1、tomcat 的优化方式？

回答：Tomcat 的优化我准备从五方面来说：

第一部分：内存优化

Tomcat 的默认内存配置比较低，不用说大项目，就算是小项目，并发量达到一定程度也就可能会抛出 OutOfMemoryError 异常，

为了解决这个问题，我们要修改 JVM 的一些配置，在 tomcat 的 bin 目录下的 catalina 配置文件中，配置 Xms 和 Xmx，也就是

Java 虚拟机初始化时堆的最小内存和最大内存，这俩值通常会配置成一样，这样 GC 不必再为扩展内存空间而消耗性能.

除了这两个，还可以配置 XX:PermSize 和 XX:MaxPermSize，它们是 Java 虚拟机永久代大小和最大值，除了这几个参数

还可以再根据具体需要配置其他参数。

第二部分：配置优化

配置优化，主要有五方面：

1. Connector 优化

Connector 是连接器，它负责接收客户的请求，以及向客户端回送响应的消息。

默认情况下 Tomcat 只支持 200 线程访问，超过这个数量的连接将被等待甚至超时放弃，所以我们需要提高这方面的处理能力.

修改这部分配置需要修改 conf 下的 server.xml，找到 Connector 标签项，修改 protocol，默认的协议类型是 BIO，也就是阻塞式 I/O 操作，

简单项目及在用可以采用 BIO.

第二种协议类型是 NIO，它就是一种基于缓冲区的、并能提供非阻塞 I/O 操作的 java API，它有更好的并发运行性能. NIO 更适合后台需要耗时完成请求的操作

第五种协议类型是 APR，它主要可以提高 Tomcat 对静态文件的处理性能.

选择哪个协议也是根据实际项目进行配置.

除了这个协议类型，还有一个非常重要的参数要改，就是 maxThreads，就是当前连接器能够处理同时请求的最大数目.这个数目也并非

越大越好，它也受操作系统等硬件制约，所以这个值要根据压力测试后实际数据进行配置.

2. 线程池

使用线程池的好处在于减少了创建销毁线程的相关消耗，而且可以提高线程的使用效率。使用线程池就在 Service 标签中配置 Executor 就可以了

3. Listener

还有一个影响 tomcat 性能的因素是内存泄漏，我们在 Server 标签中配置一个 JreMemoryLeakPreventionListener 就可以用来预防 JRE 内存泄漏

第五部分：组件优化

可以选用 Tomcat Native 组件，它可以让 Tomcat 使用 Apache 的 APR 包来处理包括文件和网络 IO 操作，从而提升性能及兼容性

2、http 协议有哪些部分组成？

回答：1.请求部分：1）请求行：请求方式 路径 协议及版本 2）请求头：请求

头中保存的是本地浏览器信息的，是发送到服务器，被服务器解析的 3）请求体：请求体中存储的是请求数据，请求方式是 POST 时，请求体中才有内容；

2. 响应部分：1）响应行：协议及版本 状态码 状态码描述 2）响应头：包含了服务器信息以及响应内容信息，被浏览器解析的 3）响应体：存储响应数据，给一般用户看的

3、Get 和 Post 的区别？

回答：Get 请求方式：地址栏里会显示我们提交的数据（不安全），并且地址栏中支持提交少量数据，请求的数据存在请求行中；

Post 请求方式：地址栏里不显示我们提交的数据信息（相对安全），可以提交大量数据，请求的数据存在请求正文中。

4、cookie 和 session 的区别？

回答：共同点：cookie 和 session 都是用来跟踪浏览器用户身份的会话方式；

区别：cookie 数据保存在客户端，保存数据不安全且存储数据量有限；session 数据保存在服务器端，保存数据安全且存储数据量大，session 是基于 cookie 进行信息处理的；

5、什么是 ajax，为什么要使用 ajax？

回答：Ajax 是一种创建交互式网页应用的网页开发技术；Asynchronous JavaScript and XML 的缩写；

Ajax 的优势：

1. 通过异步模式，提升了用户体验；
2. 优化了浏览器和服务器之间的传输，减少不必要的数据往返，减少了宽带占用；
3. Ajax 引擎在客户端运行，承担了一部分本来由服务器承担的工作，从而减少了大用户量下的服务器负载；

Ajax 的最大特点：可以实现局部刷新，在不更新整个页面的前提下维护数据，提升用户体验度。

注意：ajax 在实际项目开发中使用率非常高（牢固掌握）。

6、浅谈你对 ajax 的认识？

回答：同上（第 5 个问题）

7、Cookie 和 Session 以及 Servlet 的生命周期？

回答：1. Cookie 的生命周期是累计的，从创建时，就开始计时，20 分钟后，cookie 生命周期结束。

2. Session 的生命周期是间隔的，从创建时，开始计时如在 20 分钟，没有访问 session，那么 session 生命周期被销毁；但是，如果在 20 分钟内（如在第 19 分钟时）访问过 session，那么将重新计算 session 的生命周期。注意：关机会造成 session 生命周期的结束，但是对 cookie 没有影响。

3. init（）：在 Servlet 的生命周期中，仅执行一次 init（）方法。它是在服务器装入 Servlet 时执行的，负责初始化 Servlet 对象。可以配置服务器，以在启动服务器或客户机首次访问 Servlet 时装入 Servlet。无论有多少客户机访问

Servlet，都不会重复执行 init()。

service()：它是 Servlet 的核心，负责响应客户的请求。每当一个客户请求一个 HttpServlet 对象，该对象的 Service() 方法就要调用，而且传递给这个方法一个"请求"(ServletRequest)对象和一个"响应"(ServletResponse)对象作为参数。在 HttpServlet 中已存在 Service() 方法。默认的服务功能是调用与 HTTP 请求的方法相应的 do 功能。

destroy()：仅执行一次，在服务器端停止且卸载 Servlet 时执行该方法。当 Servlet 对象退出生命周期时，负责释放占用的资源。一个 Servlet 在运行 service() 方法时可能会产生其他的线程，因此需要确认在调用 destroy() 方法时，这些线程已经终止或完成。

8、说一下你熟悉的常用 Linux 命令？

回答：

1. 列出文件列表：ls
2. 创建目录和移除目录：mkdir rmdir
3. 用于显示文件后几行内容：tail
4. 打包：tar -xvf
5. 打包并压缩：tar -zcvf
6. 查找字符串：grep
7. 显示当前所在目录：pwd
8. 创建空文件：touch
9. 编辑器：vim vi

10、后台传过的 json 数据前台怎么接收？

回答：在前台可以使用 js 代码接收，也可以通过 ajax 接收，也有专门的前端框架接收

11、后台传过来一个集合前台怎么接收？

回答：使用 el 表达式或者 ognl 表达式，或者根据实际情况从域中取数据