

网络

CDN

是利用最靠近每一位用户的服务器，更快、更可靠地将文件发送给用户分发网络。

CDN 优点

提速：会给用户指派较近、较顺畅的服务器节点，将数据传输给用户

低成本：服务器被放到不同地点，减少了互连的流量，也降低了带宽成本

高可用度：当某个服务器故障时，自动调用邻近地区的服务器

CDN 回源

回源是指浏览器访问 CDN 集群上静态文件时，文件缓存过期，直接穿透 CDN 集群而访问源站机器的形为。

CDN 缓存

三级缓存：浏览器本地缓存、CDN 边缘节点缓存、CDN 源站缓存

缓存设置：缓存时间设置的过短，CDN 边缘节点缓存经常失效，导致频繁回源，增大了源站负载，访问也慢；缓存时间设置的过长，文件更新慢，用户本地缓存不能及时更新；所以结合业务情况而定
不同静态资源类型缓存时间：

- HTML：3分钟
- JS、CSS：10分钟、1天、30天

```
1 html
2 <script src="./react.js" > // 30
```

缓存设置

```
1 server {
2   listen 80;
3   server_name 123.com
4   location ~* \.(jpg|jpeg|gif|bmp|png){
5     expires 30d;
6   }
7 }
```

```
8 // koa
9 app.use(KoaStatic('public', {
10   cacheControl: 'max-age: 30'
11 })))
12 app.listen(3000)
```

DNS

浏览器并发数限制，分布设置成多个域名

- 用户访问：Java、PHP 等 API 接口 [api.com](#)
- 页面和样式：HTML/JS/CSS [css.com](#)
- 图片：jpg、png、gif 等 [image.com](#)

cookie：你请求的url在 cookie 生效范围之内，cookie 追加到请求头里面去了

如何减少 HTTP 请求数

- 图片使用 DataURI、Web Font
- 接口存储 LocalStorage

缓存

开启和配置 gzip 压缩

对文本进行压缩，减少 50%~70%

```
1 Nginx 配置：nginx.conf 文件增加
2 gzip on
```

生效检测：

Response header 查看是否有 Content-Encoding: gzip，代表服务端已开启 gzip。

https

nginx 配置

```
1 server {
2   listen 443 ssl; # 启用HTTPS
3   server_name 123.com; # 这里是刚才的域名
4   ssl_certificate 123+3.pem;
```

```
5  ssl_certificate_key 123+3-key.pem;  
6  ...  
7 }
```

http 2.0

HTTP/2 站点的优势

- 可以降低服务器压力
- 提升网站访问速度
- 保护网站安全

