

Akash 网络：去中心化云

基础设施市场

Overclock 实验室

2018 年五月 13 日

版本 0.0.3

备注：Akash 网络是一个正在研发中的项目，本文的新版本将会在 akash.network 发布。若有评论和建议，请联系 research@akash.network

摘要

云计算——将运算工作转移至远程服务器的过程——有着原生性的缺陷。尽管大部分时间它像广告所说的那样工作着，我们还是发现效率低下的问题仍在困扰着这个系统。由主要云供应商提供的产品虽然可用，但他们有一些局限性，这些局限性可以通过先进的箱化技术和强大的通证经济来解决。本白皮书的目的就是提出我们关于云服务市场的计划，可称之为 Akash 网络，世界首个云计算现货市场。

我们认为在未来，全球的云基础设施是去中心化的，分布在所有云服务供应商上；有一个以安全、快速和透明的现货定价方式部署和清算（日益商品化）数据中心计算的市場。服务在民主但统一的生态中销售，任何人都可以使用。

本文，我们将介绍 Akash，一个去中心化、有竞争性、可在多个云服务供应商间分布应用的全球云基础设施网络。文章会介绍现有市场的情况，指出我们将如何通过无服务器箱化结构的最新发展来解决现有问题，解释网络原生通证 AKASH 的基础和必要性，最后展示我们产品发布的路线图。

目录

一	介绍.....	4
(一)	一个遭遇问题的行业.....	4
二	Akash 网络.....	5
(一)	Akash 区块链.....	6
(二)	Akash 通证, AKASH.....	8
三	市场.....	8
四	部署.....	9
(一)	清单分布.....	10
(二)	覆盖网络.....	10
五	自动化.....	11
(一)	示例: 延迟优化部署.....	12
(二)	示例: 机器学习部署.....	13
	参考资料.....	15

图片清单

图 1 Akash 网络中链上和链下不同参与者的互动展示	7
图 2 市场采购流程总结。(1) 用户的部署订单发布到订单簿 (2) 数据中心为部署订单发布符合条件的履行订单 (3)将最佳履行订单与部署订单匹配, 创建一个新的租约。	9
图 3 Akash 覆盖网络展示	11
图 4 单个数据中心部署给分布全球的终端用户带来较多延迟而导致性能下降的展示	12
图 5 通过在接近终端用户的数据中心之间动态地分配工作负载及其状态, 改进网络性能的展示.....	13
图 6 一个机器学习批处理作业在更少的负载下运行一个主节点和一个从节点	14
图 7 一个机器学习批处理作业在负载下运行一个主节点和多个从节点	14

一 介绍

Akash 网络 (Akash) 是一个安全、透明、去中心化的云计算市场，连接需要计算资源的人 (客户) 和那些有能力出租计算力的人 (供应商)。

Akash 就像一个“超级”云平台 (supercloud) ——在市场上的供应商之上提供一个统一的层，为客户展示的是单个云平台，客户不需要考虑他们用的供应商具体是哪家。

客户使用 Akash 是因为其成本优势、可用度、在不同云供应商间转换的灵活性，以及全球部署的性能优势。供应商使用 Akash 是因为他们可以通过提供专属容量或暂时不用的容量来赚取利润。

(一) 一个遭遇问题的行业

到 2020 年，云基础设施供应商将会战局全球互联网流浪的 53% [Cisco(2016)]，其中亚马逊、谷歌、微软将会提供 80% 的有效载荷 [Forrester(2017)]。

虽然云计算将提供大部分的工作负载，但互联网的未来面临着任由这三家供应商整合、中心化、摆布的风险。

采用云服务的首要动力是其承诺的灵活性和成本优势，但事实上，云供应商提供的产品价格过高、复杂，并将客户锁定在特定生态系统中，限制他们的创新、竞争能力，且供应商还拥有对客户基础设施需求的主导权。

在云上和自我 (内部配置) 购买硬件及租用数据中心的资本支出差异很小；但是，云供应商有显著的运营成本优势，因为他们大力投资在自动化方面，把人工操作降到最低。

尽管在内部运行计算可以有好得多的灵活性、性能、安全性，但大部分组织正逐步放弃自己运营数据中心，向云迁移，因为他们发现由于缺乏自动化，利用率又低，越来越难证明相关运营成本的合理性。闲置、未被充分利用的服务器带来高昂的成本和诸多浪费。研究估计现实中多达 85% 的服务器容量未被充分利用 [Glanz(2012)] [Kaplan 等 (2008) Kaplan, Forrest, 和 Kindler] [Liu(2011)] [Kooimey 和 Taylor(2015)]。

云供应商通过建立超大规模设施提升利润空间，即把资源整合到几个数据中心，提高经济效率，交叉销售完全管理的后端服务，如数据库、缓存存储、API 网关等。

超大规模允许他们超量订阅客户，提升利润空间，但也造成了单点失败。工作量的地理分布提供更高的可靠性和终端用户性能；但是，云供应商让用户很难做到跨区域使用云服务，因为这不符合供应商们的最佳利益。

云供应商希望用户在单一数据中心部署应用，并经常通过高额的带宽费用和不同地区的定价对跨地区或多地区的应用程序进行惩罚。这就是 AWS 对各个地区的相同资源制定不同价格体系的原因。

尽管销售实例利润丰厚，但云服务供应商通常对实例收取的费用往往少于后台管理服务（PaaS）收取的额外费用；类似于旧的汉堡和薯条模式，餐馆需要亏本销售汉堡，这样他们才能以高利润率销售更让人上瘾的薯条。

供应商出售的 PaaS 服务往往是白标的开源项目，在这些项目中，原作者从未受到激励，云供应商也没有动力改进产品。例如，AWS 的 ElastiCache 是一个白标的开源软件，名为 Redis。Redis 是一个开源项目，深受开发者喜爱，由 Salvatore Sanfilippo 编写，由 Redis 实验室维护。

在本文撰写之时，一个在美国东部（俄亥俄州）运行于 r3.8xlarge 的托管 Redis 服务器的价格为 31,449 美元/年[亚马逊(2017a)]，而相同的实例但不运行 Redis 的价格为 18,385 美元/年[亚马逊(2017b)]。额外的 13,064 美元仅仅是为了让顾客“放心”。Sanfilippo 和 Redis 实验室都没有为此而受到激励。

同时，更多的服务意味着客户更加依赖云供应商。使用非标准 API 增加的特性、服务可用性和编码带来的复杂性导致客户被云供应商锁定，阻止客户在市场上探索其他更好的选择，抑制了创新。

供应商采用的这种模式抑制了创新，因为它大大降低了开源项目成功的机会。云服务供应商实际上扮演着中间人的角色，为行业制定管理规则，但总体上对社会没有任何贡献。

二 Akash 网络

Akash 网络的基本设计目标是为供应商保持一个较低的准入门槛，同时确保客户可以信任平台提供给他们资源。为了实现这一点，系统需要一个公开的可验证的网络交易记录。为此，Akash 网络引入区块链技术，作为对分布式数据库准确性形成共识的一种手段。

首先且最重要的是，Akash 是一个允许客户从供应商那里获取资源的平台。这是一个基于区块链的分布式交易所实现的，在这个交易所中，客户发布他们想要的资源，供应商对其竞标。这个市场的货币是一种数字通证，即 Akash (AKASH)，其账本存储在区块链上。

Akash 是一个用于实际应用程序的云平台。这些应用程序的要求包括：

- 许多工作负载部署在任意数量的数据中心。
- 连接限制，防止不必要的访问工作负载。
- 自我管理，操作人员不需要经常关注部署。

为了支持在采购的资源上运行工作负载，Akash 加入了一个点对点协议，用于在客户端供应商之间分配工作负载和部署配置。

Akash 中的工作负载被定义为 Docker 容器。Docker 容器允许高度隔离和可配置的执行环境，已经成为当今许多基于云的部署的一部分。

(一) Akash 区块链

Akash 区块链在去中心化和去信任的环境中提供了一层信任。客户对当今大型基础设施供应商的信任本质上是基于他们多年来建立起来的品牌资产。Akash 没有也不应该要求同样的信任，因为任何有能力的供应商都可以竞争在 Akash 平台上提供服务。相反，区块链通过一个开放透明的平台赢得信任。链上的数据是不可篡改的，所有交易都有公开记录，包括每个供应者的过往服务履行情况。

Akash 在政治上也是去中心化的。没有单一的实体控制网络，也没有中介促进交易。因此，没有实体被激励去控制或从网络中提取边际收益。例如，像可口可乐这样的大公司可以作为供应商参与到网络中，为另一个大公司或单个开发者提供计算服务，但这三方在网络中是平等的。

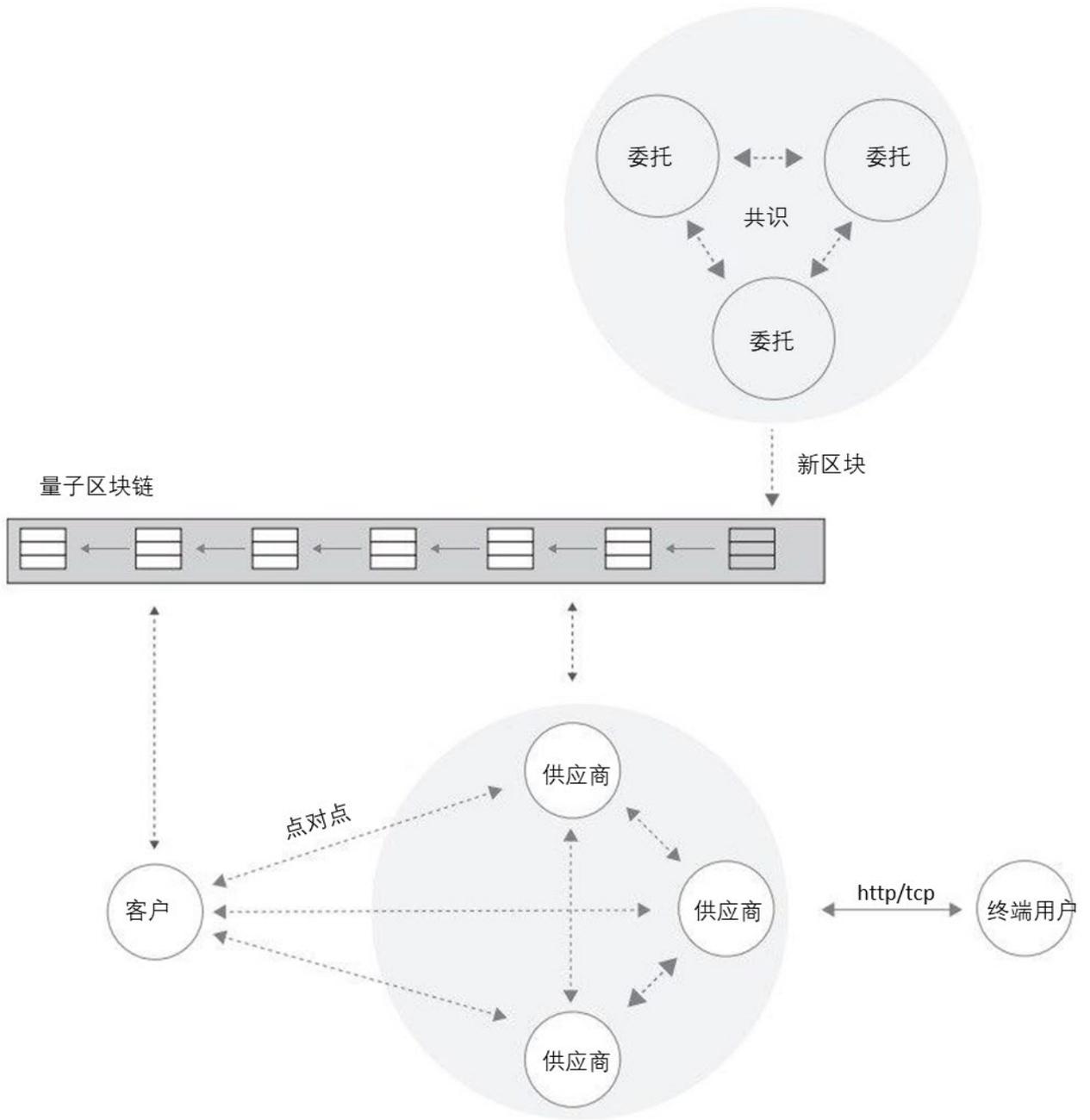


图1 Akash 网络中链上和链下不同参与者的互动展示

(二) Akash 通证, AKASH

Akash 通证 (AKASH) 是用于简化价值交换, 并将经济激励与适当的用户行为结合起来。Akash 通证是市场的流通货币, 用于支付在 Akash 去中心化网络中租用的计算基础设施。该通证在 Akash 生态系统中主要提供两个功能。

市场规模预计将达到 7370 亿美元, 年增长率超过 21% [Gartner(2017)], AKASH 的流动性将与计算能力的需求相匹配。沿着这条思路, 我们对网络和早期采用者和终端用户的实现最大 AKASH 流动性充满信心。

1. 质押

Akash 网络的稳定性依赖于一个可以防范不良行为者滥用的质押系统。该质押系统给考虑参与网络的不良行为者提供一个反向的金钱激励。当新的、未知的供应商加入我们的网络时, 欺诈行为的最高风险。不同于要求对新帐户进行集中或联合的审批过程, Akash 网络允许任何人加入。

当一个新的供应商选择在 Akash 网络上提供其资源而不是获得批准时, 它必须质押一个对网络有意义的 Akash 数额。没有最低的质押规定数量, 但是参与 Akash 网络治理与供应商质押的数量成正比, 其质押量被视为所有质押总量的一小部分。而且, 质押数量是一个衡量因素, 计入供应商的声誉评分中, 租户可以将其用作部署标准。

2. 全球支付

Akash 通证将减轻通常由跨境支付导致的外汇风险。在这些交易中, Akash 通证将取代了法币, 简化了云基础设施行业的价值交换。我们的匹配引擎比照主流市场价格, 对每一个箱化计算进行有竞争性的定价, 用 Akash 通证标价。当租户与供应商匹配时, 租户将向网络支付 Akash 通证, 随后根据租约条款将其支付给供应商。

三 市场

基础设施采购——客户从供应商租赁基础设施的过程——在 Akash 上是通过一个去中心化交易所 (市场) 实现的。

市场由公共订单簿和配对算法组成。客户发出部署订单, 其中包含客户服务需求的具体参数, 数据中心发出履行订单, 对部署订单进行投标。部署订单包括客户愿意在特定时间内为固定数量的计算单元 (通过内存、cpu、存储和带宽来衡量) 支付的最大金额; 履行订单声明供应商提供资源的价格。

部署订单在客户以秒为单位指定的时间长度内是开放的。当部署订单是开放时，供应商可以发布履行订单对其进行投标。

如果履行订单满足部署订单的所有最低规格要求，则履行订单可以与部署订单匹配。如果一个部署订单有一组合格的履行订单竞标，提供最低价格的履行订单将与部署订单匹配。如果多个履行订单符合匹配条件并提供相同的价格，那么首先下的履行订单将与部署订单匹配。

企业和个人消费者将希望并且需要保护他们公开展示的计算能力的使用。为了防止竞争对手的数据挖掘和其他攻击载体，增加了同源加密层。

当部署订单和履行订单之间出现匹配时，将创建 *租约*。租约包含对部署和履行订单的引用。租约是履行合同的约束因素。

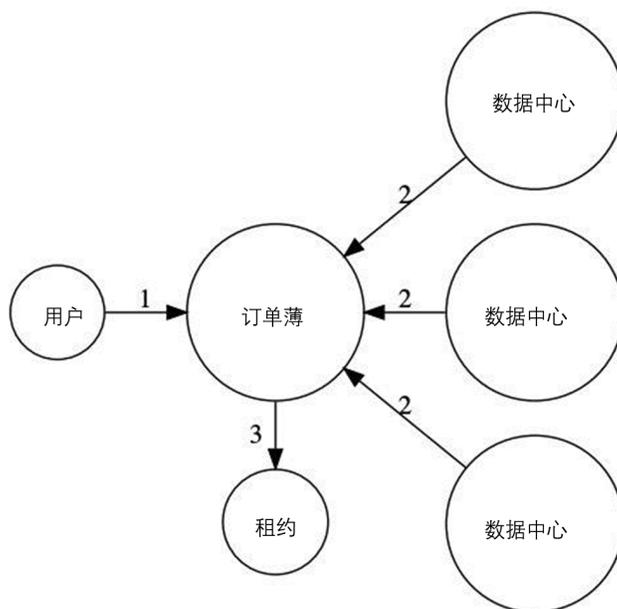


图2 市场采购流程总结。(1) 用户的部署订单发布到订单簿 (2) 数据中心为部署订单发布符合条件的履行订单 (3) 将最佳履行订单与部署订单匹配，创建一个新的租约。

四 部署

一旦采购了资源，客户必须将其工作负载分配给供应商，以便它们可以在租用的资源上执行。我们将 Akash 网络上客户工作负载的当前状态称为 *部署*。

用户在清单中描述他们想要的部署。清单是用声明式文件格式编写的，其中包含工作负载定义、配置和连接规则。供应商使用工作负载定义和配置在自己提供的资源上执行工作负载，并使用连接规则构建覆盖网络和防火墙配置。

清单的哈希称为部署版本，并存储在基于区块链的分布式数据库中。

(一) 清单分布

清单包含的敏感信息应该只与部署的参与者共享。这给自我管理部署带来了一个问题——Akash 必须自主地分发工作负载定义，而不将其内容透露给不必要的参与者。

为了解决这些问题，我们设计了一个点对点文件共享方案，在该方案中，租约参与者根据需要相互分发清单。协议在 TLS 连接上链下运行；每个参与者都可以通过计算其哈希值，并将其与存储在基于区块链的分布式数据库上的部署版本进行比较，从而验证它们收到的清单。

除了提供私有的、安全的、自主的清单分发之外，点对点协议还支持大型清单快速分发到大量数据中心中。

(二) 覆盖网络

默认情况下，工作负载的网络是隔离的——没有任何东西可以与之连接。虽然这是安全的，但对于真实的应用程序来说是不实际的。例如，考虑一个简单的网页应用程序：终端用户浏览器应该能够访问网页层工作负载，而网页层需要与数据库工作负载通信。此外，网页层可能与数据库在不同的数据中心中托管。

在 Akash 网络上，通过在清单中定义连接拓扑，客户可以有选择地允许工作负载单向和双向的通信。数据中心使用此拓扑配置防火墙规则，并根据需要在各个工作负载之间创建安全网络。

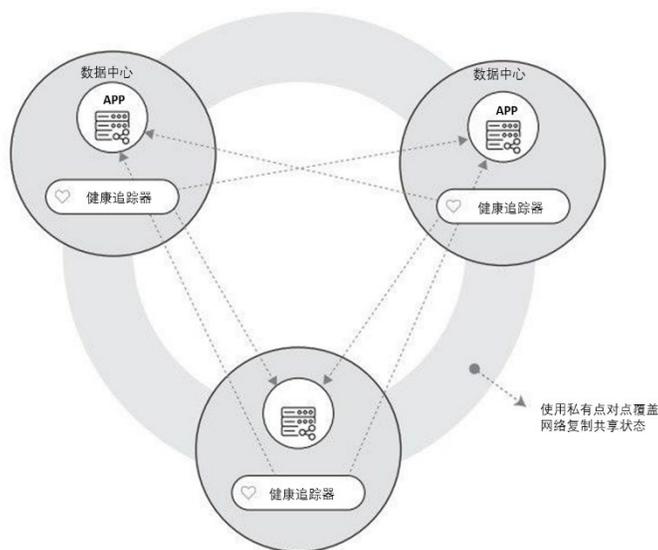


图3 Akash 覆盖网络展示

为了支持安全的跨数据中心通信，供应商通过 mTLS 隧道相互公开工作负载。每个工作负载到工作负载的连接使用一个不同的隧道。

在建立这些隧道之前，供应商为每个需要的隧道生成一个 TLS 认证，并与必要的供应商交换这些证书。每个提供者的根证书都存储在基于区块链的分布式数据库中，使对方能够验证它收到的证书的真实性。

一旦交换了证书，供应商就建立一个认证过的隧道，并将工作负载的网络连接至此。所有这些对于工作负载本身都是透明的——它们可以通过稳定地址和标准协议相互连接。

五 自动化

云基础设施的动态特性对运营管理来说既是好事，也是坏事。新的资源可以按意愿提供是一件好事；爆炸性的管理开销和上述资源的复杂性是一种诅咒。开发运营

(DevOps) 的目标——以编程方式管理部署的实践——是通过利用云基础设施的优势来减轻其痛点。

Akash 网络的建立是为了给开发运营工程师提供一个简单但强大的工具集，用于创建高度自动化的部署。该工具集由支持非管理应用程序——通用工作负载和覆盖网络——的基元组成，可用于创建自治、自管理的系统。

Akash 上的自管理部署非常简单，只需创建自己管理部署的工作负载即可。当供应商加入或离开部署时，开发运营工程师可能会使用一个工作负载来更新 DNS 条目；测试网页层应用程序的响应时间；并根据输入指标的任意数量根据需要扩展和缩小基础设施（根据客户定义的权限和约束）。“管理层”可以分散在所有数据中心进行部署，全局状态由运行在安全覆盖网络上的分布式数据库维护。

（一） 示例：延迟优化部署

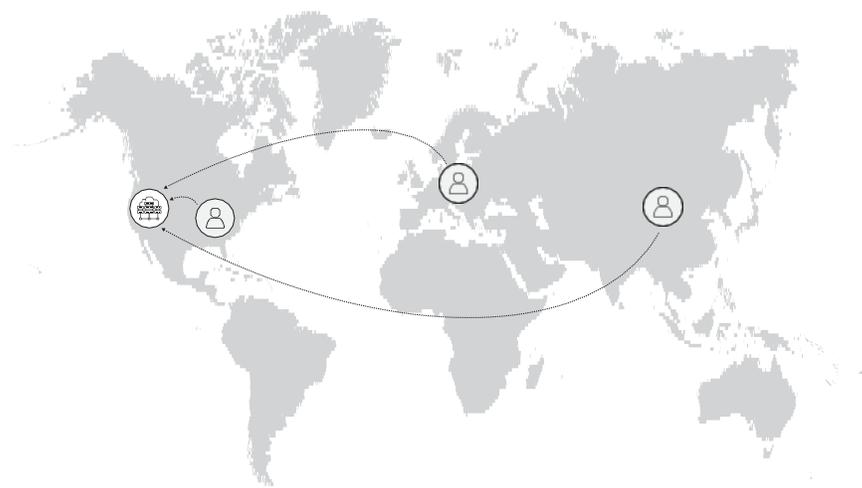


图4 单个数据中心部署给分布全球的终端用户带来较多延迟而导致性能下降的展示

许多基于网页的应用程序对延迟很敏感——应用程序服务器的较低响应时间可以显著改进终端用户体验。这类应用程序的现代部署使用内容交付网络（CDNs）来向终端用户快速交付图像等静态内容。

CDNs 通过分发内容，使其在地理位置上接近访问内容的用户，从而减少了延迟。在 Akash 网络上的部署不仅可以复制这种方法，而且可以击败它——Akash 让客户能够将动态内容放置在接近应用程序用户的地方。

为了在 Akash 上实现一个自我管理的动态交付网络，开发运营工程师将在部署中包括一个管理层，以监控客户的地理位置。这个管理层将在全球范围内添加和删除数据中心，在用户活跃度高的地区提供更多的资源，在用户参与度低的地区提供更少的资源。



图5 通过在接近终端用户的数据中心之间动态地分配工作负载及其状态，改进网络性能展示

(二) 示例：机器学习部署

机器学习应用程序使用大量节点来并行化涉及大数据集的计算。他们以“批量”的方式完成工作——不存在所需的“稳定状态”的能力。

Akash 上的机器学习应用程序可以使用管理层主动获取单个数据中心内的资源。当一个机器学习任务开始时，管理层可以为它“扩大”节点的数量；当一个任务完成时，为它指定的资源可以被释放。

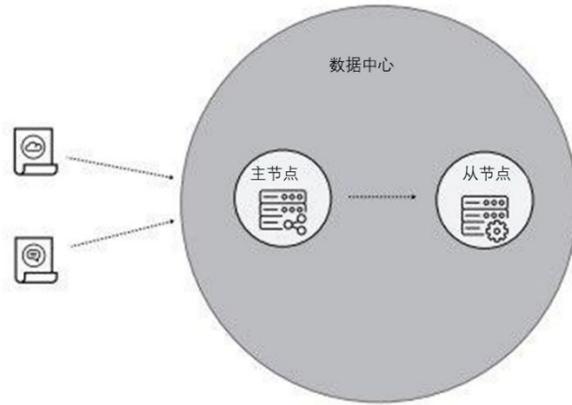


图7 一个机器学习批处理作业在更少的负载下运行一个主节点和一个从节点

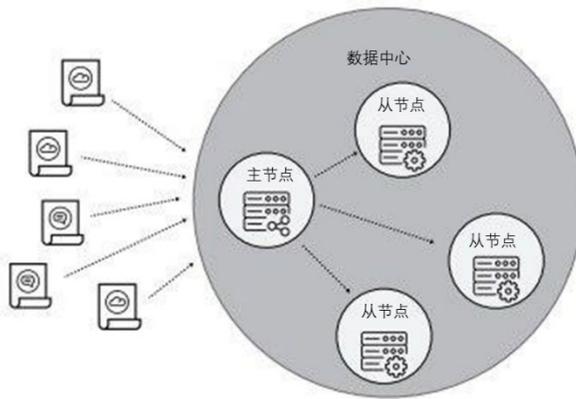


图6 一个机器学习批处理作业在负载下运行一个主节点和多个从节点

参考资料

- [Amazon(2017a)] Amazon. Amazon elasticache pricing. 2017a. URL
<https://aws.amazon.com/elasticache/pricing/>.
- [Amazon(2017b)] Amazon. Amazon ec2 pricing. 2017b. URL
<https://aws.amazon.com/ec2/pricing/>.
- [Cisco(2016)] Cisco. Cisco global cloud index: Forecast and methodology, 2015 - 2020. 2016. URL
<https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud->
- [Forrester(2017)] Forrester. Predictions 2018: Cloud computing accelerates enterprise transformation everywhere. 2017. URL
<https://www.forrester.com/report/Predictions+2018+Cloud+Computing+Accelerates+Enterpri>
- [Gartner(2017)] Gartner. Forecast analysis: Public cloud services, worldwide, 2q17 update. 2017. URL
<https://www.gartner.com/doc/3803517>.
- [Glanz(2012)] James Glanz. Power, pollution and the internet. 2012. URL
<http://www.nytimes.com/2012/09/23/technology/data-centers-waste-vast-amounts-of-energy>
- [Kaplan et al.(2008)Kaplan, Forrest, and Kindler] James Kaplan, William Forrest, and Noah Kindler. Revolutionizing data center energy efficiency. 2008. URL
https://www.sallan.org/pdf-docs/McKinsey_Data_Center_Efficiency.pdf.
- [Koomey and Taylor(2015)] Jonathan Koomey and Jon Taylor. New data supports finding that 30 percent of servers are 'comatose', indicating that nearly a third of capital in enterprise data centers is wasted. 2015. URL
https://anthesisgroup.com/wp-content/uploads/2015/06/Case-Study_DataSupports30PercentC
- [Liu(2011)] Huan Liu. A measurement study of server utilization in public clouds. 2011. URL
<http://ieeexplore.ieee.org/document/6118751/>.