Comp 4060 / 7720 - Project Proposal

Lam T. Nguyen & Danika Passler Bates

{nguyenl9, passlerd}@myumanitoba.ca

Problem Definition

In this proposal, we revisit a general abstraction of problems that share the nature of efficient movement and relocation of data. It is best known as the **k-server** problem. It was originally presented in 1990 as a means of generality for the popular page replacement algorithms during the 60s and 70s [1]. In the grand scheme, the k-server problem along with its' abstraction metrical task system, play an important role in the heart of competitive analysis in metric spaces. Through studying the k-server problem, we should encounter other interesting instances captured by the problem. Some of them are similarly studied in depth, namely the aforementioned paging problem and its extensions like weighted paging, others like two-headed disk, k-taxicab, the chasing problem, and many more [1]. The problem has received a lot of attention and research efforts for its real-life applicability. Thus, these factors are among the incentives of our project. Before diving in the algorithms, let us restate the definition of our focus:

Definition. *Given a metric space M that has n points and k servers locating at some of those points. The algorithm receives requests and moves its servers to* serve *the requests accordingly, should there not be any server at the request location. The objective is to minimize the cost, in this case is the total distance travelled by the servers [2].*

Previous Work

The offline version of the k-server problem is best optimized to $O(kn^2)$ where n is the number of requests in a sequence. This is done by reducing it to a minimum cost / maximum flow problem [3]. For the online version, the deterministic lower bound is found to be no less than k, the number of servers in a configuration [4]. The k-server conjecture suggests the existence of a deterministic online algorithm that offers a competitive ratio of k [4]. So far, Koutsoupias and Papadimitriou arrived at a competitive ratio 2k-1 with the Work Function Algorithm [2]. On the other hand, randomization allows the online player more power and hence better results, as discovered by Fiat et al that offers a competitive ratio of $O(logK)$ against an oblivious adversary [4]. When faced with a stronger adversary, however, it falls short. This reveals that the competitiveness of randomized algorithms is rather complicated, best understood through the adversary configurations.

Approach

The findings of previous works offer opportunities for researching new algorithms with various restraints. Our main focus is to survey these algorithms and compare their competitiveness when faced with adversaries with different capabilities. We propose to survey the adversary models of the k-servers problem in the following steps:

- We survey different conjectures of k-server problem, where they fail and succeed.
- We survey various adversaries against algorithms with emphasis on randomized algorithms, in particular oblivious and adaptive adversaries.
- We compare the performance of algorithms under different constraints and draw correlations, providing insights on possible direction for readers with interest for further research.

References

[1] M. S. Manasse, L. A. MacGeoch, and D. D. Sleator, Competitive algorithms for server problems.Pittsburgh, Pennsylvania : Carnegie-Mellon University. Department of Computer Science, 1988.

[2] N. Bansal, N. Buchbinder, A. Madry, and J. (S. Naor, “A Polylogarithmic-Competitive Algorithm for the k-Server Problem,” Journal of the ACM, vol. 62, no. 5, pp. 1–49, 2015.

[3] A. Borodin and R. El-Yaniv, Online computation and competitive analysis. Cambridge, U.K.: Cambridge University Press, 1998.

[4] S. Ben-David, A. Borodin, R. Karp, G. Tardos, and A. Wigderson, “On the power of randomization in online algorithms,” Proceedings of the twenty-second annual ACM symposium on Theory of computing - STOC '90, 1990.