How to Avoid the Thundering Herd Problem?

The thundering herd problem can occur when a large number of clients suddenly access a resource at the same time, such as a server or a database, causing it to become overwhelmed and potentially fail. The different types of problems could be:-

1) Cascading Failure - Cascading failure refers to a phenomenon where the failure of a component in a system triggers a chain reaction that leads to the failure of multiple interdependent components.

   TO AVOID CASCADING FAILURE WE CAN USE:

   ➔ **Rate limiting:** Limiting the rate at which requests are processed by a system. By restricting the rate at which requests are processed using a queue, rate limiting can help ensure that a system does not become overwhelmed and fail as a result of a surge in demand.

2) Going Viral - During sales like black Friday events, load can increase on the servers due to more Number of users logging in at once.

   TO AVOID CRASHING OF SERVERS WE CAN USE:

   ➔ **Pre-scale:** Use extra servers and be ready beforehand.

   ➔ **Auto-scale:** This type of services can be provided by cloud service providers. If the load on the system increases, the servers automatically scale up (increase in number) in order to distribute the load.

   ➔ **Rate Limiting**

★ Auto-scale is much better than pre-scale as installing servers requires a significant cost and If the number of users are not as much as expected, then there is no use of more servers.

3) Bulk Job Scheduling - During events like New Year or diwali etc. there are cron jobs like sending email notifications to users. If all the notifications are sent at once then the load on servers increases.

   TO AVOID THIS WE USUALLY PREFER THE METHOD OF:-

   ➔ **Batch processing:** It is a mode of operation where large quantities of data are processed in groups, rather than one record at a time. This helps to reduce load and increase efficiency.

4) Popular Post - If a famous person uploads a video and we need to send notifications to their subscribers. Notifications can be sent using Batch

Processing. But the problem can arrive when all the users start hitting the page at once increasing its load.

TO AVOID FAILURE IN THESE SITUATIONS WE CAN USE:

➔ Jittering: This method helps to decrease load by not showing some data that is not of great importance so that other important data can flow seamlessly. Like in youtube, the number of likes and views are not of much importance. So we can not show some data in real time but display an approximate number instead reducing the load.

★ GOOD PRACTICES TO FOLLOW TO AVOID THUNDERING HERD

1. **Caching -** If there are lots of common requests on the server then we can cache them to reduce load on the database.
2. **Gradual Deployments -** Servers should not be deployed all at once. We should deploy some servers then inspect them, and if there is a need to add more servers then deploy more servers based on requirement.
3. **Coupling -** To improve performance we keep some data of external service in our own service