Lecture 23

Randomized LRU Caching : I

October 6 2016

1 Observations about LRU

In order to understand *Randomized LRU Caching* we need to observe some important points related to LRU (Least Recently Used) Caching strategy.

Claim 1 : No matter which On-line Algorithm we follow, there will always be a cache miss at every step for a custom tailored sequence.Given that Cache Size **k** and number of letters **l** satisfies the relation **k = l - 1**.

Proof : Say our custom tailored sequence is **S** (Custom tailored sequence means that sequence which is build according to individual specifications or needs). Here say at any stage we will add that element to S out of l letters which is not present in our cache memory. Since at any stage there are k letters in our cache memory and we are aiming at maximum cache miss, We have total of l(=k+1) letters to choose one but to make sure a cache miss, we choose a letter apart from k letters currently present in cache memory. Hence at any stage choosing next element of our sequence wisely we ensure cache miss independent of the on-line algorithm followed to reduce cache miss. Hence any on-line algorithm on a sequence (tailored) suffer cache miss at every step.

Claim 2 : Let OALG represents an on-line algorithm and OPT represent most optimum algorithm ie.. FIF (Farthest In Future). Let OALG(S) represents number of cache miss in case of OALG and OPT(S) represents number of cache miss in case of OPT for sequence S and cache size k. Given **k = l -1**. Then,

$$OALG(S) \geq kOPT(S)$$

Proof : Let total number of phases be P in sequence S. Let the phase size be $\phi_1, \phi_2, \phi_3, ..., \phi_P$ of P phases of given sequence S. Since our OALG will have ϕ_i cache miss in i-th phase (from claim 1), Our summation is starting from index 2 because there won't be any cache miss in first phase

$$OALG(S) = \sum_{i=2}^{P} \phi_i$$

From lecture 22,

$$\phi_i \geq k$$

Therefore,

$$OALG(S) \geq \sum_{i=2}^{P} k$$

$$OALG(S) \geq k(P-1)$$

From lecture 22,

$$OPT(S) = P - 1$$

Hence

$$OALG(S) \geq kOPT(S)$$

Instead of doing this above proof we can give simple argument that in each phase our OPT will have only one cache miss and OALG will have number of cache misses equal to phase size and cache size is always greater than equal to cache size, hence for each phase number of cache miss in OALG is greater than equal to k. So for each cache miss of OPT there are at least k cache miss for OALG, hence

$$OALG(S) \geq kOPT(s)$$

Recall the definition of Competitive Ratio as

$$COMPRAT = MAX \frac{ALG(S)}{OPT(S)}$$

From Claim 2 for any on-line algorithm,

$$COMPRAT \geq k$$

In other words it gives an upper bound on how worse can an algorithm perform with respect to performance of the most optimum algorithm

Claim 3 : Competitive Ratio of Least Recently Used (LRU) algorithm is exactly equal to cache size, ie...

$$\frac{LRU(S)}{OPT(S)} = k$$

Proof : Since LRU is an on line algorithm, so from claim 2 competitive ratio for LRU is greater then or equal to k,

$$\frac{LRU(S)}{OPT(S)} \geq k$$

Let P be total number of phases in S, It can easily be observed that there are at most k cache miss per phase in LRU strategy, So for total of P phases (excluding first phase as there won't be any cache miss in first phase) becomes

$$LRU(S) \leq k(P-1)$$

From lecture 22, minimum number of cache misses for most optimum strategy is equal to total phase P - 1,

$$OPT(S) = P - 1$$

Hence

$$\frac{LRU(S)}{OPT(S)} \leq k$$

Therefore,

$$\frac{LRU(S)}{OPT(S)} = k$$