Task 11

<1> Web 应用性能分析

随着近年来国内互联网的高速发展，网络上 web 应用也呈现爆炸性增长，电子商务飞速发展，如何才能为用户提供满意的服务性能保障也成了一个新的研究课题。

在客户端向服务器发送请求到服务器响应客户端请求的这个过程中，大概经历了以下三个过程的时间：①数据在网络上传播的时间。②服务器处理请求并生成回应数据的时间。③浏览器本地计算和渲染的时间。

以下就性能方面的因素，考虑项目中可以采取的策略：

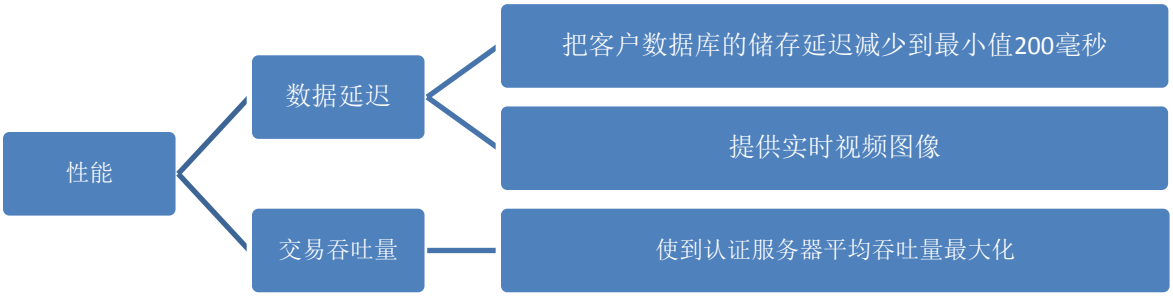
Web 的性能指标

连接时间	事物响应时间
发送时间	吞吐量
接收时间	吞吐率（Throughput）
处理时间	每秒事物处理量（TPS，Transaction Per Second）
响应时间	点击率(Hit Per Second)
并发用户	资源利用率
用户并发数量	请求响应时间

Web 应用性能提升策略

增加带宽	页面组件分离
减少 Web 页面中 Http 请求	提升 web 服务器性能
加快服务器脚本计算速度	使用负载均衡
采用缓存技术	优化数据库
采用代理服务器	优化应用程序
动态内容静态化	使用 CDN
页面容量优化	采用镜像分担流量

这里举个例子,给出一个简单的 SA 性能效用树，后面我们会具体分析，有关 web 应用的性能：



（1）性能方面的因素：

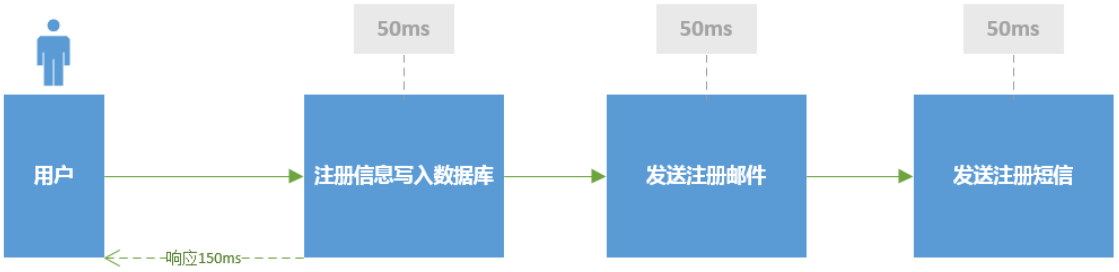
web 服务的一个性能瓶颈就是 Http 对其的影响。由于底层消息传递和传输协议的局限性，Web 服务会遇到性能瓶颈。然而，对公众普遍接受的协议（例如 HTTP 和 SOAP）的依赖却使它们成了必须承担的永久的负担。

Http 是一种尽力而为的传输服务，是一个无状态的数据转发机制，它对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。
随着运行在网络上的用户和数据量的增加和电子商务的飞速发展，在有限带宽和网络资源的条件下，Http 显然是一个制约 Web 服务性能的一个瓶颈，Http 协议无法为 Web 服务器提供区分服务和性能保证。

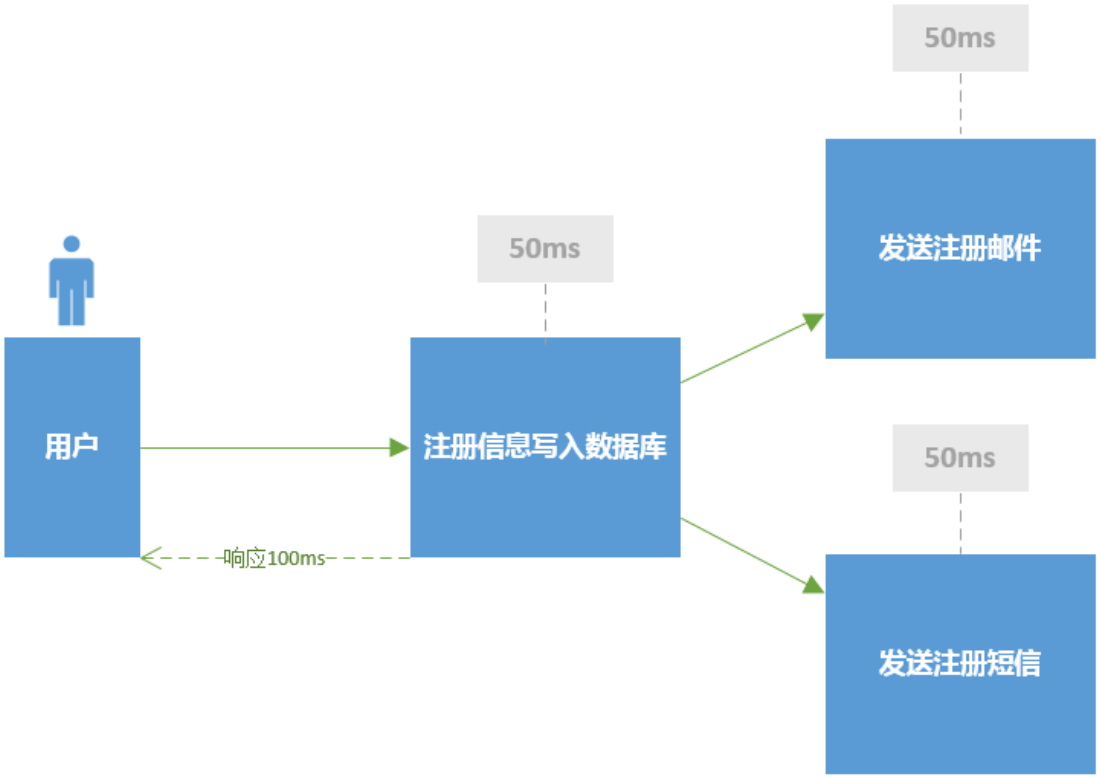
（2）项目中可以采取的策略：

1. 使用异步消息队列
   习惯上，许多应用程序使用同步消息传递，当应用程序仅在一台计算机上运行的时候，组件通信的延迟以几毫米计。但是，对于 Web 服务来说，是通过因特网进行通信，这意味着延迟要以几十、几百甚至几千微秒计。所以我们使用消息对列，消息队列有两个主要优势：
   (1)它是异步的：一个消息传递服务提供者可以在消息到达时向请求者传递消息，请求者不必为接收消息而请求消息。那些实时性要求不高，且比较耗时的任务，是队列的最佳应用场景。
   (2)它是可靠的：消息传递服务可以确保一条消息被传递一次，且仅传递一次。

   举个用户注册的例子：
   <1> 串行方式：将注册信息写入数据库成功后，发送注册邮件，再发送注册短信。以上三个任务全部完成后，返回给客户端



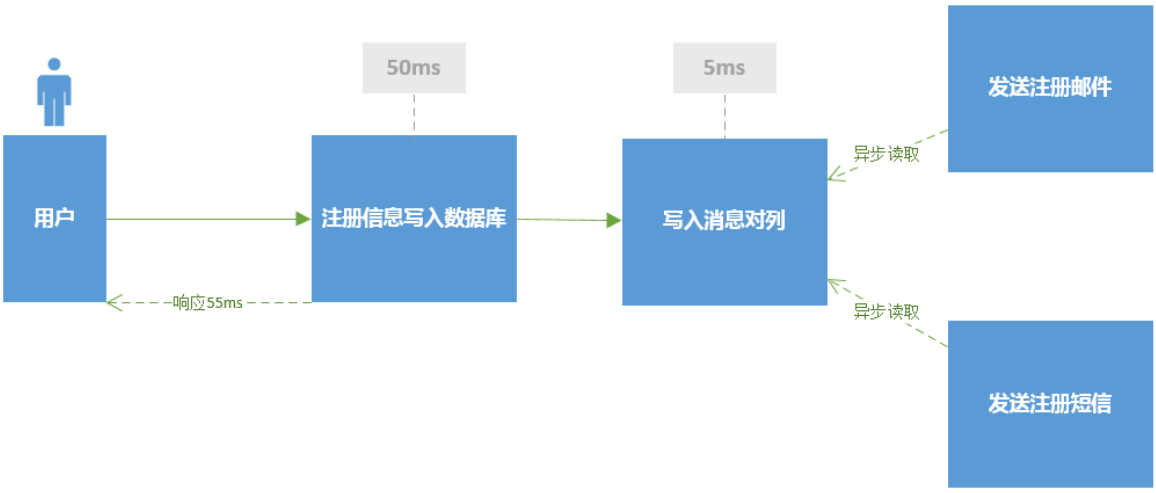
<2> 并行方式：将注册信息写入数据库成功后，发送注册邮件的同时，发送注册短信。以上三个任务完成后，返回给客户端。与串行的差别是，并行的方式可以提高处理的时间



假设三个业务节点每个使用 50 毫秒钟，不考虑网络等其他开销，则串行方式的时间是 150 毫秒，并行的时间可能是 100 毫秒。

因为 CPU 在单位时间内处理的请求数是一定的，假设 CPU1 秒内吞吐量是 100 次。则串行方式 1 秒内 CPU 可处理的请求量是 7 次（1000/150）。并行方式处理的请求量是 10 次（1000/100）

<3> 引入消息队列，将不是必须的业务逻辑，异步处理。改造后的架构如下：

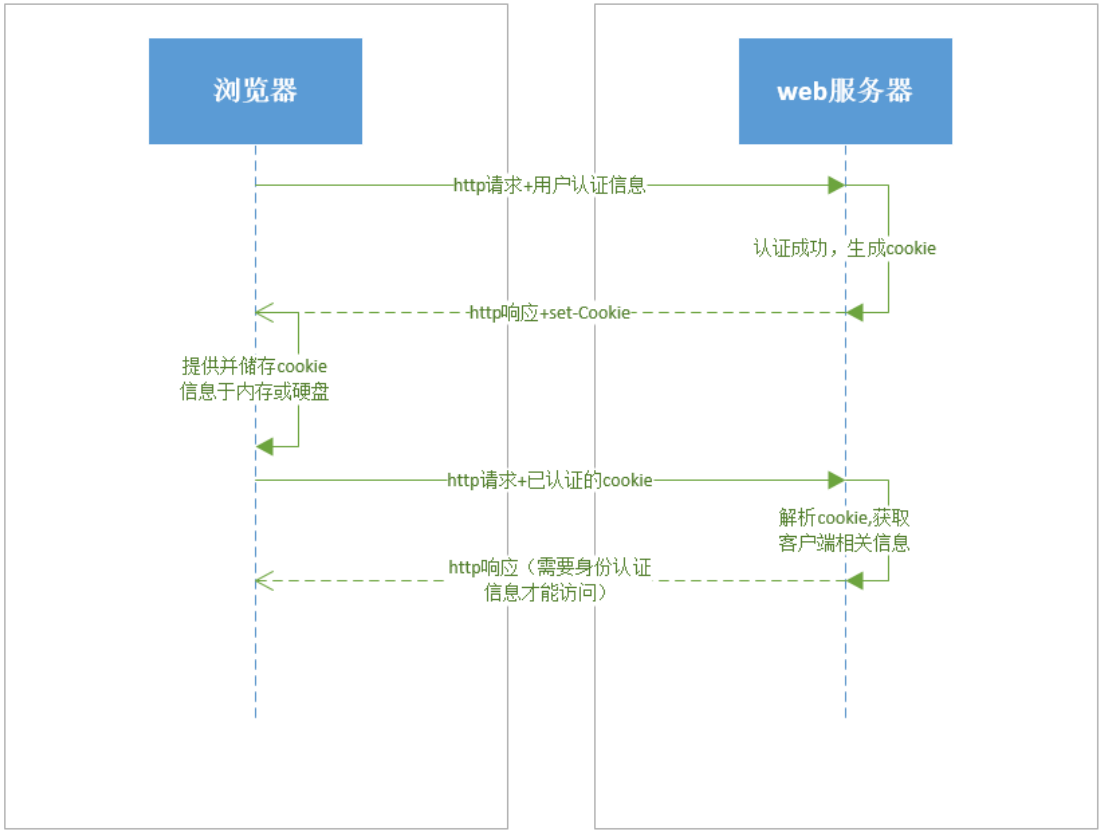


按照以上约定，用户的响应时间相当于是注册信息写入数据库的时间，也就是 50 毫秒。注册邮件，发送短信写入消息队列后，直接返回，因此写入消息队列的速度很快，基本可以忽略，因此用户的响应时间可能是 50 毫秒。因此架构改变后，系统的吞吐量提高到每秒 20 QPS。比串行提高了 3 倍，比并行提高了两倍

2. 根据不同的用户分类

   要对客户进行分类，客户可以按服务器的要求输入一定的信息，服务器以此来判断客户的身份。一种方法是用客户的 IP 地址来区分客户，这种方法是在服务质量 Web 服务器模型中，服务器可以对客户的服务请求设定不同的服务等级，按照预先定义的资源分配策略对客户的服务请求作出响应。

   这种方法具有占用带宽小，容易实现，客户等等时延小的优点。



另外一种方法是基于 Http Cookie 的分类，它是将 Web Cookie 嵌入 Http 请求内，以表明客户所属的类别。HTTP 请求中的 Cookie 是可以由服务器发送给浏览器的唯一标识符，它可以内嵌在 HTTP 请求中，用来表示不同的服务级别。

服务商可以给某个特定的服务提供一个永久的 Cookie 以供给用户使用。这样，就可以为付费用户和免费用户设置不同的优先级，利用 cookie 来识别用户信息,记录下用户在一段时间内的访问倾向,例如经常浏览哪一类的网页,或者常常购买哪一类的商品;并将有相同兴趣的用户归类分组。

当用户访问网站时,服务器可以根据他们的兴趣倾向推荐他们可能接受的网页,而且可以预测用户将来可能的行为,以此来提高服务质量。

3. 缓存静态和动态的内容

缓存可以通过加速内容的传输速度来提高 web 应用的性能。它可以采用以下几种策略：当需要的时候预处理要传输的内容，保存数据到速度更快的设备，把数据存储在距离客户端更近的位置，或者将这几种方法结合起来使用。

有两种不同类型数据的缓存：

（1）静态内容缓存。不经常变化的文件，比如图像(JPEG、PNG) 和代码(CSS,JavaScript)，可以保存在外围服务器上，这样就可以快速的从内存和磁盘上提取。

（2）动态内容缓存。很多 web 应用会针对每次网页请求生成一个新的 HTML 页面。在短时间内简单的缓存生成的 HTML 内容，就可以很好的减少要生成的内容的数量，而且这些页面足够新，可以满足用户的需要。

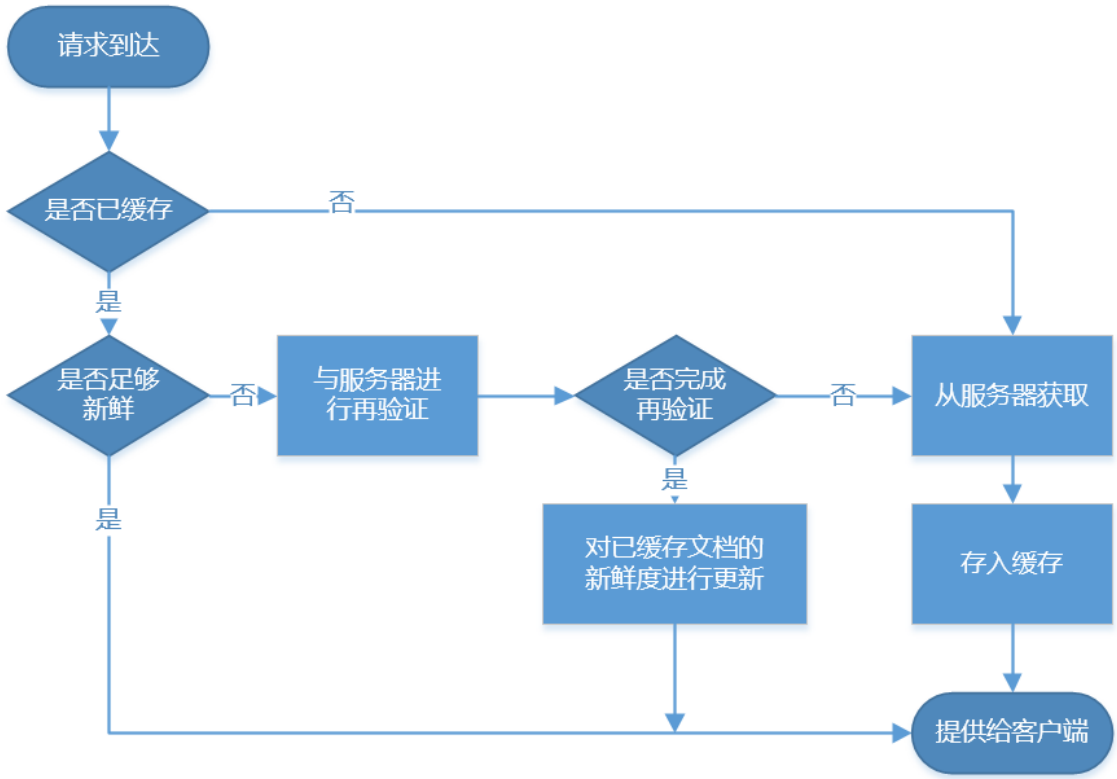
基于缓存所处的位置不同，可以把缓存技术分为三类：客户端缓存技术、基于代理服务器的缓存技术和服务端的缓存技术。

web 应用又有三种主要的缓存技术：

<1> 缩短数据与用户的网络距离。把一份内容的拷贝放的离用户更近的节点来减少传输时间。

<2> 提高内容服务器的速度。内容可以保存在一个更快的服务器上来减少提取文件的时间。

<3> 从过载服务器上移走数据。机器经常因为要完成某些其它的任务而造成某个任务的执行速度比测试结果要差。将数据缓存在不同的机器上可以提高缓存资源和非缓存资源的性能，而这是因为主机没有被过度使用。



对 web 应用的缓存机制可以在 web 应用服务器内部实现。首先，缓存动态内容是用来减少应用服务器加载动态内容的时间。其次，缓存静态内容（包括动态内容的临时拷贝）是为了更进一步的分担应用服务器的负载。而且缓存之后会从应用服务器转移到对用户而言更快、更近的机器，从而减少应用服务器的压力，减少提取数据和传输数据的时间。

改进过的缓存方案可以极大的提高应用的速度。对于大多数网页来说，静态数据，比如大图像文件，构成了超过一半的内容。如果没有缓存，那么这可能会花费几秒的时间来提取和传输这类数据，但是采用了缓存之后不到 1 秒就可以完成。

4. 使用 CDN

分布式资源体系结构就是 CDN（Content Delivery Network,资源分发网络），是指一份资源可以拥有多份复制件，而这些复制件分布在 Internet 上的不同地方，其目的在于针对用户所在的位置选取从最合适（如最近或最空闲）的地点提供服务。

分布式资源体系结构需要解决两个主要问题：①如何针对用户选取最佳的资源服务器；②如何在资源服务器之间传送资源或资源的索引。目前后者的一种解决方案是每个资源服务器维护一份独立的资源复制件，该资源的权威服务器定期对镜像服务器进行更新；另一种方案是形成资源交换网络，同路由器之间交换路由信息一样，交换网络中间的结点交换的是资源对象的信息。每个结点保存的不是资源的复制件，而是可以最快获取该资源的路由，用户对该资源的请求会被转发到该处。

<2>web 可用性分析

优秀的互联网 web 应用离不开良好的用户体验，随着用户和开发人员对 web 的可用性的重视，UED（user experience design）这个词也成为前端开发的热门。

用户体验，表明了可用性对产品生命力的重要性，而什么是可用性，可以概括为就是强调优越的用户体验，一切为了用户，为了一切用户，为了用户的一切。具体而言，即是 Web 应用在特定使用环境中为特定目标所使用，从而快速、有效、满意地完成特定任务的程度。一个高可用性的用户界面让用户全神贯注于正在进行的工作，而不用花费心思考虑如何使用该 Web 应用。

根据 Nielsen 的定义，web 的可用性属性如下：

可用性属性	描述指标
易学性和易记性	新用户能多块学会有效地完成一些基本工作；老用户能否清晰地记得如何再次有效使用，还是必须重新学习所有的东西
有效性	能否帮助用户准确地找到所要的信息，或者是用户通过 web 应用完成特定任务的准确程度；能否让用户完整地找到自己所要的信息或实现特定目的
效率	用户使用该 web 应用能多块地完成任务
容错度	在使用 web 应用的过程中，用户出错的频率是多少，这些错误有多严重，用户是如何从错误中回复的
用户满意度	用户喜欢使用该 web 应用的程度

Steve Krug 的 web 可用性三大定律

（1） 不要让我思考（Don’t make me think）。

（2） 点击多少次都没关系，只要每次点击都是无须思考且明确无误的选择。

（3） 去掉每个页面上一半的文字，然后把剩下的再去掉一半。

为了达到 Web 应用的高可用性，应当遵循的原则：

别让用户思考	尽量简洁
别浪费用户的耐心	别怕留白
抓住用户的注意	用“可视化”语言有效交流
尽量使特征明显呈现	规范是我们的朋友
有效书写	早测常测

Web 应用的可用性，具体来说，主要体现在以下几个方面：

（1）Web 应用的设计能够使用户把知觉和思维集中在自己的任务上，可以按照自己的行动过程进行操作，不必分心在寻找人机界面的菜单或理解 Web 应用结构、人机界面的结构和图标含义上，不必分析考虑如何把自己的任务转换成计算机的输入方式和输入过程。

（2）用户不必 记忆面向计算机软硬件的知识。

（3）用户不必为键盘鼠标的操作分心，操作动作简单重复。

（4）在非正常环境和情景时，用户仍然能够正常进行操作。

（5）用户理解和操作出错较少。

（6）用户学习操作的时间较短。