

# 分布式Session共享的4类技术方案，与劣势比较

## 题目标签

学习时长：20分钟

题目难度：中等

知识点标签：负载均衡

## 题目描述

服务器为每个用户创建一个会话，存储用户的相关信息

根据题目要求我们可以知道：

- 什么是session
- 什么是session一致性问题？
- 分布式session
- Session一致性解决方案

分析需要全面并且有深度

### 容易被忽略的坑

- 分析片面
- 没有深入

### 1.什么是session

服务器为**每个用户创建一个会话，存储用户的相关信息**，以便多次请求能够定位到同一个上下文。这样，当用户在应用程序的 Web 页之间跳转时，存储在 Session 对象中的变量将不会丢失，而是在整个用户会话中一直存在下去。

当用户请求来自应用程序的 Web 页时，如果该用户还没有会话，则 Web 服务器将自动创建一个 Session 对象。当会话过期或被放弃后，服务器将终止该会话。

Web开发中，web-server可以自动为同一个浏览器的访问用户自动创建[session](#)，提供数据存储功能。最常见的，会把**用户的登录信息、用户信息存储在[session](#)中，以保持登录状态。**

### 2.什么是session一致性问题？

只要用户不重启浏览器，每次http短连接请求，理论上服务端都能定位到[session](#)，保持会话。

### 3.分布式session

单服务器web应用中，session信息只需存在该服务器中，这是我们前几年最常接触的方式，但是近几年随着[分布式系统](#)的流行，单系统已经不能满足日益增长的百万级用户的需求，集群方式部署服务器已在很多公司运用起来，当高并发量的请求到达服务端的时候通过负载均衡的方式分发到集群中的某个服务器，这样就有可能**导致同一个用户的多次请求被分发到集群的不同服务器上，就会出现取不到session数据的情况，于是session的共享就成了一个问题。**

这个时候就需要解决Session一致性。

## 4.Session一致性解决方案



### 1、session复制

session复制是早期的企业级的使用比较多的一种服务器集群session管理机制。

应用服务器开启web容器的session复制功能，在集群中的几台服务器之间同步session对象，使得每台服务器上保存所有的session信息，这样任何一台宕机都不会导致session的数据丢失，服务器使用session时，直接从本地获取。

**简介：**将一台机器上的Session数据广播复制到集群中其余机器上

**session复制使用场景：**机器较少，网络流量较小

**优点：**实现简单、配置较少、当网络中有机器Down掉时不影响用户访问

**缺点：**广播式复制到其余机器有一定延时，带来一定网络开销。

这种方式在应用集群达到数千台的时候，就会出现瓶颈，每台都需要备份session，出现内存不够用的情况。

### 2、session绑定

利用hash算法，比如Nginx的ip\_hash,使得同一个Ip的请求分发到同一台服务器上。

**session绑定使用场景：**机器数适中、对稳定性要求不是非常苛刻

**优点：**实现简单、配置方便、没有额外网络开销

**缺点：**网络中有机器Down掉时、用户Session会丢失、容易造成单点故障。

这种方式不符合对系统的高可用要求，因为一旦某台服务器宕机，那么该机器上的session也就不复存在了，用户请求切换到其他机器后没有session，无法完成业务处理。

### 3、利用cookie记录session

session记录在客户端，每次请求服务器的时候，将session放在请求中发送给服务器，服务器处理完请求后再将修改后的session响应给客户端，**这里的客户端就是cookie。**

**缺点**

- 比如受cookie大小的限制，能记录的信息有限；
- 每次请求响应都需要传递cookie，影响性能，如果用户关闭cookie，访问就不正常。

**优点:**

cookie的简单易用，可用性高，支持应用服务器的线性伸缩，而大部分要记录的session信息比较小，因此事实上，许多网站或多或少的在使用cookie记录session。

#### 4、session服务器

**session服务器可以解决上面的所有的问题，利用独立部署的session服务器（集群）统一管理session，服务器每次读写session时，都访问session服务器。**

这种解决方案事实上是应用服务器的状态分离，分为无状态的应用服务器和有状态的session服务器，然后针对这两种服务器的不同特性分别设计架构。

对于有状态的session服务器，一种比较简单的方法是利用分布式缓存（redis、memcached），或者存储在数据库（如MySQL）等。在这些产品的基础上进行包装，使其符合session的存储和访问要求。

如果业务场景对session管理有比较高的要求，比如利用session服务实现单点登录（sso），用户服务器等功能，需要开发专门的session服务管理平台。

**session服务器使用场景：**集群中机器数多、网络环境复杂。

**优点：**可靠性好

**缺点：**实现复杂、稳定性依赖于缓存的稳定性、Session信息放入缓存时要有合理的策略写入。

## 2. 扩展内容

- 分布式系统全局唯一ID简介、特点、5种生成方式
- 单点登录的原理、来源、实现、以及技术方案比较
- 分布式锁的3种实现（数据库、缓存、Zookeeper）