WTiny-LFU C++ - map

Is a frequency-based cache admission policy in order to boost the effectiveness of caches subject to skewed access distributions.

- the cache consists of two areas:
  - **Window cache** is a simple **LRU**
  - **Main Cache** which is **SLRU** cache with an admission policy.
- instead of maintaining a ghost entries we use **approximate counting scheme** to maintain statistics of items frequency with periodic aging.
- Items evicted from the **Window Cache** are candidates to enter the **Main Cache.**
- Default **Window Cache** is **1%** of the cache.

- This algorithm was adapted by multiple open source library and products like Caffeine, Cassandra, neo4j etc. ..

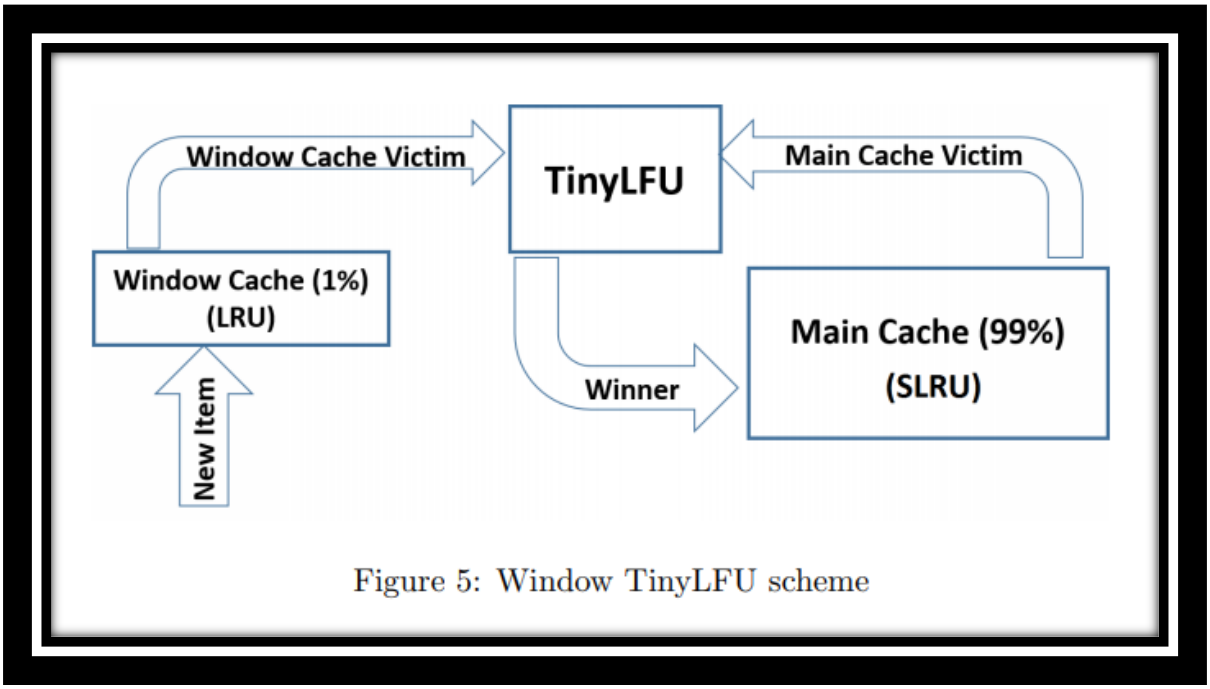


Figure 5: Window TinyLFU scheme

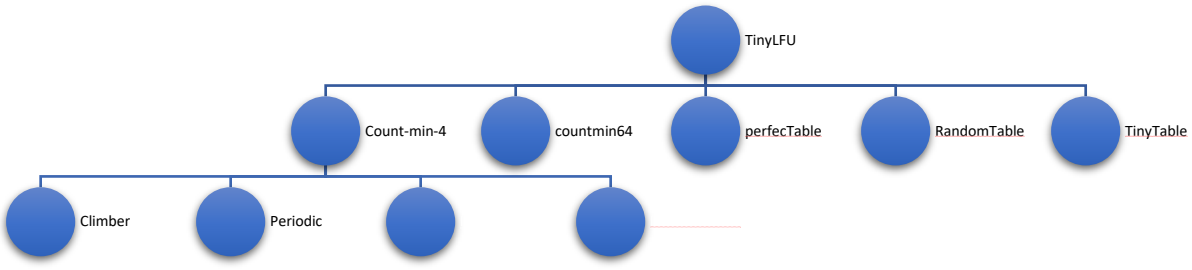
WTLFU++:

We divided the implementation into two parts:

- **Main Cache -** which consists of two parts, TinyLFU and frequency sketch w/doorkeeper, the main class in our implementation is tinyLfu class which consists of calling the wanted approximate sketch needed, easy to change and adding a new policy, by adding a new policy name to the switch.
- **Window Cache** – we implemented the two versions, adaptive hill climbing and the default percent window.
  hill climber type class is where the adaptive algorithms are offered, as tinyLfu is easy to change and add your own algorithm then heading to basicSettings.
- **BasicSettings** – the main class to maintain the project, you could choose what sketch/adaptive version you want, as instructed in the .h file.

  - Don't forget to Add your file in the Cmake to make the wanted change

TinyLfu Class :-



The main simulator working under this hierarchy :-