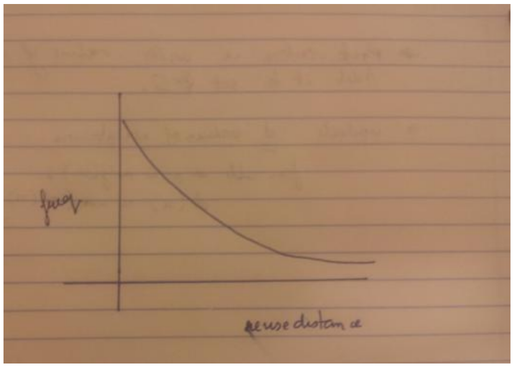
Lecture Scribe

by Nikhil Roy

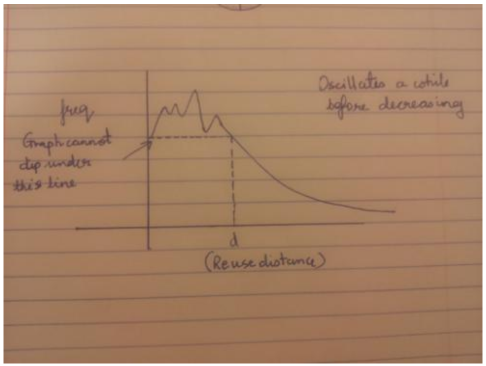
Last class stack Reference PDF (or) Reverse Distance PDF

Strong Locality Theorem



if PDF goes decreasing f() ,
Then LRU is best cache policy replacement as [we can never look ahead]
No matter how large the cache is

Weak Locality Theorem



Cannot dip under this line
up/down initially but later it starts decreasing ,
anything before 'd'

LRU is still the best if cachesize >= d (Among non-lookup ahead)

Packet Classfication is kind of expensive

Suppose you have cache : You only have to lookup for 1st packet

Objective is to treat flow-ids(sourceIP,port Number) as memory addresses so as to observe flows like above.

Programmable Priority Encoders solution are cheaper than barrel shifters

How to measure :

We need an augmented data structure to measure.

Alternative definition of re-use-distance: Naive Algorithm has the output which basically outputs the re-use distance.

When new address comes in we need to look up the stack which takes up O(N) for the given stack space.

Design of datastructure:

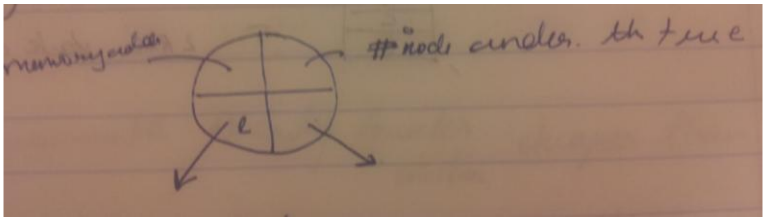
We could use a hashtable

Top of Stack
c <key1> ------><Content>
    []
b<key2> ------><Content>
    []
a<key3>-------><Content>

Use of a balanced Tree to find re-use distance
(Memory Address is not the key)
(No Explicit Key) for this awesome balanced tree

Doing an inorder Traversal of trees

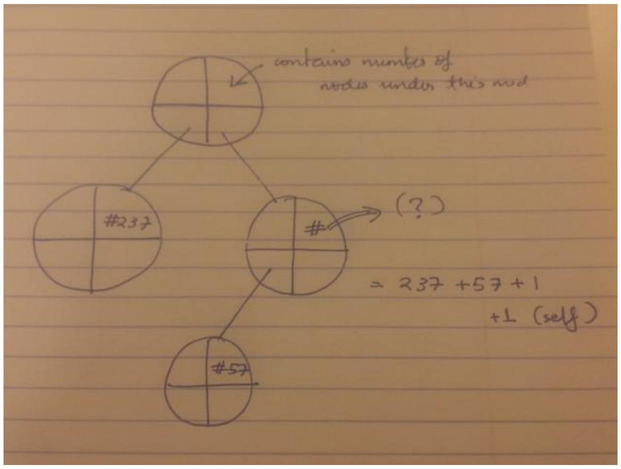
Sample node of the given tree



* Invariant #1 : Inorder Traversal of Tree -- > Order of elements in Fictional Tree

Use of AVL/ Red Black Tree

Example of the how the Tree structure is represented.



*Invariant #2:

number of nodes under the give node is the member