Project Proposal: Alleviating the Cold Start Problem in Serverless Computing

Introduction:

A well-liked cloud computing model is called "serverless computing," in which the cloud provider administers the infrastructure and allots resources automatically for the execution of user-defined tasks in response to events. The "cold start" problem, which describes the delayed initial response time of a serverless function, especially when it hasn't been invoked for a while, is one of the biggest problems with serverless computing, though. Through inter-function container sharing, this project seeks to address the serverless computing cold start issue.

Problem:

The cold start problem in serverless computing is caused by several factors, including the overhead of container startup, resource allocation, and application initialization. A serverless function runs in a container, and the container needs to be created, initialized, and prepared before the function can start executing. This process can take a significant amount of time, especially when the function has not been invoked for an extended period. The overhead of container startup, resource allocation, and application initialization can result in a slow initial response time of the function, which is known as the cold start problem.

Objective:

The objective of this project is to develop a solution to the cold start problem in serverless computing, by leveraging the concept of inter-function container sharing. The solution should reduce the overhead of container startup, improve the resource utilization, and optimize the application initialization, while balancing the trade-off between the resource utilization and performance.

Challenges:

To alleviate the cold start problem, several challenges need to be addressed, including reducing the overhead of container startup, improving the resource utilization, and optimizing the application initialization. The overhead of container startup can be reduced by sharing containers between functions, so that the container can be kept alive and reused. The resource utilization can be improved by dynamically allocating and deallocating resources based on the function's resource demands. The application initialization can be optimized by caching the application state and pre-loading the necessary resources. Additionally, the trade-off between the resource utilization and the performance needs to be carefully considered, as the container sharing and resource allocation can also introduce new challenges, such as interference, security, and fairness.

Expected Outcomes:

1. A solution to the cold start problem in serverless computing, based on inter-function container sharing.
2. Improved resource utilization and reduced overhead of container startup in serverless computing.
3. Optimized application initialization and improved response time of serverless functions.
4. Validated results through real-world use cases and performance benchmarks.
5. A contribution to the research community on serverless computing and cold start mitigation techniques.