Algorithms

Research Project Proposal

Members:
Abdullah Khalid (ak08428)
Ahtisham Uddin (eu08429)

Instructor:
Waqar Saleem

Paper Details:

- **Title**: Distributed Load Balancing in the Face of Reappearance Dependencies
- **Authors**: Kunal Agrawal, William Kuszmaull, Zhe Wang, Jinhao Zhao
- **Conference**: Proceedings of the 36th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '24)
- **Year**: 2024
- **DOI/Link**: https://doi.org/10.1145/3626183.3659968

Summary: This paper addresses the challenge of distributed load balancing in database systems, where data is replicated across servers and client requests are routed to one of these servers. The focus is on minimizing both rejection rates and latency under the condition of reappearance dependencies, which occur when the same data chunk is repeatedly accessed at different time steps, potentially leading to overload on specific servers. The research introduces novel algorithms, including a greedy approach and the delayed cuckoo routing algorithm, to overcome these dependencies and ensure low rejection rates and latencies in distributed systems.

Justification: Efficient load balancing in distributed systems is essential for performance in high-demand environments like game development and parallel computing. In large-scale applications such as MMORPG Games, the challenge of evenly distributing workload across multiple servers becomes crucial to prevent bottlenecks, reduce latency, and ensure high availability. Systems like SpatialOS, which power real-time multiplayer games, must dynamically route tasks across servers, especially as demand fluctuates. Simple load balancing isn't enough in these environments; advanced techniques like greedy algorithms and delayed cuckoo routing help distribute requests efficiently, preventing server overload and maintaining smooth player interactions. These algorithms ensure that each server is optimally utilized, which is essential in massive virtual worlds where real-time updates and data replication occur constantly. With the growing complexity of distributed systems, these strategies will be pivotal in optimizing large-scale, low-latency applications, such as those in gaming and cloud computing infrastructures.

Implementation Feasibility: This paper includes the design of several algorithms, such as the greedy algorithm and delayed cuckoo routing. Both algorithms are discussed with sufficient detail to allow for implementation. Additionally, the paper's theoretical frameworks provide a foundation for understanding the computational challenges posed by reappearance dependencies. While no direct code repository is provided, the algorithms and their analysis can be replicated through existing tools in distributed systems and load balancing research. I may need to adapt these algorithms to specific use cases or datasets, but the theoretical grounding makes this feasible.We can implement this in multiple ways by simulating local machine cores as server and if time permits we can also scale it to Spatial OS (Distributed Systems Game framework for unity), and implement our algorithm to simulate and compare against general load balancing techniques.

Team Responsibilities:

- **Member 1**: Review the paper and summarize the core concepts, especially focusing on the greedy algorithm and delayed cuckoo routing approaches. Design both greedy and cuckoo hashing approach incorporated into the LoadBalancing Algorithm in the form of pseudoCode. Derive Mathematical Time/Space Complexities and compare against general approach.

- **Member 2**: Implement the algorithms presented in the paper, particularly the greedy and delayed cuckoo routing strategies, using the provided theoretical framework. Set up distributed system either on local machine or any other technical framework like Spatial OS. Simulate the results of both LoadBalancing approaches.

Github Repository : https://github.com/Abdullah3435/Algo_Project