Problem:   
We want to dynamically move frequently used objects to high speed OSDs such as OSDs with SSDs. In general, we are trying to make CRUSH more adaptive and provide a maximum throughput in tiered installations.

Primary solution:

1. Rank the data object:  
   We rank the data objects with its access frequency and the increasing rate of access frequency in update period T1. Each OSD will maintain a list of the rank of data objects stored on that OSD.
2. Move highly ranked objects from non-SSD OSDs to SSD OSDs, without breaking the uniform distribution.  
   SSD OSDs will form a virtual cluster.
   1. Move objects to SSD OSDs:  
      The rule set RuleSetSSD will assign certain number of objects to SSD OSDs, the rest of replicas will be assigned to non-SSD OSDs.  
        
      *SSDRankthreshold = highest threshold of SSD OSDs in Crushmap*  
      *for obj in OSD:  
       if obj.rank > SSDRankthreshold:  
       new\_osd\_list = crush.run(obj, RuleSetSSD)  
       for osd in new\_osd\_list:  
       move replica to osd*
   2. SSD OSDs will record the number of objects being moved from non-SSD OSDs (these objects have a higher rank than *SSDRankthreshold*) and run a routine at a period of T2 to move same number of objects with lowest ranks to non-SSD OSDs.  
      The rule set RuleSetNonSSD will assign all replicas to non-SSD OSDs.  
        
      *num\_to\_remove = len(new\_obj\_lsit)  
      remove\_list = num\_to\_remove objects in OSD with lowest rank  
      for obj in remove\_list:  
       new\_osd\_list = crush.run(obj, RuleSetNonSSD)  
       for osd in new\_osd\_list:  
       move replica to osd  
      threshold = f(min\_rank\_of \_objs)  
      update threshold in Crushmap*
3. Assumptions:
   1. The access frequency of a file is a stationary random process.
   2. OSDs is capable of giving the access time of each object
4. Evaluation metrics:
   1. Throughtput  
      Since we are moving more frequently used objects to faster storage, so the overall throughput should be improved. But according to Amdahl’s Law, the best improvement we can achieve is limited by:
      1. percentage of faster storage devices (how many SSD vs. how many storage in total)
      2. the speed up of faster storage devices (speed of SSD vs. speed of HDD)
   2. Reliability  
      Since SSD suffer a lower reliability than HDD, so if all of an object’s replicas have been placed at SSD OSDs (which may happen a lot in Crush when a storage cluster has a large percentage of SSD OSDs), it may suffer a lower level of reliability.
   3. Data distribution

Crush has done a good job on building an uniform distribution for large number of objects. It is obviously that what we are doing might have some effect on this uniform distribution. So we need to study on this.