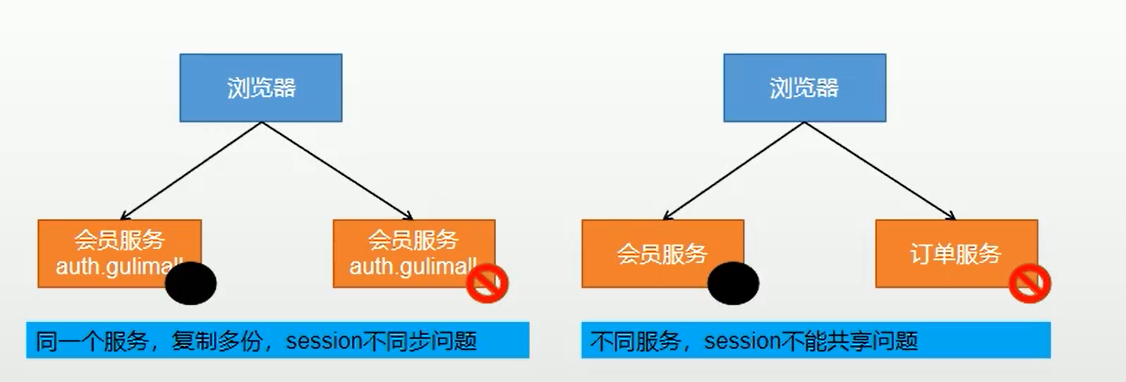
分布式存在的问题：



**集群复制出现：不大可以**



优点：

- web-server (Tomcat)原生支持，只需要修改配置文件

缺点：

- session同步需要数据传输,占用大量网络带宽，降低了

服务器群的业务处理能力

·任意一台web-server保存的数据都是所有web-server的

session总和，受到内存限制无法水平扩展更多的web-server

·大型分布式集群情况下，由于所有web-server都全量保

存数据，所以此方案不可取。

**第二种方法：客户端存储，不大可以**



优点

·服务器不需存储session，用户保存自己的session信息

到cookie中。节省服务端资源

缺点

·都是缺点,这只是种思路。·具体如下:

.每次http请求，携带用户在cookie中的完整信息，浪费网

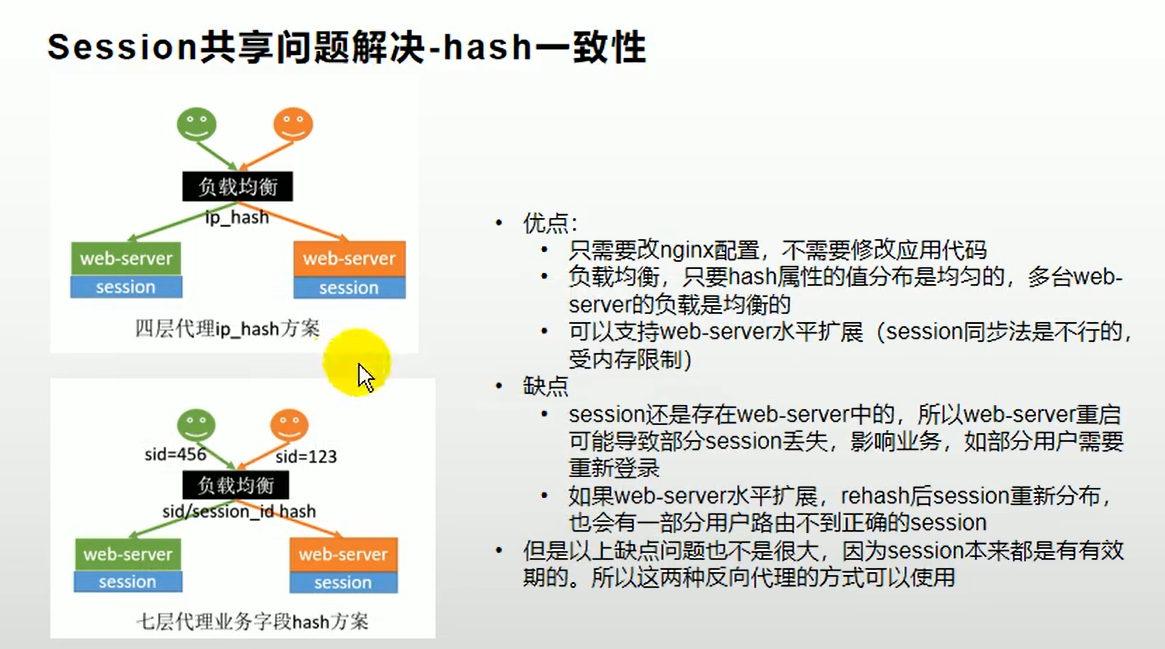
络带宽

- session数据放在cookie中，cookie有长度限制4K，不能保存大量信息

. session数据放在cookie中，存在泄漏、篡改、窃取等安全隐患

这种方式不会使用。

**使用hash一致性：还行**



优点:

·只需要改nginx配置，不需要修改应用代码

·负载均衡，只要hash属性的值分布是均匀的，多台web-server的负载是均衡的

·可以支持web-server水平扩展(session同步法是不行的,受内存限制)。

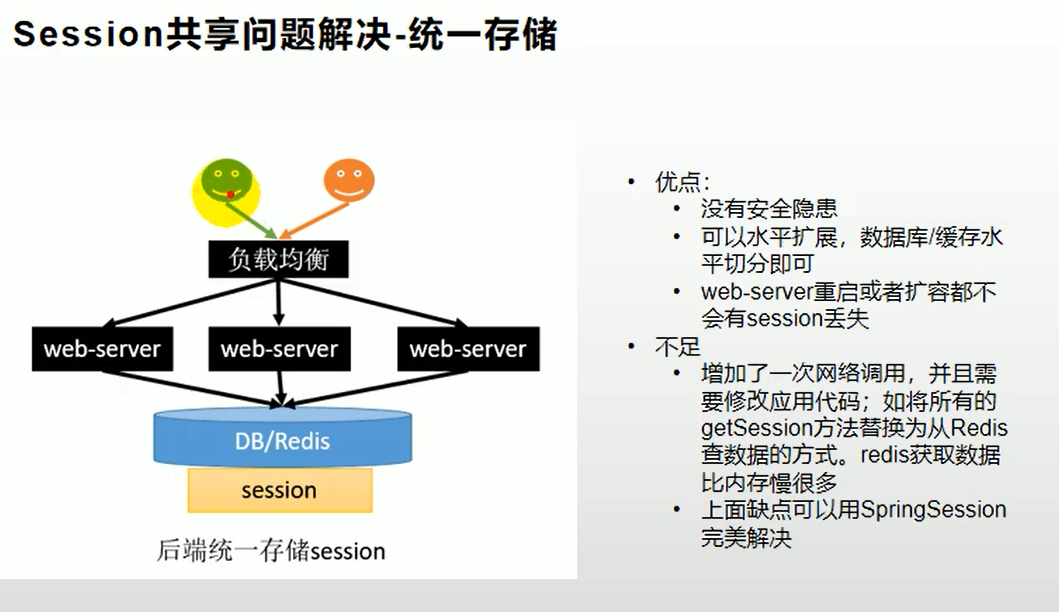
缺点：

session还是存在web-server中的，所以web-server重启可能导致部分session丢失，影响业务，如部分用户需要重新登录

·如果web-server水平扩展，rehash后session重新分布，也会有一部分用户路由不到正确的session

·但是以上缺点问题也不是很大，因为session本来都是有有效期的。所以这两种反向代理的方式可以使用

**使用redis解决：较好**

优点:

·没有安全隐患

·可以水平扩展，数据库/缓存水平切分即可

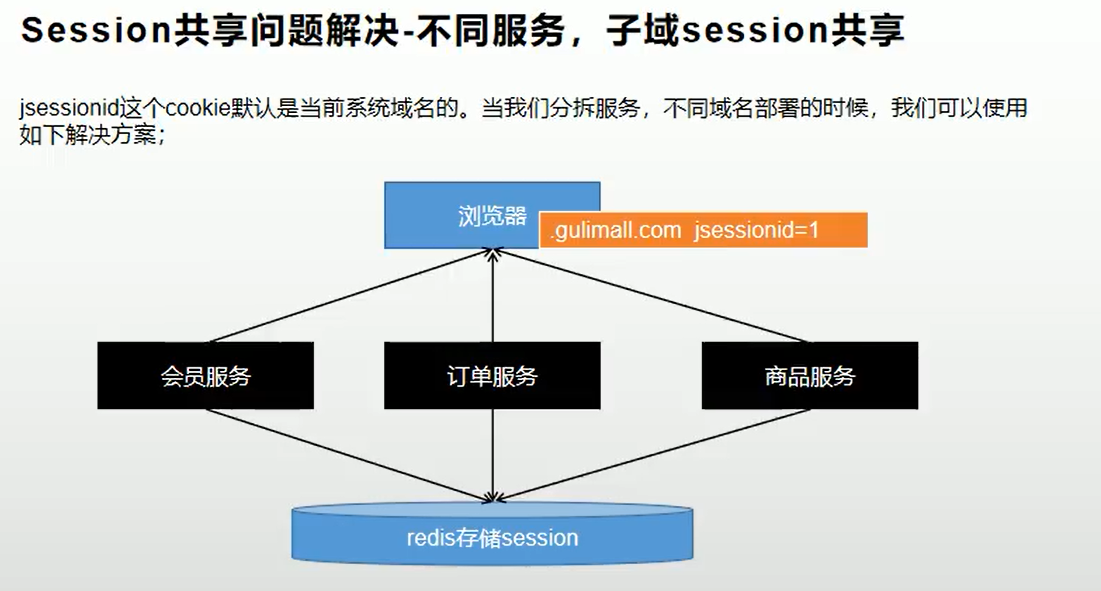
·web-server重启或者扩容都不会有session丢失

不足

·增加了一次网络调用，并且需要修改应用代码;如将所有的getSession方法替换为从Redis查数据的方式。redis获取数据比内存慢很多

·上面缺点可以用SpringSession完美解决

子域session共享问题：



扩大session的子域域名即可。

步骤：记得导入redis和session

*<!-- springsession依赖-->*<dependency>  
 <groupId>org.springframework.session</groupId>  
 <artifactId>spring-session-data-redis</artifactId>  
 <version>2.2.6.RELEASE</version>  
</dependency>

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-cache</artifactId>  
</dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-redis</artifactId>  
</dependency>

yml文件书写：

spring:

session:  
 store-type: redis