



CACHES AND CDNs

Alia Asheralieva

Department of Computer Science and Engineering (CSE)

SUSTech

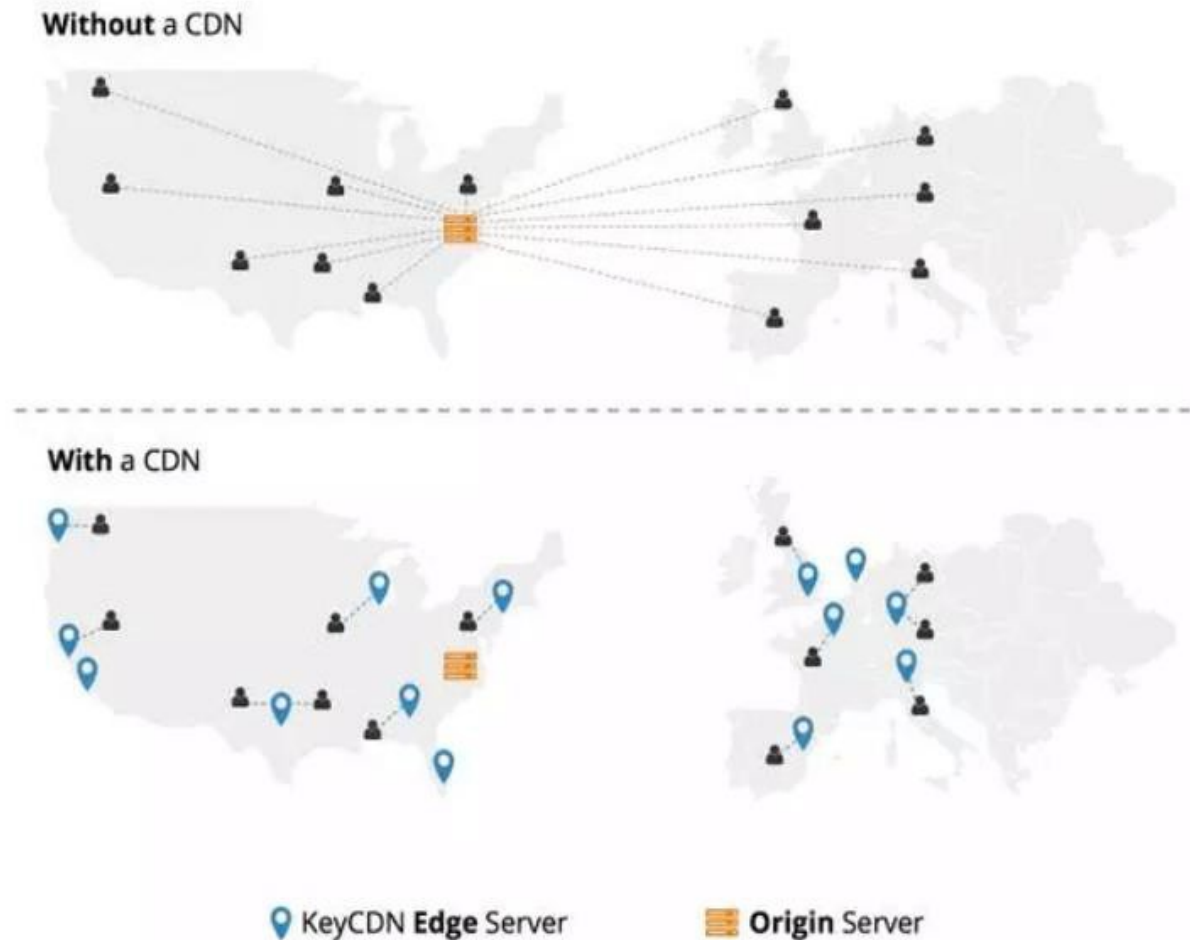
CONTENT DELIVERY NETWORK (CDN)

- CDN is designed for cache content on a node closer to edge users to improve their experience.
- What are the scenarios of CDN?
 - Big flow website, such as: online video, games, pictures, audio, social, e-commerce, download stations, etc.
- CDN is suitable for a certain level of static resource access (html, js, css, jpg, gif, etc).



How does CDN work?

A CDN server is a **reverse proxy cache server**.



CDN EXAMPLE

A static resource which is cached on a CDN node of aliyun

```
C:\WINDOWS\system32\cmd.exe
C:\Users\vivi>curl -I https://wx4.sinaimg.cn/mw690/6fa017c71y1fw55ou2gb2j21001qqb29.jpg
HTTP/1.1 200 OK
Server: edge-esnssl-1.12.1-12.1
Date: Sun, 14 Oct 2018 07:42:30 GMT
Content-Type: image/jpeg
Content-Length: 302723
Connection: keep-alive
x-oss-request-id: 5BBFF2EEB4DE0B1E77F7AEB5
ETag: "C3893605B7AD4E4A34D1D4E0E5C008CB"
Last-Modified: Fri, 12 Oct 2018 00:55:57 GMT
x-oss-object-type: Symlink
x-oss-storage-class: Standard
x-oss-hash-crc64ecma: 2801485700556859094
Via: cache48.12cm12-1[0,200-0,H], cache16.12cm12-1[0,0], cache8.cn1009[0,200-0,H], cache14.cn1009[1,0],
http/1.1 cmcc.guangzhou.ha2ts4.137 (ApacheTrafficServer/6.2.1 [cMsSf ])
Age: 196728
Ali-Swift-Global-Savetime: 1539306777
X-Swift-SaveTime: Fri, 12 Oct 2018 01:12:57 GMT
X-Swift-CacheTime: 8640000
Timing-Allow-Origin: *
EagleId: b7f0d5a215395029501395216e
X-Cache: MISS.137
X-Via-CDN: f=edge, s=cmcc.guangzhou.edssl.95.nb.sinaedge.com, c=183.232.197.103;f=edge, s=cmcc.guangzhou.ha2ts4.103.nb.sinaedge.com, c=183.232.24.95;f=Edge, s=cmcc.guangzhou.ha2ts4.137, c=183.232.24.103;f=alicdn, s=cache14.cn1009, c=183.232.24.137;
Access-Control-Allow-Origin:
X-Via-Edge: 153950295013267c5e8b7de18e8b734b573f6
```

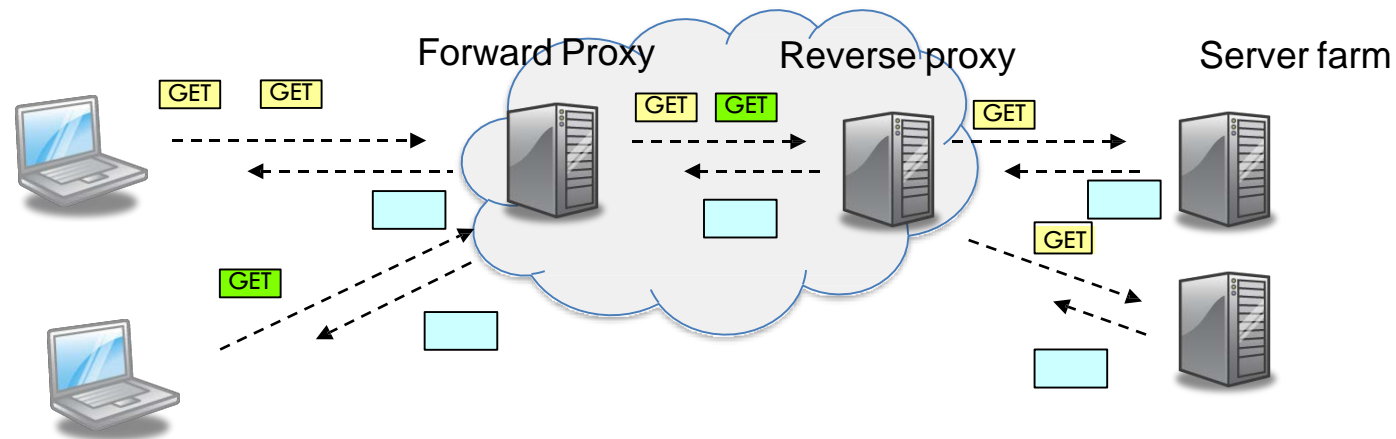
WEB PROXIES

- Web proxies are intermediaries between web clients and web servers that fulfill transactions on clients' behalf.
- A client sends a request to the proxy, which forwards the request to the server. When the proxy receives a response from the server, it forwards the response back to the client.
 - Proxies act like servers to web clients
 - Proxies act like clients to web servers



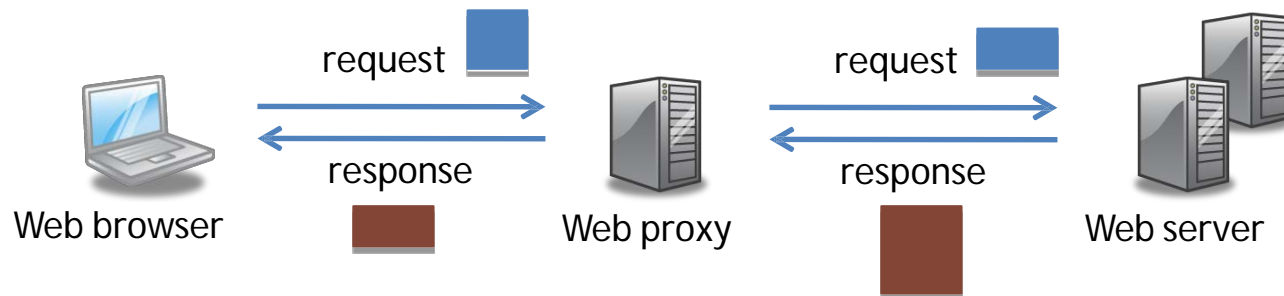
WEB PROXIES

- There can be multiple proxy servers between a browser and the origin web server (that produces HTTP responses)
- Proxy servers are transparent to end-users.



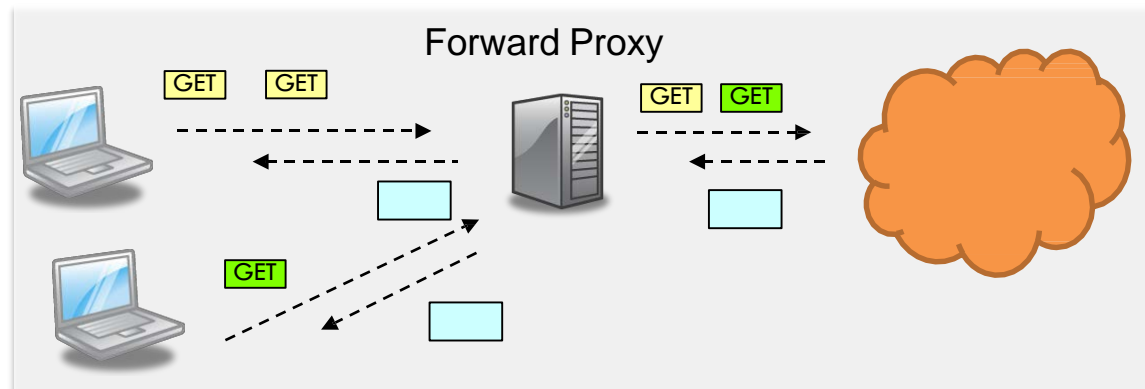
FUNCTIONS OF PROXIES

- A web proxy may scan and change requests and responses in transit. It may also route a request to different servers or handle the request itself.
- Some possible functions:
 - Content filtering – block access to inappropriate content
 - Security firewall – block malicious software like virus
 - Web caching – improve HTTP performance
 - ... two more examples follow



FORWARD PROXY

- A **forward proxy** acts on behalf of a client (or other forward proxy) to access web servers in the Internet
- Forward proxy server can concentrate HTTP traffic.
 - Requests from different sessions may be transmitted in one TCP connection
 - Reduce outgoing bandwidth usage and concurrent TCP connections
- Possible to reuse cached response for different users



MORE ABOUT FORWARD PROXY

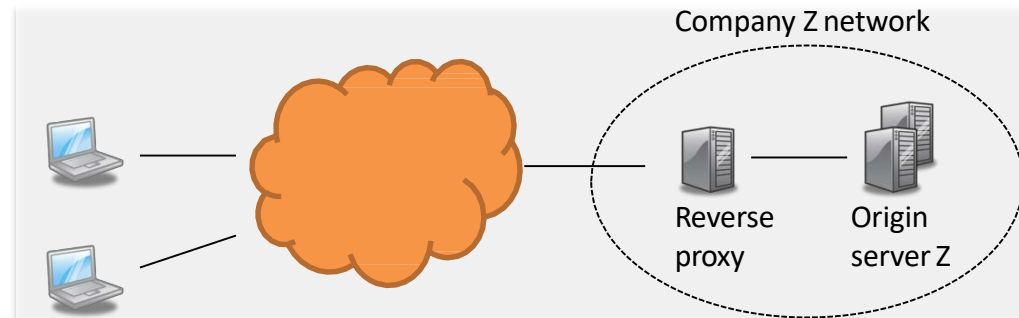
- A forward proxy often hides the IP of the clients
 - TCP connection between the proxy and the server, but not between the client and server
- Some proxies add a request header X-Forwarded-For to reveal client's IP address
 - Experiment*: <http://whatismyipaddress.com/>

* If you are behind a NAT (network address translation), the result will not correspond to your IP address, which doesn't mean you are behind an HTTP proxy.



REVERSE PROXY

- Reverse proxy is a proxy server that retrieves resources on behalf of a client from one or more servers
 - Usually, a reverse proxy only connects to web servers of a web site
 - Popular software: Nginx, lighttpd
- For client, the reverse proxy works as the web server of the company
 - Client cannot connect to the 'real' web servers behind. The reverse proxy uses IP address of a web site.
 - For clarity, the 'real' web servers are sometimes called origin server



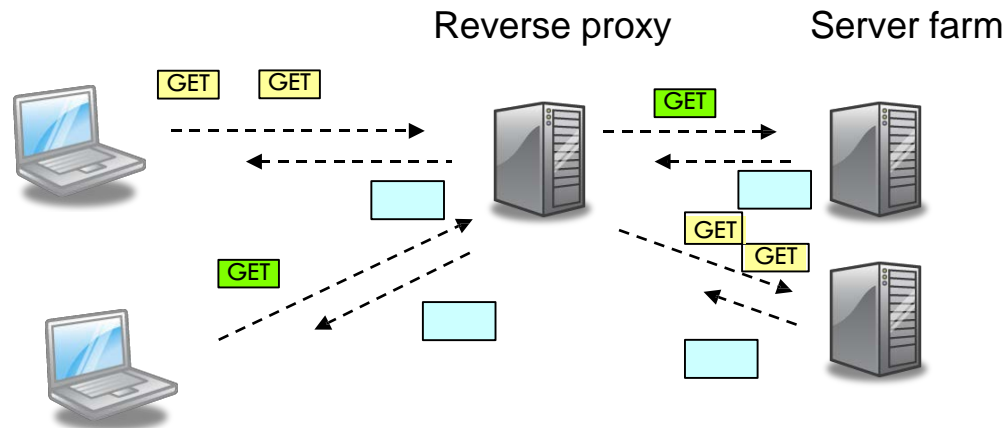
FUNCTIONS OF REVERSE PROXY

- **Reduce workload of origin servers**
 - Caching
 - Load balancing
 - Serving static resources (e.g. images)
 - HTTP Compression and encryption
- **Protect against common web-based attacks**

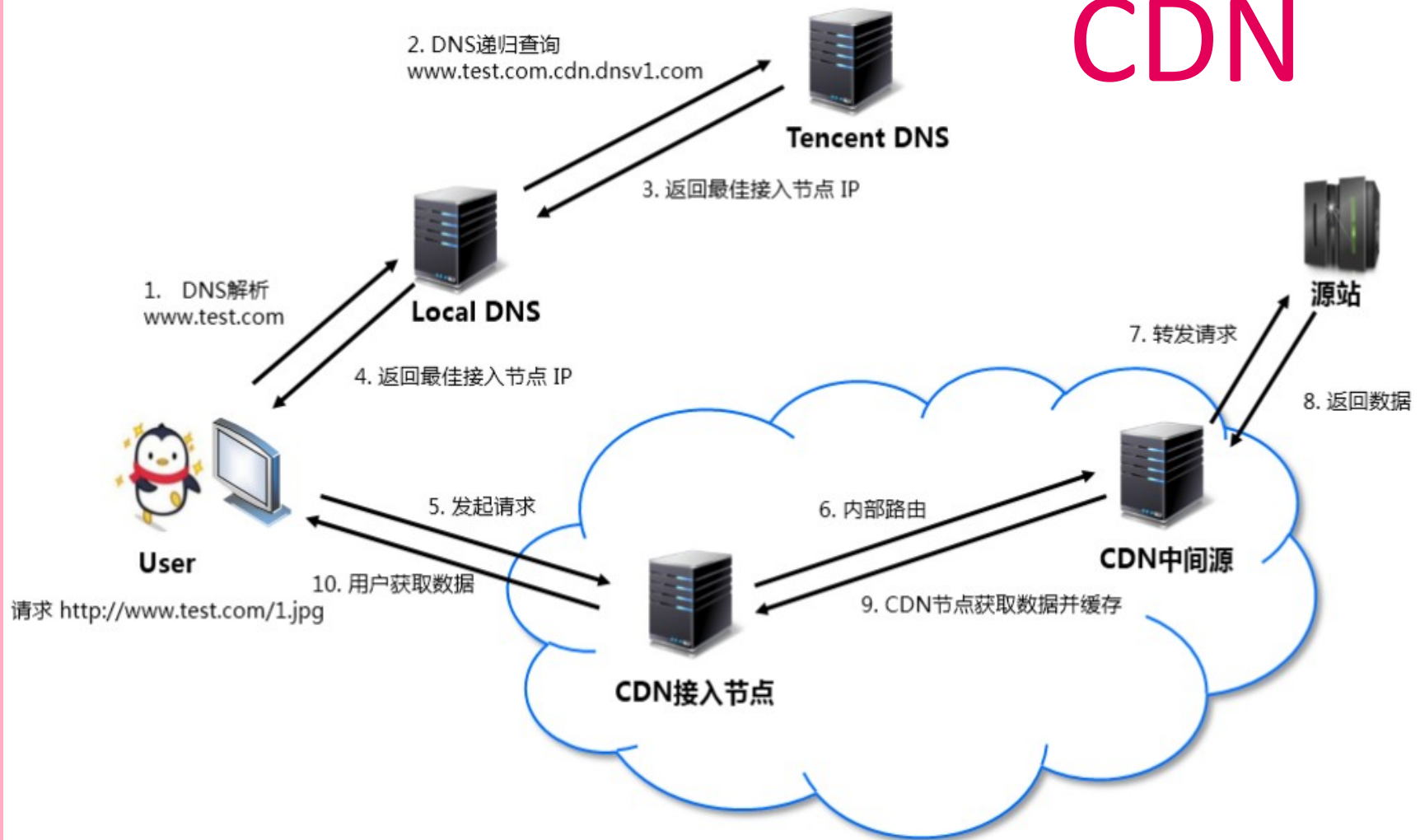


REVERSE PROXY FOR LOAD BALANCING

- A reverse proxy usually sits before a **server farm**
 - Each server in the farm **duplicates** databases, programs and static resources (e.g. images)
 - The reverse proxy dispatches request from the same user to one of the servers in the server farm.
 - Each server maintains client sessions.




CDN



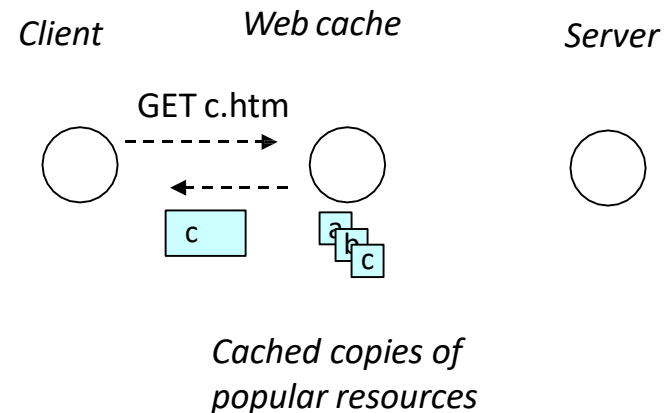
CACHING EVERYWHERE (EXAMPLES)

Xiaolin has accessed the web site via Chrome on her MacBook:

- **Web server:** cache rendering result (for dynamic pages).
 - **CDN server:** cache static content.
 - **Web browser:** cache web resources*.
 - **Operating System:** likely cache file of browser cache in memory.
 - **Hybrid Drive:** cache recent access blocks of HDD in SSD.
- 

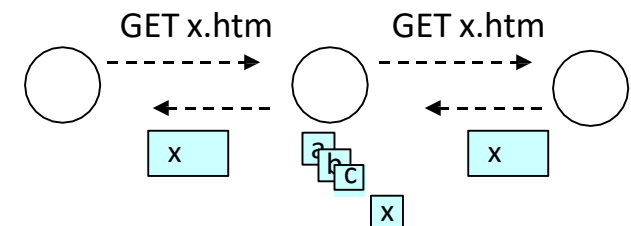
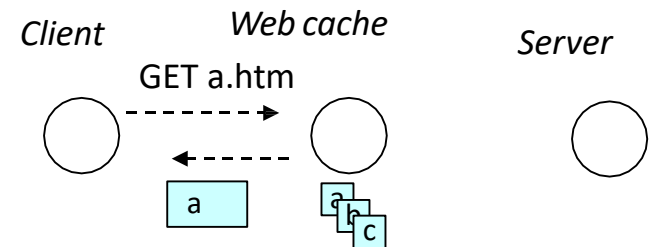
WEB CACHING:

- Some resources are retrieved frequently when users browse the web
 - e.g., images, popular web pages, JavaScript libraries
- Web cache saves such resources (in memory or file system) and use them to satisfy future requests from clients.



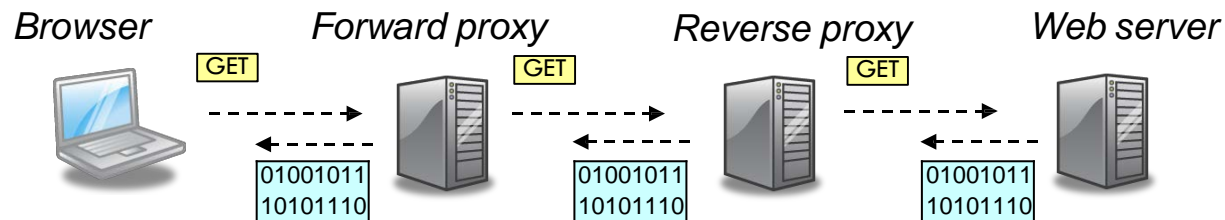
CACHE HIT VS. CACHE MISS

- **Cache hit:** the requested resource is available in the cache
 - **return the cached copy as a response**
- **Cache miss:** the requested resource is not available
 - **forward the request to the web server**
 - **save response in the cache**
 - **return response to the client**



VARIETIES OF WEB CACHING

- Web request / response travel through several machines from a client to a server. Web caching is implemented in several places:
 - Web browser
 - Web proxies:
 - forward proxy (Cache server)
 - reverse proxy
 - Web server



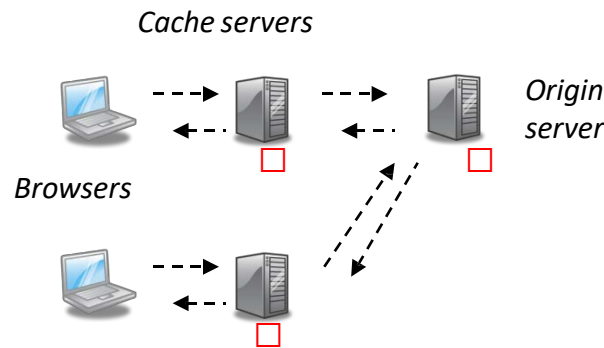
WEB BROWSER CACHE

- Built-in caching of browsers.
 - It saves cached copies in memory and disk on the client machine.
 - Can cache private resources (response with Cache-Control: private)



WEB CACHE CONSISTENCY

- Resources in an origin server may be modified after a cache server saves a copy
- If the cache server returns such an outdated copy, clients will have inconsistent view of the web site

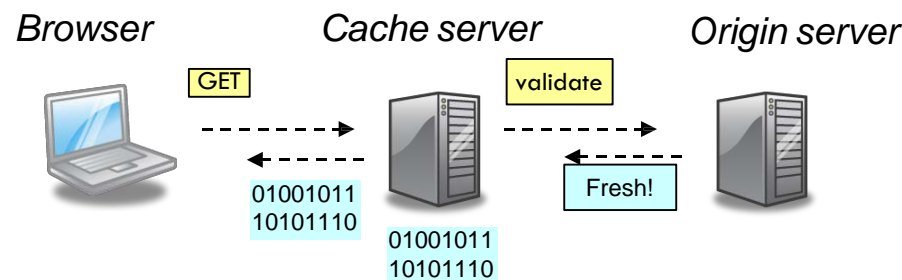


1. Cache server A saves a copy of a resource at 8:00
2. The resource is modified at origin server at 9:00
3. Cache server B retrieves the resources and saves a copy at 10:00



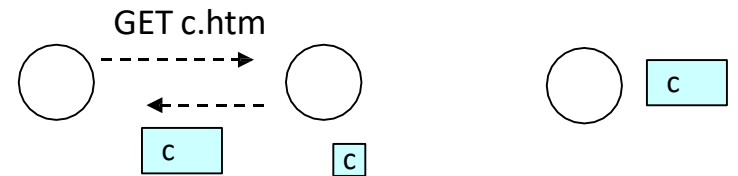
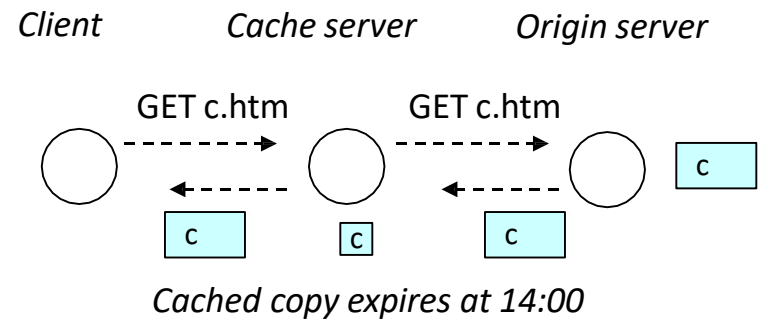
ENSURING CONSISTENCY BY VALIDATION

- **Validation:** a cache server inquires the origin server whether the cached copy is 'the same' as the resource in the origin server
- The cache server may validate a cached copy before satisfying client's request
 - But validation for every request would be too expensive.



FRESH CACHED COPY

- The origin server may specify an **expiration** time of its response. Before a cached copy expires, it remains **fresh**.
- The resource is not likely to change before it expires
- The cache server considers it safe to satisfy a client's request with a fresh cached copy.

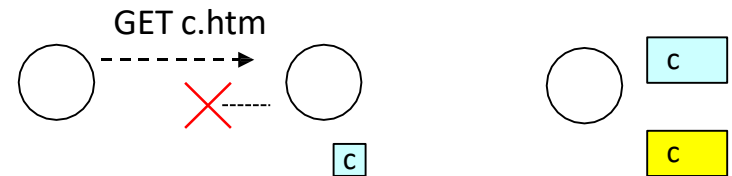
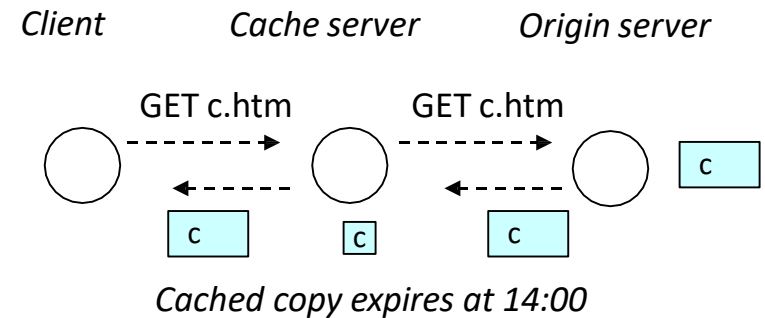


At 13:30, the cached copy is still fresh.
So the cache server can return the
copy to the client.



STALE CACHED COPY

- If a cached copy has expired, it becomes **stale**.
- It is likely that the resource in the origin server has changed.
- The cache server cannot satisfy a client's request with a stale cached copy.
- But the stale cached copy may still be the same as the resource in the origin server.



At 15:00, the cached copy has expired. The cache server cannot use the copy immediately.



VALIDATION:

- Before reusing a stale cached copy, a cache server should **validate** it with the origin server.
- If there is no change in the resource after the cache server obtained the cached copy, it is safe to return to a client.



OPERATION OF CACHE SERVER

