结构决定功能：AO 与 Nostr 的分析比较

AO 是超并行计算，Nostr 是去中心化的社交协议，怎么进行比较呢？各自有着怎样的定位和发展路径？

当你看到标题时可能会觉得奇怪：AO 代表超并行计算，Nostr 则是去中心化的社交协议，这两者似乎属于完全不同的领域，不容易进行比较。然而，如果我们深入技术实现的层面，会发现它们本质上都是“消息传输协议”，这样一来，它们就具有了可比性。

作为消息传输协议，其中的核心要素自然是“消息”本身，一切围绕消息的传递展开。那么在 AO 和 Nostr 网络中，各自的消息是如何定义的？它们支持消息传输的网络架构又是如何搭建的？又如何与其他协议集成？它们各自的定位、主要应用场景和发展趋势又是什么？

本文旨在深入比较 AO 与 Nostr 这两种协议，重点探讨协议结构对功能的影响，并围绕上述问题展开详细分析。

消息的概念和特征

AO 中的 Message

在 AO 网络架构中，消息（Message）是各个网络单元（MU、SU、CU）之间或各个进程之间传递的基本信息单元。它们通过发送和接收消息来实现信息交换和协作。

AO 被设计为一个消息驱动的异步通信网络。首先，AO 在启动工作（启动一个进程）时需要消息来驱动，这些消息可以来自外部用户或其他进程。其次，AO 的进程间通信是异步的，这意味着消息的发送和接收是独立于发送者和接收者的操作。发送消息的进程不需要等待接收者的响应或确认，可以立即继续执行其他操作。这种特性显著提高了 AO 并行计算的效率。

在 AO 网络中，消息的异步传输和无需等待的特性使其成为处理大规模并行计算任务的理想选择，因为它允许系统的各个部分能够并行执行，而无需长时间等待其它进程的响应。

在 AO 中，每条消息都遵循 Arweave 生态系统的 ANS-104 标准，这是一种数据打包协议。ANS-104 通过将多个交易以二进制方式序列化到一笔原生交易中，显著提升了数据读写的吞吐量。这个协议不仅仅是简单的数据打包，还在打包数据中添加了所有者、签名、目标地址、标签、数据等字段。这种设计使得 ANS-104 能够支持多种数据类型，包括文档、图片、音视频、游戏、数据模型、程序代码以及全息状态等。此外，它还支持数据所有权和签名验证，为数据的安全性和完整性提供了保障。

ANS-104 标准的这些特性对 AO 尤为重要，因为它们使得 AO 能够针对不同的数据类型建立多样化的应用场景。统一的消息格式也极大地促进了进程间的高效通信和无缝协作，还可以提高在 Arweave 上进行存储和结算处理的效率，使得 AO 能够有效地构建数据可用性层和数据共识，从而支持其广泛的应用需求。

Nostr 中的 Event

在 Nostr 协议中，采用了基于 JSON 结构定义的一种标准消息格式，用于描述各种类型的数据。这种消息格式被称为事件（Event），是 Nostr 网络中的基本数据对象。

最常用的消息结构正在被整合到一个共同遵循的标准中，称为 NIPs（Nostr Implementation Possibilities）协议标准。这种一致性和标准化大大提升了数据处理和管理的效率，有助于增强系统的互操作性和稳定性。通过 NIPs，用户可以在 Nostr 网络中执行各种操作和互动，而无需担心数据格式的差异性或不兼容性问题。

JSON 结构用于定义 Nostr 协议中的事件数据格式，在此结构中包含不同的字段，每个字段具有不同的作用，举例来说：

pubkey 字段：表示发送事件的用户公钥，用于标识用户的身份。此公钥将用于对事件进行数字签名，以确保事件的真实性和完整性。

kind 字段：用于指示事件的类型。它可以表示不同的消息类型，如聊天室消息、钱包信息等，也可以表示用户的特定行为，例如推荐中继列表、执行某项操作等。

content 字段：包含事件的具体内容。这个字段支持多种数据类型，例如社交媒体贴子、论文、音视频等。用户可以在这里表达他们想要传达的各种信息和数据。

sig 字段：用于存储事件的数字签名。这个签名由发送者使用他们的私钥生成，然后由接收者的客户端使用相应的公钥进行验证。签名能够确保具有对应公钥的用户在特定日期确实发送了该事件，从而增强了事件的可信度和可验证性。

……

详细的事件数据结构描述可参见：https://nostr.how/zh/the-protocolcontent。Nostr 协议通过这些字段和结构，为事件的发送、接收和验证提供了一套清晰的框架。这种设计有助于确保数据的安全性、一致性和可靠性。

所有字段组合在一起，以此来表达一个完整的、发挥特定作用的事件。概括来说，事件是一个包含任意内容并由用户签名的数据结构，这个结构反映出 Nostr 的定位、特征和作用：

Nostr 是一个信息发布、存储和接收系统。Nostr 通过采用 JSON 结构和 NIPs 协议标准，建立了一个高效的数据交换和管理框架，确保了信息的一致性和可解析性，为用户提供了稳定可靠且统一的信息通信环境；

支持客户端验证：该数据结构支持在客户端进行验证，因此无需信任任何中继服务器或第三方，可以直接验证事件的真实性和完整性。

构建去中心化、抗审查、可验证的社交网络：这种数据结构的特性使得 Nostr 可以作为一个去中心化的平台，用户可以在其中自由交流和共享信息，而无需担心审查或信息篡改问题。

支持消息传输的网络结构

AO：MU/SU/CU 构成协作网络

AO 网络由 MU、SU 和 CU 三个模块化基本单元构成，并通过消息和进程来运行和协作，其网络架构如图 1-1 所示。



图 1-1：模块化且相互协作的网络单元形成AO的网络架构（图片来自AO白皮书）

进程（Process）是 AO 网络中的计算单位，启动一个应用程序在 AO 上等同于启动一个或多个进程，系统会为每个进程分配和调度资源，如 MU、SU、CU、虚拟机、内存等，以执行该进程：

MU（Messenger Unit）：这是信使单元，负责将信息发送到合适的 SU 进行处理，然后交付给 CU 执行计算，计算结果再返回 SU，信使单元不断重复这一过程；

SU（Scheduler Unit）：这是调度单元，负责调度和消息排序，并将消息上传至 Arweave；

CU（Compute Unit）：这是计算单元，接受消息，执行计算，实现状态转换。

如上所述，AO 的网络结构和运行方式表明：

AO 是一个消息传输系统

消息是进程中的核心要素，是 MU、SU 和 CU 的唯一的工作对象。整个进程围绕消息而展开，进程就是消息集合的运行活动，包含了从接受消息、消息传输、消息调度和排序、执行计算（消息状态转换）、输出并存储计算结果的完整过程。

所以 AO 是一个消息传输系统，可以致力于构建信息发布和实时交流互动、内容分发等方面的应用，如去中心化的社交网络、社交媒体以及去中心化的音视频点播/直播平台等。

AO 是一个超并行计算网络

AO 是一个模块化网络，其中计算在链下执行，不受区块共识的约束，使得计算单元（节点）能够根据需要进行无限扩展，从而极大提升了计算性能。

在 AO 的环境中，可以同时开启任意数量的计算任务（并行进程），这些进程能够在不同的计算节点上独立运行，并在本地完成验证。这使得 AO 成为一种分布式、可验证的超并行计算机。

尽管各个计算进程可以在不同的节点上独立运行，它们之间可以通过统一的信息格式（ANS-104）进行消息传递和协作。这种方式将独立运行的计算进程连接到一起，形成一个统一的网络。

AO 是一个开放平台

AO 的核心是一个信息协议，允许运行在 Arweave 上的不同应用之间相互通信。每个应用可以通过 AO 网络向其他应用发送信息，利用 AO 进行组合性运算，实现跨链信息交换。

AO 网络运行在链下，能够与 Web2 应用实现无缝连接。通过调用 AO 协议接口，Web2 应用可以参与到这个去中心化网络中。这种特性使得 AO 能够消除 Web2 和 Web3 应用之间的隔阂，实现可信的信息交换和应用之间的互操作。AO 的通信协议设计机制使其成为一个开放的平台，为开发者带来了无限可能性。

综上所述，AO 的网络架构支持其成为一个可组合、互操作、可扩展、可验证、去中心化和开放的计算机网络平台。它不仅适用于以信息发布和交流互动为主题的社交应用，还能够支持对计算性能要求更高、业务逻辑更复杂的应用，例如机器学习、自主决策代理、图形渲染、在线游戏和 DeFi 应用等。

Nostr：Client-Relay 结构

Nostr 是“Notes and Other Stuff Transmitted by Relays”的缩写，意思是“通过中继传输注释和其他内容”。网络中有两个主要的组件，其网络结构如图 1-2 所示。



图 1-2 Nostr的网络结构

客户端（Client）

这是在用户端运行的应用程序，用于将数据读写到中继服务器。客户端使用公钥作为用户发送和接收事件的地址，而私钥则用于在发送事件时进行签名，以证明是用户本人的操作并防止篡改。在接收事件时，客户端使用私钥验证签名，以确认事件的来源和完整性。

客户端允许用户连接到分布在不同地点、任意数量的中继服务器。用户可以在一个中继上发布信息，并在另一个中继上检索信息。这意味着客户端（用户）不必依赖于任何特定的中继服务器，从而有效保护用户数据和行为。

中继服务器（Relay）

中继服务器具有监听、抓取和存储与其连接的客户端事件的能力，并将这些事件转发给已订阅的客户端。

任何人都可以运行中继服务器，多个中继服务器可以互相替代，这种设计弱化了单个中继的重要性，降低了单点故障的风险，提升了抗审查能力。此外，多个中继之间的竞争能够推动服务质量的提升，如提供更大的存储容量、更快的响应速度以及垃圾信息的过滤等服务。

中继服务器可以根据自身需求选择存储用户全部或部分内容，并决定存储的时间长度，这为中继的定位和商业行为提供了更大的灵活性。与此同时，中继服务器之间无需进行相互通信，因此不存在共识层面的问题，也无需进行数据同步。数据同步是通过客户端之间的事件发送和接收来实现，这与区块链节点有根本的不同。

这样的架构不仅能够提升系统的灵活性和效率，还能够有效地应对不同的使用场景和需求。

可见，Nostr 采用了 Client-Relay 这样的轻量级网络结构，不仅能够提升系统的灵活性和效率，还能够有效地支持构建一个去中心化、抗审查、可验证的信息发布系统，满足人们对言论自由、流畅交流和数据安全与隐私控制的需求。这种设计有效应对了中心化社交媒体带来的种种挑战和弊端。因此，Nostr 被誉为去中心化社交协议，广受开发者青睐，用于构建各类去中心化社交应用，例如 Damus、YakiHonne、Iris 等。

与其他协议集成

AO+Arweave：去中心化的世界计算机

AO 运行在 Arweave 之上，实现了与 Arweave 的无缝集成，如图 3-1 所示。



图 3-1 AO与Arweave无缝集成（图片来自AO白皮书）

这是存储共识范式（SCP）的一种实现。在这种创新的范式中，存储（共识）与计算被有效地分离，使得链下计算和链上共识成为可能。这种架构带来了显而易见的优势：

高性能计算： 智能合约的计算在链下执行，不再受制于链上的区块共识过程，从而大大扩展了计算性能，使高性能计算成为可行。

超并行计算： 不同节点上的各个进程可以独立地执行并行计算和本地验证，而无需像传统的 EVM 架构中那样等待所有节点完成重复计算和全局一致性验证。这种设计使得 AO 能够实现超并行计算。

自定义计算：Arweave 为 AO 提供了所有指令、中间状态和计算结果的永久存储，作为 AO 的数据可用层和共识层。每个应用程序（智能合约）的执行与 Arweave 中存储的数据密切相关，计算逻辑和数据资源可以根据应用在本地节点执行的需求进行自定义。这种灵活性远远超过了传统的 EVM 模型，其中所有节点必须同时执行预定义操作以追求全网状态的一致性。

综上所述，AO 为 Arweave 增添了超并行计算层，而 Arweave 为 AO 提供了存储即共识的支持。这两者的结合创造了一个去中心化的世界计算机，为去中心化世界开启了广阔的应用创新空间。

Nostr + Lightning：构建去中心化信息

和价值网络

因为 Nostr 的开发者 fiatjaf 同时也是闪电网络的开发者，因此 Nostr 原生支持闪电网络。闪电网络（Lightning Network）是比特币区块链的第二层解决方案，通过使用通道（Channels）在链下扩展区块链的功能。它有效地解决了比特币交易速度慢、吞吐量有限以及交易成本高昂等问题，使得高频、低成本的比特币小额支付成为可能。

Nostr 与 Lightning Network 的结合，最直接的应用之一是在社交应用中实现“打闪”。Nostr 的流行客户端 Damus 内置了比特币闪电网络支付功能，用户只需填入 Nostr 的公钥，即可轻松调用闪电网络支付 Relay 的一次性费用。支付完成后，用户会收到一个闪电网络发票。有关其详细工作流程，请参阅：https://nostr.how/zh/zaps，网页中详细解释了操作步骤。

在资产发行方面，比特币一层的资产发行协议 Taproot Assets（TAP）与闪电网络兼容，可以将 Taproot 资产和比特币的最小单位 Satoshis 引入到 Nostr 生态系统中，并利用闪电网络进行即时和经济实惠的资产转移。这不仅丰富了 Nostr 的资产种类，还为社交网络、支付和 DeFi 等应用场景提供了更多可能性。

此外，CKB 社区的成员还提出了 Nostr 绑定协议，利用 RGB++ 技术将 Nostr Event 与 CKB CELL 进行同构绑定。这一举措使得用户可以在 Nostr 社交网络中创建和分发原生资产，有效解决了社交网络中的原生支付问题。

更重要的是，Nostr 与 Lightning Network 的结合正在为去中心化应用开辟一条全新的商业模式，即 V4V（Value 4 Value）。

V4V 的理念认为，将非稀缺信息货币化是一项非常困难的任务。传统网络世界默认的货币化模式通常依赖于广告，这种模式严重依赖于中心化监控和用户行为分析。V4V 提供了一种替代方案，允许信息和价值的自由流动，无需受到限制或中介。这种方法不仅是在线货币化比特（bit）和字节（byte) 的一种新方法，也是内容创造和价值传递的新途径。

V4V 的解决方案正在为基于 Nostr 构建的社交应用、播客和直播平台等带来创新价值，例如：

去中心化媒体信息交互协议 YakiHonne，支持了 Nostr 与闪电网络的集成，使用 SATS 作为打赏支付，全年支付总额超过 9000 万 SATS。

Nostrwatch.live 是一个运行在 Nostr 和闪电网络上的去中心化直播平台，构建了一个“Value for Value”的双向流价值交换平台。当播放者向观众发送实时媒体流时，同时也从观众那里接收 SATs 支付流。一旦支付停止，流媒体信号也随之停止播放。这与传统的预付费模式不同，观众无需事先订阅或预付费用。

Podverse 是一款 Podcasting 2.0 应用，能够与 Alby 集成，使用闪电网络向播客发送 boostagram（一种向创作者发送消息的捐赠方式）和 sat 支付流。当在 Podverse 中“流式传输 Satoshis”时，该应用程序会按每分钟的收听时间将 Satoshis 发送给用户正在收听的播客。

可以看出，通过 Nostr 与 Lightning 的结合，Nostr 从一个去中心化信息传输网络逐步演变为一个融合信息和价值的去中心化传输网络。这一进化不仅能够保护个人言论免受侵犯，还能确保个人资产的安全，使其成为价值交换的载体和网络。这种演进为规模化和消费级应用的创新开辟了新的可能性和发展机遇，可能成为实现Web3 大规模采用的一条可行途径。

总结：结构决定功能

我们在上文中主要从数据结构和网络结构两个方面对 AO 和 Nostr 协议进行了比较分析，并根据“结构决定功能”的原则，深入分析了各协议支持的主要功能和应用场景：

从数据结构角度来看

从数据结构角度来看，AO 和 Nostr 协议有一些共同点。它们都是信息传输协议，支持各种数据类型的信息发布、交流互动和内容分发，能够构建去中心化社交网络和去中心化社交媒体应用。此外，它们都具备去中心化、抗审查、可签名验证以及隐私和安全保护的功能特征。

然而，它们也有显著的不同之处。Nostr 协议的定位和主要应用场景侧重于上述这类应用，而这些只是 AO 数据结构所支持功能和应用的一个小子集。AO 协议的重点则在于支持超并行计算，其应用领域更为广泛和深远。

从网络结构角度来看

从网络结构的角度来看，AO 协议具备模块化、相互协作和可扩展的网络单元，允许进程在不同节点上独立运行并进行本地验证，这些特性为实现超并行计算提供了基础条件。

同时，AO 协议通过基于 SCP 范式与 Arweave 的无缝集成，突破了区块链技术的不可能三角限制。它可以根据需求无限扩展存储资源和计算资源，并利用 Arweave 永久存储的具有所有权保护的共识数据进行任意进程（应用）间的信息交换和协作。因此，AO 协议能够构建面向全球的高性能、超并行计算网络，为 Web3 甚至 Web2 应用带来了创新空间。

举例来说，它支持需要大语言模型（LLMs）和高密度计算的机器学习应用；支持能够满足具有复杂业务逻辑、可预定义需求和多元化自主策略的 AgentFi；支持强调数据所有权和内容变现的版权管理与创作者市场（ContentFi）；以及能够支持有数据共识要求，需要进行信息跨链通信、资产跨链流动、数据跨链共享以及智能合约跨链互操作等去中心化应用等。

相比之下，Nostr 协议则主要由 Cliet-Relay 两个网络组件和 Event 数据结构以及公私钥体系构成，它建立了一个轻量级的信息发布和接收网络。当 Nostr 协议与 Lightning 集成后，整个网络融合了去中心化信息网络和去中心化价值网络的特征，更适合构建规模化、消费级应用网络的需求。

从协议定位角度来看

从协议定位角度来看，虽然 AO 和 Nostr 都是消息传递协议，它们的定位和重点有所不同。AO 协议的主要侧重点在于构建“去中心化世界计算机”这样的基础设施，定位下层，但却打开了支持各类应用的广阔空间，能够捕获更广泛的价值，带来不断向上发展的力量。

相比之下，Nostr 协议最初设计时就倾向于支持社交应用，定位为一个轻量级的去中心化社交协议，其应用场景更为特定和聚焦。

综上所述，AO 和 Nostr 在数据结构、网络结构以及协议功能上各具特色和优势，有着不同的定位和适用场景。它们将在不同的发展路径上展现出各自的潜力和价值。