Akash Mete

52 TE Comp F

Page :
Date :

DS Assignment - 4

Q.1] What are the two basic file system used in distributed systems? Compare these two file systems

→① Two distributed systems that have been in widespread used for a decade or more:
1) Sun Network File System (NFS)
2) Andrew File System (AFS)

② Comparing between NFS and AFS
- AFS has stateful servers. whereas NFS has stateless servers
- AFS Provides both:
1. Location Independence (the physical storage location of the file can be changed without having to change the path of the file etc.
- 2. Location transparency (the file name does not hint as its physical storage location). But as was seen in the last lecture, NFS provides only location transparency.

- Callback: Stateful servers in AFS allow the server to inform all clients with open files about any updates made to that file by another client
- Callbacks to all clients with a copy of that file is ensured as a callback promise is issued by the server to a client when it requests for a copy of a file.

Camlin

Q.2] Explain different methods for name resolution with suitable example.

→ DNS is desigred as a client server application. A host that needs to map an address to a name or a name to an address calls a DNS client named a resolver.

1] Recursive Resolution

- A client request complete translation
- If the server is authority for the domain name, it checks its database & responds.
- If the server in not authority, it sends the request to another server & waits for the response
- When the query is finally resolved, the response travel back until it finally reaches the requesting client. This is called recursive resolution

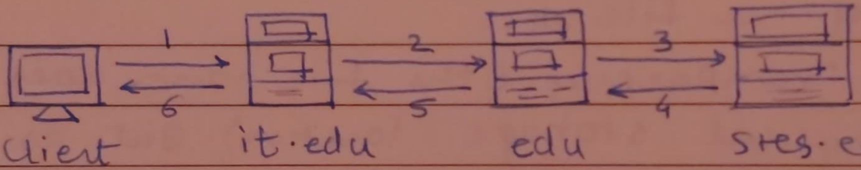
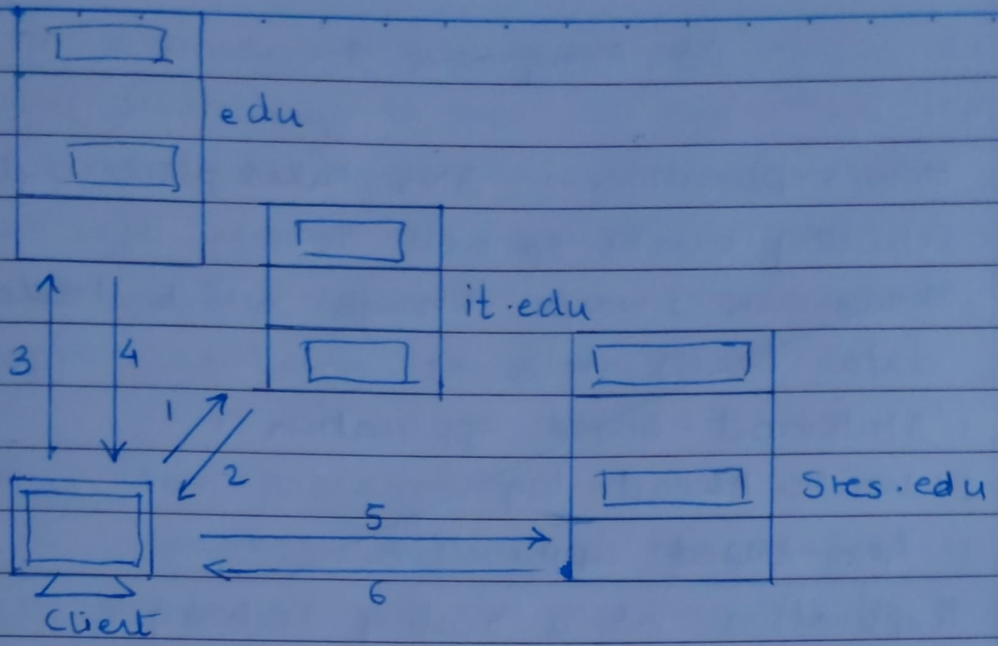


Fig. Example of Recursive Resolution

2] Iterative Resolution

- Only a single resolution is made & returned (not recursi)
- Client must now explicity contact different name servers if further resolution needed
- If the server is and authority for name, it sends the answer. If it is not, it returns the IP addr of the server that is thinks can resolve the query The client is responsible for address of server that is second server. This process is called iterative resolution because the client repeats the same query to multiple Serves.



Iterative Resolution

3] Reverse Name Resolution :

- Reverse Name resolution is important task of DNS on the Internet or the translation of IP addresses back to domain names. For examples servers car determine & record the full domain name of machine connecting to them over the network.
- Reverse name Resolution fields use the PTR resource record, which points to the correct position in the normal DNS space. The hierarchy under "IN-ADDR.ARPA" can be delegated of course just like any other domain
- To obtain IP address of a named server, each host has a client protocol known as name resolver.
- Resolver then creates a resolution req msg in standard msg format of domain name server protocol.

Akash Mete

52 TE Comp I

Page :
Date :

DS Assignment 5

Q.1] How replication in DNS takes place and why it actually works so well.

→ ~~There are 3 ways through which leaders replicate data to its follower:~~

~~1. Statement-based application~~

~~2. Write-Ahead log~~

~~3. Row-based application.~~

→① Replication as a scaling Technique :

- Placing copies of data close to client processes can help with scaling. But keeping copies up to date requires more network bandwidth. Updating too often may be a waste. Not updating often enough is the flip side.
- Replication & caching is used for system scalability. Scalability issue generally appears in the form of performance problem.
- Performance is increased by reducing access time. This is possible when multiple copies of data is placed near to the object.
- 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 consistancy overhead & wasted messages.

② Reasons why Replication works so well:

i) 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 the other replicas.

- better protection against corrupted data.
- can be safe against single, failing write operation by considering the value that is returned by at least two copies as being the correct one.

2) Replication for performance

- Scaling in numbers: Replication for performance is important when distributed system needs to scale in numbers & geographical area.
- Scaling in geographical area

Q.2] 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 browser always check whether it is displaying the most recent version of a page. this requires sending a request to the ~~imp~~ web server.
This scheme is simple as it is already implemented by many systems.

Example: updating your web page & guaranteeing that your web browser shows the newest version instead of its catched copy.

Location 1: Write[x1] - - -
Location 2: write[x1;x2] - - → Read [x2]

(a) Data store that provides read-your-writes consistency

Location 1: Write[x1] - -
Location 2: Write[x2] - - → Read[x2]

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

Akash Mere

52 TE Comp I

Page :
Date :

DS Assignment 6

Q 1] What is a failure model? Explain the different types of failures.

→ 1) Failure models defines the way in which failure may occur in order to provide an understanding of its effects.

2) A taxonomy of failures which distinguish between the failures of processes & communication channels is provided :-

1. Omission Failures : Process or channel failed to do something.
2. Arbitrary Failures : Any type of error can occur in processes or channels
3. Timing Failures. Applicable only to synchronous DS where time limits may not be met.

1. ~~Omm~~ Omission Failures - A process or communication channels fails to perform actions that it is supposed to do.

a) Process omission failures : Process has crashed and can be detected using timeouts. Fail-stop process crash is one that can be detected with certainty by other processes.

b) Communication omission failures : Communication primitives are send & received.

- Send Omission : loss of messages betn sending process & outgoing message buffer.
- Channel Omission : loss of message in communication ch.
- Receive Omission : loss of messages betn incoming message buffer & receiving Process.

Page:
Date:

2. Arbritrary Failures:

- Arbitrary process failures: Arbitrarily omits intended processing steps or takes unintended processing steps.

- Arbitrary channel failures: Messages may be corrupted, duplicated, delivered out of order, incur extremely large delays, or non-existent messages may be delivered.

- Arbitrary failures in processes cannot be detected by seeing whether the process responds to invocations because it might arbitrarily omit to reply.
- Communication channel also suffer from arbitrary failures.

3. Timing failures:

- Timing failures are applicable in synchronous DS where time limits are set on process execution time, message delivery time & clock drift rate.
- In an asynchronous DS, an overloaded server may respond too slowly, but we can not say that it has a timing failure since no guarantee has been offered

1) Clock failure: Affects Process's clock exceeds the bounds on its rate of drift from real time.

2) Performance: Affects process exceeds bounds on the interval between 2 steps

3) Performance Channel: A msg's transmission takes longer than the stated bound.

Page :
Date :

q.2] Explain reliable client server communication.

→ 1) A communication channel may lose and/or corrupt messages.

2) Techniques for reliable communication:

a. Use redundant bits to detect bit errors in packet

b. Use sequence numbers to detect packet loss

c. Recover from corrupted / lost packets using

3) Five types of failures can occur in RPC

① client cannot locate server

② server crashes after receiving a request

③ client request is lost

④ server response is lost

⑤ client crashes after sending a request.

4) Communication using TCP

- A reliable point-to-point communication can be established by using TCP protocols
- TCP masks omission failures
- TCP does not mask crash failures.

5) communication using RPC (Remote Procedure Calls)

- the goal of RPC is to hide communication by making remote procedure calls that look just like local ones
- the RPC mechanism works well as long as both the client and server function properly.