

# 如何通俗地理解「分布式系统」；Vue 是否可以在一个项目中使用多个 UI 框架；大厂上线流程：先上前端还是后端 | 极客观点

SegmentFault思否 2022-10-08 12:18 发表于山东

**#极客观点** 聚焦于技术方向、程序员职业发展、个人成长等主题，致力于发起有价值的讨论，输出有价值的观点。

在本栏目中，我们将为大家推荐在 **#极客观点** 版块被热烈讨论的话题，甄选出有趣的观点为你呈现。期待我们一起成长和进步呀 🥰🥰

今日关键词：**#分布式系统 #UI 框架 #大厂上线流程**

## 如何通俗地理解「分布式系统」？

话题发起人：Alluxio

如何通俗地理解「分布式系统」，它解决了哪些问题，有什么优缺点？

有趣的观点：

关于“分布式系统”的定义，《分布式系统原理和范型》一书中是这样定义分布式系统的：“分布式系统是若干独立计算机的集合，这些计算机对于用户来说就像是单个相关系统”。关于这个定义，我们直观的感受就是：首先，这种系统相对来说比较牛逼，起码由好几台主机组成。以谷歌、亚马逊等服务商而言，他们的数据中心都由上万台主机支撑起来的。其次，虽然很牛逼，但对于外人来说，是感觉不到这些主机的存在。也就是说，我们只看到是一个系统在运作。

"宕机事件"为例，平时，我们压根不知道所提供的服务背后是由多少台主机组成，但是等到宕机才知道，是多么严重的事故。从进程角度看，两个程序分别运行在两个台主机的进程上，它们相互协作最终完成同一个服务（或者功能），那么理论上这两个程序所组成的系统，也可以称作是“分布式系统”。当然，这个两个程序可以是不同的程序，也可以是相同的程序。如果是

相同的程序，我们又可以称之为“集群”。所谓集群，就是将相同的程序，通过不断横向扩展，以提高服务能力的方式。“分布式系统”和“集群”的定义够都简单吧。分布式系统有哪些优势那么，为啥我们要用分布式系统？说起分布式系统，我们就不得不说下分布式系统的祖先——集中式系统。集中式系统跟分布式系统是完全相反的两个概念。集中式系统就是把所有的程序、功能都集中到一台主机上，从而往外提供服务的方式。集中式系统最容易理解了。比如，我们主机的 PC 电脑，或者手机，我们把各种软件都安装在一台机子上，当我需要什么功能，我就从这台机子上去获取。再比如，我们在学生时代做的课程设计或者开发时的小应用，我们把 Web 服务器、数据库等都会安装到一台电脑上。好处是，易于理解、方便维护，想要的东西我都放到了一个地方，东西好找啊。当然弊端也是显而易见的，如果这台机子崩了，或者硬盘坏了，那相当与整个系统就奔溃了，而且如果备份也是在这个硬盘上，那相当于招了灭顶之灾。

有个名言，就是“不要把鸡蛋放在一个篮子里”。对于系统而言也是如此。厂商的机子不可能永远保证永远不坏，我们也无法保证黑客不会来对我们的系统搞基，最为关键的是，我们自己无法保证自己的程序不会出 bug。所以问题无法避免，错误也不可避免。我们只能鸡蛋分散到不同的篮子里，来减轻一锅端的风险。这就是为什么需要分布式系统的原因。使用分布式系统的另外一个理由是可扩展性。毕竟任何主机（哪怕是小型机、超级计算机）都会有性能的极限。而分布式系统可以通过不断扩张主机的数量以实现横向水平性能的扩展。

——社区用户：bucai

### 有趣的观点：

我自己的理解：分布式系统能广为人知，跟社会的发展脱不了干系，每年的网民数量在逐步递增，随即带来的就是访问量以及数据量的问题，以前的单体架构不足以满足社会的发展，分布式系统存在的意义就是解决这样的事情，高并发、高可用、高、高性能，这也是常说的三高；至于优点前面已经说了，缺点的话我个人理解是取决于物理因素了，比如网络带宽等等，也会因为一些环境架构原因，通常最让人头疼的优化就是实时数据更新的问题。

——社区用户：小乘字节



## Vue 可以在一个项目中使用多个 UI 框架吗？

话题发起人：mmkk\_ccvv

Vue 的 UI 框架有 ElementUI, AntUI 等，可以在一个项目中使用多个 UI 框架吗？

### 有趣的观点：

先说明：没问题，但是不推荐。

有几点考虑：

全局样式不同。会导致冲突，虽然可能很少，但是只要出现一个，就会很烦。

站点 UI 设计通常是基于一个组件库。多个组件的风格混合会让呈现效果显得格格不入，因为每套组件库都有自己一套设计模式，组合使用效果更佳。

组件齐全。当前来说，应该不会出现组件不够用，要去隔壁借一个用的场景，真出现了，那么可以考虑那种，专门做一种组件的插件，会更加专业。

——社区用户：YangFong

### 有趣的观点：

可以，但完全没有必要。

首先一个 UI 框架打包下来，即使按需加载，再怎么也得 100K 往上了。你用几个框架，打出来的包会相当大。

不同 UI 框架使用的 CSS 预编译不同，有些是 less，有些是 scss，混用会相当混乱，并且你

必须安装两个 loader。无形之中你的开发环境会变得非常臃肿。

设计语言不同，不同的 UI 框架遵循的设计语言和规范不同，容易出现放在一起不和谐的情况。

主题换肤功能不好做，需要统一两个框架的换肤方式。比如 Ant Design 可以直接用 less.js 换肤，但是 Quasar 是 scss，没法通用。

可能引起样式冲突。

——社区用户：Gomi



## 大厂上线流程：是先上前端，还是先上后端？怎么保证平滑上线？

话题发起人：孟思行

- 1、如果先上前端，还没上后端，就请求不到新的接口。
- 2、如果先上后端，还没上前端，使用老接口的页面可能会报错。

有趣的观点：

一般情况都是后端先上线，这样现网对上线的感知也是最小的。毕竟前端上线之后，客户有了交互的界面，配合早已发布的后端环境，可以流畅体验新功能。

——社区用户：这个杀手不太冷静

有趣的观点：

如果是一个新项目，无所谓前后端先上后上

如果是老项目，在开发的时候，就该考虑到比如接口的兼容性了，然后后端先上线，然后前端上线。

——社区用户：牙羽

### 有趣的观点：

正常情况后端->bff->前端，如果是特殊情况比如前后端没有关联，先后上线不影响线上功能等就可以不关注顺序，想怎么上就怎么上。

还有一种情况就是没有规范的小公司了，可能是线上流量不大不在乎宕机就可以不关注上线顺序。

——社区用户：bucai



他们的观点和讨论是否也能带给你启发呢？你又有什么有趣的观点，希望与大家分享？

快扫描二维码加入我们，一起交流成长吧，等你哦 🍷🍷🍷 欢迎在评论区留下你的观点呀~



阅读原文

喜欢此内容的人还喜欢

里程碑！用自己的编程语言实现了一个网站

crossoverJie



浅谈deepin自研编程语言Unilang

铁君



手写编程语言-实现运算符重载

crossoverJie

