为什么ZelFlux不用“牛逼技术”就能实现去中心化通信？[Chinese Version]

关于ZelFlux的介绍，可以看一下这段视频***[1]***或文章***[2][3]***，它是Zel生态系统***[4]***中的一个重要组成部分。你对这标题感兴趣，说明跟作者一样，想知道或者评估一下ZelFlux或Zel生态系统相关的技术状况。

如果你是一个技术宅，你一定会有一些疑惑，为啥ZelFlux的技术，感觉没有其它项目那么高大上？它的技术实现手段跟传统的Web开发没啥区别，比如采用Javascript，MEVN软件栈(MongoDB,Express,Vue.js,Node.js)，这是不是太普通了？“算力操作系统”或文雅一点的说法“Dapp管理平台”，这不是太扯淡了吗？或者说，VPS虚拟机主机方案，不是挺好用的吗，为啥还需要一个所谓的去中心化算力网络，这不是多此一举吗？

如果你有这些疑惑，我们是同道中人，因为这也困惑我好久，至少大半年，那么请耐住性子，看看我的看法是不是有一定道理。如果你有观点，也可以到Github***[5]***上提问题或递交更新版本，大家一起协同编写本文的后续。

我们先扯一些题外话，如果你要去弄一个去中心化的平台，或者做一个连接去中心化世界的网关，你会如何去做？如果把目标缩小一点，不要说将传统世界应用跟去中心化世界对接，仅仅局限于金融领域，那是不是就变成了DeFi这个话题的讨论？哪怕还是聚焦在区块链领域，那也可以聊聊如何跨链啥的？如果你可以通过ICO/IEO，筹集到大把钱，你会用什么方案，但如果没有ICO/IEO，如同当年中本聪搞比特币时的那种状况，仅仅只有少数支持者，你会如何弄？我认为这样一思考，就会很有趣，会有很多不同的现实挑战和实施方案，对吧？

我想你已经知道了ZelNodes，它是可以在家或办公室中搭建，也可以租用VPS搭建，这里有相关视频***[6][7]***介绍。会到开头提到的疑惑，为啥是去中心化算力网络而不是我们熟悉的VPS，况且你的某些节点也是构建在VPS上的？作者认为，最大的区别是，ZelNodes的节点拥有者是可以自由或退出整个网路，节点所有权和使用权实现分离。但在传统的网络和集群中，操作系统管理员权限对于系统来说是至关重要，它是由某几个人和某几个组织所控制和维护，一旦泄漏，将会造成重大但安全事件和损失。但去中心化网络环境中，这些权限掌握在很多人不知来历的人手上，也就是说在完全不可信的环境下，我们认为管理员权限是不重要的，即便某些机器、人、组织想要做坏事，但对于整个网络来说仍然还是安全可靠，这是区块链技术所带来的新世界。2000年左右，要实现Web资源共享，主机托管服务商会在IIS这类Web服务器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（虚拟机、Docker）成为主流后，基于Web服务器共享的模式就不是最安全最好的解决方案。从这点可以看出，虽然最终用户的共享需求（经济模型）没有发生本质的改变，但它所依赖的互联网技术世界却已发生了天翻地覆的变革。套用这个模式，前面提到的那个关于VPS的话题，换句话说，可以理解为，虽然最终用户在享受或购买计算/应用服务这个需求没有变化，但其支撑的技术世界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六年前的演讲视频***[8]***。简单说，从主机时代的集中化处理、到集群网络客户/服务或主/从的服务模式，到互联网站点的开放分散，到大型互联网公司的集中垄断，到云服务商/数据中心的集中化，再到当下的去中心化，都是历史中的一种演变趋势。 **古人说过：“天下大势，合久必分，分久必合！”**（中文版特有）。

就算认可了“**去中心化算力网络**”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。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。

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。”DARPA的分散计算项目（被称为DCOMP）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。

我们可以了解一下，继网格计算、云计算、雾计算、边缘计算之后的新型计算范式：分散计算(Dispersed computing)，推荐阅读这篇论文**[9]**。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）。我们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，都是计算范式演变中的前置技术。也就是说，“去中心化算力网络”在利用区块链技术构建，尤其是计算节点不局限在云环境中，在万物互联的新型环境中时，它已经具备分散计算范式的雏形。由于这是一种新型计算范式，因而它的外在形式应当是多样化的。有理由相信绝大多数人都没有意识到这点，最大可能，就是利用区块链技术，在赋能已有应用/服务/系统等方面场景展开想象。作者的主要兴趣领域是人工智能，物联网，安全对等网络等。将来，将专注于基于“分散计算范式”的智能流对等网状网络服务。也非常期待在ZelNodes计算网络和Flux平台的帮助下构建实验项目，以展示其超出我们当前想象的巨大潜力。也可以关注作者的GitHub账号***[5]***，一起共同进步。

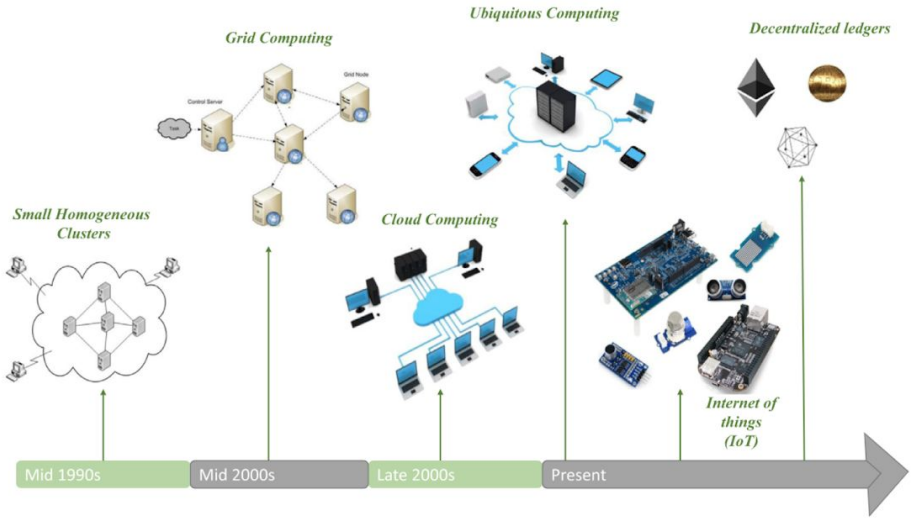

Fig. 1. Evolution of computing: a general view on the evolution of personal devices over the years.

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、网格计算、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。网格计算、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的目的是：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算力网络，实际上是分散计算范式的一个正在进行中的社会化实践或者说雏形，它拥有下一代网络或计算范式的强大潜力，这点至关重要！现在感觉逼格应该够了！**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]***. 当我看到这段，“*Flux使用websocket进行内部通信，通信过程是...扒拉扒拉...*”的时候，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，也不可能通过所有广播或遍历的方式进行信息解锁，节点是不保证稳定在线的，因而节点的进入退出对现有的网络没有影响，节点也需要去主动或被动发现其它的对等节点。但为什么Flux居然跟传统模式一样，通过访问API，就可以查找确定性节点的IP地址，而不需要路由表，如同在一个局域网或传统Web环境中的操作方法。这里，是看不出去中心化系统的特殊地方。仔细去跟踪一下，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，同时需要满足ZelBench的定期测试，对应节点按照确定性次序，可以获取对应的节点收益。而在这个过程中，它必须保证稳定上线。无论是基准测试、还是网络连接，这些都会进行上链操作，这跟一般的主节点抵押还是有很大差别的。也就是说，基于经过久经考验的比特币核心程序代码的支撑、基于零知识证明的隐私技术，这条区块链驱动了一个低门槛准入，可自由进出的持续服务的分布式节点网络。而Flux在这个网络上，完全不需要DHT这类技术，可以直接从区块链上获取确定性节点的IP地址。由此可见，复杂的区块链技术被隐藏起来了，这样很少有人关心ZelChain、ZelBench等后端技术，也无需关心ZelNodes是部署在什么地方，数据中心还是家庭/办公环境中。这样，去中心化网络也就变得跟传统网络/集群非常类似，对开发人员足够友好，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。但这一点，恰恰是让很多技术人员发生疑惑的地方，Flux的技术方案是如此普通。

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接口，让广大开发人员从事AI建模方面的工作，而将复杂的利用C/C++编写的核心代码功能模块隐藏起来。从MS Windows和Android的开放接口这种商业生态，也可以看出，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。从SpaceX发射火箭的程序，利用很多主流的软件开发栈包括Python脚本语言的道理一样。

当然ZelCore的多币种钱包、ZelCore+中的集成交易所功能，看起来都很普通。随着下一个阶段ZelCore和ZelFlux的深度整合，普通钱包用户，将会跟ZelFlux上的应用/服务进行了连接，从而使得整个生态门槛进一步降低，也是如此的道理。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，当我们想到构建计算力市场的时候，就会想到供给方和需求方，一个项目中介方或者平台，两个方向都去主攻的时候，无论技术路线如何美好，实际运营将非常具有挑战性。如果搞定某个方向无论是供给方还是需求消费方，主攻一个方向的时候，项目的商业成功性就会大大提升。Zel通过类似主节点的模式，按照一定的规格设置（基于未来扩展的可能性，一方面会提高规格要求，另一方面也会有更低抵押数要求），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）的计算节点供给。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，便使得计算力网络这个愿景容易在运营中取得成功。而采用ICO/IEO的基于市场的项目，可以看看它会否面临两个方面的挑战，如果存在，那该项目运营的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，可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，依此利用区块链技术驱动实际世界中的多种应用场景。显得不那么“绚烂多彩”，但以踏实稳健的进取态度，用时间换取成长空间，在投机成风的大环境下，也不失为一种理性的态度。

到此，也就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，那种低门槛要求或主流的软件开发栈随处可见，便在情理之中了。基于理性投资，我会选择采用这种保守主义类型的项目。

当然世界本来就该是多样性的，除了上面所说的那种方向，还有类似Apple那种垂直整合的商业形态。在大数据时代，除了Python以外，还有Julia***[17][18]***这种特殊的语言，也非常流行。这个方向，在某种程度上会提高技术门槛、牺牲开发人员友好度和受众的普及率，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。也反映了“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，在分散式计算范式的试验性项目（更高风险）中，作者本人偏好于选择这个方向。

参考链接：

[1]. Flux, The Gateway To a Decentralized World！https://www.youtube.com/watch?v=FeauNYxEzx4
带中文字幕视频： 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
(b) Streaming-Peer-Mesh-Network https://github.com/thesmallboat
(c) Streaming-AI https://github.com/flowsmind
[6]. ZelNode Setup Guide Part 1: https://www.youtube.com/watch?v=aExUEjFH1IQ
中文字幕：https://weibo.com/tv/show/1034:4536042402021383
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
[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
[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
[13]. OpenDHT: A Public DHT Service and Its Uses
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
[15]. S/Kademlia: A practicable approach towards secure key-based routing
https://github.com/TheSmallBoat/Technical_Survey/blob/master/pdf/skademlia_4319659.pdf
[16]. Kademlia protocol succinctly
kademlia_protocol_succinctly.https://github.com/TheSmallBoat/Technical_Survey/blob/master/DDS/Succinictly_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