Ans(1) (a) HTTP Get.

eg: $\overset{\text{URI}}{\overbrace{\text{GET /index.html}}}$ HTTP/1.0

In this example GET request for document named "/index.html" and uses "HTTP/1.0" protocol. In "get" request method only data is retrieved and should have no other effect. In this server will recieve the request and maps the request URI do document under its document directory. If document is available server will return document with response status code "200 OK", or other suitable ~~res~~ response code.

(b) HTTP Head :

eg: HEAD /index.html HTTP/1.0

This request method is similar do "GET" request method, however in this server only returns header without the response body. It is useful for checking the

the header, such as following field:

Last-Modified
Content-Type
Content-Length.

ⓒ HTTP PUT:

eg PUT /index.html HTTP/1.0

HTTP PUT request method creates a new ~~the~~ resource or replace a representation of the target resource with request method. So for above eg: If the target resource does not have current representation, the server will respond to the request by sending "201" ("Created") status.
If the representation is modified is according to enclosed representation, then the server will send "200" (OK")

(d) HTTP POST:

eg POST $\overbrace{\text{/bin/login}}^{\text{URI}}$ HTTP3/1.1

user= Amandeep & pw= 123456 action= login

body

POST request method is used to post addition data to the server (eg submitting form data" We can trigger POST request by using HTML forms with attribute method="Post" When the submit button of form is clicked, post request method is created with ~~p~~ username and password in body of request method. Server recieves the string that will be parsed in order to get data.

(e) HTTP DELETE

eg DELETE /file.html HTTP/1.1

Delete method is used to ~~dlt~~ delete specified resource.

So for above eg:
server will locate ~~or~~ resource "file.html"
and will delete ~~and~~ the resource and
following response code ~~is~~ is possible

ⓐ 202 (Accepted) -> action will likely succeed but not yet enacted

ⓑ 204 (no content) -> action has been enacted and no further information is to be supplied

ⓒ 200 (OK): action has been enacted and the response message include a representation describing status.

④ HTTP trace:

eg: TRACE http://www.amazon.com/ HTTP/1.1

TRACE request method is used to ask the server to return a diagnostic trace.
so for eg: we are asking for trace of amazon.com.

(9) HTTP OPTIONS.

Client can use an OPTIONS request method to query the server which request are supported.

for eg:

OPTIONS http://www.amazon.com/ HTTP/1.1

will get 200 OK response and no. of request methods supported by amazon.com like.

" Allow: GET, HEAD, POST, OPTIONS, TRACE."

Sol(2)

	Global cache	Distributed Cache.
(a)	All nodes use the same single cache space	(a) All nodes own a part of the cached data.
(b)	In this each node will check global chache first, if the data is not in the cache, then the node will	(b) In this, for each request, node checks the cache based on the item key (using hashing algorithm), then the

retrieve data from origin	the data origin

Sol ②

ⓑ

Local Resturant Food Ordering System

Reason to use Global cache:

① Similar type of data retrival requests

Since system is food ordering system mostly request will be data retrival request regarding "Food Menu" Option and price list for food items. All the request are almost similar and limited.

② Reason for not choosing distributed Cache:

(a) Data is small and predictable
Since the data is small and predictable, Global cache is good option because it will perform much better than a distributed cache system

(b) For small to medium size application where data is small, implementing distributed cache is difficult to implement.

(c) Distributed Cache system requires an interprocess call for each cache access, which will be futile in case of short to medium range Applications.

(2) (c)

eg: LinkedIn system: storing large no. of user profile
Reason to use distributed Cache system

~~(a) Data is huge: Because our data in this case ie no. of user profiles, is large, we have~~ ~~large~~ ~~our object~~

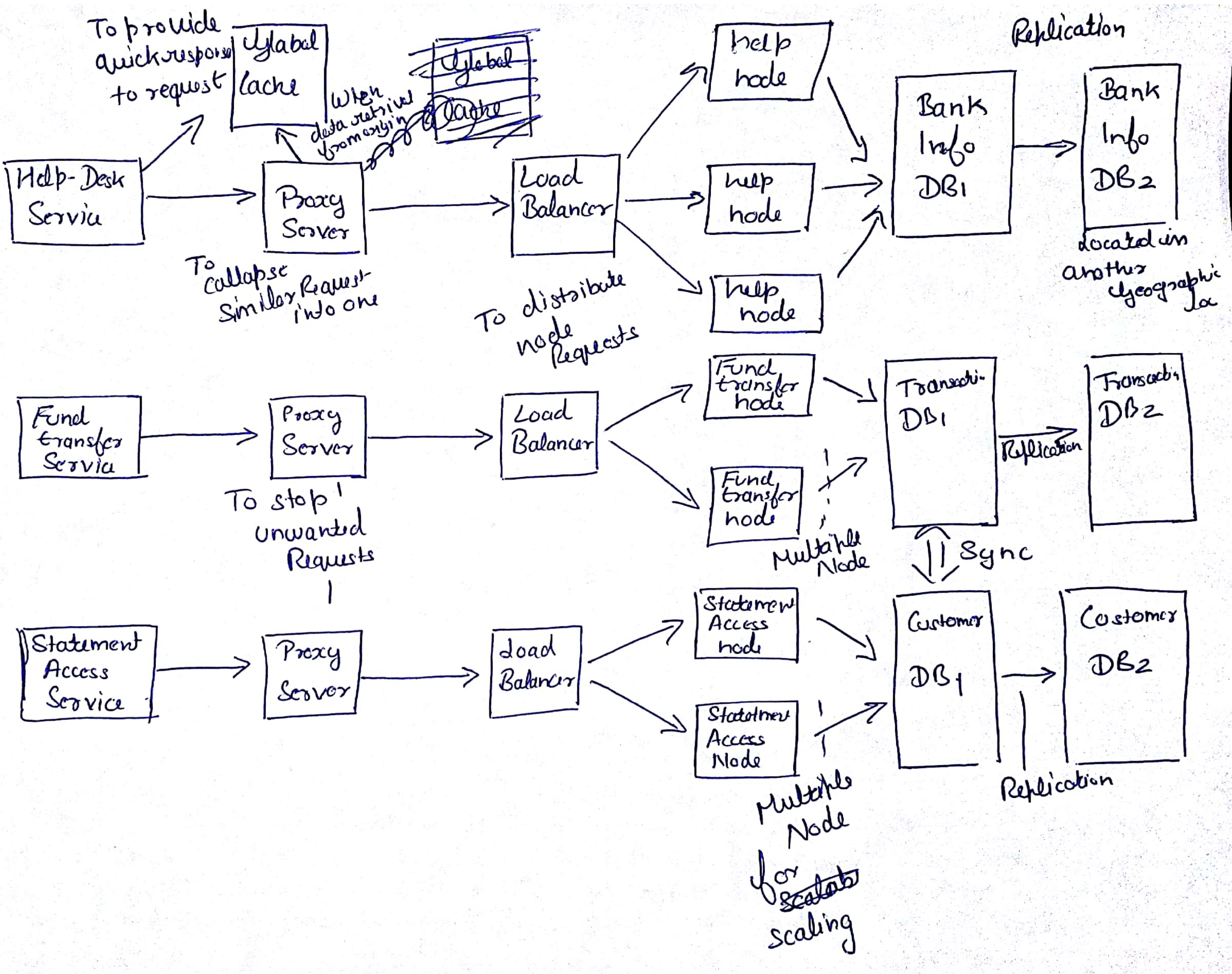
, and unpredictable.

(a) Data is large: As we have large number of users profiles to handle, our objects that needs to be stored or ~~tha~~ cached are large and unpredictable ∴ There fore if we use global chache in place of distributed cache our consistency of read won't be good.

(b) Distributed cache system are better scallable in this case:

⇒ Since we can increase of cache memory in distributed system by just adding extra nodes. There fore incase of large system distributed cache system are much more scalable than global cache systems.

Q3



To provide quick response to request
Global Cache
When data retrival from origin
~~Global Cache~~
Help-Desk Service
Proxy Server
To collapse similar Request into one
Load Balancer
To distribute node Requests
help node
help node
help node
Replication
Bank Info DB1
Bank Info DB2
Located in another Geographic lo
Fund transfer Service
Proxy Server
To stop unwanted Requests
Load Balancer
Fund transfer node
Fund transfer node
Multiple Node
Transaction DB1
Transaction DB2
Replication
Sync
Statement Access Service
Proxy Server
Load Balancer
Statement Access node
Statement Access Node
Multiple Node for ~~scalab~~ scaling
Customer DB1
Customer DB2
Replication