

# 为什么ZelFlux不用“牛逼技术”就能实现去中心化通信？

[Chinese Version]

Abe.Chua

2020/08/21

关于ZelFlux的介绍，推荐这段视频[1]和相关文章[2][3]，它是Zel生态系统[4]中一个重要的组成部分。如果读者对这标题感兴趣，大概率说明跟作者一样，也许想知道或者评估一下关于ZelFlux或Zel生态系统的相关技术状况。

一个技术宅一定会有某些疑惑，如：为啥ZelFlux技术，感觉没有那么高大上？技术实现手段跟传统Web开发好像没啥区别？为啥采用Javascript和大家熟知的MEVN软件开发栈(MongoDB+Express+Vue.js+Node.js)？而“算力操作系统”或换种说法“DApp资源管理平台”，是不是有点扯淡？也许还有，VPS虚拟主机方案，不也是挺好的，为啥还需要一个“去中心化计算网络”，是不是多此一举，感觉也没啥想象空间啊？跟某些读者一样，这让作者困惑了大半年。那么，接下来读者可以看看作者的某些思考或看法是不是有一定的道理。如果读者想进一步进行交流、更新、补充、修正某些方面的内容，那么可到Github[5]上提问题或递交更新版本，通过协作可以一起完善本文的后续工作[5]。

让我们先扯一些题外话，如果读者也想去弄一个去中心化平台，或者开发一个连接去中心化世界的网关，会如何设计？再把目标缩小一点，不要说将传统应用/系统跟去中心化世界对接，而仅仅局限于金融领域，那是不是就变成了DeFi这个话题？哪怕还是聚焦在区块链领域，是不是会去思考“跨链”？如果可以通过ICO/IEO，筹集到大把钱，一般会用什么技术方案或架构，但如果没有ICO/IEO，如同当年中本聪搞比特币时那种状况，仅有少数支持者，又会如何弄？这样去思考一下，会发现很多现实的挑战和不同的实施方案，对吧？

关于ZelNodes，它是可在家中或办公室中搭建，也可通过租用VPS搭建。推荐一下搭建节点的介绍视频[6][7]。回到本文开头所提到的困惑，为啥去中心化算力网络而不是大众熟悉的VPS，况且某些Zel节点也会构建在VPS上？作者认为，最大的区别是，ZelNodes的节点拥有者可以自由加入或退出网路，某种意义上，实现了节点所有权和使用权的分离。在传统网络或集群中，操作系统管理员权限是至关重要的存在，它是由某些人或某些组织控制和维护，一旦密码遭到泄漏，那将会造成重大安全事故和不可估量的损失。但在去中心化网络环境中，这些权限掌握在很多不知来历的匿名人士手上。也就是说，在完全不可信的环境下，可以认为管理员权限是不重要的。即便某些机器、人、组织想做坏事，对整个网络而言，仍然不构成危险，网络依然保证安全可靠，这是区块链技术所带来的全新赛博世界。在2000年左右，想对Web资源进行共享，那么主机托管服务商会在IIS、Apache这类Web服务软件上设置网页共享功能，这样很多小网站就可以用便宜的价格实现供公共访问的Web服务。随着云服务（如虚拟机、Docker容器）成为主流技术后，基于Web服务软件共享的模式就不是最好最安全的解决方案。从这点可以看出，虽然最终用户的共享需求或经济模型

没有发生本质的改变，但它所依赖的互联网赛博世界却已发生了天翻地覆的变革。套用这个模式，前面所涉及的关于VPS话题，换句话说，可理解为虽然最终用户在享受或购买计算/服务这个需求没有本质变化，但其支撑的技术已发生了某些颠覆性变化。关于“未来是去中心化的”这个话题，建议浏览一下TED的六年前演讲视频[8]。简单说，从主机时代的集中化处理、到集群网络架构的客户端/服务器或主/从模式，到互联网站点的开放分散，再到大型互联网公司的资源集中垄断，或云服务商/数据中心的进一步集中化，发展到当下去中心化热潮，都是历史发展过程中的一种演变趋势。古人说过：“天下大势，合久必分，分久必合！”（中文版特有）。

即便认可“去中心化计算网络”这个说法，但逼格感觉还是不够高大上，没有特点！嗯，那我们再发散一下，看看这个世界发生了什么。如果接下来的某些内容，感觉有点烧脑，可直接跳到个人观点部分的段落。

那个发明了互联网的美国国防高级研究计划局(DARPA)，在2017年开展“分散计算”(DCOMP)项目，意在利用已经开始遍布全球各地的计算资源，包括智能手机、平板电脑、联网汽车、物联网终端等等，以便“彻彻底底地重新思考如何联网和计算。它给包括雾计算、边缘计算和分布式计算在内的一系列新兴技术增加了另一个名称。不过，DCOMP让这些范式更进一步，设想这么一个网络：能够从它的众多节点那里借取处理资源和通信资源，随时满足它的用户的资源需求，进而帮助完成他们可能会抛给它的任何任务。（本段内容摘录自互联网）

想了解分散计算(Dispersed Computing)，这一继网格计算、云计算、雾计算、边缘计算之后的新型计算范式，推荐阅读这篇学术论文[9]以及它提及的参考文献。在这篇论文中，提及了这个领域所可能涉及的各种技术，也理清了各种术语所涉及的内涵。尤其提供了一张非常棒的计算演变图：“计算的演变：多年来个人设备发展的普遍性观点”（见下图）。在此图中可以看到集群技术、网格计算、云计算、普适计算、物联网、去中心化账本等，都是计算演变中的前置技术。也就是说，“去中心化计算网络”在利用区块链技术激励节点运营者构建，尤其是这些计算节点不局限在云环境中，而是融入万物互联这一新型环境时，它已经具备分散计算范式(Dispersed Computing Paradigm)的雏形。由于这是一种新型计算范式，因而它所表现的外在形式可以是多样化的。有理由相信，绝大多数人没有意识到这点。最大可能的情况，就是在利用区块链技术赋能已有应用/服务/系统等方面的场景中展开想象。本文作者的主要兴趣领域是人工智能，物联网，安全对等网络等领域的融合上，尤其会专注于基于分散计算范式的智能流对等网状网络的服务。非常期待在ZelNodes计算网络和Flux平台的帮助下构建相关的实验项目，以展示其超出我们当前想象的巨大潜力。如果感兴趣，可以关注作者另外两个GitHub账号[5]，一起交流，共同进步！

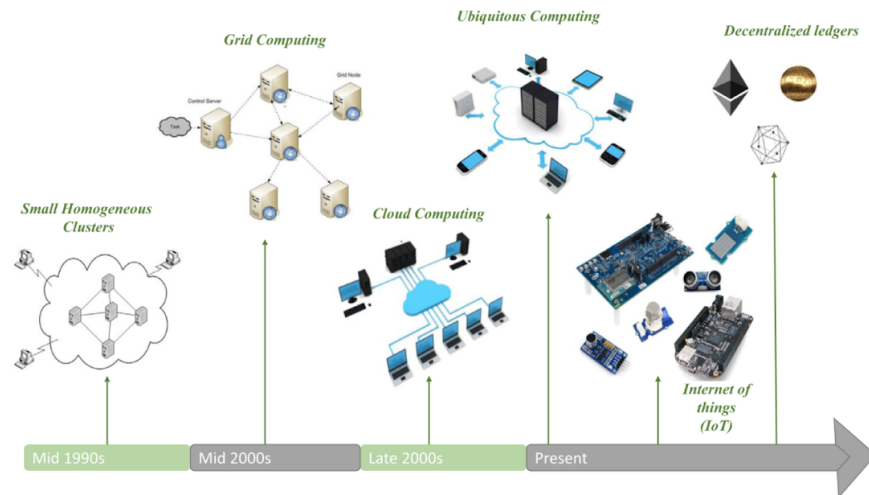


Fig. 1. Evolution of computing: a general view on the evolution of personal devices over the years.

读者也可以阅读这篇中文文献[10](中文版特有)。分散计算是完全分散的架构，将每个计算设备都看作是一个网络计算节点。这种架构更快，更高效，更具可扩展性。旨在提供可扩展且强大的计算网络，将所有设备连接成一个网络化的有机体，充分利用网络中的闲置计算资源，使用异构计算平台来处理大量数据。尽管分散计算、网络计算、雾计算和边缘计算不尽相同，但它们都反映了互联网计算范式在万物互联时代的新需求。网络计算、雾计算和边缘计算都是对云计算的补充，然而，分散计算将这些计算范式又向前推进了一步。通过分散计算最终想要达到目的是：1. 利用无处不在但物理上分散的计算平台，将应用程序和网络性能提高几个数量级。这种计算架构包括网络元件，无线电，具有可编程执行环境的智能电话或传感器，以及不同形式的便携式微云。2. 能够支持用户跨众多不同计算平台进行安全、集体的计算任务分配。3. 能够在网络连接存在变数或降级的环境下运行，使用户能够在代码和数据之间切换，以更好地满足用户、应用程序和任务需求。综上所述我们可知，分散计算是一种新的计算范式，用于设计能够在高度可变和不可预测(也可能是退化的)网络连接条件下运行的系统。（备注：摘录自中文文献[10]作者在知乎上的一段描述）

作者的个人观点：Zel生态系统所提及的“去中心化计算网络”，实际上是分散计算范式(Dispersed Computing Paradigm)的一个正在进行中的社会化实践或者说雏形，它拥有下一代互联网计算范式的强大潜力，这点至关重要！读者是不是感觉到这种逼格要好多了？我们晓得，最终的表现形式并不是一成不变的，如同当年的淘宝或QQ，除非是穿越者，否则无法联系到当下的阿里巴巴/蚂蚁金服或腾讯帝国。但作为投资者/从业者，不就是寻找那种潜力劲头吗？（中文版特有）

扯了那么远，回到ZelFlux这个话题，我们先看看这篇文章[11]. 当读者这段内容时：“Flux使用websocket进行内部通信，通信过程是...扒拉扒拉...”，就在想，为啥没看到“牛逼”的分布式哈希表DHT技术呢？它可是对等网络所依赖的核心技术啊！感兴趣的朋

友，可以参考一下这类论文[12][13][14][15]和资料[16]，也可自行网络搜索，[知乎上](#)也有很多这方面文章（中文版特有）。

在分布式网络上，每个节点不可能存储全部的信息，也不可能通过广播全部节点的方式进行信息检索，同时节点是不保证稳定在线的，还需要保证节点进入/退出对现有网络产生最小的影响，节点也需要通过主动或被动的方式发现其它对等节点。但为什么Flux跟传统模式一样，不需要路由表，通过访问区块链提供的API，就可查找到确定性节点IP地址，如同在一个局域网或传统Web环境中的那种操作方法。这里，是看不出去中心化系统开发的特殊地方。如果仔细跟踪一下，我们可以看到，ZelNodes是由ZelChain驱动，在锁定抵押币之后，通过在钱包上激活对应的确定性节点，在满足ZelBench定期测试后，对应节点讲按照确定性次序获取节点收益。而在这个过程中，它们必须保证稳定上线。无论是基准测试、还是网络连接，都会进行上链操作，这跟一般主节点抵押还是有很大差别的，Flux在这个网络上，完全不需要DHT这类技术，采用跟传统系统类似的方法，实现了去中心化节点间的通信。基于经过久经考验的比特币核心程序代码的支撑、通过零知识证明的隐私技术和确定性节点技术，这条Zel区块链驱动了一个低门槛准入，可自由进出的能提供持续服务的分布式节点网络。由此可见，复杂的区块链技术被隐藏起来了，这样Flux开发者不需要关心诸如ZelChain、ZelBench等后端技术，也无需关心ZelNodes部署在什么地方，数据中心还是家庭/办公环境中。这样，去中心化网络开发也就变得跟传统网络/集群的开发非常类似，类似传统Web开发，可通过MEVN软件栈或Docker技术，就可实现DApp或DService这一类系统开发，这样对Flux开发人员足够友好，并不需要额外的技术门槛。恰恰是这点，以前反而让很多技术人员产生疑惑的地方，为啥Flux技术方案会是如此普通。

可以这么说，让开发人员远离复杂，降低技术门槛，通过常规的技术栈或工具，同样可以实现复杂系统，这是一种非常聪明的做法。这种做法，我们也可以从AI领域的机器学习/深度学习框架看出，它们通过提供诸如Python语言的开发接口，让广大开发人员可以非常方便的从事AI建模/训练等方面的工作。这种方式，将C/C++编写的复杂的核心代码功能模块给隐藏起来了。从微软Windows和Google Android的开放接口这种商业生态，也可以看出端倪。通过降低外部开发人员的门槛，是一种很聪明、很主流的工程与商业结合的方案。SpaceX发射火箭的程序，同样也利用很多主流的软件开发栈包括Python脚本语言，其中的道理是完全一样的。

ZelCore多币种钱包、ZelCore+中集成交易所API的方式，同样看起来都很普通。随着下一个阶段ZelCore和ZelFlux的深度融合，那些钱包用户，将会跟ZelFlux上的应用/服务进行了连接，从而使得整个Zel生态的进入门槛大幅度降低，同样是这样的道理。由此，未来一定可以见证，“平凡之中孕育不平凡的种子”这句格言！

说个题外话，当我们想到构建算力市场的时候，就会想到供给方和需求方。如果一个项目的中介方或者平台，在两个方向都去主攻的时候，无论技术路线如何美好，实际运营

都会面临非常大的挑战。如果搞定其中某个方向，无论是供给方还是需求消费方，剩下仅主攻一个方向时，那么在商业上，项目成功性就会大大提升。Zel通过类似主节点的质押奖励模式，按照某种规格设置（基于扩展需求，可以猜测未来，一方面会提高规格要求，另一方面也会有更低的抵押数量要求），提供了一个物理位置不限定（无论是云数据中心、还是家庭或公司网络）的计算资源供给。这是一个非常巧妙的办法，解决了市场经济模型中的一个难题。加上上述讨论的，降低需求方（开发人员）的进入门槛，便使得“去中心化计算网络”这个愿景容易在运营中取得成功。而采用ICO/IEO的基于市场概念的项目，可以看看它是否会面临供需两方的挑战，如果存在，可以肯定，其运营难度系数将会超级高。

如果项目前期募集到很多资金，可以组建比较大一些的团队，项目上线之前有很长一段的研发周期，那么可以面向未来，选择一些诱惑力高的技术方案。但对于没有ICO/IEO的那些项目，而且团队规模也较小，往往这些项目都处在实际运行中的，而不是测试或演示环境中，如同飞机一边飞一边升级维护的场景下，往往需要选择稳健保守方案。如看起来不出众，但极其稳定、久经考验的比特币代码核心，借助于区块链技术驱动实际世界中的多种应用场景。尽管显得不那么“绚烂多彩”，但其以踏实稳健的进取态度，用时间换取足够的成长空间。这在投机成风的大环境下，不失为一种理性态度。

到此，啰哩啰嗦，也大致回答了作者一直以来关于技术选型的困惑。即便未来要更深程度的去实践分散计算范式，无论应用到什么领域或场景，那种低门槛要求或采用主流软件开发栈的方式将会随处可见。基于理性长期投资态度，作者选择保守主义倾向的项目。

当然世界本该是多样的，除了上面所说的那种方向，也有类似Apple垂直整合的商业形态。在大数据时代，除了Python外，还有Julia[17][18]这种特殊的语言也非常流行。这种商业策略，在某种程度上会提高技术门槛、牺牲开发人员友好度以及受众普及率，但可以获取更好的性能、更高的效率、更低的消耗、更精简的实现，同样也可以获得成功，它反映了“天下武功、唯快不破！”的道理（中文版特有）。当看到其它一些区块链项目，选择走这条路，也在情理之中了。由此，在分散计算范式的高风险试验性项目中，作者本人基于兴趣的原因，在开发方面会偏好这个模式。

综上，本文主要想阐述两个话题，其一：分散计算范式，是面向下一代互联网的新兴技术领域，而ZelNodes和ZelFlux所尝试的去中心化计算网络，将是一个有意义的尝试；其二：ZelFlux采用Web开发人员所熟悉的开发栈和较低的技术门槛，隐藏了复杂的区块链技术，这是一种非常聪明的商业策略。

参考链接：

[1]. Flux, The Gateway To a Decentralized World!

<https://www.youtube.com/watch?v=FeauNYxEzx4>

中文字幕：<https://weibo.com/tv/show/1034:4539004771827736>

[2]. Everything To Know About ZelFlux OS— Zel Computational Network

<https://medium.com/@ZelOfficial/everything-to-know-about-zelflux-os-zel-computational-network-35528eed87b7>



中文翻译: <http://www.niubiquan.com/article/1279>

- [3]. The ZelNodes Operating System — ZelFlux — Will Debut Sept. & Oct.  
<https://medium.com/@ZelOfficial/the-zelnodes-operating-system-zelflux-will-bring-useful-work-to-the-zel-computational-network-9ec340492763>
- [4]. Zel Website: <https://zel.network/>
- [5]. (a) Talking\_about\_ZEL : [https://github.com/jniva/Talking\\_about\\_ZEL](https://github.com/jniva/Talking_about_ZEL)  
(b) Streaming-Peer-Mesh-Network <https://github.com/thesmallboat>  
(c) Streaming-AI <https://github.com/flowsmind>
- [6]. (a) ZelNode Setup Guide Part 1: <https://www.youtube.com/watch?v=aExUEjFH1IQ>  
中文字幕: <https://weibo.com/tv/show/1034:4536042402021383>  
(b) ZelNode Setup Guide Part 2: <https://www.youtube.com/watch?v=P8ZbjWvljvw>  
中文字幕: <https://weibo.com/tv/show/1034:4536954990297093>
- [7]. Zelnode (VPS/Server) setup using Zelcore Control Wallet  
[https://www.youtube.com/watch?v=5\\_KapYFgtjo](https://www.youtube.com/watch?v=5_KapYFgtjo)
- [8]. The future will be decentralized | Charles Hoskinson | TEDxBermuda  
<https://www.youtube.com/watch?v=97ufCT6lQcY>
- [9]. Introducing the new paradigm of Social Dispersed Computing: Applications, Technologies and Challenges [https://github.com/TheSmallBoat/Technical\\_Survey/blob/master/pdf/S1383762118301036.pdf](https://github.com/TheSmallBoat/Technical_Survey/blob/master/pdf/S1383762118301036.pdf)
- [10]. 分散计算: 技术、应用与挑战 <http://fcst.ceaj.org/CN/abstract/abstract2185.shtml>  
PDF: <http://fcst.ceaj.org/CN/article/downloadArticleFile.do?attachType=PDF&id=2185>
- [11]. Let's talk Flux <https://medium.com/@kmentat/lets-talk-flux-a1f085baec0a>  
中文翻译: 让我们谈谈助焊剂 <http://www.niubiquan.com/article/2761>
- [12]. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications  
[https://github.com/TheSmallBoat/Technical\\_Survey/blob/master/pdf/chord.pdf](https://github.com/TheSmallBoat/Technical_Survey/blob/master/pdf/chord.pdf)
- [13]. OpenDHT: A Public DHT Service and Its Uses  
[https://github.com/TheSmallBoat/Technical\\_Survey/blob/master/pdf/opendht-sigcomm2005.pdf](https://github.com/TheSmallBoat/Technical_Survey/blob/master/pdf/opendht-sigcomm2005.pdf)
- [14]. Kademlia: A Peer-to-peer Information System Based on the XOR Metric  
[https://github.com/TheSmallBoat/Technical\\_Survey/blob/master/pdf/kademlia\\_2492563.pdf](https://github.com/TheSmallBoat/Technical_Survey/blob/master/pdf/kademlia_2492563.pdf)
- [15]. S/Kademlia: A practicable approach towards secure key-based routing  
[https://github.com/TheSmallBoat/Technical\\_Survey/blob/master/pdf/skademlia\\_4319659.pdf](https://github.com/TheSmallBoat/Technical_Survey/blob/master/pdf/skademlia_4319659.pdf)
- [16]. Kademlia protocol succinctly  
[https://github.com/TheSmallBoat/Technical\\_Survey/blob/master/DDS/Succinctly\\_ebook/kademlia\\_protocol\\_succinctly.pdf](https://github.com/TheSmallBoat/Technical_Survey/blob/master/DDS/Succinctly_ebook/kademlia_protocol_succinctly.pdf)
- [17]. Julia Website: <https://julialang.org/>
- [18]. Julia: A fresh approach to technical computing  
<https://medium.com/dev-genius/julia-a-fresh-approach-to-technical-computing-1904a7e6b023>