT in phase i. This is because there can be atmost k distinct pages in cache.

$$\implies \text{OPT} \geq \frac{\sum_i m_i}{2} \tag{13.2.1}$$

Claim 13.2.1 *Expected # of misses for marker in phase $i \leq m_i H_k$, where $H_k = \sum_{i=1}^{k} \frac{1}{i}$ is the k^{th} Harmonic number.* ***NOTE:*** $\log k \leq H_k \leq \log k + 1$

Proof: Firstly we see that the worst case would be when the new m_i pages arrive at the start of the phase. This is because this gives a chance that an old page gets evicted which is later requested in the same phase. Thus, as we will see we can get misses even after all the m_i pages have arrived. On the other hand if the new pages arrive later, the first few requests would be cache hits. Thus, in the worst case we would want the new pages to arrive at the beginning of this phase.
Now, all m_i pages have arrived. Note that all these are marked 1. Thus, we will not evict these. Now, in the worst case none of these m_i pages are requested again in this phase. Thus, we assume only the old pages are requested. At time step $m_i + 1$ since the beginning of this phase, we get an old page and the probability that it was evicted is $\frac{m_i}{k}$. Now, if another old page comes, the probability that it got evicted by this time is $\frac{m_i}{k-1}$ because the previous old page is now in cache. Similarly, it can be seen that the probability of a miss when the $x + 1^{th}$ page is accessed for the first time in this phase $= \frac{m_i}{k-x}$. The phase ends at $x = k - m_i - 1$. Thus,

$$\mathbb{E}[\text{total \# of misses}] \leq \sum_{x=0}^{k-m_i-1} \frac{m_i}{k-x} \leq m_i \left(\frac{1}{k} + \frac{1}{k-1} + \cdots + 1 \right) = m_i H_k$$

■

Claim 13.2.1 together with Equation 13.2.1 proves Theorem 13.1.2.

13.3 Recent work - What if we have access to predictions?

Example: Let $k = 2$ and we have three pages - a, b, c. We initially have a and b in cache. Consider the following sequence of events and predictions:

- Cache $= a, b$. Now, c arrives.

Predicted sequence = $a, b, c, b, c, b, c, b, c \ldots$. Actual sequence = $b, c, b, c, b, c, \ldots$
According to FIF, we should evict b.

- Cache = a, c. Now, b arrives.
  Predicted sequence = $a, b, c, b, c, b, c, b, c \ldots$. Actual sequence = $c, b, c, b, c, \ldots$
  According to FIF, we should evict c
- This cycle repeats

We see that bad prediction led to a cache miss every time. ∎

13.3.1 Marker with prediction (MwP)

The following is the algorithm for Marker with prediction:

- Initialize bit (mark) for every page in cache to 0.
- When new page arrives
  - If in cache, set bit = 1.
  - If not in cache, evict ~~uniformly at random~~ an unmarked page that is FIF according to the prediction and set bit = 1.
  - If all pages marked, reset all bits = 0.

Claim 13.3.1 *If the predictions are prefect, Marker with prediction has at most m_i misses in phase i.*

We define a chain of evictions $P_1 \to P_2 \to P_3 \to \ldots$ which starts when a new page arrives (that is P_1 is a new page) and $P_i \to P_{i+1}$ if P_{i+1} gets evicted when P_i arrives. Therefore,

$$\#\text{ misses in MwP } = \sum_{\text{new pages}} (\text{length of eviction chain of the page})$$

Also note that every old page can belong to at most 1 chain. If it causes a cache miss, then this page had a misprediction.

Claim 13.3.2 *# misses $\leq m_i$ + # mispredictions. Here, # mispredictions is the edit distance between true and predicted sequences*

Therefore, ALG $\leq$ 2OPT + Edit distance. This is because OPT $\geq m_i/2$.
But, this can be really bad if the predictions are bad!
Idea: Keep track of all chains to ensure their lengths $\leq \log k$

13.3.2 Marker with prediction 2

The following algorithm handles bad predictions:

- Initialize bit (mark) for every page in cache to 0.

- When new page arrives
  - If in cache, set bit $= 1$.
  - If not in cache,
    - If chain containing this page is of length $< \log k$ then evict FIF unmarked page
    - Else evict random unmarked page
  - If all pages marked, reset all bits $= 0$.

In this case, $\mathbb{E}[\text{length of any chain}] \leq 2 \log k$. Therefore, ALG $\leq 4 \log k$ OPT.

13.4 Next lecture: Online Bipartite Matching

Setup: One side is known (offline), the vertices in the other side arrive one by one (online).