

面试常问：为什么 Vite 速度比 Webpack 快？

前言

最近作者在学习 `webpack` 相关的知识，之前一直对这个问题不是特别了解，甚至讲不出个123....，这个问题在面试中也是常见的，作者在学习的过程当中总结了以下几点，在这里分享给大家看一下，当然最重要的是要理解，这样回答的时候就不用死记硬背了。

原因

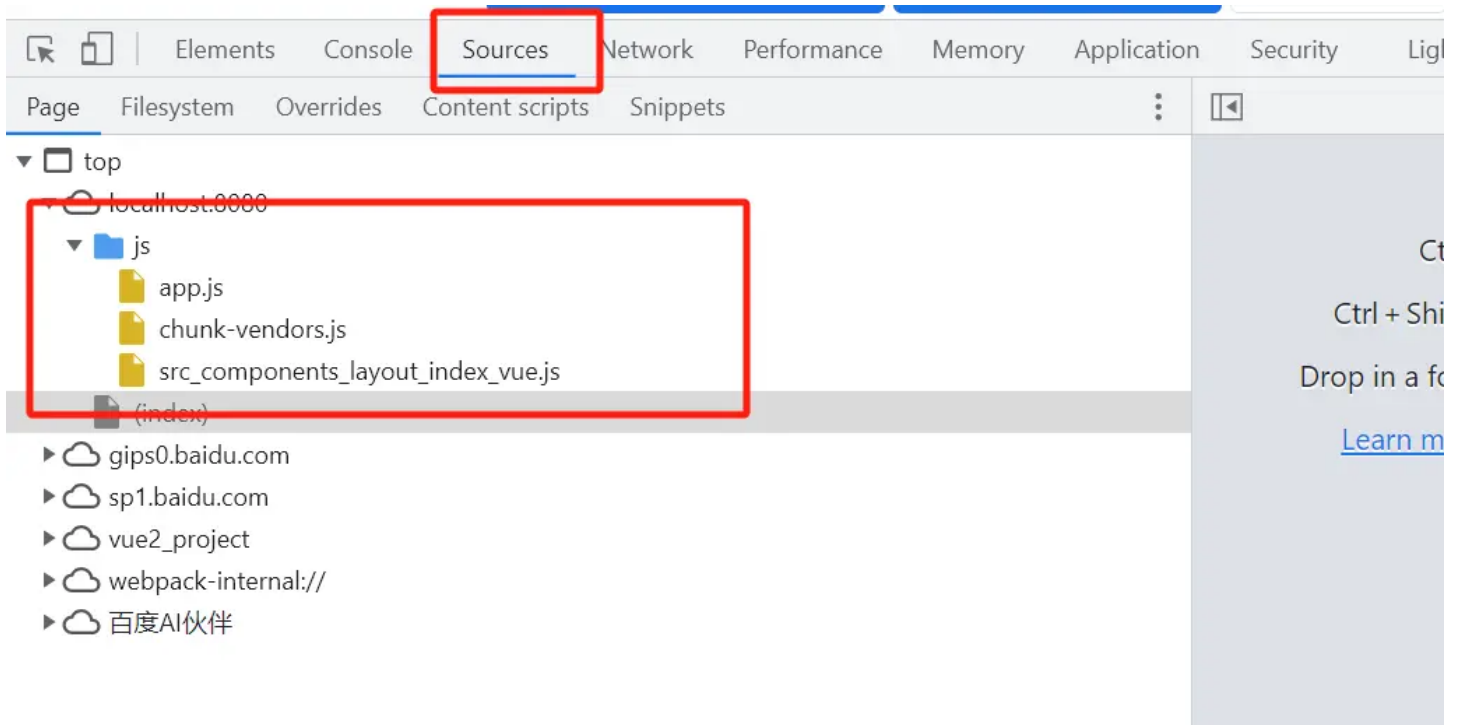
1、开发模式的差异

在开发环境中，`Webpack` 是先打包再启动开发服务器，而 `Vite` 则是直接启动，然后再按需编译依赖文件。（大家可以启动项目后检查源码 `Sources` 那里看到）

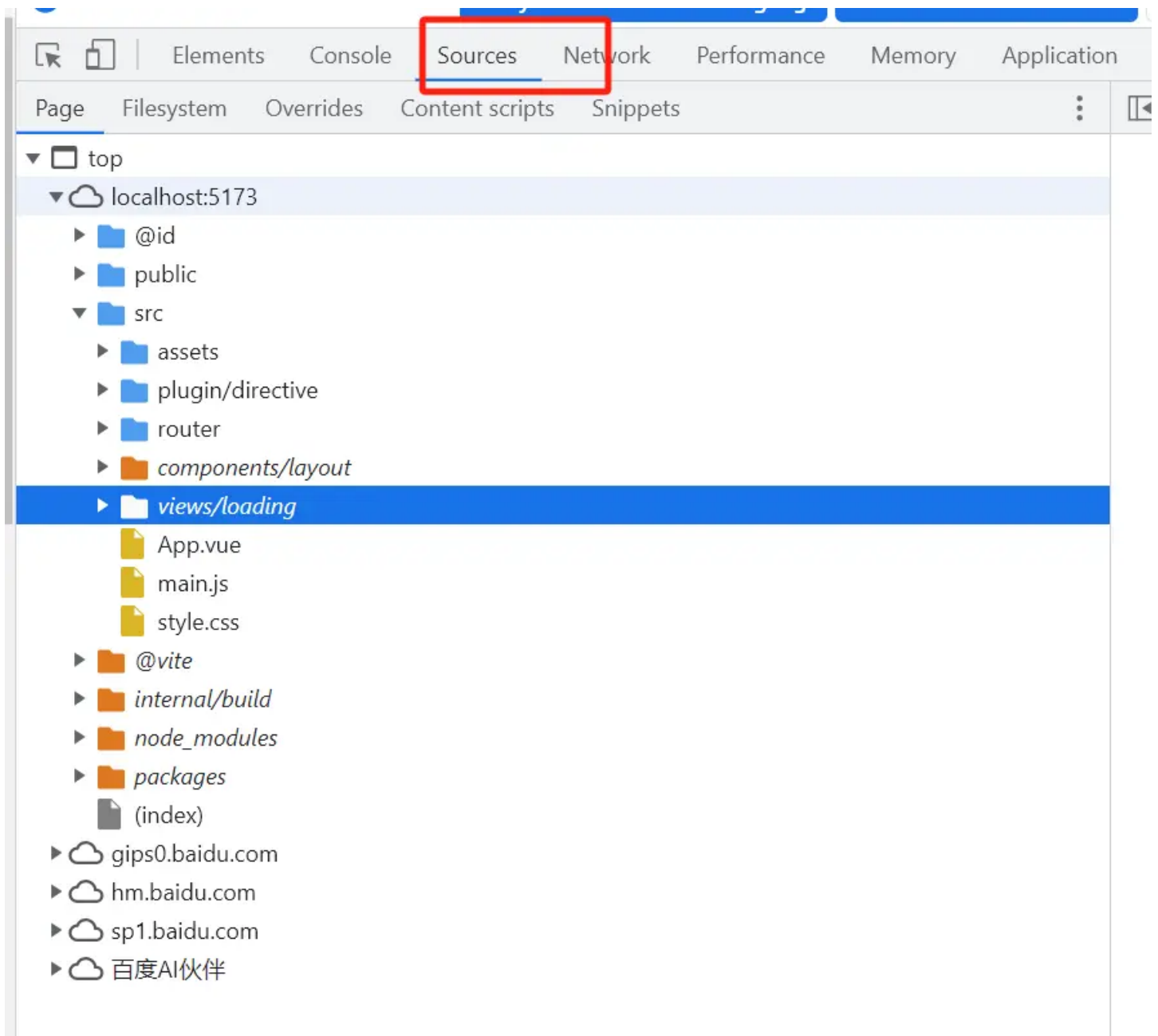
这意味着，当使用 `Webpack` 时，所有的模块都需要在开发前进行打包，这会增加启动时间和构建时间。

而 `Vite` 则采用了不同的策略，它会在请求模块时再进行实时编译，这种按需动态编译的模式极大地缩短了编译时间，特别是在大型项目中，文件数量众多，`Vite` 的优势更为明显。

Webpack启动



Vite启动



2、对ES Modules的支持

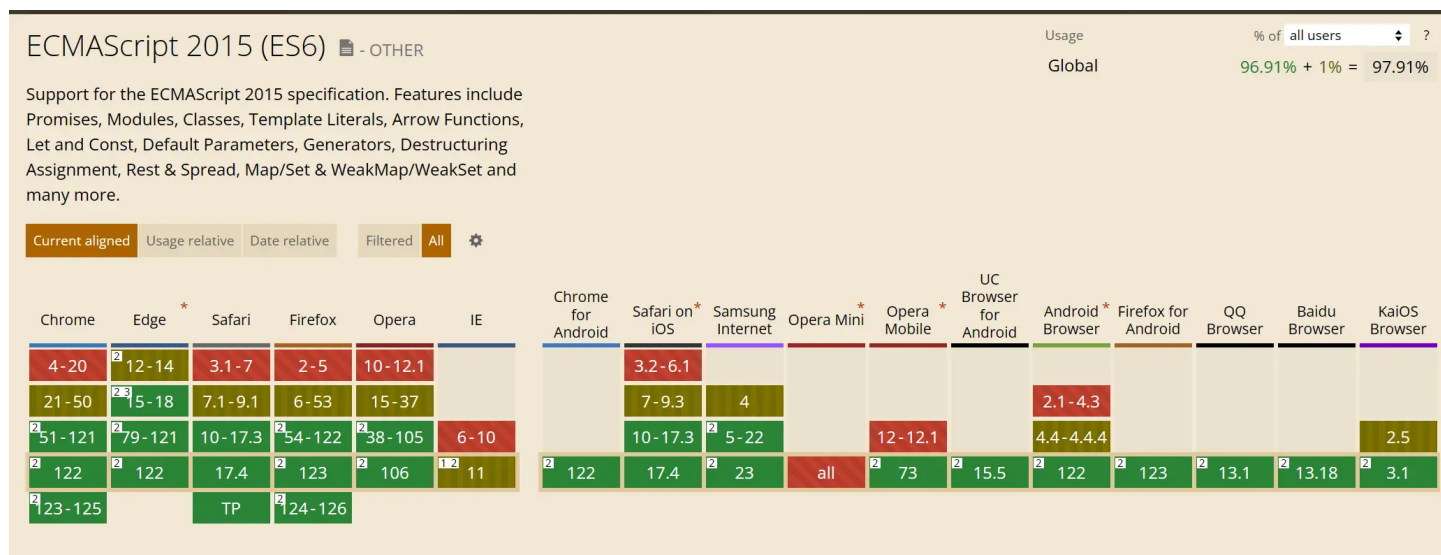
现代浏览器本身就支持 **ES Modules**，会 **主动发起** 请求去获取所需文件。Vite充分利用了这一点，将开发环境下的模块文件直接作为浏览器要执行的文件，而不是像 Webpack 那样 **先打包**，再交给浏览器执行。这种方式减少了中间环节，提高了效率。

什么是ES Modules?

通过使用 **export** 和 **import** 语句，ES Modules 允许在浏览器端导入和导出模块。

当使用 ES Modules 进行开发时，开发者实际上是在构建一个 **依赖关系图**，不同依赖项之间通过导入语句进行关联。

主流浏览器（除IE外）均支持ES Modules，并且可以通过在 script 标签中设置 **type="module"** 来加载模块。默认情况下，模块会延迟加载，执行时机在文档解析之后，触发DOMContentLoaded事件前。



3、底层语言的差异

Webpack 是基于 `Node.js` 构建的，而 Vite 则是基于 `esbuild` 进行预构建依赖。`esbuild` 是采用 `Go` 语言编写的，Go 语言是 `纳秒` 级别的，而 Node.js 是 `毫秒` 级别的。因此，Vite 在打包速度上相比 Webpack 有 `10-100` 倍的提升。

什么是预构建依赖？

预构建依赖通常指的是在项目 `启动或构建` 之前，对项目中所需的依赖项进行预先的 `处理或构建`。这样做的好处在于，当项目实际运行时，可以 `直接使用` 这些已经预构建好的依赖，而无需再进行实时的编译或构建，从而提高了应用程序的运行速度和效率。

4、热更新的处理

在 Webpack 中，当一个模块或其依赖的模块内容改变时，需要 `重新编译` 这些模块。

而在 Vite 中，当某个模块内容改变时，只需要让浏览器 `重新请求` 该模块即可，这大大减少了热更新的时间。

总结

总的来说，Vite 之所以比 Webpack 快，主要是因为它采用了 `不同的开发模式`、`充分利用了现代浏览器的 ES Modules 支持`、`使用了更高效的底层语言`，`并优化了热更新的处理`。这些特点使得 Vite 在大型项目中具有显著的优势，能够快速启动和构建，提高开发效率。