**Common Failure Causes in Distributed Systems**

1. **Single Point of Failure:** Example, a service that starts by reading configuration from a non replicated data store. If the data store is unavailable, service won’t be able to start. They need to identified when the system is arhictected. Examine every component to see what would happen if the component were to fail.
2. **Unreliable Network:** If a request sent by the client doesn’t arrive, client has two options – either wait or fail the request with exception. It doesn’t know whether the response will eventually arrive or not. Why a response couldn’t come yet – server slow, request drops by network switch or router, server crashed, response drops by network switch or router.
3. **Slow Processes:** Resource leaks are the most common causes of slow processes. Example, memory leak. When there is no memory left, OS starts swapping memory pages to disk constantly and garbabe collector kicks in more frequently which eats up CPU cycles. Other example of resource leak. If a process uses thread pool, the thread can be lost if it executes a synchronous HTTP call that never returns, the thread won’t be returned to the pool.
4. **Unexpected System Load:** DDOS, Seasonality, Expensive requests taking more time like scrapers
5. **Cascading failures.** Example, Initially two replicas handling equal load. If replica B becomes unavailable, LB removes it from the pool and all requests hit replica A. Replica A struggles and clients start to receive timeouts which can then retry causing further load. Eventually replica A becomes unavailable and taken out of the pool. So failure to replica B cascades to A as well.