

[登录 \(//ask.dcloud.net.cn/account/login/\)](https://ask.dcloud.net.cn/account/login/)[注册](https://ask.dcloud.net.cn/register/) [帮助 \(//ask.dcloud.net.cn/register/\)](https://ask.dcloud.net.cn/register/)

DCloud_heavensoft ([//ask.dcloud.net.cn/people/DCloud_heavensoft](https://ask.dcloud.net.cn/people/DCloud_heavensoft)) 
([//ask.dcloud.net.cn/people/DCloud_heavensoft](https://ask.dcloud.net.cn/people/DCloud_heavensoft))
发布: 2019-07-02 09:12 更新: 1 天前 阅读: 117529

flutter、rn、uni-app比较

分类: uni-app

[flutter](https://ask.dcloud.net.cn/topic/flutter) ([//ask.dcloud.net.cn/topic/flutter](https://ask.dcloud.net.cn/topic/flutter)) [rn](https://ask.dcloud.net.cn/topic/rn) ([//ask.dcloud.net.cn/topic/rn](https://ask.dcloud.net.cn/topic/rn))

[React Native](https://ask.dcloud.net.cn/topic/React+Native) ([//ask.dcloud.net.cn/topic/React+Native](https://ask.dcloud.net.cn/topic/React+Native)) [weex](https://ask.dcloud.net.cn/topic/weex) ([//ask.dcloud.net.cn/topic/weex](https://ask.dcloud.net.cn/topic/weex))

前言

每当我们评估新技术时要问的第一个问题就是“它会给我们的业务和客户带来哪些价值？”，工程师们很容易对闪闪发光的新事物着迷，却经常会忽略这些新事物其实可能对我们的客户没有任何好处，反而只会让现有的工作流程更加复杂。

flutter最近比较热闹，毕竟是Google出品。

但我们不是炒作热点的媒体，也不是忽悠你交学费的培训机构，我们作为实际的跨平台开发者，冷静的分析下这个东东。

flutter是Google为Fuchsia操作系统设计的应用开发方式。

Fuchsia OS要兼容廉价物联网设备，要求对硬件的消耗降低，并且为了避免与oracle的java打官司，Fuchsia 使用了dart语言+flutter界面库的方式。

从设计上来看，这套方案的性能确实够高。dart虽然属于大前端范畴，但dart是和java一样的强类型语言，这让dart虚拟机可以做很多优化，性能方面超出了js。

dart曾经与typescript竞争，谁才是更好的js？但不幸输给了typescript，chrome也放弃了内置dart虚拟机的计划。

不过dart团队没有解散，几年后，他们借助flutter，再次出现在公众面前。

性能分析和写法的对比

flutter作为界面库（注意它只是界面库，dart语言是另一个项目），它唯一要干的事情就是渲染界面。不像HTML5，flutter界面库连视频、定位等都没有，就是一个纯排版引擎，绘制文字、按钮、图片等常用界面控件。

这个排版引擎的特点是简单、高性能。

在3大主流渲染引擎里，webview、react native/weex、flutter，复杂度依次降低，渲染性能依次上升。（uni-app是双渲染引擎，webview和weex都内置了，随便开发者使用切换）

所以我们要清楚，提升性能是有代价的，你究竟想要灵活丰富的css3，还是想要固定flex模式排版，抑或是最简单但高性能的flutter排版？开发便利性和运行性能不可兼得。

同时我们要明白，性能的差别，并不是因为Google的chrome团队、Android团队的技术比同公司的flutter团队差。而是flutter提供的布局写法是被限制过的，解析快，所以渲染快。别忘了webview的排版引擎也是世界级工程师用c写的。

但通过这种方式提升性能的代价，就是布局复杂的界面时，flutter的代码嵌套的让人崩溃。

我们先举个例子，同样的界面，用HTML和flutter如何实现：

```
<div class="greybox">
  <div class=redbox>
    smaple text
  </div>
</div>

.greybox {
  display: flex;
  align-items: center;
  justify-content: center;
  background-color: #e0e0e0; /* grey 300 */
  width: 320px;
  height: 240px;
  font: 18px
}
.redbox {
  background-color: #ef5350; /* red 400 */
  padding: 16px;
  color: #ffffff
}
```

```
var container = new Container( // grey box
  child: new Center(
    child: new Container( // red box
      child: new Text(
        "smample text",
        style: new TextStyle(
          color: Colors.white,
          fontSize: 18.0,
        ),
      ),
    ),
    decoration: new BoxDecoration(
      color: Colors.red[400],
    ),
    padding: new EdgeInsets.all(16.0),
  ),
),
width: 320.0,
height: 240.0,
color: Colors.grey[300],
);
```

可以看出，从代码的写法来说，flutter没有tag和样式的说法，更没有选择器，从头到尾只有dart语言，它的界面控件是用dart代码new出来的，每个控件的样式，是在new的时候设置的类json写法的参数。

如果我们要嵌套布局，就要不停的在dart里写child，同时在dart里给child们设样式参数。上面的代码，只是嵌套了1层，实际开发中，dom要嵌套好多层，想象那样的代码。。。所以大家都诟病dart是“嵌套地狱”。

或者，你可以这么理解，这是一个只有js，没有html和css的浏览器。你需要用js createElement来创建元素，用js的style方法给每个element设style，反正就是不能写html和css代码。前端都已经发展到各种mvc等视图逻辑分离的架构了，也有了vue组件这种组件化模式方便用各种轮子快速完成界面。你是否能适应dart这种低效的界面开发模式？从开发模式来讲，这确实是一种倒退。

浏览器的html提供了tag和样式分离的写法，还有各种各样的选择器，但其实这也是有代价的。它导致webview初始化时要同时先启动webkit排版引擎来解析这些编写随性的html、css，同时还要启动一个js引擎比如v8或jscore来解析里面的js。

而dart就很简单，只启动一个dart引擎，解析严格的dart语法，它不会去操心有些标签未闭合要如何容错，不会判断宽度320后面是px还是rem或者是动态计算百分比。

对比这2个引擎初始化时要干的事，差别简直太大了。

所以从解析效率上，flutter肯定比webview要高。但从编码灵活性上，flutter写的代码，嗯，难看而低效！

flutter使用的也是flex布局思想，这是一个强嵌套布局模型，比web常规排版引擎的嵌套更多。当界面复杂时，flutter的代码要嵌套几十层，每层的元素的json样式都和元素一起混写在dart代码里，让人崩溃。

有人提出是否可以通过一种预编译的dsl来简化写法，让flutter的开发不那么痛苦。

但这个难度太大了，从严格转换为松散是简单的，从松散转换为严格几乎是不可能的。

什么意思呢？比如flutter代码转换web代码，是很简单的，flutter已经自带了这个功能。但是想反过来，那可难了。

类似的还有，把typescript转为js是容易的，反之，不是绝对不可行，但会复杂到你宁愿去重写一套typescript代码。

flutter的性能高，除了简单严格，还有一个特点，就是逻辑层与视图层统一，运行在同一套dart虚拟机下。

我们知道rn和weex，也是原生渲染的，它们的性能高于webview。但同为原生渲染的，怎么会慢于flutter呢？其实不是原生渲染慢，而是js和原生通信慢。

rn和weex都采用了独立的js引擎（iOS是jscore，Android是v8，最新版rn开始在Android上搞自己的js引擎Hermes），从js与dart的比较上，性能稍逊一筹。但这不是主要问题，因为v8的jit不是盖的，也是编译为原生代码解析的。性能上的主要问题是：**rn、weex的js引擎和原生渲染层是两个运行环境。**

当js引擎联网获取到数据后，通知原生视图层更新界面时，有一个跨环境的通信折损。同样，当用户在屏幕上操作原生视图层时，要给js引擎发送通知，也会产生这个通信折损。

不过这种性能差别，在大多数场景中，用户是感受不到的。比较影响的场景，是跟手式的js响应操作绘制帧动画，或者说js连续操作界面元素方面，flutter折损更少。

这个通信折损，其实普遍存在于所有逻辑和视图分离的框架中，包括各家小程序也有这个问题。

为了解决react native上js绘制动画卡的问题，曾经的react native拥趸aribnb搞了一个lottie的动画库，但lottie只能静态执行，无法跟手交互。weex更进一步，搞了个BindingX，这个技术很赞，它可以预定义规则，让用户界面在原生层交互时通过预定义规则直接响应，**而无需传递给js层。**在需要短时间内来回通信的场景时，可以使用BindingX这类解决方案。它的性能和灵活性比rn更强了一些。

在uni-app里，nvue页面可以直接使用bindingx (<https://uniapp.dcloud.io/nvue-api?id=nvue-%e9%87%8c%E4%bd%bf%E7%94%a8-bindingx>)。至于uni-app的vue页面不是基于weex渲染的，它遇到通信折损时，解决方案叫wxs (<https://uniapp.dcloud.io/frame?id=wxs>)，wxs是一种运行在视图层的js，它的性能和和灵活性都非常高，完全可以达到flutter的水准。

说回来flutter，它只有一个dart引擎，没有来回通信产生的性能问题。不过任何事情都是有利有弊的，flutter在普通的界面绘制上效率虽然高，但一旦涉及原生的界面，反而会遇到更多问题。

前面已经说过，flutter只是一个基础排版引擎，缺少很多能力，当我们需要在flutter界面上内嵌一个原生的视频播放扩展控件时（flutter没有内置视频播放能力），或者原生的高德地图sdk，那么在拖动视频进度时、拖动地图时，flutter一样会产生原生和dart之间的通信，造成性能损耗。

事实上，由于flutter是在一个类canvas环境绘制的，想把一个原生控件嵌入flutter的布局里某些元素之间去排版，还不是一件容易做到的事情，坑很多。

每个人都想要一个像css3那样灵活写法的布局引擎，他们给react native和weex提需求，给flutter提需求。殊不知，让这些产品团队实现了css3时，他们的性能优势已经不再了，他们相当于又实现了一遍webview。这种无意义的需求，他们是不会受理了。

性能好，有个度，客观地讲，rn/weex调用原生渲染的性能，和flutter的渲染性能，在用户体验上并没有明显区别，甚至在很多场景下，和webview渲染的小程序也没有明显区别。

也简单说说webview渲染小程序，为什么性能高，核心是预载。点击一个新页面时，webview是提前创建好的，不会走复杂的webkit、v8的初始化流程，连开发者的js代码，也是预载好的。所以点击新页面时，它的渲染速度和原生应用没什么差别。当然也有个坏处，就是启动慢。微信里启动小程序速度看着还行，其实是微信在启动小程序之前，就已经提前初始化了小程序运行环境。

即便是排版引擎，ui库好用吗？

不管是rn还是flutter，有一个设计，很不中国化。它们在iOS和Android平台上，使用2套ui库。

比如flutter，在iOS上写一个button，要用CupertinoButton，是iOS风格的控件，在Android上则要用RaisedButton，是Material风格的控件。

rn也是如此，它的官方说法是：learn once，write anywhere。它都不敢说：write once，run anywhere。因为它确实要求开发者写2套代码。

中国的开发者可没有这种习惯，中国的每个开发者，为了避免用户换手机后不会用自己的app，都会使用中性的设计。

就连微信Android版，底部的tab也是仿iOS而不是Material风格（Material风格是把底tab放在顶部的，并且左右滑动，微信曾经有这样一个临时版本，因为被用户吐槽，很快就下掉了）。

这种中外差异怎么造成的？

国外Android手机，其手机主界面就是强烈的Material风格。用户在Android主界面习惯的风格和使用方式，如果启动一个App后不是这样，会导致用户不会用了。

Google也一再给Android开发者强调，App必须使用Material风格。这其实也是一个防止用户切换脱离Android生态的策略设计。

所以国外开发者的App，Android上都会遵循Material风格，当然，这种Material风格的App是上不了Apple的Appstore的。

这就导致他们默认就是要写2套ui的，所以rn和flutter都是iOS、Android各自1套ui控件。

但在中国，我们的国产Android Rom，根本不是Material风格，很多rom以仿iOS体验为卖点。

所以中国的App，全都是贴近iOS的中性风格，中国的用户换了手机，不管是手机os本身，还是App的使用，都不会造成切换障碍。

rn和flutter这种“跨平台”排版引擎，其跨平台性，对于中国开发者而言，又打了折扣。

其实类似小程序那样的ui风格，是能够良好的跨iOS和Android的体验的，不管用什么手机，打开小程序都不会觉得有问题。

uni-app默认也是这种通用ui风格。uni-app的开发者只需要写一套界面ui，就可以适应不同手机的用户，真正的write once, run anywhere。

动态性

webview、rn/weex，都有一个特点，可以远程动态载入js代码，可以更新本地的js代码。前端开发者认为动态性是天经地义的，但其实flutter并不支持。

flutter是有编译优化概念的，如果它提供动态性支持，会影响它的性能。

业内有些开发者，改造了flutter，用一个独立的v8/jscore来加载动态js代码，去操作flutter布局引擎的渲染。好像还有些人在追捧这样的方案，简直是闲得蛋疼。

flutter本来没有跨环境通信的问题，结果又弄了一个js引擎进来搞出了通信问题，造成性能下降，还把包体积增加了很大，还不如直接用rn/weex。

除了flutter，rn/weex/uni-app都可以动态热更新。

跨平台排版引擎和跨平台应用开发引擎的区别

有些人说他们的App用rn/weex、flutter。但是具体用它们做了什么呢？

是整个App用了它们，还是某个页面用了它们？

一个页面跨平台，和一个应用跨平台，是完全不同的2个概念。

webview、rn/weex、flutter全部是渲染引擎，webview因为HTML5的发展，还算是多了一些能力比如位置服务、多媒体等。而rn/weex、flutter真的只是一个纯粹的排版引擎，没有任何原生能力。

如果一个原生应用里，某个不涉及原生能力的界面想跨平台，那么这几个引擎都可以，并且flutter的性能最高。所以能看到一些公司尝试把App中的个别原生交互较少页面使用flutter实现。

但如果一个完整的应用，想用跨平台工具开发，那就不是排版引擎的范畴了，它需要应用开发引擎。

什么是跨平台应用开发引擎？不但排版部分要跨平台，开发API也要跨平台。

应用开发离不开os或三方sdk的能力调用，如果是单纯的排版引擎，一旦涉及os能力和sdk调用，就必须iOS、Android的工程师配合，编写不同的原生代码整合在一起。这就不跨平台了。

Airbnb曾是React Native 框架的倡导者和开发者代表。但他们于2019年正式发公告，弃用了react native。

原因是什么？

很简单，react native并不能提升Airbnb的开发效率，反而降低了他们的效率。

“本来我们可以只维护Android和iOS两套代码，但现在我们要维护三套（指多了一套react native的js代码），这让我们很疲惫” -- aribnb

开发者选用跨平台开发引擎，本来是为了提高效率、降低成本。Airbnb正是在实践了几年后，发现rn根本无法实现他选用跨平台引擎的初衷时，无奈放弃了rn，用原生开发重写。

要想真的提升开发效率，降低开发成本，那么跨平台开发引擎，需要提供一个完整的应用开发平台，包含所有常用的应用开发能力的跨平台。在不常用的部分，提供插件市场以及免原生介入的插件使用方式。

在react native、flutter的社区，也有不少三方提供的原生插件，但是连Airbnb这样的国外开发者对此都不满意。更何况对于很多中国开发者常用的场景，其对应的插件的质量、跨端性都难以商用。

更麻烦的是如果你不会原生开发，就没法把这些插件与你的前端代码集成起来。

uni-app，它的设计目标不是跨平台排版引擎，而是跨平台应用开发引擎。

所以uni-app的排版部分，可以选择小程序强化webview引擎和weex引擎，可根据自己的需求切换。而能力层面，uni-app提供了htmlplus API、Native.js、插件市场，解决了原生能力js化的问题。

uni-app让开发者真的不用懂原生开发就能做出完整的跨平台应用。遇到极个别的需求，开发者也可以去插件市场找人订做一个原生插件，自己仍然使用js来集成，仍然可以云端直接打包。

技术学习成本和难度

rn，要求开发者学习react，要求精通flex布局，要求原生开发协作。

flutter，要求开发者学习dart，了解dart和flutter的API、要求精通flex布局，要求原生开发协作。

weex已经内嵌到uni-app中，就不单独提了。

uni-app，要求开发者学习vue，了解小程序。

很明显uni-app的学习成本太低了，它没有附加专有技术，全部使用公共技术。

学习成本和难度，直接意味着：开发成本、招聘成本、上线速度、上线风险。

另外，dart究竟值不值得学，是一个大问题。

Google的天才工程师也发明了go语言，它确实有很多理论优势，但实际上市场的主流，仍然是c和c++。

质量

这几种跨平台引擎，整体的质量水平是差不多的，都是可以商用，但是有各种小坑。

2020年7月6日的数 据，各家的github的issues如下：	框架	issues
uni-app	557	
react native	816	
flutter	7840	

代码量是flutter最多，对应它的issues的数量也越多。

对于uni-app和react native来讲，渲染层是基于相对成熟的原生渲染或webview渲染来做的，不像flutter一切都是自己实现。而逻辑层uni-app也使用相对成熟的v8和jscore，出问题的点相对要少。

各框架的更新都比较频繁，真的有影响商用的大问题，肯定都会及时修复。

目前各个框架都有较大的用户使用量，质量方面均是可控的。

生态

任何开发引擎，都离不开生态。

对于国外的开发者，rn、flutter的生态肯定比uni-app好，比如facebook登陆分享、Google地图等。

但对于国内的开发者，那是反过来的，中国开发者需要的全端推送（UniPush集成了iOS、华为、小米、OPPO等众多原厂推送）、各种国内登陆、支付、分享SDK、各种国内地图、各种ui库、以及Echart图表等，都是在uni-app体系里，这方面生态可比rn、flutter丰富多了。uni-app的插件市场有数千款插件，不能说应有尽有，但确实是最丰富的跨端开发框架生态了。

另外，uni-app的生态还比其他竞品强在如下方面：

- App和H5提供了renderjs技术，使得浏览器专用的库也可以在App和H5里使用，比如echart、threejs等参考 (<https://uniapp.dcloud.io/frame?id=renderjs>)
- 兼容微信小程序 JS SDK，丰富的小程序生态内容可直接引入uni-app，并且在App侧通用，参考 (<http://ask.dcloud.net.cn/article/35070>)
- 兼容微信小程序自定义组件，并且App、H5侧通用，参考 (<http://uniapp.dcloud.io/frame?id=%E5%B0%8F%E7%A8%8B%E5%BA%8F%E7%BB%84%E4%BB%B6%E6%94%AF%E6%8C%81>)

这些丰富的生态兼容，是rn和flutter无法享受的。

更新：2020年11月google全球开发者大会公布的数据，flutter全球月活开发者50万，中国占11%，也就是5.5万。这个数字太小了，撑不起好生态。DCloud的开发者月活100万+，生态更繁荣。

其他端的跨端性

flutter和rn，都是支持web技术的。但都是仅限于普通界面排版，涉及定位、摄像头、相册什么的，是要单独写代码的。

另外flutter的H5版，嗯，作为中国开发者，你不会想要一个如此浓郁的Material风格的H5版的。。。

更何况这个Material ui库大的很，编译出来的H5版要十几M的体积。

uni-app的H5端是包含完善的能力引擎的，丰富能力都可以直接跨端使用，风格也是跨端风格。uni-app的H5引擎体积只有1百K，gzip后只剩下30k（不含vue、vue-router），比其他工具的引擎体积要小的多。

另外，中国离不开小程序，rn、flutter官方都不会支持小程序，由于架构差异太大，国内三方也做不到把rn代码良好的编译为小程序代码。uni-app则可以一套代码，同时编译为iOS、Android、H5、微信小程序、支付宝小程序、百度小程序、头条小程序、QQ小程序。

结论

每种技术的诞生，有其背后公司的目的。

但凡没有明确公司战略的技术，除非是特别简单的技术，否则很难商用，因为为了商用要投入公司非常多资源。

flutter诞生的目的，是为了Fuchsia OS，是为了在下一个互联网大潮，即万物互联的物联网年代，提供一个类似Android在移动互联网位置的垄断性操作系统。

因为Google已经很明确不会在下一个时代使用Android+java的路线了。

至于在Android上去java化，那是Kotlin的使命，与flutter无关。

跨iOS和Android平台开发，这不是Google的战略目标。

但万物互联何时到来？Fuchsia OS何时流行？这在现实中是一个问号，在Google内部，也只是战略储备项目。

一个语言的流行，不是一件简单的事情，不是有优点，就会流行，它需要天时地利人和。

6年前我们就知道dart比js更好，dart不应该消亡，但想成为主流技术，太难太难了。

同样我们也知道go比c++更好，但go还是起不来。

想靠flutter驱动dart流行是不现实的，甚至是反过来的。跨iOS、Android开发在国外不是主流市场，这点价值造就不出一个这么难建的生态。

所以dart能否流行，是要打一个大大的问号的，它可能会像go语言一样，叫好不叫座。

一个跨平台公司，应当是中立的。而flutter在这个位置上很尴尬，它是google出品的、同时跨iOS和Android的开发引擎。

如果这个引擎做大了，Apple会怎么看？那可不是像微信搞个小程序那么简单了，微信是中立三方，且只在中国。Apple不会让google的flutter在iOS上做大的。大家也都知道，即便是chrome，在iOS上也只能使用iOS提供的内核，容不得google在iOS上乱搞的。

目前，flutter在国内一些大厂的原生App里得到了局部应用。这个应用场景，目的不是为了节省成本，flutter开发成本很高，生态也不完善，但因为性能高，一些大公司愿意负担这个代价来使用flutter，在原生app里部分页面使用flutter制作。但是这个场景，有个尴尬，就是flutter页面无法动态更新，但很多大app对动态发版的需求极强，有些厂商改造了flutter使其可动态发布，但又降低了flutter的性能。目前还没有一个完美的解决方案。

写了这么多，最后总结下flutter与uni-app的比较：

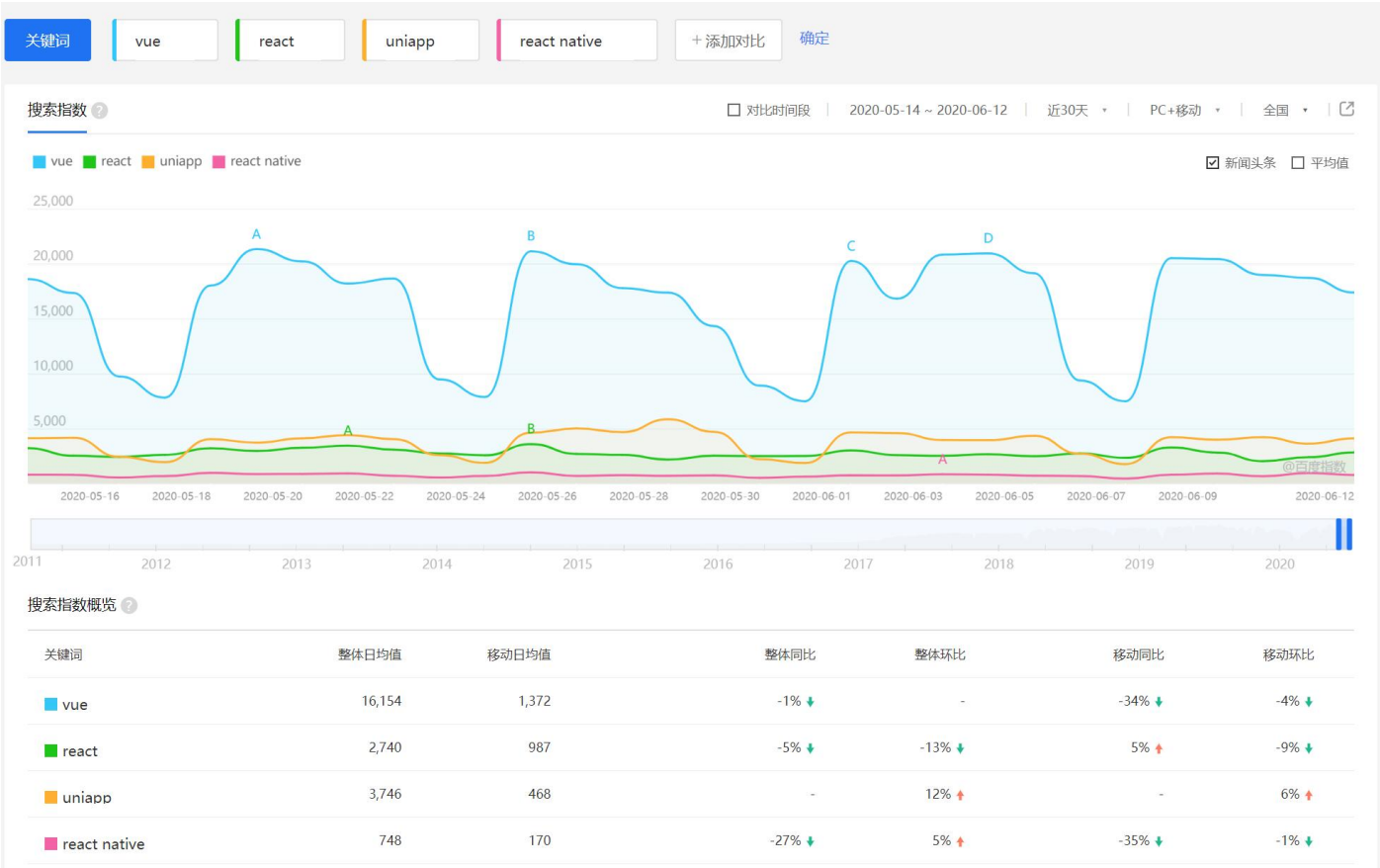
- flutter与uni-app的相对优势:
 - 性能好一丢丢。比rn有优势，但比拥有bindingx和wxs的weex/uni-app，在实际开发中没有很明显的差距。
- flutter与uni-app的相对劣势:
 - 需要原生协作，维护3套代码，无法有效降低开发成本，提升开发效率
 - 嵌套地狱，代码难看难维护
 - 不支持热更新
 - 目前质量和成熟度很低
 - 原生可视控件融合不好，比如webview、video、map
 - ui库不适合国情
 - 学习成本高
 - 应用场景有限，dart未来扑朔迷离

更新：uni-app里其实也可以使用flutter，有插件作者提供了插件，使得uni-app的应用在某些页面可以打开flutter页面，详见 (<https://ext.dcloud.net.cn/plugin?id=4302>)

再总结下rn和uni-app的比较

- rn与uni-app的相对优势:
 - rn的坑虽然比weex的少，但uni-app已经填了weex的很多坑。这方面差别不大。
 - rn的生态虽然比weex丰富。但uni-app是反过来的，uni-app的国内应用生态丰富度超过了rn。
 - rn是纯单页的，嵌入原生App比较灵活。而uni-app是应用整体的概念，如果要内嵌入其他原生应用的话，要求原生应用内嵌uni-app应用整体进来。即集成uni小程序sdk (<https://ask.dcloud.net.cn/docs/#https://ask.dcloud.net.cn/article/36941>)。
- rn与uni-app的相对劣势:
 - 需要原生协作，维护3套代码，无法有效降低开发成本，提升开发效率
 - 不支持小程序，发布到h5也无法直接发
 - 性能不如uni-app
 - 国内的插件生态不如uni-app丰富
 - ui库不适合国情，learn once, write anywhere
 - 学习成本高，用人成本高，不利于开发商降低开发成本
 - rn是纯单页应用，如果一个应用的页面很多，用rn写会很崩溃，变量污染和干扰严重。而weex/uni-app支持多页面，页面之间上下文隔离，写页面较多的大型应用更合适
 - 另外react在中国的市场占有率远不如vue。这也是中国与国外不同的特色情况。

贴个vue、react、react native的百度指数对比，无论总体量的差距，还是发展趋势的下滑程度，可以明显看出react系在中国确实不行了。



(//img-cdn-tc.dcloud.net.cn/uploads/article/20200614/8a0ac21db5feca970bc890e7d3ccde01.jpg)

中国的开发者，过去总会想：

1. 小程序那套webview优化的技术，我能不能用到我的App里？现在uni-app已经为你解决了这个问题。
2. weex能不能坑少点，API和插件多点？现在uni-app已经为你解决了这个问题。

如果你是一个资源充沛的大公司，原生App中部分不要求动态更新、也没有太多原生交互的页面，可以尝试使用flutter实现。但如果大范围使用，你也会遇到和aribnb一样的问题，维护3套代码还不如维护2套代码。

如果你开发uni-app选用了原生渲染，那App的性能足够好，且你得到了切实的开发效率的提升、成本的下降、快速和低风险的上线。
















































选择跨平台工具而不是原生开发，本质目的不就是为了成本和效率吗？在跨平台开发里能真正解决你本质需求的，就是uni-app。


当然，uni-app的app引擎并没有吊炸天。App平台，所有跨平台工具都还比不过原生，这是客观事实。只是，如果uni-app不能满足你的需求，你没有必要去用其他跨平台工具，直接上原生吧。

👍 98 赞

★ 关注

↻ 分享

 ([//ask.dcloud.net.cn/people/wenju](https://ask.dcloud.net.cn/people/wenju))  ([//ask.dcloud.net.cn/people/485516](https://ask.dcloud.net.cn/people/485516))  ([//ask.dcloud.net.cn/people/%E4%B8%8A%E4%BA%A4%E5%9B%BD%E5%AE%B6](https://ask.dcloud.net.cn/people/%E4%B8%8A%E4%BA%A4%E5%9B%BD%E5%AE%B6))  ([//ask.dcloud.net.cn/people/957718](https://ask.dcloud.net.cn/people/957718))  ([//ask.dcloud.net.cn/people/%E5%89%91%E5%BF%83%E6%97%A0%E9%97%B4](https://ask.dcloud.net.cn/people/%E5%89%91%E5%BF%83%E6%97%A0%E9%97%B4))  ([//ask.dcloud.net.cn/people/belowfox](https://ask.dcloud.net.cn/people/belowfox))  ([//ask.dcloud.net.cn/people/112139](https://ask.dcloud.net.cn/people/112139))  ([//ask.dcloud.net.cn/people/986720](https://ask.dcloud.net.cn/people/986720))  ([//ask.dcloud.net.cn/people/sean1658](https://ask.dcloud.net.cn/people/sean1658))  ([//ask.dcloud.net.cn/people/1046058](https://ask.dcloud.net.cn/people/1046058))  ([//ask.dcloud.net.cn/people/mShrne](https://ask.dcloud.net.cn/people/mShrne))  ([//ask.dcloud.net.cn/people/shinvey](https://ask.dcloud.net.cn/people/shinvey))  ([//ask.dcloud.net.cn/people/jokerzhou](https://ask.dcloud.net.cn/people/jokerzhou))  ([//ask.dcloud.net.cn/people/%E7%99%BE%E4%B8%87%E5%AF%8C%E7%BF%81%E8%B5%B0%E8%B5%B7](https://ask.dcloud.net.cn/people/%E7%99%BE%E4%B8%87%E5%AF%8C%E7%BF%81%E8%B5%B0%E8%B5%B7))  ([//ask.dcloud.net.cn/people/1017006](https://ask.dcloud.net.cn/people/1017006))  ([//ask.dcloud.net.cn/people/duo123](https://ask.dcloud.net.cn/people/duo123))  ([//ask.dcloud.net.cn/people/aQiu](https://ask.dcloud.net.cn/people/aQiu))  ([//ask.dcloud.net.cn/people/275762](https://ask.dcloud.net.cn/people/275762))  ([//ask.dcloud.net.cn/people/421387](https://ask.dcloud.net.cn/people/421387))  ([//ask.dcloud.net.cn/people/1294710](https://ask.dcloud.net.cn/people/1294710))  ([//ask.dcloud.net.cn/people/1294724](https://ask.dcloud.net.cn/people/1294724))  ([//ask.dcloud.net.cn/people/aca](https://ask.dcloud.net.cn/people/aca))  ([//ask.dcloud.net.cn/people/qins](https://ask.dcloud.net.cn/people/qins))  ([//ask.dcloud.net.cn/people/skysowe](https://ask.dcloud.net.cn/people/skysowe))  ([//ask.dcloud.net.cn/people/%E5%92%B8%E9%B1%BC%E9%85%B1%E4%B8%8D%E6%98%AF%E9%85%B1](https://ask.dcloud.net.cn/people/%E5%92%B8%E9%B1%BC%E9%85%B1%E4%B8%8D%E6%98%AF%E9%85%B1))  ([//ask.dcloud.net.cn/people/1177345](https://ask.dcloud.net.cn/people/1177345))  ([//ask.dcloud.net.cn/people/%E8%9C%A1%E7%AC%94%E5%B0%8F%E4%BC%99](https://ask.dcloud.net.cn/people/%E8%9C%A1%E7%AC%94%E5%B0%8F%E4%BC%99))  ([//ask.dcloud.net.cn/people/%E4%BA%8E%E6%88%90%E9%BE%99](https://ask.dcloud.net.cn/people/%E4%BA%8E%E6%88%90%E9%BE%99))  ([//ask.dcloud.net.cn/people/665013](https://ask.dcloud.net.cn/people/665013))  ([//ask.dcloud.net.cn/people/313741](https://ask.dcloud.net.cn/people/313741))  ([//ask.dcloud.net.cn/people/1344143](https://ask.dcloud.net.cn/people/1344143))  ([//ask.dcloud.net.cn/people/1325681](https://ask.dcloud.net.cn/people/1325681))  ([//ask.dcloud.net.cn/people/surbye](https://ask.dcloud.net.cn/people/surbye))  ([//ask.dcloud.net.cn/people/%E5%9C%B0%E8%A1%A8%E8%8D%A3%E8%80%80](https://ask.dcloud.net.cn/people/%E5%9C%B0%E8%A1%A8%E8%8D%A3%E8%80%80))  ([//ask.dcloud.net.cn/people/vonfly](https://ask.dcloud.net.cn/people/vonfly))  ([//ask.dcloud.net.cn/people/biaoov](https://ask.dcloud.net.cn/people/biaoov))  ([//ask.dcloud.net.cn/people/1047281](https://ask.dcloud.net.cn/people/1047281))  ([//ask.dcloud.net.cn/people/%E8%BF%91%E6%B0%B4%E6%A5%BC%E5%8F%B0](https://ask.dcloud.net.cn/people/%E8%BF%91%E6%B0%B4%E6%A5%BC%E5%8F%B0))  ([//ask.dcloud.net.cn/people/%E6%B0%B5%E6%B0%B5%E6%B5%B7](https://ask.dcloud.net.cn/people/%E6%B0%B5%E6%B0%B5%E6%B5%B7))  ([//ask.dcloud.net.cn/people/437164](https://ask.dcloud.net.cn/people/437164))  ([//ask.dcloud.net.cn/people/%E8%96%9B%E5%86%9B%E5%86%9B](https://ask.dcloud.net.cn/people/%E8%96%9B%E5%86%9B%E5%86%9B))  ([//ask.dcloud.net.cn/people/323762](https://ask.dcloud.net.cn/people/323762))  ([//ask.dcloud.net.cn/people/678788](https://ask.dcloud.net.cn/people/678788))  ([//ask.dcloud.net.cn/people/411141](https://ask.dcloud.net.cn/people/411141))  ([//ask.dcloud.net.cn/people/1017526](https://ask.dcloud.net.cn/people/1017526))  ([//ask.dcloud.net.cn/people/700272](https://ask.dcloud.net.cn/people/700272))  ([//ask.dcloud.net.cn/people/%E4%BE%A0%E5%AE%A2%E8%AF%B7%E6%8E%A5%E5%88%80s](https://ask.dcloud.net.cn/people/%E4%BE%A0%E5%AE%A2%E8%AF%B7%E6%8E%A5%E5%88%80s))


(//ask.dcloud.net.cn/people/944992)  (//ask.dcloud.net.cn/people/zihanoo) 
(//ask.dcloud.net.cn/people/1445168)

160 条评论


要回复文章请先登录 (//ask.dcloud.net.cn/account/login/)或注册 (//ask.dcloud.net.cn/account/register/)

 匆匆 (//ask.dcloud.net.cn/people/foreverglory)
(//ask.dcloud.net.cn/people/foreverglory)
所以 xamarin 怎么样?


1 天前

 3***@qq.com (//ask.dcloud.net.cn/people/977647)
(//ask.dcloud.net.cn/people/977647)
111



2021-10-02 19:46

 DCloud_heavensoft (//ask.dcloud.net.cn/people/DCloud_heavensoft) ▼ (作者)
(//ask.dcloud.net.cn/people/DCloud_heavensoft)
回复 DoublePD : (//ask.dcloud.net.cn/people/DoublePD)
和dom数量没关系吧, 如果是js频繁操作dom, app端要用renderjs来解决


2021-10-02 01:11

 DoublePD (//ask.dcloud.net.cn/people/DoublePD)
(//ask.dcloud.net.cn/people/DoublePD)
uniapp APP端dom一多就卡了 甚至不如之前H5+ 这是开倒车吗


2021-09-30 17:15

  maozai (//ask.dcloud.net.cn/people/maozai)
(//ask.dcloud.net.cn/people/maozai)
回复 1***@qq.com : (//ask.dcloud.net.cn/people/919955)
有社区反馈的

2021-08-09 14:45

 叶耳朵 (//ask.dcloud.net.cn/people/%E5%8E%B6%E8%80%B3%E6%9C%B5)
(//ask.dcloud.net.cn/people/%E5%8E%B6%E8%80%B3%E6%9C%B5)
回复 倾斜的水瓶座 : (//ask.dcloud.net.cn/people/netcore)
问题是 我写的好几个APP都会在不知名的情况下白屏 即时最新的iphone也会出现。

2021-07-28 09:03

 html5来了 (//ask.dcloud.net.cn/people/html5%E6%9D%A5%E4%BA%86)
(//ask.dcloud.net.cn/people/html5%E6%9D%A5%E4%BA%86)

说了那么多，客户要的流畅丝滑。不然你告诉我怎么说，这个东西省成本？但是程序员需要天天加班，不是在改bug，就是在改bug的路上。

2021-07-22 11:31



s***@sikebaoan.com (//ask.dcloud.net.cn/people/1760573)

回复 1***@qq.com : (//ask.dcloud.net.cn/people/1735880)

用了就知道 很多鸡肋

2021-07-22 11:13



1***@qq.com (//ask.dcloud.net.cn/people/919955)

回复 DCloud_heavensoft : (//ask.dcloud.net.cn/people/DCloud_heavensoft)

建议可以在用户群体里面征集一些技术好的开发者，让他们也参与到社区技术贴的跟踪中，把能回复和解决的解决了，不能的统一建一个群进行反馈，这样就把问题进行集中化了，这样也避免有的地方没能及时回复和给用户解决得麻烦，给每个参与的用户开发者账号给一个单独的标，还可以从开发者中吸取优秀的人才。

2021-06-02 10:00



banro (//ask.dcloud.net.cn/people/banro)

好好梳理下文档，而不是让用户遇到问题都要在网站的大量搜索结果中苦苦寻找，以及找不到清楚的文档，而误报bug，这比什么都重要。

对比下flutter的文档，甚至让人觉得flutter上手似乎更容易

2021-05-28 19:13

1	2 (//ask.dcloud.net.cn/article/id-36083__page-2)	3 (//ask.dcloud.net.cn/article/id-36083__page-3)
4 (//ask.dcloud.net.cn/article/id-36083__page-4)	> (//ask.dcloud.net.cn/article/id-36083__page-2)	
>> (//ask.dcloud.net.cn/article/id-36083__page-16)		

友情链接

W3Cschool (<https://www.w3cschool.cn/>) 优势智云 (<https://www.yszyun.com/>)

黑马程序员 (<http://bbs.itheima.com/>) layui前端框架 (<https://www.layui.com/>)

程序员客栈 (<https://www.proginn.com/>) 个推 (<https://www.getui.com/>)

锐亚教育 (<http://www.insideria.cn/>) 撩课教育 (<http://itlike.com/>) 即速应用 (<https://www.jisuapp.cn/>)

上线了小程序 (<https://www.sxl.cn/xiaochengxu>) 慕课网 (<https://www.imooc.com/>)

DCloud新闻 (<https://news.dcloud.net.cn>)

InfoQ (https://www.infoq.cn/?utm_source=dcloud&utm_medium=youlian)

美信拓扑 (<https://maximtop.com/>) 环信IM (<https://www.easemob.com/product/im>)

申请友链 (<mailto:bd@dcloud.io>)



DCloud 即数字天堂(北京)网络技术有限公司是W3C成员及HTML5中国产业联盟发起单位

京ICP备12046007号-4 | 国家信息安全等级保护三级, 证书编号: 11010813802-20001

|