Plan for graph based file prediction

* Month 1 (7.9 – 8.6) – Is it a reasonable way.  
  Study the property of static file relation graph first and then study online version of algorithms for evolving graphs.
  + Week 1 (7.9 – 1.16)  
    Study the relation between the threshold and the ssd/hdd ratio
    - How to control the portion of nodes being placed on ssd
    - Are majority of nodes in file relation graph connected?
    - How fast will the algorithm converge? – this determines how effective online algorithms will be and how many communication overheads there will be.
  + Week 2(7.16 – 7.23)  
    Study the property of nodes being placed on ssd, compare results with prediction based on access frequency only and markov model.
    - The performance in terms of system throughput
    - The overhead in terms of data movement
    - Do we need to use self-edges to take access frequency into account?
  + Week 3(7.23 – 7.30)  
    Replace static graph with evolving graph, how graph changes effect placement. Come up with the online version of the algorithm
    - The performance
    - The overhead, especially those introduced by frequent changing of graph
  + Week 4(7.30 – 8.6)  
    Improve the online algorithm.
    - If overhead introduced by graph change are huge, how to reduce them.
    - Is the algorithm capable of catching fast graph changes?
* Month 2 (8.6 – 9.3) – How to make it better  
  Based on Month 1’s result, improve the algorithm, and consider following questions as well.
  + Is it possible to control the communication overhead at a very low level so that a distributed version of algorithm is possible?
  + Extensive comparison between graph based algorithm and markov based algorithm.
  + If we change the graph node from file to chunks, will the graph be much different? If so, how to handle the difference and how will they effect the performance and overhead.
* Month 3 (remaining time) – Paper writing, extensive experiments