

原 java初、中、高级面试题必备——javaWeb

2019年06月22日 18:15:33 在IT中穿梭旅行 阅读数 47

编辑

76.请问JAVA应用服务器都有那些?

考察点：服务器

参考回答：

BEA WebLogic Server, IBM WebSphere Application Server, jBoss, Tomcat, Jetty

Apache Tomcat

Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，支持最新的Servlet 和JSP 规范。因为Tomcat 技术先进、性能稳定，而且免费，成为目前比较流行的Web 应用服务器。

优点：

- 1、免费的开源代码
- 2、轻量级应用服务器，在中小型系统和并发访问用户不是很多的情况下被普遍使用
- 3、占用的系统资源小，启动快
- 4、扩展性好，支持负载均衡与邮件服务等开发应用系统常用的功能

Jetty

Jetty 是一个开源的WEB容器，它为JSP和servlet提供运行环境。使用Java语言编写的，它的API以一组JAR包的形式发布。开发人员可以将Jetty容器实例化成一个对象，为一些独立运行的Java应用提供网络和web连接。

优点：

- 1、免费的开源代码
- 2、轻量级，运行速度快。
- 3、易用。默认配置可以满足大部分的需求；将 Jetty 嵌入到应用程序当中只需要非常少的代码；
- 4、可扩展。Jetty 设计了非常良好的接口，用户可以非常方便地对 Jetty 的某些实现进行修改，使得 Jetty 适用于特殊的应用程序的需求。

5、易嵌入。Jetty 设计之初就是作为一个优秀的组件来设计的，这也就意味着 Jetty 可以非常容易的嵌入到应用程序当中。从某种程度上，你也可以把 Jetty 理解为一个嵌入式的Web服务器。

目前最新版本是：9.1.4

JBoss

JBoss不仅是一个基于J2EE的开放源代码的应用服务器。还是一个管理EJB的容器和服务，支持EJB 1.1、EJB 2.0和EJB3.0的规范。但JBoss核心服务不包括支持servlet/JSP的WEB容器，一般与Tomcat或Jetty绑定使用。

优点：

1. 标准的开源J2EE容器
2. 性能好，执行效率高
3. 支持集群，理论上无最大访问量限制一说。
4. 可伸缩性强。可伸缩性应该是架构选择的主要标准，所谓可伸缩性，只在小型系统、一台服务器情况下，我的系统也可以良好运转，多台服务器扩展后，我的系统只需通过增加硬件就可以实现性能扩展，无需修改太多软件。
5. 支持热部署
6. 目前JBOSSE的最新版本号为7.1。

BEA WebLogic (收费)

WebLogic是Oracle的主要产品之一，是商业市场上主要的Java (J2EE) 应用服务器软件 (application server) 之一，是世界上第一个成功商业化的J2EE应用服务器。它是一套基于JAVA功能强大的电子商务套件，提供了许多功能强大的中间件以方便编程人员编写的JSP、SERVLET 等电子商务应用，可以为企业提供一个完整的商务应用解决方案。

优点：

1、标准

对业内多种标准的全面支持，使Web应用系统的实施更为简单。

2、可扩展性

WebLogic Server以其高扩展的架构体系闻名于业内，包括客户机连接的共享、资源pooling以及动态网页和EJB组件群集。

3、快速开发

凭借对EJB和JSP的支持，以及BEA WebLogic Server 的Servlet组件架 构体系，可加速投放市场速度。

4、更趋灵活

BEA WebLogic Server的特点是与领先数据库、操作系统和Web服务器紧密集成。

5、可靠性

其容错、系统管理和安全性能已经在全球数以千记的关键任务环境中得以验证。

目前WEBLOGIC最新版本号为：12c

IBM WebSphere (收费)

WebSphere Application Server 是一种功能完善、开放的Web应用程序服务器，是IBM电子商务计划的核心部分。WebSphere 是随需应变的电子商务时代的最主要的软件平台。使用它，你可以开发、部署和整合新一代的电子商务应用，如B2B电子商务，并支持从简单的网页内容发布到企业级事务处理的商业应用。

目前WEBSPPHERE最新版本号为：8.5.5

总结：

开发大型的、开放的、需要安全级别较高的系统，选用WebLogic 或Websphere等商用应用服务器，

开发分布式的，而且仅仅在公司内部使用，那么你可以选择JBoss。

如果开发的系统较小型，不涉及分布式等，选择Tomcat

77.1分钟之内只能处理1000个请求，你怎么实现？

我知道的有两种方式可以实现：

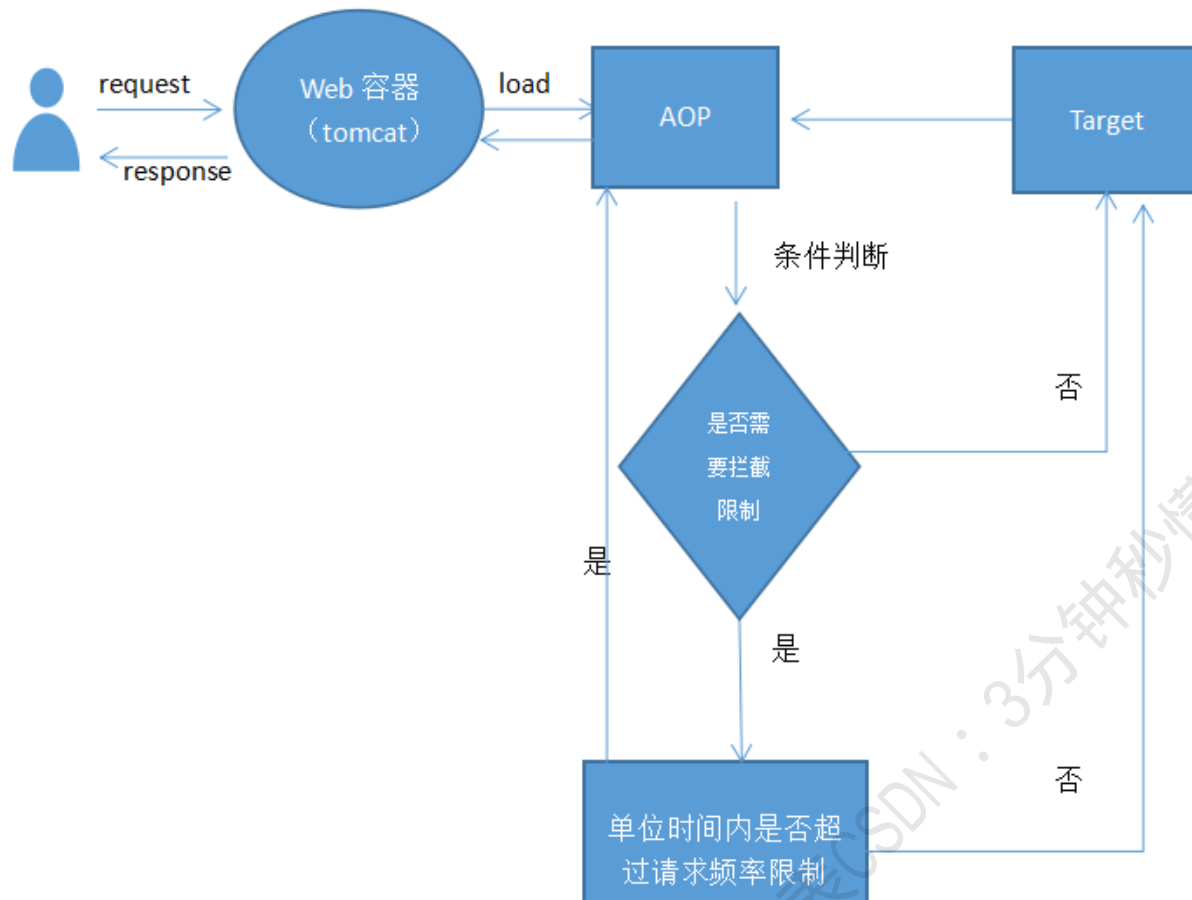
a). Application 对所有用户访问的次数计数。同时定义一个计时器，单位为一分钟。如果Application 中的用户在单位时间内超出请求次数，就拒绝处理该请求。一分钟再刷新application的值为0。

使用一个Map 维护变量：

```
1 // 泛型 String 表示用户标识，List 中存放用户不同请求的时间戳。  
2 private Map<String, List<Long>> map = new ConcurrentHashMap<>();
```

我们只需要在单位计数中判断 List中数量是否超出限制即可。

b). 使用 aop 实现请求的限制，在需要限制的请求方法上加上 aop 逻辑。即可实现，思路如下：



自定义注解类实现请求限制的拦截逻辑，在需要限制的方法上使用注解，超出限制后拒绝处理请求。

78.jsp 和 servlet 有什么区别？

相同点：JSP和Servlet本质上都是Java类。

不同点：JSP侧重于视图，Servlet主要用于控制逻辑

Servlet中没有内置对象,Jsp中的内置对象都是必须通过HttpServletRequest对象， HttpServletResponse对象以及HttpServlet对象获得。

联系：JSP是Servlet技术的扩展，本质上就是Servlet的简易方式。

79.jsp有哪些内置对象？这些对象的作用分别是什么？

JSP有9个内置对象：

- request：封装客户端的请求，其中包含来自GET或POST请求的参数；
- response：封装服务器对客户端的响应；
- pageContext：通过该对象可以获取其他对象；
- session：封装用户会话的对象；
- application：封装服务器运行环境的对象；
- out：输出服务器响应的输出流对象；
- config：Web应用的配置对象；
- page：JSP页面本身（相当于Java程序中的this）；
- exception：封装页面抛出异常的对象。

基于Java的Web开发很多都同时使用了Servlet和JSP。JSP页面其实是一个Servlet，能够运行Servlet的服务器（Servlet容器）通常也是JSP容器，可以提供JSP页面的运行环境，Tomcat就是一个Servlet/JSP容器。第一次请求一个JSP页面时，Servlet/JSP容器首先将JSP页面转换成一个JSP页面的实现类，这是一个实现了JspPage接口或其子接口HttpJspPage的Java类。JspPage接口是Servlet的子接口，因此每个JSP页面都是一个Servlet。转换成功后，容器会编译Servlet类，之后容器加载和实例化Java字节码，并执行它通常对Servlet所做的生命周期操作。对同一个JSP页面的后续请求，容器会查看这个JSP页面是否被修改过，如果修改过就会重新转换并重新编译并执行。如果没有则执行内存中已经存在的Servlet实例。

80.请说明一下jsp有哪些动作？这些动作的作用又分别是什么？

JSP共有以下6种基本动作 作用如下：

jsp:useBean：寻找或者实例化一个JavaBean。

jsp:setProperty：设置JavaBean的属性。

jsp:getProperty：输出某个JavaBean的属性。

jsp:plugin：根据浏览器类型为Java插件生成OBJECT或EMBED标记。

jsp:forward：把请求转到一个新的页面。

jsp:include：在页面被请求的时候引入一个文件。

includ两种方法的实现：

有两种实现方法，动态，静态。

动态：用于包含动态页面，并且可以随时检查页面的变化，采用jsp:include动作可以实现，例如：

```
<jsp:includepage="***.jsp" flush="true"/>
```

静态，适合于包含静态页面，不检查页面的变化，采用include伪码实现

```
<%@include file="***.html"%>
```

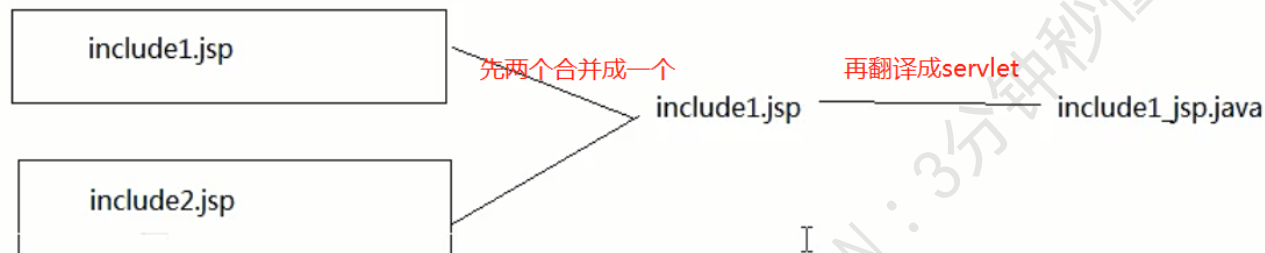
81.JSP中的静态包含和动态包含有什么区别？

在jsp当中经常需要嵌套其他的jsp页面，以提高代码利用率跟工作效率。

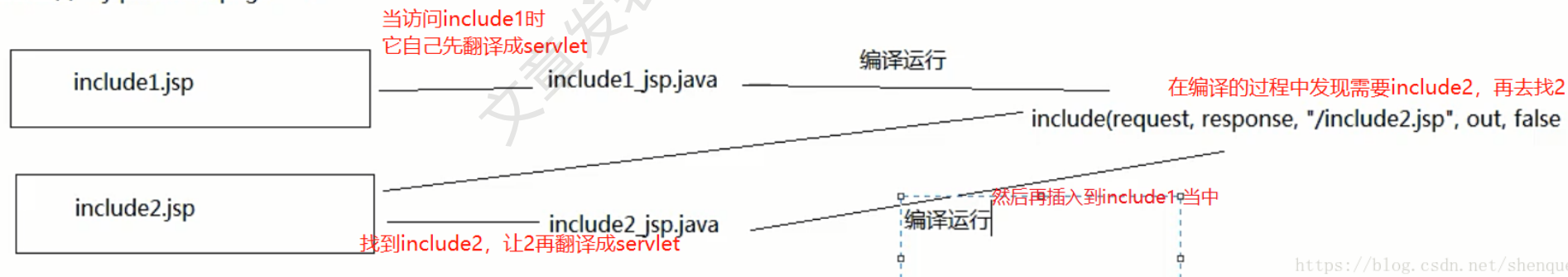
常用的jsp嵌套包括静态包含跟动态包含两种方式，那这两种方式有啥区别呢。

下面这张图很好的解释了两者的区别：

静态包含 <%@ include file=""%>



动态包含 <jsp:include page="">



<https://blog.csdn.net/shenqueying>

82.请谈谈你对Javaweb开发中的监听器的理解?

考察点：监听器

参考回答：

Java Web开发中的监听器（listener）就是application、session、request三个对象创建、销毁或者往其中添加修改删除属性时自动执行代码的功能组件，如下所示：

- ①ServletContextListener：对Servlet上下文的创建和销毁进行监听。
- ②ServletContextAttributeListener：监听Servlet上下文属性的添加、删除和替换。
- ③HttpSessionListener：对Session的创建和销毁进行监听。

session的销毁有两种情况：1). session超时（可以在web.xml中通过<session-config>/<session-timeout>标签配置超时时间）；2). 通过调用session对象的invalidate()方法使session失效。

- ④HttpSessionAttributeListener：对Session对象中属性的添加、删除和替换进行监听。
- ⑤ServletRequestListener：对请求对象的初始化和销毁进行监听。
- ⑥ServletRequestAttributeListener：对请求对象属性的添加、删除和替换进行监听。

83.请问过滤器有哪些作用？以及过滤器的用法又是什么呢？

考察点：过滤器

参考回答：

Java Web开发中的过滤器（filter）是从Servlet 2.3规范开始增加的功能，并在Servlet 2.4规范中得到增强。对Web应用来说，过滤器是一个驻留在服务器端的Web组件，它可以截取客户端和服务端之间的请求与响应信息，并对这些信息进行过滤。当Web容器接受到一个对资源的请求时，它将判断是否有过滤器与这个资源相关联。如果有，那么容器将把请求交给过滤器进行处理。在过滤器中，你可以改变请求的内容，或者重新设置请求的报头信息，然后再将请求发送给目标资源。当目标资源对请求作出响应时候，容器同样会将响应先转发给过滤器，在过滤器中你可以对响应的内容进行转换，然后再将响应发送到客户端。

常见的过滤器用途主要包括：对用户请求进行统一认证、对用户的访问请求进行记录和审核、对用户发送的数据进行过滤或替换、转换图象格式、对响应内容进行压缩以减少传输量、对请求或响应进行加解密处理、触发资源访问事件、对XML的输出应用XSLT等。

和过滤器相关的接口主要有：Filter、FilterConfig和FilterChain。

84.说说对HTTP协议有什么了解？以及8中请求方式

HTTP(超文本传输协议)是一个属于应用层的面向对象的协议,是一个客户端和服务端请求和应答的标准(TCP)。

http请求由三部分组成, 分别是: **请求行、请求头、请求体**

请求行: 请求行包含http请求方式, 请求资源名称, http版本 具体如下:

Method Request-URI HTTP/1.1 CRLF

其中 Method表示请求方法; Request-URI是一个统一资源标识符; HTTP/1.1表示请求的HTTP协议版本; CRLF表示回车和换行(除了作为结尾的CRLF外, 不允许出现单独的CR或LF字符)。

请求方法包含8种:

get	请求获取Request-URI所标识的 资源
post	在Request-URI所标识的资源后 附加新的数据
head	请求获取由Request-URI所标识的资源的 响应消息报头
put	请求服务器 存储一个资源 , 并用Request-URI作为其标识
delete	请求服务器 删除Request-URI所标识的资源
trace	请求服务器回送收到的 请求信息 , 主要用于测试或诊断
connect	保留将来使用
options	请求查询服务器的性能, 或者查询与资源相关的选项和需求

请求头: 请求头用于描述客户端请求哪台主机, 以及客户端的一些环境信息, 以键值对的形式传递

Accept text/html, application/xhtml+xml, */*

Accept-Language en-US,zh-CN;q=0.5

Host localhost:8888

请求体: 请求体代表着浏览器在post请求方式中传递给服务器的参数, 请求体中参数以键值形式传递, 多个用&链接, 服务器接收到后再解析

username=admin&userpwd=123

85.get和post的请求的区别:

①get请求用来从服务器上获得资源，而post是用来向服务器提交数据；

②get将表单中数据按照name=value的形式，添加到action 所指向的URL 后面，并且两者使用"?"连接，而各个变量之间使用"&"连接；post是将表单中的数据放在HTTP协议的请求头或消息体中，传递到action所指向URL；

③get传输的数据要受到URL长度限制（1024字节）；而post可以传输大量的数据，上传文件通常要使用post方式；

④使用get时参数会显示在地址栏上，如果这些数据不是敏感数据，那么可以使用get；对于敏感数据还是应用使用post；

⑤get使用MIME类型application/x-www-form-urlencoded的URL编码（也叫百分号编码）文本的格式传递参数，保证被传送的参数由遵循规范的文本组成，例如一个空格的编码是"%20"。

86.介绍一下cookie机制和session机制

cookie机制： cookie技术是客户端的解决方案，

1. 用户使用浏览器访问一个支持Cookie的网站，用户会提供包括用户名在内的个人信息并且提交至服务器；
2. 服务器在向客户端回传相应的超文本的同时也会发回这些个人信息，然后将这些个人信息存放于HTTP响应头（Response Header）；
3. 客户端浏览器接收到来自服务器的响应之后，对于Windows操作系统，浏览器会将这些信息存放在系统盘中的Cookies目录；
4. 客户端再次向服务器发送请求的时候，都会把相应的Cookie再次发回至服务器。而这次，Cookie信息则存放在HTTP请求头（Request Header）了。

session机制： session技术则是服务端的解决方案，通常都会把Session翻译成会话，我们可以把客户端浏览器与服务器之间一系列交互的动作称为一个 Session。

1. 服务器端程序运行时，调用HttpServletRequest的getSession方法创建session,服务器会为该Session生成唯一的Session id，而这个Session id在随后的请求中会被用来重新获得已经创建的Session；
2. 在Session被创建之后，就可以调用Session相关的方法往Session中增加内容了，而这些内容只会保存在服务器中，发到客户端的只有Session id；
3. 当客户端再次发送请求的时候，会将这个Session id带上，服务器接受到请求之后就会依据Session id找到相应的Session.

87.cookie 和session的区别：

1. cookie数据存放在客户的浏览器上，session数据放在服务器上。
2. cookie不是很安全，别人可以分析存放在本地的COOKIE并进行COOKIE欺骗 考虑到安全应当使用session。

3. session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能
4. 考虑到减轻服务器性能方面，应当使用COOKIE。
5. 单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。

88.在前端提交请求时，cookie中经常有个jsessionId，这个jsessionId是用来干什么的？

jsessionId就是客户端用来保存session id的变量，jsessionId是保存在内存cookie中的，在一般的cookie文件中是看不到它的影子的。内存cookie在打开一个浏览器窗口的时候会创建，在关闭这个浏览器窗口的时候也同时销毁。

session跟jsessionId紧密不可分割的联系。只有通过jsessionId才能使session机制起作用，而jsessionId又是通过cookie来保存

89.服务器一般保存了一个session,浏览器为什么知道我多次请求在一个session里面，为什么能找到我之前的session？

因为当在服务器中创建session的时候，服务器会为该Session生成唯一的Session id，并在该session id返回给客户端，然后通过

jsessionId来在客户端的内存中保存session id，每次客户端请求时都会把这个id传到服务器，就能找到之前的session。

90 请说一下表达式语言（EL）的隐式对象以及该对象的作用

考察点：EL

参考回答：

EL的隐式对象包括：pageContext、initParam（访问上下文参数）、param（访问请求参数）、paramValues、header（访问请求头）、headerValues、cookie（访问cookie）、applicationScope（访问application作用域）、sessionScope（访问session作用域）、requestScope（访问request作用域）、pageScope（访问page作用域）。

91.请说说你做过的项目中，使用过哪些JSTL标签？

考察点：JSTL

参考回答：

项目中主要使用了JSTL的核心标签库，包括<c:if>、<c:choose>、<c: when>、<c: otherwise>、<c:forEach>等，主要用于构造循环和分支结构以控制显示逻辑。

虽然JSTL标签库提供了core、sql、fmt、xml等标签库，但是实际开发中建议只使用核心标签库（core），而且最好只使用分支和循环标签并辅以表达式语言（EL），这样才能真正做到数据显示和业务逻辑的分离，这才是最佳实践。

92.请谈一谈，网站在架构上应当考虑哪些问题

- 分层：大型网站的软件系统都使用分层的理念将其分为持久层（提供数据存储和访问服务）、业务层（处理业务逻辑，系统中最核心的部分）和表示层（系统交互、视图展示）。需要指出的是：（1）分层是逻辑上的划分，在物理上可以位于同一设备上也可以在不同的设备上部署不同的功能模块，这样可以使用更多的计算资源来应对用户的并发访问；（2）层与层之间应当有清晰的边界，这样分层才有意义，才更利于软件的开发和维护。

- 分割：分割是对软件的纵向切分。大型网站是将不同功能和服务分割开，形成高内聚低耦合的功能模块（单元）。在设计初期可以做一个粗粒度的分割，将网站分割为若干个功能模块，后期还可以进一步对每个模块进行细粒度的分割，这样一方面有助于软件的开发和维护，另一方面有助于分布式的部署，提供网站的并发处理能力和功能的扩展。

- 分布式：除了上面提到的内容，网站的静态资源（JavaScript、CSS、图片等）也可以采用独立分布式部署并采用独立的域名，这样可以减轻应用服务器的负载压力，也使得浏览器对资源的加载更快。数据的存取也应该是分布式的，传统的商业级关系型数据库产品基本上都支持分布式部署，而新生的NoSQL产品几乎都是分布式的。当然，网站后台的业务处理也要使用分布式技术，例如查询索引的构建、数据分析等，这些业务计算规模庞大，可以使用Hadoop以及MapReduce分布式计算框架来处理。

- 集群：集群使得有更多的服务器提供相同的服务，可以更好的提供对并发的支持。

- 缓存：所谓缓存就是用空间换取时间的技术，将数据尽可能放在距离计算最近的位置。使用缓存是网站优化的第一定律。我们通常说的CDN、反向代理、热点数据都是对缓存技术的使用。

- 异步：异步是实现软件实体之间解耦合的又一重要手段。异步架构是典型的生产者消费者模式，二者之间没有直接的调用关系，只要保持数据结构不变，彼此功能实现可以随意变化而不互相影响，这对网站的扩展非常有利。使用异步处理还可以提高系统可用性，加快网站的响应速度（用Ajax加载数据就是一种异步技术），同时还可以起到削峰作用（应对瞬时高并发）。"能推迟处理的都要推迟处理"是网站优化的第二定律，而异步是践行网站优化第二定律的重要手段。

- 冗余：各种服务器都要提供相应的冗余服务器以便在某台或某些服务器宕机时还能保证网站可以正常工作，同时也提供了灾难恢复的可能性。冗余是网站高可用性的重要保证。

下一章节：https://blog.csdn.net/weixin_38201936/article/details/93378748

