
BonusCloud

区块链的未来基础设施

版本 1.0

2018

翻译：教主

校对：Eimo

目录

BONUSCLOUD	1
摘要	3
一、 背景介绍	5
1. 云计算平台	5
2. 中心化云计算平台	5
3. 去中心化的云计算平台	6
4. 什么是 BONUSCLOUD.....	7
二、 技术架构	8
1. 下一代区块链 (BONUSCHAIN)	8
2. 开放硬件平台 (BONUSNODE)	9
3. 分布式网络 (BONUSNET)	10
4. 分布式计算平台 (BONUSCOMPUTING)	12
<i>BonusContainer</i>	14
<i>BonusAppEngine</i>	14
<i>BonusServerless</i>	15
5. 分布式存储 (BONUS STORAGE)	16
<i>BonusCloud 对象存储 (BOS)</i>	17
6. 生态经济模型	18
<i>工作者</i>	20
<i>记账者</i>	20
<i>应用开发者</i>	21
<i>应用使用者</i>	21
<i>资源交易</i>	21
<i>开发者激励协议</i>	22
<i>共识算法</i>	22
<i>节点贡献度</i>	24
三、 应用场景	25
1. 性能监控和压力测试.....	25
2. P2P 计算和云存储.....	25
3. VPN 边缘接入.....	25
4. IoT 数据聚合分析.....	25
5. 公有链承载.....	26

摘要

社会的发展伴随着人类文明的发展，人类文明的发展越来越向信任、协作的方向演进。计算机技术的快速发展，已经在很多领域将规则数据化与数字化，能够让规则更加清晰，也能够使规则快速复制，取得社会效应的提升。

区块链技术的发展，其生态效应与数据可信性让信任与协作上升到一个新的标准，也符合人类社会对于信任与协作的期望。

BonusCloud 致力于打造一个去中心化的、可信任的、结合区块链与云计算技术的、全球化的基础设施平台，并构建一个基于此的开放式共享生态。旨在连接和利用各种具备计算能力的资源，转换为数字资产价值，在数字经济时代，让 DApp 赋能区块链应用创业者。

BonusCloud 通过构造一个全球化的算力交易平台，能够将全球包括现有云计算、企业数据中心以及个人 PC 及终端设备的 CPU/GPU/带宽等资源有效连接起来，集平台上千千万万个资源贡献者、平台使用者、应用开发者之力，提供超低价格、超大范围、超强算力的可信任的云计算平台服务，彻底改变传统的中心化云计算商业模式及资源分布结构。BonusCloud 的算力资源平台具备分布式、低延时、智能化的特点，在网络访问、数据存储、数字货币挖矿、3D 渲染、直播转码、AI 学习、IoT 防护、AR 及 VR 等行业应用前景广泛。BonusCloud 平台更靠近设备端，使得平台在满足近场计算类需求方面有无与伦比的优势，不仅可以为数据采集做出贡献，平台在实时、短周期的数据分析及智能化处理方面的显著优势，可以更好的支撑本地业务及时处理执行，使得计算服务更加高效且安全。

在这个全新的分布式计算平台上，可以建设 DApp 应用商店，从而构筑一

个基于分布式计算和边缘计算的应用生态圈，创造巨大的社会和经济价值。

目前 BonusCloud 已经完成区块链基础服务的 50%的代码工作，并完成网络测试的小型原型系统验证，后续将逐步进行相关功能的上线使用。同时，BonusCloud 开始构建基于容器与区块链结合的计算资源框架及资源调度原型。

一、 背景介绍

1. 云计算平台

基于美国国家标准与技术研究院（NIST）的定义，云计算是一种按资源使用量付费的模式，这种模式提供可用的、便捷的、按需的网络访问，进入可配置的计算资源共享池，资源包括网络、服务器、存储、应用软件与服务，这些资源能够快速被提供，只需投入很少的管理工作，或服务供应商进行很少的交互。

通俗来讲，云计算提供了 IT 基础资源与能力，并将资源与能力与设备进行解耦，达到快速交付与使用的体验。同时，云计算将传统 IT 的 Capex 模式变为 Opex 模式，减少企业的现金流占用。尤其对于不可预期的资源需求提供了非常友好的供给模式。

2. 中心化云计算平台

由于云计算产业的初始投资规模巨大，使得云计算变成巨头玩家的游戏，因此导致了云计算平台的诸多问题。

缺少行业标准

云计算行业的供应商大多是巨头玩家，从商业价值角度考量，每个玩家都希望能够成为行业的标准，因此在给用户提供的应用标准上，每个云计算供应商都有自己独立的一套标准体系。商业产品的竞争无形中给用户带来了使用成本，当用户为了成本与稳定的需要使用多家云计算平台时，需要适配多家供应商。而为了满足巨头的需要，云计算产业中的上游供应链公司不得不向每家巨头提供定制化的产品，以满足客户的需求。

运行环境黑盒化

中心化的云计算平台作为一个黑盒环境提供给用户，用户无法控制自身资源与应用的可信运行状态。为了实现云计算企业的管控或其他诉求，云计算平台均会以侵入式的方式获取用户的相关资源。而在这种环境下，也出现过多起云计算平台服务商由于自身业务 BUG 问题，导致用户数据丢失、应用运行异常的情况（如某公有云安全产品删除用户文件事件）。

成本过高

由于云计算行业初始投资规模巨大，并且是多寡头垄断，云计算厂商对于营利的诉求会比较高。从 Amazon 的财报上来看，AWS 的 EBITDA（毛利润）达到近 45%，其运营利润（净利润）在 25% 左右。而从服务模型来讲，云计算厂商对于稳定的资源使用与增长能够较好的满足，而对于临时的大规模资源使用是不予以服务的，主要原因是由于资源使用量的不确定性会导致大量的投入不能及时得到回报，因此这方面的计算需求现有的云计算平台无法满足，这也是商业与技术概念的冲突所在。同时这方面的需求被压抑，是一个较大的蓝海市场。

3. 去中心化的云计算平台

区块链技术的产生与发展给予众人一个新的可信任、可协作的技术平台与生态模式。同时区块链与云计算技术的结合能够解决中心化云计算平台所存在的问题。并且去中心化的云计算平台也是未来区块链应用 Dapp 的理想的应用承载平台。

4. 什么是 BonusCloud

随着个人家庭网络从 ADSL, 宽带, 光纤的不断普及, 以及移动端从 2G 到 4G、5G 的发展, 网络的接入能力提高, 模糊了用户终端与 IDC 内服务器的互联边界。而用户终端通用芯片计算能力的提高, 专有芯片的普及, 用户终端与服务器算力的差异也远非天差地别。可以预见当网络接入能力和终端算力进一步提升, 同在一个地区的用户终端与 IDC 服务器将不会再有明显的区别, 而当基础设施普及超过 20%之后, 产业将迎来全面的发展。

BonusCloud 致力于打造一个去中心化的、可信任的、结合区块链与云计算技术的、全球化的基础设施平台, 并构建一个基于此的开放式共享生态。旨在连接和利用各种具备计算能力的资源, 转换为数字资产价值, 在数字经济时代, 让 DApp 赋能区块链应用创业者。

BonusCloud 通过构造一个全球化的算力交易平台, 能够将全球包括现有云计算、企业数据中心以及个人 PC 及终端设备的 CPU/GPU/带宽等资源有效连接起来, 为网络访问、数据存储、数字货币挖矿、3D 渲染、直播转码、AI 学习、IoT 防护等行业, 提供超低价格、超大范围、超强算力的通用计算服务。同时通过生态回馈所有有贡献的参与者, 进一步推动基础设施的建设和升级。

二. 技术架构

1. 下一代区块链 (BonusChain)

传统的云计算服务中，资源交易的匹配，任务的分发和执行，均是围绕云计算服务提供商，以中心化方式提供的。虽然在云计算技术和服务创立之初，分布式是主要特色，但是以数据中心为维度，相对于小型机和超级计算机的分布式，在如今遍及全球的互联网面前，已经过于集中。并且伴随这种集中式资源服务提供方式的，是信息的不透明和不对称。使用者很难真正了解到资源和任务在云计算系统中的分配和使用方式，除了带来各种安全隐患，诸如超售等现象也是普遍存在。

而区块链技术的产生，尤其是以太坊代表的区块链 2.0，通过以太坊虚拟机 (Ethereum Virtual Machine, EVM) 运行的智能合约机制，将区块链的应用场景大大增加。其所代表的信息透明公开不可篡改的特点，恰恰可以解决传统云计算所面临的问题。

BonusCloud 希望能在此基础上更进一步。一方面将云计算领域的虚拟化技术与区块链结合起来，在保证安全性的基础上，增强区块链虚拟机功能，以支持更丰富的 PaaS 应用。不仅可以执行智能合约，还可以执行网络应用，存储应用，计算应用。在此基础上，FaaS 和 Edge Computing 都可以基于 BonusChain 上的虚拟化技术发展起来。另一方面，将所有的交易信息，资源信息，均放在区块链中，用户可以公开查询和审计。由于不同信息产生的时效性，数据量均有所不同，因此 BonusChain 采用主链和侧链结合的方式。主链进行交易的撮合，记录交易信息等核心内容，一条或者多条侧链记录资源信息，日志信息等数据量大，

时效性低的内容。侧链可以根据需要采用与主链不同的技术，如 DAG 等

2. 开放硬件平台（BonusNode）

BonusCloud 作为去中心化的云计算基础设施平台，采用软硬件解耦方案，以具备快速迭代、开放共享的能力。硬件方面使用开源、开放的硬件平台方案，对终端硬件进行适配与验证，以最大化的使用共享资源，打造开放平台。

BonusCloud 的硬件适配主要分为三种类型：瘦终端（Lite）、普通终端（Normal）和超级终端（Super）。不同的硬件方案适用于不同的任务应用场景，以给予客户不同的平台能力。

Lite：瘦终端，以单片机，家用路由器为代表的瘦终端硬件作为轻型网络相关任务的承载，主要应用于网络接入与测量、数据获取、移动 APP 测试等。任务以批量任务为主，强调终端的分布能力与接入能力，对于单节点稳定性要求较低。任务形态为短周期轻型任务。

Normal：普通终端，以 PC+GPU 为代表的普通终端作为计算任务的承载，主要应用于 HPC、DeepLearning、Rendering 的计算。任务以批量任务为主，强调终端的计算能力与任务分解与调度能力，对于单节点稳定性要求一般。任务形态为短周期重计算任务。

Super：超级终端，以 Server 为代表的超级终端作为网络枢纽、快速存储接入、规模数据处理与计算、企业应用的承载，适应场景较为广泛。任务以实时性需求的企业服务为主，强调终端的稳定性及快速处理能力，对于单节点稳定性要求较高，一般部署在 IDC 内部。任务形态为长周期、实时性的企业服务。

用户可根据需求来自主选择硬件平台方案进行服务的开发与承载，以灵活

配置资源，将资源使用合理化。

3. 分布式网络（BonusNet）

BonusCloud 致力于连接和利用各种符合能力要求的资源，因此构建一张能够连接一切资源的安全可靠的网络，是所有服务的基础。

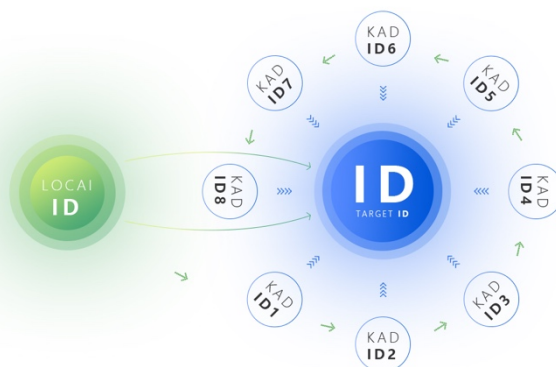
BonusNet 是基于 Internet 之上的新形态网络空间。在其网络空间中，将设计独立的相关基础组件，如 DNS、Router。BonusNet 使用 L2+L3 over Internet 的模式，在 L2 层采用类 Kademlia 协议的 DHT 架构，L3 层使用加密隧道协议，并且辅助使用新的命名空间与寻址模式以代替 Internet 中的 DNS 协议，同时对于 DNS 协议会兼容支持。标准的网络空间协议，无论是公有协议（如 TCP/IP）还是用户的私有网络协议均可无缝从 Internet 网络空间迁移至 BonusNet 网络空间，具备足够的兼容性。

Kademlia 是一种通过分散式杂凑表实现的协议算法，为非集中式 P2P 计算机网络而设计的，参与通讯的所有节点形成一张完全无中心化的虚拟网（或者叫做覆盖网）。这些节点通过一组数字（或称为节点 ID）来进行身份标识。节点 ID 不仅可以用来做身份标识，还可以用来进行值定位（值通常是数据的散列或者关键词）。

Kademlia 使用异或来定义距离，两个节点 ID 的异或（或者节点 ID 和关键字的异或）的结果就是两者之间的距离，因此在实际的节点距离的度量过程中计算量将大大降低。一个基本的具有 2^n 个节点的 Kademlia 网络在最坏的情况下只需花 n 步就可找到被搜索的节点或值。

同时，尽可能重用已有节点信息、并且按时间排序是 Kademlia 网络节点更

新方式的主要特点。从启发性的角度而言，这种方式具有一定的依据：在线时间长一点的节点更值得我们信任，因为它已经在线了若干小时，因此，它在下一个小时以内保持在线的可能性将比我们最新访问的节点更大。



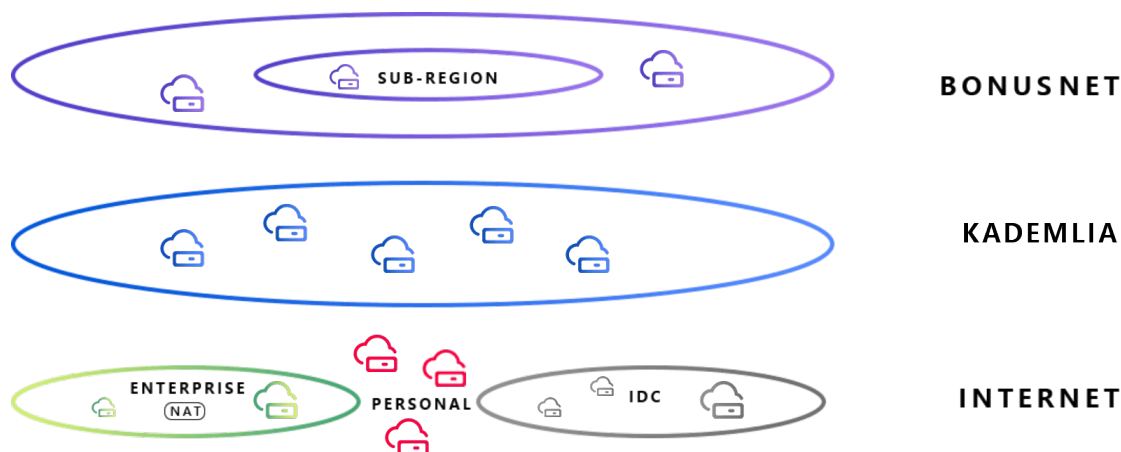
非集中式网络结构还有更大的优势，是它能够显著增强抵御拒绝服务攻击的能力。即使网络中的一整批节点遭受泛洪攻击，也不会对网络的可用性造成很大的影响，通过绕过这些漏洞（被攻击的节点）来重新编织一张网络，网络的可用性就可以得到恢复。

在 Kademia 算法中，节点选择“远”和“近”与物理距离无关，只与异或计算结果有关，因此很有可能选择的节点会在不同的地区甚至国家。BonusNet 中引入物理距离概念，在每次发现邻居节点的循环中同时在物理距离上收敛。还可以根据用户基础网络的特性，如区域，运营商等，对 BonusNet 进行一定的划分和聚合，提供更高效和健壮的网络访问能力。在交易撮合的过程中，应用使用者可以对工作者在网络中的特性提出选择性要求，而不同特性的资源也可以有不同的价格。

在传统的 Kademia 网络中，节点之间的连接是私有的网络协议，而应用

需要进行改造传输协议才能够适配和使用 Kademlia 网络。为了给予应用更高的兼容能力，同时为了解决地址管理、数据传输安全等方面的问题，BonusNet 在 Kademlia 网络基础上构建了一张专有加密网络。该网络采用 IPv6 地址，以保证足以容纳所有的节点。所有节点通过加密隧道连接，保证传输数据的安全性。

这样我们可以将 BonusCloud 中的节点全部连接起来，建立起一套无缝的平行于现有基础网络的网络系统。即使节点位于 NAT 或者防火墙之后，也可以通过 NAT 穿透技术，打破用户基础网络的区域限制。



BonusNet 作为 BonusCloud 的基础网络平台，将是云平台自身与平台应用的承载基石。

4. 分布式计算平台（BonusComputing）

BonusCloud 以开放硬件平台（BonusNode）为基础，通过对中心化云平台软件的去中心化演进与实施，为社区提供三种不同的计算环境来满足不同开发者的需求，BonusContainer、BonusAppEngine 和 BonusServerless。用户可以自己构建 DApp 或者购买由应用开发者提供的 DApp, BonusCloud 负责将 DApp 运行环境封装成 Docker 文件，并根据需求方的指令，安全、可靠的分发到租赁的

资源节点上，启动并运行。

BonusChain 虚拟机 (BonusVM) 整体负责计算节点中任务生命周期的管理和分配，在 BonusNetwork 中各个节点建立安全的连接。

虽然容器已彻底改变了开发、打包和部署应用程序的方式。但是安全性是在公有资源调度平台中使用容器所面临的主要问题之一。传统 Linux 容器中运行的应用程序访问系统资源的方式与常规（非容器化）的应用程序一模一样：直接对主机内核进行系统调用。内核在特权模式下运行，因而得以与必要的硬件交互，并将结果返回给应用程序。内核通过使用 Linux 控制组 (cgroup) 和命名空间对应用程序所能访问的资源施加一些限制，然而并非所有的资源都可以通过这种机制来加以控制。此外，即使有这样的限制，内核仍然暴露了很大的攻击面，恶意应用程序可以直接攻击。

提高容器隔离效果的一种方法是让每个容器在其自己的虚拟机里面运行，与主机完全隔离。即使访客系统 (guest) 存在漏洞，虚拟机管理程序仍会隔离主机以及主机上运行的其他应用程序/容器。BonusCloud 的容器技术使用具备虚拟化隔离技术的容器实现，运行时可以让容器具有虚拟机级别的隔离和安全特性，同时保留容器级别的灵活性和性能。

镜像仓库用来存储和分发用户上传的 Docker 镜像，镜像仓库本质上并不是一个中心化的仓库，只提供上传和下载接口。在 BonusCloud 这种分布式的计算环境中，中心化的镜像存储与分发是无法满足系统需求的。因此镜像仓库以 BonusCloud 所提供的分布式存储服务为依托，同时结合了智能压缩、智能流控、P2P 分发等技术，以实现镜像的低延时加速分发。

BonusContainer

BonusContainer 提供容器资源服务，主要提供用户对资源使用的需求。用户可以申请指定 CPU、内存、存储规格的实例，平台通过调度选择最适合的计算节点运行容器。使用 BonusContainer，用户首先将 Docker 镜像上传至 BonusRegistry，创建容器时选择 BonusRegistry 中的镜像名称和标签。创建时用户可以选择一定的调度策略，例如地域位置、亲和性、资源需求等，BonusController 会根据用户的调度需求选择最合适的节点创建资源。

BonusContainer 优势

安全	运行在专用内核中，提供 CPU、内存、网络、IO 的隔离，并可以利用虚拟化 VT 扩展实现硬件强隔离。
兼容	支持OCI容器格式，Kubernetes CRI界面等行业标准，以及传统的虚拟化技术。
简单	不需要在完整虚拟机中嵌套运行容器，使用方式和 Docker 类似。
性能	提供与标准 Linux 容器一致的性能。

BonusAppEngine

BonusContainer 适合有一定容器使用背景的用户，要求按照一定格式创建容器镜像等，决定资源的使用方式，最灵活但是有一定技术门槛。为了让应用开发者更方便的使用平台，BonusCloud 也提供 BonusAppEngine 功能，BonusAppEngine 是一个 DApp 应用程序运行时沙箱，用户可使用 BonusCloud 提供的不同语言的 SDK，编写好代码后上传至平台即可运行。

BonusAppEngine 具有以下特点：

-
- (1) 支持主流的编程语言：可以使用 Java、Node.js、Ruby、Go、Python 或 PHP 编写您的应用。
 - (2) 使用方便：让开发者只关注于代码的实现，不需要关注基础设施架构的问题。BonusAppEngine 会全面管理应用的流量变化、负载均衡、运行状况监控及故障恢复，以及基础设施的自动更新。
 - (3) 版本控制和 A/B 测试：可以做到灰度发布不同的版本以完成 A/B 测试。

BonusServerless

BonusServerless 相比 BonusAppEngine 更进一步, BonusServerless 提供函数级程序运行时环境, 相比 BonusAppEngine, BonusServerless 提供业务逻辑的支持, 用户可以通过配置 BonusApiGateway 将网络请求转发到自定义函数。BonusServerless 将只按照函数的运行时间收取资源费用, 函数不运行时不收取费用。

BonusServerless 优势：

无需管理服务器，灵活扩展和高度的自动化扩容及故障恢复。

更细粒度的计费, 更优的成本控制, 相比 BonusAppEngine 不需要为 AppEngine 的空闲时间付费。

提供第三方工具和开源项目的生态系统来简化开发到生产过程的构建、测试和部署。

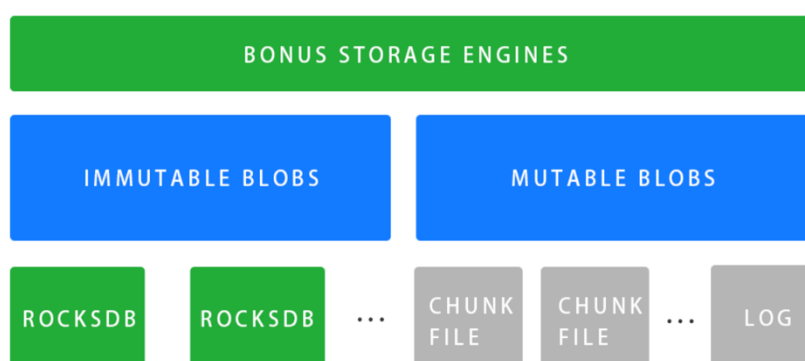
5. 分布式存储（Bonus Storage）

为了方便应用开发者和社区用户，BonusCloud 提供基于分布式节点的对象存储 BOS（Bonus Object Storage），并且在后续计划推出基于分布式节点的块存储 BBS（Bonus Block Storage）

分布式存储系统中有两种主流数据的数据分布方式，一种是哈希分布，如 Kademlia DHT, amazon dynamo 等，一种是由中心节点分配进行分布，如 google GFS, hadoop hdfs. 在实现中，对象存储和 KV 存储一般采用哈希分布比，块存储和分布式数据库采用往往采用中心节点分配。Bonus Storage 同时支持两种数据分布策略，可以在资源申请时进行协商。

为了简化提供不同类型的存储资源，BonusCloud 实现一个基于本地文件存储的存储引擎 Bonus Storage Engines，为 BOS 和 BBS 服务提供底层的存储。

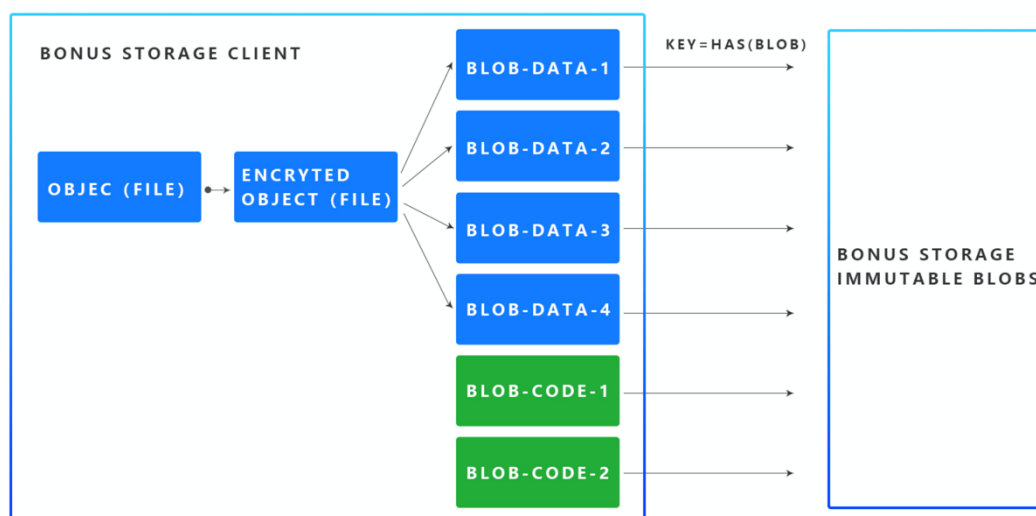
Bonus Storage Engines 提供对两类 blob 的支持，及 immutable blob 为 BOS 提供存储支持，mutable blobs 为 BBS 提供存储支持。



对象存储是一次写多次读，所以底层采用 Immutables blob，具体实现可

以采用在本地磁盘上维护一组 rocksdb, key 为 blob 的 hash 值。

块存储要求支持随机读写, 所以底层采用 mutables blob, 具体实现通过维护一组 chunkfile 和 log 来实现, 每个 chunkfile 对应一个 volume 的某个 chunk 内容, 每个 chunkFile 要维护当前的 version, 每次写请求完成后要更新, Log 用于记录写请求的日志, 偏于数据的校验和恢复。



BonusCloud 对象存储 (BOS)

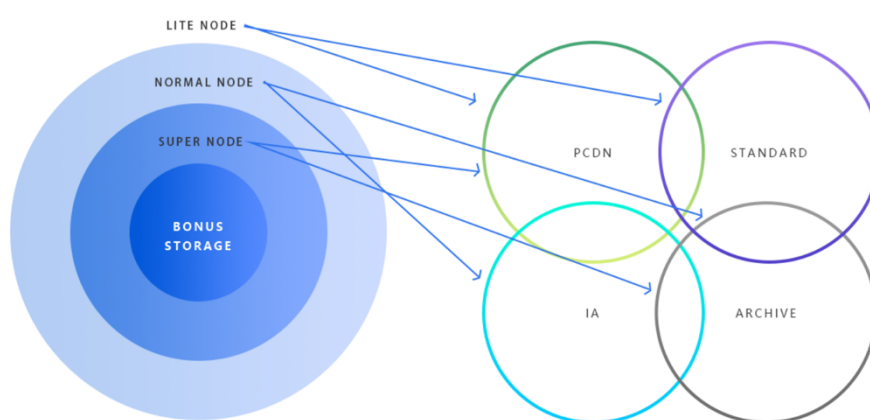
BOS 存储模型

BOS 提供提供去中心化的对象存储服务, 并且对文件(Object)进行加密存储。在 BonusStorage 的客户端上, 如果要存储一个对象(文件),在客户端会首先进行加密, 然后对文件进行拆分, 拆分为多个不可修改的 blob-data,然后根据这些 blob-data 计算纠删码的校验块 blob-code,然后把数据 blob 和校验 blob, 以 blob 的 hash 值作为 key 存储在 Bonus 存储网络中。在 Bonus 存储网络中会对这组 blob 进行多副本和纠删码的组合存储, 具体策略在创建时协商。

BOS 数据分布

BOS 的业务使用场景覆盖 PCDN 分发，标准类型，低频类型，归档类型，结合了传统 CDN 和云对象存储的能力，并且支持灵活转换。

BonusCloud 支持三种硬件方案，各个工作者的存储、带宽能力不同，这里根据 BOS 的业务场景，构成多个业务类型的资源网络，每个工作者在加入 BonusCloud 时，根据自身的存储、带宽能力，考虑加入不同类型的资源网络。



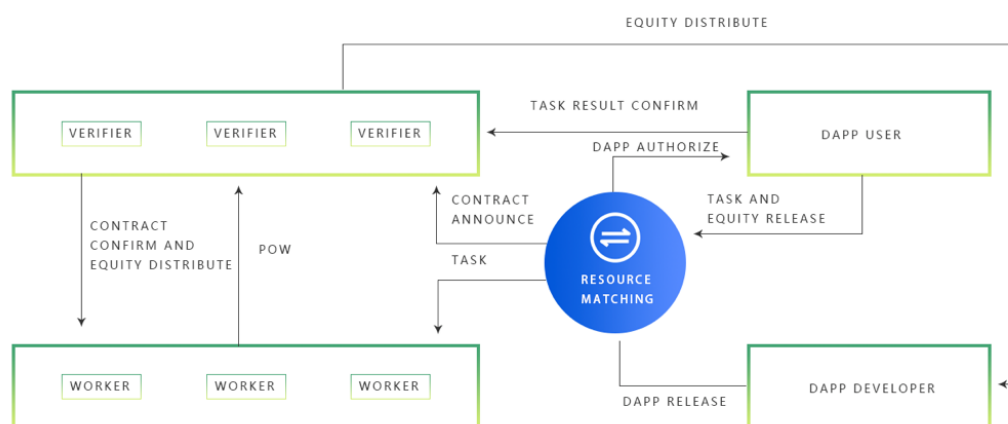
Bonus Storage Client 在上传 Object 时，可以指定存储类型，并且之后可以更改存储类型，在 Object 存储类型转换的过程中。

6. 生态经济模型

BonusCloud 是分布式的通用计算平台，提供通用计算的交易与分发，通过开源的云平台方案，给予客户一个公开可信任的环境，进行有价值的计算能力交换。

BonusCloud 将构造一个双边（多边）市场的生态，主要由以下几种类型的参与者构成：工作者、记账者、应用开发者、应用使用者以及资源交易平台。

在双边市场会促使出现“双边网络效应”，即跨边网络效应(cross-side network effect)：市场一边的用户在市场获取的价值取决于另一方的用户数量，以 Uber 的商业模式为例子，乘客的用户价值取决于司机的多少（司机越多，乘客等待时长越短，体验越好），司机的价值取决于乘客的多少（乘客越多，司机空驶时长越短，收入越高），两者相互促进，放大网络效应。因此，BonusCloud 在未来打造的生态中，第一步将关注加大供给量，例如，提高工作者作为资源供给方的数量及在线时长，激励记账者及应用开发者提供更完善、透明、多样的产品及服务，其次，通过对全球市场的宣传和渗透，大力拉动使用者需求，使供给及需求互相促进、双向驱动，在竞争中形成强规模效应，构建竞争壁垒。BonusCloud 整个生态作为一个经济体系，创始团队的目标是增加整个经济体系中的 GDP 收益。



BonusCloud 的生态中，主要有以下几种角色：

工作者

工作者是 BonusCloud 的基础资源贡献者，拥有特定的算力、存储、网络等资源。任务在交易撮合平台发布后，现有空闲资源可以满足应用使用者的条件的工作者会竞争领取任务。

工作者通过执行 DApp 完成任务获取收益，新加入工作者与老工作者会有不同的收益算法。异常中断节点将会扣除收益，并且受到一段时间内降低获得收益或者领取任务概率的惩罚。

工作者根据提供资源的级别不同（瘦终端，普通终端，超级终端），可以领取不同类型的任务，获得不同的收益。

工作者除拥有获得应用使用者可付出的收益外，还可同时担任记账者角色，从而获得相应的记账收益。

记账者

记账者负责为应用开发者确权，负责对每笔交易记账。记账者将区块内容及 Hash 值进行广播、其它记账者对区块进行确认无误后，该区块将入链。

工作者领取任务后，记账者负责将任务领取信息入链登记，并对应用使用者公开服务 IP 及端口。任务完成后工作者向记账者提供工作证明，证明自己完成的工作量。记账者通知应用使用者查看工作者提供的工作证明记录，并对应用完成情况进行确认。

工作者完成任务后，记账者根据工作者提供的计算资源的服务贡献量和服务质量，依据智能合约，自动为工作者的工作收益进行记账，对应用使用者的付出进行记账，向应用开发者派发授权对应的收益分配并记账。执行本次记账操作

的记账者也将获得收益奖励。

应用开发者

应用开发者是 BonusCloud 的重要参与方，在 BonusCloud 提供的资源和服务之上，开发各种功能的 DApp，并将 DApp 封装成 Docker 镜像，完成分发前的所有准备工作。DApp 使用收益由系统算法决定，在 DApp 被使用时写进智能合约，应用使用者获得使用授权后，从应用使用者账户向应用开发者账户转账。DApp 的质量和使用情况，代表着 DApp 对社区的贡献，会影响 DApp 的使用收益计算结果。

应用使用者

应用使用者获取应用开发者的使用授权后，可根据实际需求发布资源及计算任务，包括任务的资源配置需求及价格。应用使用者提交包含任务需求的容器镜像给资源平台，并提交相应数量的权益。

资源交易

BonusCloud 负责撮合应用使用者与工作者之间的交易。工作者/应用使用者将要出租/租赁的资源以及相应的收取/支付的 BNC 报价公布之后，BonusCloud 帮助供需双方快速完成匹配。一旦双方认可交易，智能合约将会自动触发，从而确保交易的公平公正性。根据应用类型不同，报价的因素包括但不限于 CPU 使用量、占用的存储空间、内存使用空间、单位时间消耗的流量等。在交易撮合中还会引入用户因素和随机因素，以保证一定的随机性和分布性，避免出现任务分配过于集中的情况。

在每次交易撮合中，BonusCloud 会收取一定的技术服务费用，为该笔交易金额的 1%，用作基金会未来发展。

开发者激励协议

BonusCloud 提供面向 DApp 开发者的开发者激励协议(Developer Impel Protocol, DIP)。每隔一定的区块数量，选举最近周期内达到一定贡献（使用次数，使用者数量等）的 DApp，对其开发者进行奖励，新上线 DApp 和已有 DApp 会有不同的奖励策略，以达到不同的激励效果。借助 DIP 的激励机制，更多的开发者将被持续鼓励创造更高价值的 DApp，从而构建面向开发者社区的正向反馈。

共识算法

BonusCloud 采用 DPoS+PBFT 的混合共识算法。记账者定期由工作者根据贡献值选举产生，贡献值根据一系列的因素（提供的计算能力，网络带宽能力，传输数据量，数据存储容量，数据存储时间，节点健康程度等等）计算得出，并且当成为记账者后，一些因素的值会被清零，以鼓励在下一个周期中继续为应用使用者提供服务。记账者会上交一定数量的权益作为保证金，不再担任记账者后退回。如果被证实作恶，其保证金会被扣除回馈社区。当记账者选举完成后，对当前的记账者列表进行随机的排序（保证每一轮的记账者顺序不同，也无法预测下一轮记账者顺序），然后依次让每个记账者创建区块。如果记账者获得了本次记账权却不在线，本次记账权将按顺序移交到下一个记账者，不扣除保证金也不获得记账权益，但是会降低下次选举时的相关因素值。

基于以上的分析，系统共识协议工作流程如下：

1) 验证人注册

验证人注册时，需要提供押金 V_b 。押金必须大于系统设定的一个最小押金参数 B_Min 。在任何时候，注册人可以退出注册，但是押金将有一个解冻期，一般可以设置成一个月。验证人参与竞选的贡献值为 V_c 。

2) 权益人投票选举验证人

权益人可以对验证人进行投票，假设投票人生成的权重是 St ，验证人的排序权重 $W_c = St * V_c$ 。如果验证没有获得任何投票，其权重 $W_c = 0$ 。

3) 合格记账者竞选

注册的验证节点维持一个表，按排序权重排列前 n 个记账者，该表中的记账者称为合格记账者，可以参与竞争出块。当选合格记账者的贡献值 V_c 会进行一定的扣除。

5) 合格记账者打包交易

用父区块头哈希值 H_p ，当前交易 Merkle 根 H_m 以及时间戳 T ，一个随机数 $nonce$ 生成候选区块头哈希 H_n ，然后广播一个 Propose 交易。每个合格记账者每轮只能发一个 Propose 交易，在同一轮中重复发 Propose 交易者会被认为作恶节点。除非在一定的时间内（系统的共识超时参数），系统没有达成共识，合格记账者才可以变更 $nonce$ 以重新生成一个新区块的哈希值，并发起 Propose 交易。

6) 交易信息确认

各合格记账者在接收到 Propose 交易后，其他记账者首先对发起 Propose 交易的节点进行验证，以及对该区块内的交易进行验证。通过验证后，记账者计算一个投票权重值 V_w ，对当前最大的 V_w 对应的提议交易进行投票，并发 VoteBlock

消息。原提议者在收到超过 2/3 记账者的 VoteBlock 消息后, 发起 CommitRequest 消息, 各验证节点将验证该 CommitRequest 消息对应的 VoteBlock 消息是否超过 2/3, 如果超过 2/3, 则对该 CommitRequest 消息进行 Commit 并广播。各合格验证节点在收到超过 2/3 个 Commit 交易后, 广播 NewHeight 交易, 各节点在收到 NewHeight 交易后会确认该交易并更新各自的本地区块链。NewHeight 交易也使得新区块得到最终的确认, 而不可在该区块前分叉。

节点贡献度

用字母 P 来代表设备贡献度, P 包括设备在线贡献度 P_O 及设备任务贡献度 P_T , 其中 P_T 是任务在网贡献度 P_W 和任务完成贡献度 P_B 的总和, 在线贡献权重为 W_O , 任务贡献权重为 W_T 。对于特定用户 X , X 用户的总贡献度为 $P^X = W_O P_O + W_T P_T = W_O P_O + W_T (P_W + P_B)$ 。

持续在网的设备会不断积累在线贡献度 P_O , 并在某个阶段性区块产生后根据 P_O 的阶段性全网占比情况, 获得该阶段新区块产生的同等份额收益。同时, 用 Power 值来表示不同硬件成本、能耗、计算能力的差异, 不同类型节点 Power 值不同, P_O 的积累速度也不同。

T 是 CPU 使用量(C)、内存使用量(M)、带宽使用量(B)、存储空间使用量(D)函数。那么该任务当前的贡献度为 $P_w = P(C, M, B, D)$ 。以区块生成时间作为单位时间, 对于特定任务 S , 设它的开始时间为 t_0^S , 当前时间为 t^S , 如果某设备同时服务于多个应用, 那么整个设备在 t 时刻的贡献度则为所有任务的贡献度之和 $P^X(t) = \sum P^x(t^S, t_0^S)$ 。

三、 应用场景

1. 性能监控和压力测试

通过分布在边缘计算节点上的代理应用程序组成的大规模监控测试网络，为企业级客户提供应用性能监控、网络性能监控和拥有独立公网 IP 节点的 APP 和 Web 压力测试能力

2. P2P 计算和云存储

按需和弹性的内容/存储/应用分布式架构，我们称之为 CDN-aaS，自愿接入边缘计算节点设备的用户通过共享空闲的剩余算力、提供存储资源和网络带宽为媒体内容分享和分布式加密文件存储提供服务并获得可量化的收益。

3. VPN 边缘接入

用户在自愿接入的边缘节点上运行异构平台兼容的客户端应用程序，该应用接入到云端并进行加密通讯，边缘节点作为组成 SD-WAN 的 CPE 设备。高带宽的用户可以通过分享带宽作为超级节点并组成集群。

4. IoT 数据聚合分析

传统的云端计算模式可以发出计算指令，中心化的后台运算完并响应结果，这种模式可以满足大部分的应用场景，然而一些需要在毫秒之间实时响应的应用，如果按照现有云计算模式，将数据通过延时、抖动和距离不可控的网络传输到几千公里之外的云端，运算完成再传回结果，显然无法满足实时计算应用需求。此

时如果将靠近设备端或数据源头的网络、计算、存储、应用、数据融合起来，采取边缘就近计算并实时响应结果，

边缘计算应用程序使用物联网设备的处理能力来过滤、预处理、聚合或评分物联网数据，并利用云的强大计算能力和灵活性对这些数据进行复杂的分析。

5. 公有链承载

区块链的公有链需要部署在去中心化的分布式平台（如 BTC、ETH），这也是区块链最初形成的模式。而当前并没有一个去中心化的平台可以支撑公有链的部署与运行。从当前市场情况来看，有很多做公有链的企业和团队都面临这个问题，其中部分人想通过自行分发设备来解决这一问题。而这种部署模式无疑增加了公有链的投入成本与运营成本。去中心化的云计算平台能够很好的解决公有链的分布式部署和运行的需求。现在及未来的 Dapp 也会面临与公有链相同的问题。

随着区块链产业的快速发展，是需要有一个去中心化的云计算平台来做坚实的支持，以能够让 Dapp 具备快速部署、测试和上线运行的能力。