TimeRose：

基于IPFS-FAN的计算网格

Computing mesh over IPFS

taoshengshi @KEN Labs

内容

- 在IPFS中引入计算:IPFS-FAN
  - 内容寻址网络
  - 计算寻址网络
- 在IPFS-FAN中引入计算网格:TimeRose
  - 当前问题: 目前Web3领域的所有计算类型都无法提供通用计算
  - 解决方案: 在IPFS-FAN之上构建计算网格
  - TimeRose 计算网格的架构
  - TimeRose 计算网格的功能
  - TimeRose 计算网格的用例
- 问题与讨论

IPFS-FAN是什么，为什么要引入IPFS-FAN？

在IPFS中引入计算

在IPFS中引入计算

IPFS-FAN中引入计算网格

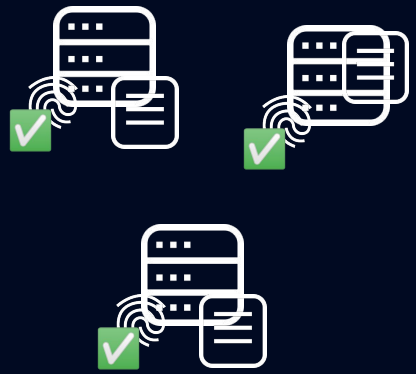
问题与讨论

IPFS: 内容寻址网络

内容的地址(即它的指纹)来自内容本身，而不是来自它存储的位置。

基于内容的寻址:

询问多个主机，选择最近的主机



cat.png

ipfs://QmW98pJrc6FZ6/cat.png

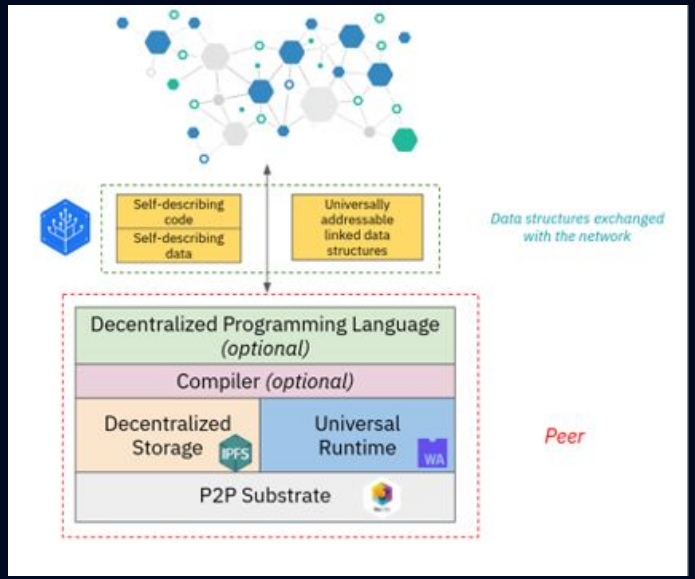
IPFS-FAN: 函数(计算)寻址网络

IPFS-FAN: A Function-Addressable Computation Network

用CID标识的WASM函数

用CID标识的数据

生成CID表示数据
依然存储在IPFS
网络中



IPFS-FAN架构

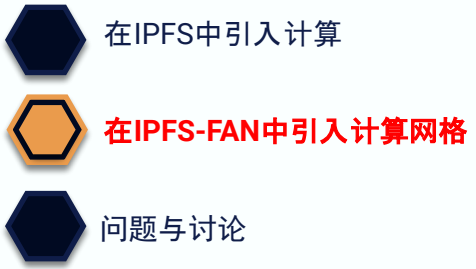
综上，我们可以得到一个结论：

在IPFS中引入计算

CID，之前主要*应用于文件内容的DAG表示*——内容寻址。现在CID也可以表示一个函数以及函数的代码。函数在CID标识的内容上运行CID标识的代码并执行计算。结合计算和存储，IPFS就具备了操作系统的能力。在这样的系统中，内容和函数都是唯一可识别的，并可以被全局处理。

在IPFS-FAN中引入计算网格

1 当前问题: 目前Web3领域的所有计算类型都无法提供通用计算
2 解决方案: 在IPFS之上构建计算网格
3 TimeRose 计算网格的架构
4 TimeRose 计算网格的功能
5 TimeRose 计算网格的用例



当前问题:

目前Web3领域的所有计算类型都无法提供通用计算

- 计算类型1:虚拟机。EVM、FVM和IPFS-FAN 都没有调度能力。来自 ethereumbook:“The EVM, therefore, has no scheduling capability, because execution ordering is organized externally to it—Ethereum clients run through verified block transactions to determine which smart contracts need executing and in which order. In this sense, the Ethereum world computer is single-threaded, like JavaScript.”

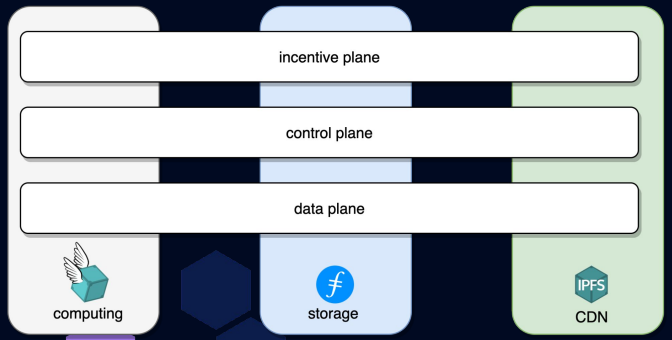
- 计算类型2:去中心化计算市场，例如Golem 和 iExec，并不是通用计算，而只是计算offloading 的一种方式。

- 计算类型3:通用去中心化计算网络，Fluence 网络和Dfinity互联网计算机，两者都使用了和PFS-FAN类似的概念:编程语言、WASM、部署、编排等。这两个网络但缺乏自描述的策略来标识代码，缺乏全局访问数据结构能力。

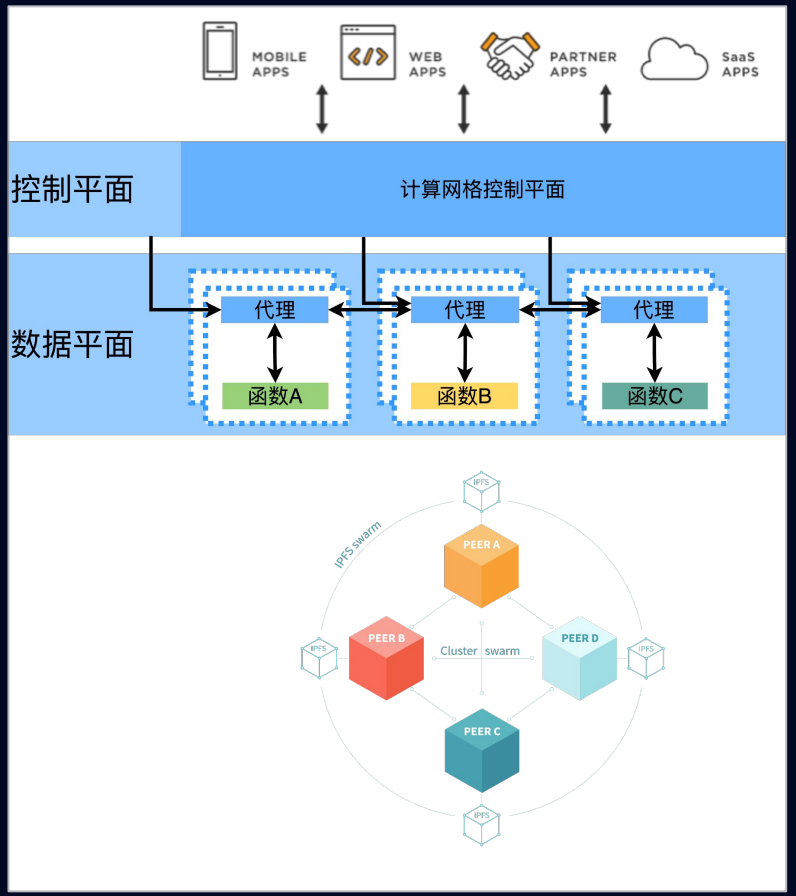
解决方案:

引入基于IPFS-FAN的计算网格:TimeRose

- 控制平面:TimeRose计算网格
- 数据平面:IPFS-FAN
- TimeRose计算网格的架构



KEN Cloud（Cloud 2.0 over IPFS）总体架构

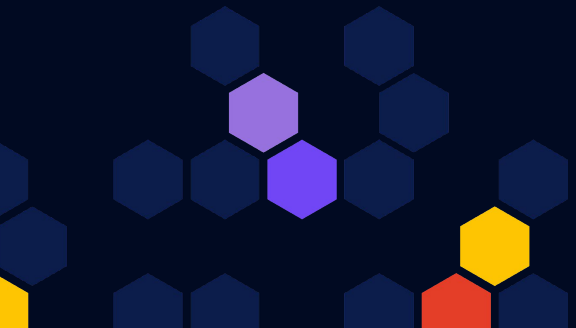


TimeRose计算网格总体架构

TimeRose计算网格的控制平面:

单独的控制平面，运维开销大

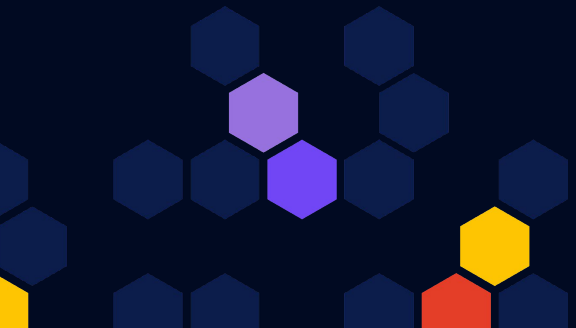
连接	安全	控制	观察
控制平面数据流	传输加密	配置	可视化
部署策略	认证	策略	监控
熔断	授权	编排调度	日志
虚拟服务	安全命名		分布式跟踪



TimeRose计算网格的用例:

构建 Web 3 分布式执行环境

- 全球无服务器基础设施。通过CID来部署、调用、运行代码的能力，使得应用开发者不需要服务器基础设施。
- 负载均衡和就近服务。根据DHT分布计算和存储，迁移计算靠近存储。
- 提升后端服务的高可用性。可以分别设置计算和存储的副本数，依此来控制服务的高可用性。
- 协作计算和计算外包。联邦机器学习。新型云计算企业。



问题与讨论

在IPFS中引入计算

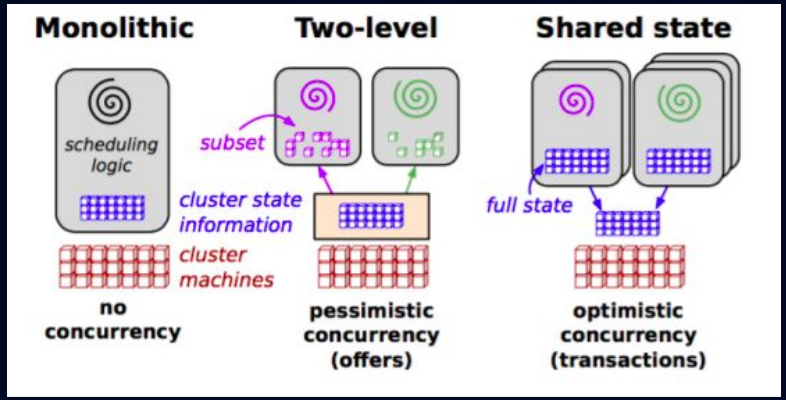
在IPFS-FAN中引入计算网格

问题与讨论

问题与讨论：

构建 Web 3 分布式执行环境

- Web2 有哪些成功的分布式执行框架？
  - Borg
  - Mesos
  - YARN
  - Omega
  - Kubernetes
  - Istio
- 计算执行环境未来会有怎样的发展？



Solomon Hykes
@solomonstre

If WASM+WASI existed in 2008, we wouldn't have needed to created Docker. That's how important it is. Webassembly on the server is the future of computing. A standardized system interface was the missing link. Let's hope WASI is up to the task!

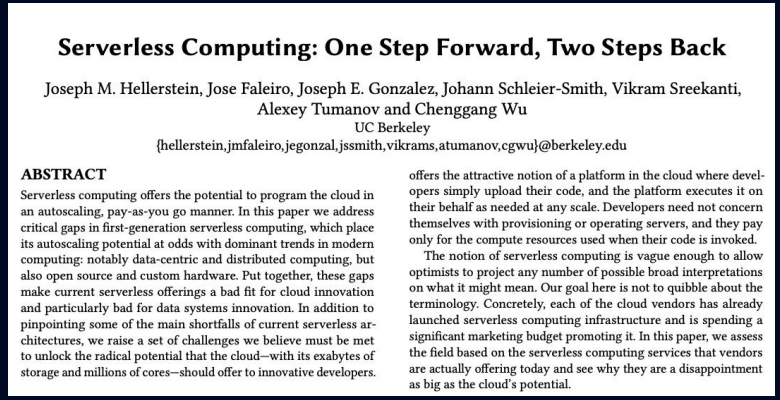
Lin Clark @linclark · Mar 27, 2019
WebAssembly running outside the web has a huge future. And that future gets one giant leap closer today with...

📢 Announcing WASI: A system interface for running WebAssembly outside the web (and inside it too)

hacks.mozilla.org/2019/03/standa...
Show this thread

4:39 AM · Mar 28, 2019 · Twitter Web Client

792 Retweets **139** Quote Tweets **2,029** Likes



Serverless Computing: One Step Forward, Two Steps Back

Joseph M. Hellerstein, Jose Faleiro, Joseph E. Gonzalez, Johann Schleier-Smith, Vikram Sreekanti, Alexey Tumanov and Chenggang Wu
UC Berkeley
{hellerstein,jmfaleiro,jegonzal,jssmith,vikrams,atumanov,cgwu}@berkeley.edu

ABSTRACT

Serverless computing offers the potential to program the cloud in an autoscaling, pay-as-you go manner. In this paper we address critical gaps in first-generation serverless computing, which place its autoscaling potential at odds with dominant trends in modern computing: notably data-centric and distributed computing, but also open source and custom hardware. Put together, these gaps make current serverless offerings a bad fit for cloud innovation and particularly bad for data systems innovation. In addition to pinpointing some of the main shortfalls of current serverless architectures, we raise a set of challenges we believe must be met to unlock the radical potential that the cloud—with its exabytes of storage and millions of cores—should offer to innovative developers.

offers the attractive notion of a platform in the cloud where developers simply upload their code, and the platform executes it on their behalf as needed at any scale. Developers need not concern themselves with provisioning or operating servers, and they pay only for the compute resources used when their code is invoked.

The notion of serverless computing is vague enough to allow optimists to project any number of possible broad interpretations on what it might mean. Our goal here is not to quibble about the terminology. Concretely, each of the cloud vendors has already launched serverless computing infrastructure and is spending a significant marketing budget promoting it. In this paper, we assess the field based on the serverless computing services that vendors are actually offering today and see why they are a disappointment as big as the cloud's potential.

凯云实验室
KEN Labs

构建面向Web3的下一代云平台

如有任何问题或者意见，请电子邮件联系我们

info@kencloud.com

https://github.com/kenlabs