

Node.js 开发 Web 应用

从零到百万PV

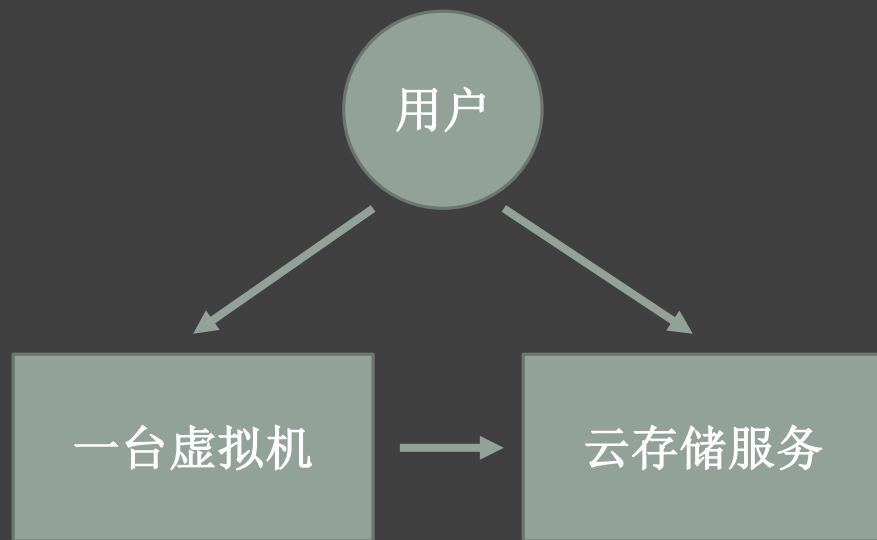
2015年7月5日

这就是那个从零到百万PV的东东



<http://wx.aituwen.com/>

开始的时候，它是这样的



All In One

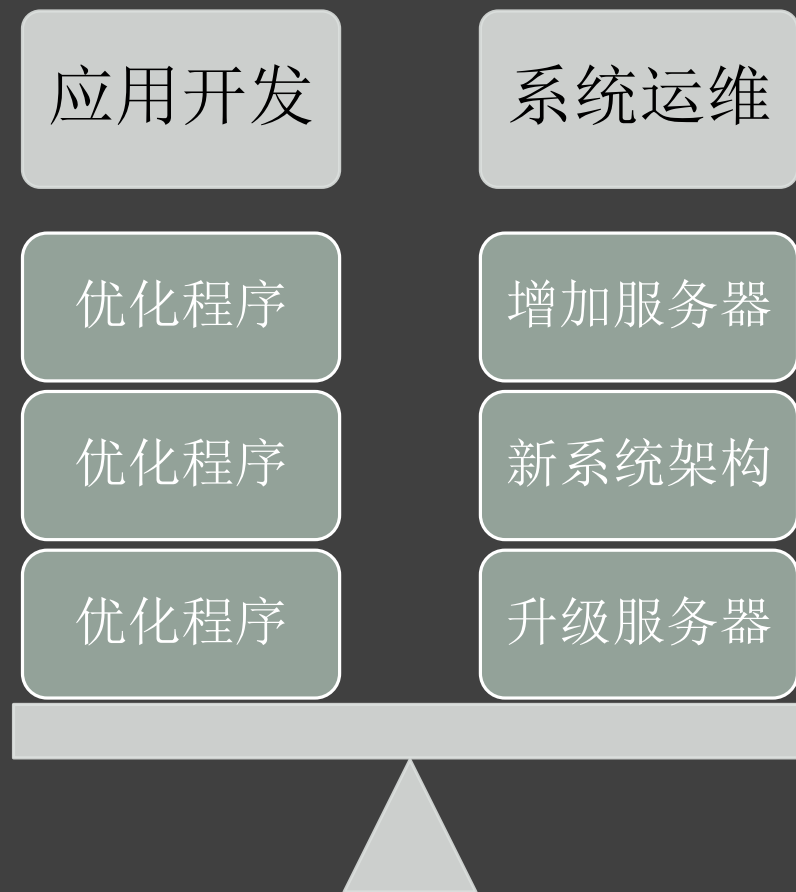
(Nginx + PM2 + Node.js App + MongoDB)

(CPU: 1核, 内存: 1GB, 带宽: 2Mbps)

后来，它是这样的



谁解个中心酸？



一点儿心得体会

设计易于横向扩展的架构：分层、解耦、精简

聚焦核心功能，尽可能使用优秀的第三方服务

充分发挥已有组件如Redis的功能，减少复杂度

使用云存储服务

安全存储

分发加速

数据处理

前端图片直传

```
function (localFile, id, next) {  
    var pic = localFile.replace(/^data:image\/\w+;base64/, "");  
    var url = "http://up.qiniu.com/putb64/-1";  
    var xhr = new XMLHttpRequest();  
  
    xhr.onreadystatechange=function(){  
        if (xhr.readyState==4){  
            var data = JSON.parse(xhr.responseText);
```


前端图片直传（续）

```
    if(data.key !== ""){
        $('#'+id).find('input[name=key]').val(data.key);
    }
    next();
}
}

xhr.open("POST", url, true);
xhr.setRequestHeader("Content-Type", "application/octet-stream");
xhr.setRequestHeader("Authorization", "UpToken " + app.v.uploadToken);
xhr.send(pic);
}
```

HTML5上传图片及预览

FileReader 方法读取图片

reader.onload();中对图片的处理

```
var exif;  
// 转换二进制数据  
var base64 = result.replace(/^.*?/, /, "/");  
var binary = atob(base64);  
var binaryData = new BinaryFile(binary);  
// 获取exif信息  
exif = EXIF.readFromBinaryFile(binaryData);  
//canvas画图处理变形 旋转问题  
var mplmg = new MegaPixImage(img);  
mplmg.render(canvas, { maxWidth: targetW,  
maxHeight: targetH, orientation:orientation});  
dataurl = canvas.toDataURL(type, 1);
```

图片上传遇到的问题

Canvas压缩图片ios会变形

`megapix-image.js`

将图片数据转换成二进制数据，方便获取图片的exif信息

`binaryajax.js`

获取图片的exif信息

`exif.js`

应用 Redis 的典型场景

缓存

- 尽可能缓存数据，减少高并发的压力

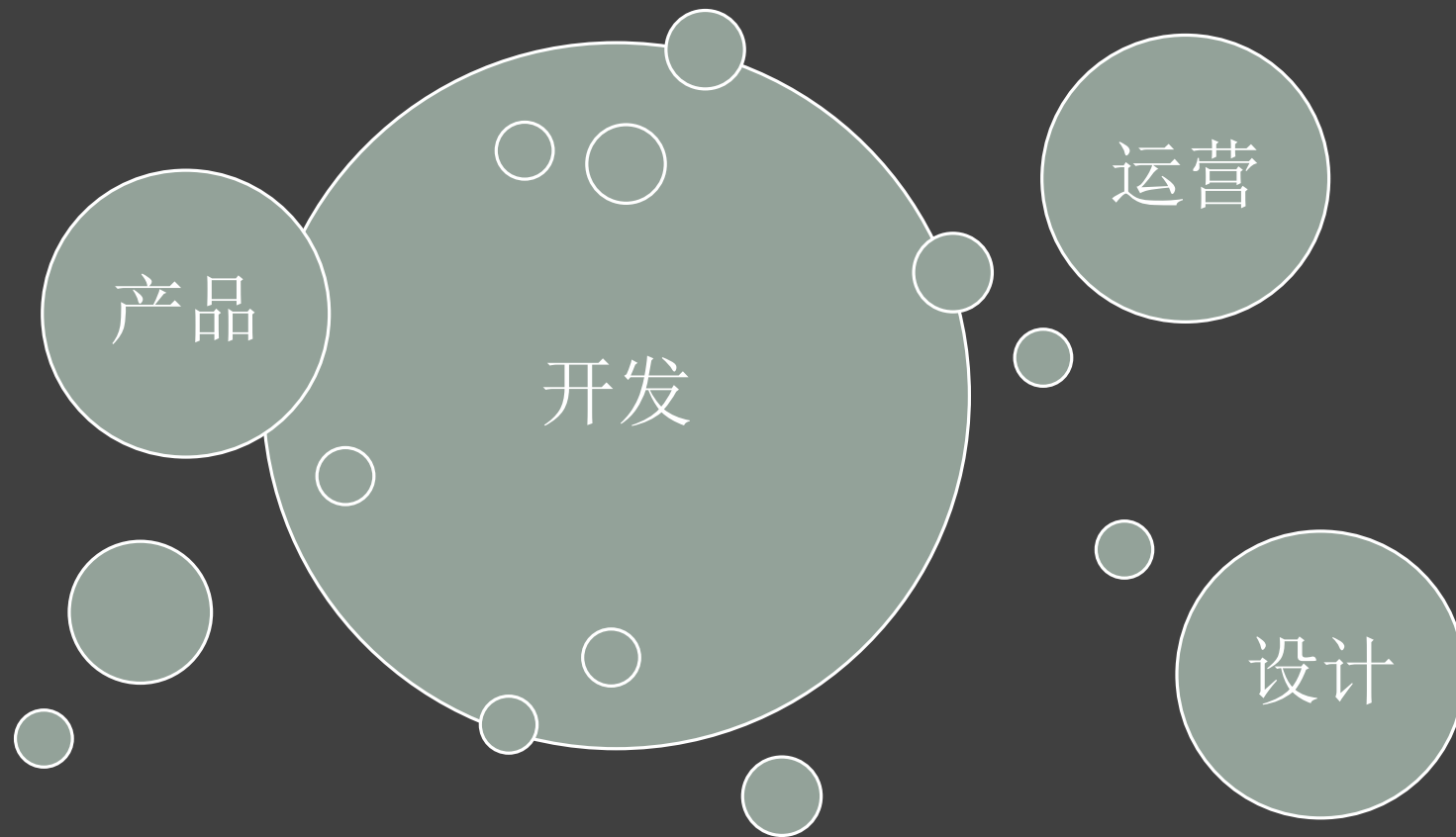
排行

- 使用有序集合在内存中维护排名列表

队列

- 使用监听列表的方式，实现消息队列

这里是广告：其实，我们也在招人



求才不拘一格，有意请发简历和作品到 david@aituwen.com