* Distributed System (DS) - Assignment Number - 5.

Name:- Kaustubh Shrikant Kabra.
Class:- Third Year Engineering.
Div:- A Roll Number:- 38
Batch:- T-2
Department:- Computer Department
College:- AISSMS's IOIT.

★ How replication in DNS takes place and why it actually works so well.

→ ① Replication as a scaling technique:

- Placing copies of data close to client processes can help with scaling. But keeping copies up to data requires more network bandwidth. Updating often too may be a waste, not updating often enough is the flip side.
- Replication and caching is used for system scalability, scalability issue generally appears in the forms of performance problem.
- It is necessary to keep up to date of data but it requires more bandwidth.
- Example: Object is replicated N times, we consider R is read frequency and W is write frequency. If R<<W, it gives high consistency overhead and wasted messages.

② Reason's why replication works so well:

1) Data are replicated to increase the reliability of a system.
- If a file system has been replicated it may be possible to continue

working after one replica crashes by simply switching to one of other replicas.

- better protection against corrupted data.

2) Replication for performance

- Scaling in numbers: replication for is important when distributed system needs to scale in numbers and geographical area.
- Scaling in geographical area.

★ Describe a simple implementation of read your writes consistency for displaying web pages that have just been updated.

→ The simplest implementation is to let the brower always check whether it is displaying the most recent version of a page. This requires sending a request to web server. This scheme is simple as it is already implemented by many systems.

Example: Updating your web page and guaranteeing that your web brow-ser shows the newest version instead of its catched copy.

Location 1: Write [x_1] - - -
Location 2: Write [$x_1 ; x_2$] - - -> Read [x_2]

(a) Data store that provides read-tour-writes consistency.

Location 1: Write [x_1] - - -
Location 2: Write [x_2] - - -> Read [x_2]

(b) Data store that does not provide read-your-writes consistency.