Failure Modes for the Cloud Computing System

Introduction: The cloud computing system is designed to handle user tasks efficiently and scale the worker nodes based on demand. However, in a real-world project, there can be various failure modes that may impact the system's performance, reliability, and availability. This document outlines potential failure modes and suggests mitigation strategies to address them.

1. Server Failure:
   Failure Mode: One or both servers become unresponsive or crash.
   Mitigation Strategy:
   - Implement a health monitoring system to continuously check the status of servers.
   - Using a load balancer to distribute incoming requests between the servers. If one server fails, the load balancer can automatically redirect traffic to the healthy server.
   - Use redundant servers in different geographical regions to ensure high availability.
   - Implement automatic server recovery mechanisms, such as server restart or failover, to minimize downtime, let say if the server is not responsive for 5 min restart.

2. Worker Node Failure:
   Failure Mode: Worker nodes become unresponsive or crash.
   Mitigation Strategy:
   - Implement a heartbeat mechanism between the server and worker nodes to monitor their status. If a worker node fails to respond, it can be marked as inactive and removed from the pool of available workers.
   - Implement auto-scaling mechanisms to dynamically provision new worker nodes when the demand increases or existing nodes fail, the metric can scale according to the number of tasks in the working-queue per the number of workers.
   - Lost of tasks, worker took a task and did not complete it before encounter an error or shut down, we need to use a buildup solution to return task to the queue if the task did not complete, Redis and Service-Bus as such mechanism for example.

3. Network Connectivity Issues:
   Failure Mode: Loss of network connectivity between servers, workers, or clients.
   Mitigation Strategy:
   - Use distributed network architectures, such as multi-region deployment (discuss in class), to reduce the impact of network connectivity issues in a specific location.
   - Implement retry mechanisms for failed network requests to handle temporary network disruptions.

- Monitor network performance and latency to detect potential issues and address them promptly.

4. Increased Workload and Scalability:
   Failure Mode: The system experiences a sudden increase in workload beyond its capacity, leading to performance degradation or service unavailability.
   Mitigation Strategy:
   - Implement auto-scaling mechanisms to dynamically provision additional worker nodes based on the demand. This can be achieved by monitoring key metrics like CPU utilization, queue length, or response time.
   - Implement load testing and performance tuning to identify bottlenecks in the system and optimize resource allocation.

5. Data Loss and Data Corruption:
   Failure Mode: Loss or corruption of user data or completed work items.
   Mitigation Strategy:
   - Implement regular backups of critical data and work items to a separate storage system or a distributed file system.
   - Utilize fault-tolerant storage solutions that offer replication, redundancy, and data integrity checks (discuss in big data course).
   - Implement data validation mechanisms to ensure the integrity of user data and work items during processing and storage, there are many implementations.

6. Security Vulnerabilities:
   Failure Mode: System vulnerabilities exploited by malicious actors, leading to unauthorized access, data breaches, or service disruption.
   Mitigation Strategy:
   - Implement strong authentication and authorization mechanisms to control access to the system components and APIs.
   - Implement security monitoring and intrusion detection systems to detect and respond to potential security threats.