

아카시 네트워크(Akash Network)

해당 번역본은 이용자가 디지털자산 관련 정보를 편리하게 확인할 수 있도록
단순 참고용으로 번역된 것이며, 투자 권유를 목적으로 하지 않습니다.
그 내용이 정확하지 않을 수 있으므로 정확한 정보 습득을 위해서는
원문을 참고하시거나 원문 작성 주체 측에 문의하시기 바랍니다.
두나무는 해당 번역본 내용의 진실성 및 정확성을 보장하지 않고,
제공된 정보에 의한 투자 결과에 대하여 어떠한 책임도 지지 않으며,
투자 손실은 투자자에게 귀속됩니다

아카시 네트워크: 탈중앙화 클라우드 인프라 마켓플레이스

오버클릭 랩스

2018년 5월 13일

버전 0.0.3

주: 아카시 네트워크는 현재 활발히 진행 중인 연구 프로젝트이며, 본 백서의 최신 버전은 akash.network에서 확인할 수 있습니다. 의견이나 제안은 research@akash.network 로 보내주시기 바랍니다.

초록

클라우드 컴퓨팅 — 즉, 작업을 원격 서버로 이전하여 처리하는 과정 — 은 본질적으로 결함을 안고 있다. 광고된 대로 대부분 작동하긴 하지만, 우리는 여전히 시스템에 비효율이 만연하다는 것을 발견했다. 주요 클라우드 제공업체가 만든 제품들은 사용 가능하지만, 오늘날 컨테이너 기술의 발전과 강력한 토큰 경제를 통해 해결할 수 있는 한계들을 가지고 있다. 본 백서의 목적은 아카시 네트워크(Akash Network)라 불리는 클라우드 서비스 시장, 즉 세계 최초의 글로벌 클라우드 컴퓨팅 스팟 마켓에 대한 우리의 계획을 제시하는 것이다.

우리는 세계의 글로벌 클라우드 인프라가 탈중앙화되고 모든 클라우드 서비스 제공업체들 사이에 분산되는 미래를 본다. 이 시장은 점점 더 상품화되는 데이터센터 컴퓨팅을 안전하고, 빠르며, 투명하게 스팟 가격으로 배포하고 청산한다. 서비스는 누구나 사용할 수 있는 민주적이면서도 통합된 생태계에서 판매된다.

본 백서에서는 탈중앙화되고 경쟁적이며 전 세계 여러 클라우드 서비스 제공업체 간에 애플리케이션을 분산시킬 수 있는 클라우드 인프라 네트워크인 아카시(Akash)를 소개한다. 본 문서는 기존 시장의 현황을 설명하고, 이러한 문제를 해결하기 위해 우리가 무서버(serverless) 컨테이너 오케스트레이션의 최신 발전을 어떻게 활용하고 있는지, 네트워크의 네이티브 토큰인 AKASH의 기본 개념과 필요성, 마지막으로 출시 로드맵을 제시할 것이다.

목차

- 1. 서론
 - 1.1 문제 많은 산업
- 2. 아카시 네트워크
 - 2.1 아카시 블록체인
 - 2.2 아카시 토큰, AKASH
- 3. 마켓플레이스
- 4. 배포(Deployment)
 - 4.1 매니페스트 배포
 - 4.2 오버레이 네트워크
- 5. 자동화
 - 5.1 예시: 지연시간 최적화 배포
 - 5.2 예시: 머신러닝 배포

1 서론

아카시 네트워크(Akash)는 컴퓨팅 자원이 필요한 이들(클라이언트)과 임대할 수 있는 컴퓨팅 용량을 가진 이들(프로바이더)을 연결하는 안전하고, 투명하며, 탈중앙화된 클라우드 컴퓨팅 마켓플레이스이다.

아카시는 "슈퍼" 클라우드 플랫폼(supercloud)으로 작동한다. 즉, 마켓플레이스의 모든 프로바이더 위에 통합 계층을 제공하여, 클라이언트가 어떤 특정 프로바이더를 사용하든 단일 클라우드 플랫폼으로 인식할 수 있게 한다.

클라이언트는 비용적 이점, 사용 편의성, 클라우드 프로바이더 간 이동의 유연성, 그리고 글로벌 배포의 성능적 혜택 때문에 아카시를 사용한다. 프로바이더는 전용 용량이든 일시적으로 사용하지 않는 용량이든 간에 이를 통해 수익을 얻을 수 있기 때문에 아카시를 사용한다.

1.1 문제 많은 산업

2020년까지 클라우드 인프라 제공업체들은 전 세계 인터넷 트래픽의 **53%**를 차지할 것이며 [Cisco(2016)], 그 중 아마존, 구글, 마이크로소프트가 전체 전송량의 **80%**를 제공할 것이다 [Forrester(2017)].

클라우드가 대다수의 워크로드를 제공할 것이지만, 인터넷의 미래는 이 세 제공업체에 의해 집중되고 중앙화되어, 그들의 영향력에 좌우될 위험에 처해 있다.

클라우드 도입의 주된 동인은 유연성과 비용적 이점의 약속이지만, 실제로 클라우드 제공업체가 제공하는 제품들은 고가이며 복잡하고, 클라이언트를 혁신하거나 경쟁하거나 자신들의 인프라 수요에 대한 주권을 유지할 수 없게 하는 생태계에 가둔다.

클라우드에서 운영하는 것과 자체 관리(on-premise) 방식으로 하드웨어를 구입하고 데이터센터를 임대하는 것 사이의 자본적 지출 차이는 미미하다. 그러나 클라우드 제공업체들은 최소한의 인력 개입으로 자동화에 투자하여 운영비용 측면에서 상당한 우위를 가진다.

온프레미스 방식으로 컴퓨팅을 운영하면 훨씬 더 나은 유연성, 성능, 보안을 제공할 수 있음에도 불구하고, 조직들은 충분한 자동화의 부재와 낮은 활용률로 인한 운영비용을 정당화하기 어려워 데이터센터 운영을 포기하고 클라우드로 이전하고 있다. 유헤 상태이거나 저활용 서버는 비용이 많이 들고 낭비적이다. 분석가들은 실제로 사용되는 서버의 최대 85%가 저활용 상태라고 추정한다 [Glanz(2012)] [Kaplan et al. (2008)Kaplan, Forrest, and Kindler] [Liu(2011)] [Koomey and Taylor(2015)].

클라우드 제공업체들은 초대규모(hyper-scale) 설치를 구축하여, 즉 경제적 효율성을 위해 소수의 데이터센터에 자원을 집중시킴으로써 마진을 확보하고,

데이터베이스, 캐시 스토어, API 게이트웨이 등과 같은 완전 관리형 백엔드 서비스를 교차 판매한다.

초대규모(hyper-scale) 구조는 고객을 초과 할당(oversubscribe)할 수 있게 하여 더 높은 마진을 가져다주지만, 동시에 단일 장애 지점을 만들어낸다. 지리적으로 분산된 워크로드는 더 높은 신뢰성과 최종 사용자 성능을 제공하지만, 클라우드 제공업체들은 클라이언트가 다지역(multi-regional)으로 운영하는 것을 극도로 어렵게 만들며, 이는 그들의 이익에 부합하지 않기 때문이다.

클라우드 제공업체들은 고객이 애플리케이션을 단일 데이터센터에 배포하도록 선호하며, 다지역 또는 다중가용 영역(multi-zonal)으로 배포할 경우 가혹한 대역폭 요금과 가변적인 지역별 가격으로 불이익을 준다. 이것이 동일한 리소스임에도 불구하고 AWS의 가격 모델이 지역마다 다른 이유다.

인스턴스 판매가 수익성이 있긴 하지만, 클라우드 제공업체들은 일반적으로 인스턴스에 대해 적은 비용만을 청구하며, 대신 관리형 백엔드 서비스(PaaS)에 대해서는 훨씬 더 높은 프리미엄을 청구한다. 이는 과거 레스토랑이 버거는 손해를 보고 판매하면서 더 중독성 있는 감자튀김을 높은 마진으로 판매하는 모델에 비유될 수 있다.

제공업체들이 판매하는 PaaS 서비스는 대체로 화이트 라벨로 제공되는 오픈소스 프로젝트이며, 원 저자는 전혀 보상받지 못하고 클라우드 제공업체 역시 해당 제품을 발전시킬 동기가 없다. 예를 들어, AWS의 ElastiCache는 Redis라는 오픈소스 소프트웨어를 화이트 라벨로 제공하는 서비스다. Redis는 개발자들에게 큰 사랑을 받는 오픈소스 프로젝트이며, Salvatore Sanfilippo가 작성하고 Redis Labs가 유지 관리하고 있다.

이 백서 작성 시점에서, 미국 동부(오하이오) 지역에서 *r3.8xlarge* 인스턴스로 실행되는 관리형 Redis 서버의 가격은 연간 \$31,449 [Amazon(2017a)]이며, 동일한 인스턴스를 Redis 없이 사용할 경우 가격은 연간 \$18,385 [Amazon(2017b)]이다. 고객이 단지 "심리적 안도감"을 위해 추가로 지불하는 비용이 \$13,064에 이른다. 그러나 Sanfilippo나 Redis Labs는 그 어떤 보상도 받지 못한다.

또한 서비스가 많아질수록 고객은 클라우드 제공업체에 더 의존하게 된다. 기능 수, 서비스 가용성 증가, 그리고 비표준 API를 통한 코드화로 인한 복잡성은 고객을 클라우드 벤더에게 종속시켜, 더 나은 시장 대안을 탐색하지 못하게 하고 혁신을 억제한다.

제공업체들이 채택한 이러한 모델은 오픈소스 프로젝트가 성공할 가능성을 극적으로 낮추면서 혁신을 억제한다. 클라우드 제공업체들은 사실상 산업의 규칙을 정하는 중개인으로 행동하지만, 사회 전체에는 아무런 기여도 하지 않는다.

2 아카시 네트워크

아카시 네트워크의 근본적인 설계 목표는 프로바이더에게 낮은 진입 장벽을 유지하면서 동시에 클라이언트가 플랫폼이 제공하는 자원을 신뢰할 수 있도록 보장하는 것이다. 이를 달성하기 위해, 이 시스템은 네트워크 내 거래의 공개적으로 검증 가능한 기록을 필요로 한다. 이러한 목적을 위해 아카시 네트워크는 분산 데이터 베이스의 진실성을 합의하기 위한 수단으로 블록체인 기술을 사용하여 구현된다.

아카시는 무엇보다 먼저, 클라이언트가 프로바이더로부터 자원을 조달할 수 있도록 하는 플랫폼이다. 이는 클라이언트가 원하는 자원을 게시하면 프로바이더들이 입찰할 수 있는 블록체인 기반 분산 거래소를 통해 가능하다. 이 마켓플레이스의 화폐는 디지털 토큰인 아카시(AKASH)이며, 그 원장은 블록체인에 저장된다.

아카시는 실제 애플리케이션을 위한 클라우드 플랫폼이다. 이러한 애플리케이션의 요구사항은 다음과 같다:

- 여러 데이터센터에 걸쳐 배포되는 다수의 워크로드.
- 원치 않는 접근을 방지하는 연결 제한.
- 운영자가 배포를 지속적으로 관리할 필요가 없는 자기 관리(Self-managed) 기능.

조달된 자원에서 워크로드를 실행하기 위해, 아카시는 클라이언트의 프로바이더 간 및 프로바이더와의 워크로드 및 배포 구성을 분산시키는 피어-투-피어 프로토콜을 포함한다.

아카시에서 워크로드는 도커(Docker) 컨테이너로 정의된다. 도커 컨테이너는 고도로 격리되고 구성 가능한 실행 환경을 제공하며, 이미 많은 클라우드 기반 배포에서 사용되고 있다.

2.1 아카시 블록체인

아카시 블록체인은 탈중앙화되고 신뢰할 수 없는 환경에서 신뢰 계층을 제공한다. 클라이언트들은 본질적으로 오늘날 대규모 인프라 제공업체를 신뢰하는데, 이는 주로 이들이 수년에 걸쳐 쌓아온 브랜드 자산에 기반한다. 아카시는 동일한 신뢰의 도약(leap of faith)을 요구하지도, 요구해서도 안 된다. 왜냐하면 용량을 가진 어떤 프로바이더라도 아카시에서 서비스를 제공하기 위해 경쟁할 수 있기 때문이다. 대신 블록체인은 개방

적이고 투명한 플랫폼을 통해 신뢰를 획득한다. 체인에 기록된 데이터는 모든 거래를 포함하여 각 프로바이더의 이행 내역까지 담고 있는 불변의 공개 기록이다.

아카시는 정치적으로도 탈중앙화되어 있다. 어떤 단일 주체도 네트워크를 통제하지 않으며, 어떤 중개자도 거래를 촉진하지 않는다. 따라서 그 어떤 주체도 네트워크를 통제하거나 한계 수익을 추출할 유인이 없다. 예를 들어, 코카콜라와 같은 대기업은

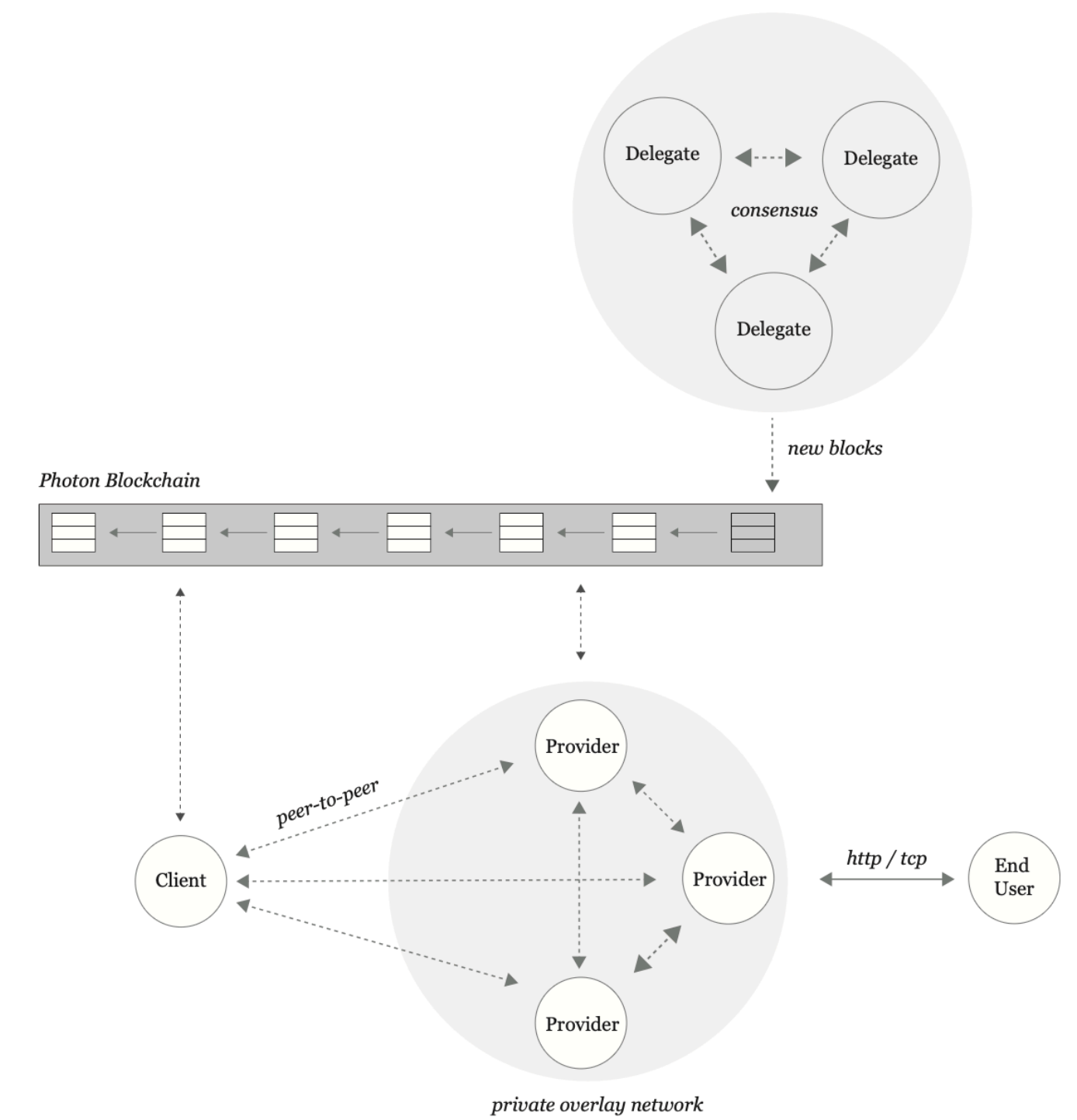


그림 1: 아카시 네트워크에서 다양한 참여자 간 온체인 및 오프체인 상호작용 그림

네트워크에서 프로바이더로 참여하여 또 다른 대기업이나 개별 개발자에게 컴퓨팅을 제공할 수 있으며, 이 세 주체 모두 네트워크에서 동등한 입장을 가진다.

2.2 아카시 토큰, AKASH

아카시 토큰(AKASH)은 가치 교환을 단순화하고 올바른 사용자 행동에 경제적 인센티브를 부여하기 위해 사용된다. 아카시 토큰은 아카시의 탈중앙화 네트워크에서 임대된 컴퓨팅 인프라에 대한 대가를 지불하는 데 사용되는 마켓플레이스 통화이다. 이 토큰은 아카시 생태계에서 두 가지 주요 기능을 수행한다.

7,370억 달러 규모로 예상되는 시장에서, 연간 21%가 넘는 성장률을 보이며 [Gartner(2017)], AKASH의 유동성은 컴퓨팅 파워 수요에 의해 보장될 것이다. 이러한 맥락에서,

2.2.1 스테이킹(Staking)

아카시 네트워크의 안정성은 우리의 시스템을 악용하려는 악의적 행위자를 차단하는 스테이킹 시스템에 의존한다. 스테이킹 시스템은 네트워크 참여를 고려하는 악의적 행위자에게 금전적 불이익을 제공하여 강력한 억제력을 발휘한다. 사기적 행위의 위험은 새로운, 알려지지 않은 프로바이더가 네트워크에 참여할 때 가장 높다. 아카시 네트워크는 새로운 계정에 대해 중앙화되거나 연합된 승인 절차를 요구하는 대신 누구나 참여할 수 있도록 허용한다.

새로운 프로바이더가 아카시 네트워크에서 자원을 제공하기로 선택할 때, 승인 절차 대신 반드시 아카시 토큰을 네트워크에 일정 가치 이상 스테이킹해야 한다. 최소 스테이킹 금액은 없지만, 아카시 네트워크 거버넌스 참여도는 전체 스테이크 합계에서 프로바이더의 스테이크가 차지하는 비율에 비례한다. 추가적으로, 스테이킹 기여도는 프로바이더의 평판 점수에 반영되며, 이는 테넌트가 배포 기준으로 사용할 수 있다.

2.2.2 글로벌 결제

아카시 토큰은 일반적으로 국경 간 결제에서 발생하는 환율 위험을 완화한다. 이러한 거래에서 법정화폐를 대체함으로써, 아카시 토큰은 클라우드 인프라 산업에서의 가치 교환을 단순화한다. 우리의 매칭 엔진은 각 컨테이너 컴퓨팅을 시장에서 통용되는 아카시 토큰 양에 따라 경쟁적으로 가격을 책정한다. 테넌트가 프로바이더와 매칭되면, 테넌트는 아카시 토큰을 네트워크에 지불하고, 이후 임대 계약 조건에 따라 프로바이더에게 지급된다.

3 마켓플레이스

아카시에서의 인프라 조달 — 즉, 클라이언트가 프로바이더로부터 인프라를 임대하는 과정 — 은 탈중앙화 거래소(마켓플레이스)를 통해 구현된다.

마켓플레이스는 공개 오더북과 매칭 알고리즘으로 구성된다. 클라이언트는 자신의 서비스 요구 사항이 명시된 **배포(deployment)** 주문을 제출하고, 데이터센터는 배포 주문에 입찰하기 위해 **이행(fulfillment)** 주문을 제출한다. 배포 주문에는 클라이언트가 일정 시간 동안 메모리, CPU, 스토리지, 대역폭으로 측정되는 고정된 컴퓨팅 유닛 수에 대해 지불할 의사가 있는 최대 금액이 포함된다. 이행 주문은 프로바이더가 해당 자원을 제공하기 위해 제시하는 가격을 명시한다.

배포 주문은 클라이언트가 정의한 기간 동안 *열린 상태*로 유지되며, 초 단위로 측정된다. 배포 주문이 열려 있는 동안 프로바이더들은 입찰을 위해 이행 주문을 제출할 수 있다.

이행 주문은 배포 주문의 모든 최소 사양을 충족할 경우 배포 주문과 매칭될 자격이 있다. 배포 주문과 적격 이행 주문 집합이 주어지면, 가장 낮은 가격을 제시한 이행 주문이 배포 주문과 매칭된다. 여러 이행 주문이 매칭 조건을 충족하고 동일한 가격을 제시한 경우, 가장 먼저 제출된 이행 주문이 배포 주문과 매칭된다.

기업 및 개인 소비자들은 자신들이 컴퓨팅 파워를 어떻게 사용하고 있는지 공개적으로 드러나는 방식에 대해 보호하고자 할 것이다. 경쟁사 데이터 마이닝 및 기타 공격 벡터를 방지하기 위해, 동형(homomorphic) 암호화 계층이 추가된다.

배포 주문과 이행 주문이 매칭되면 *임대(lease)*가 생성된다. 임대에는 배포 주문과 이행 주문에 대한 참조가 포함된다. 임대는 배포를 이행하는 데 있어 구속력 있는 요소가 될 것이다.

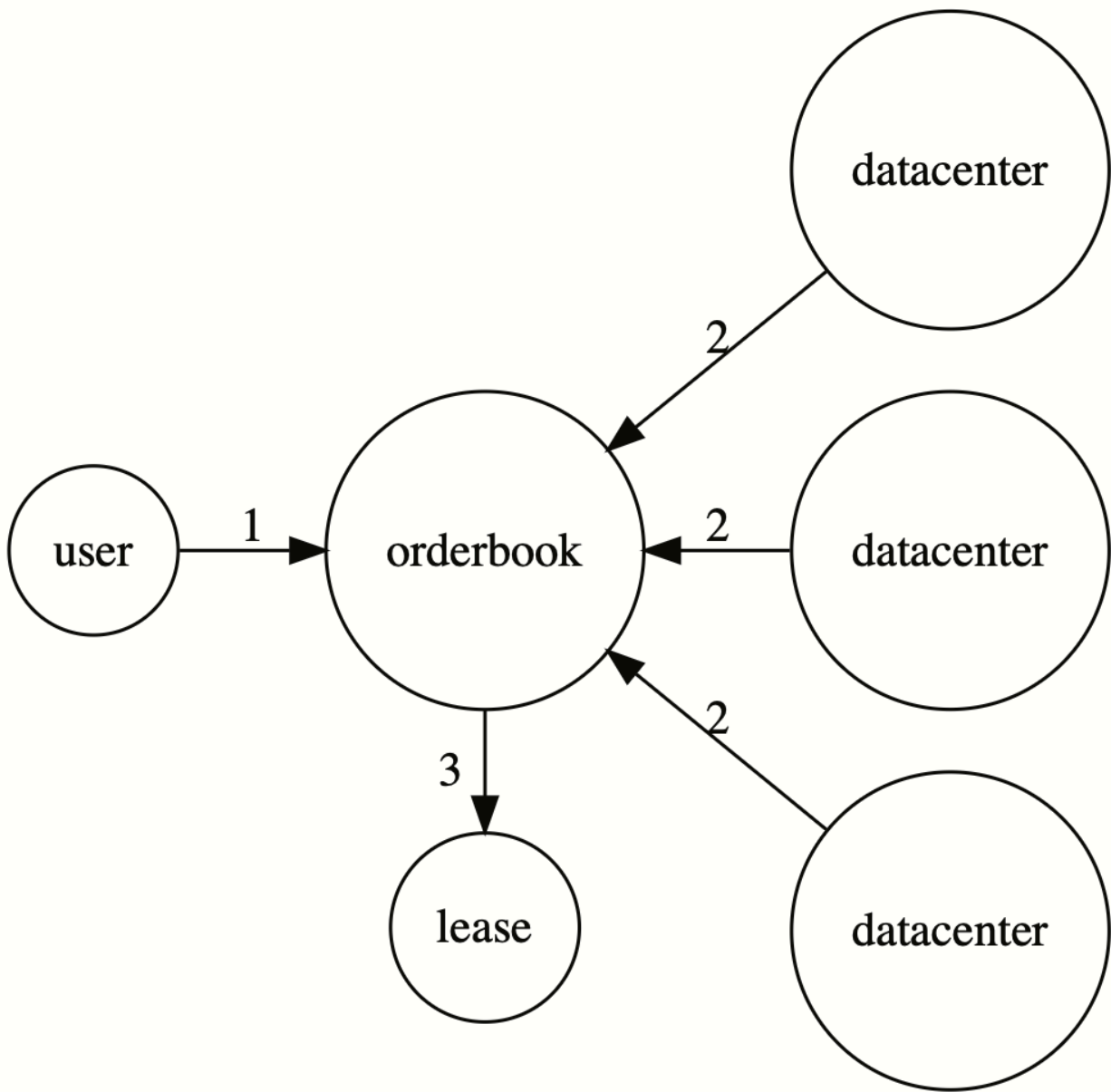


그림 2: 마켓플레이스에서의 조달 과정 요약. (1) 사용자의 배포 주문이 오더북에 게시됨 (2) 데이터센터가 배포 주문에 대해 적합한 이행 주문을 게시 (3) 가장 좋은 이행 주문이 배포 주문과 매칭되어 새로운 임대 생성됨.

4 배포(Deployment)

자원이 조달되면, 클라이언트는 자신의 워크로드를 프로바이더에게 배포하여 임대한 자원에서 실행할 수 있도록 해야 한다. 아카시 네트워크에서 클라이언트의 워크로드가 현재 실행되고 있는 상태를 배포(deployment)라고 부른다.

사용자는 자신이 원하는 배포를 매니페스트(manifest)로 기술한다. 매니페스트는 워크로드 정의, 구성(configuration), 연결 규칙을 포함하는 선언적 파일 형식으로 작성된다. 프로바이더는 워크로드 정의와 구성을 사용하여 자신이 제공하는 자원 위에서 워크로드를 실행하고, 연결 규칙을 사용하여 오버레이 네트워크와 방화벽 구성을 구축한다.

매니페스트의 해시는 배포 버전(version)으로 알려지며, 블록체인 기반 분산 데이터베이스에 저장된다.

4.1 매니페스트 배포

매니페스트에는 배포에 참여하는 이들만 공유해야 하는 민감한 정보가 포함되어 있다. 이는 자기 관리(self-managed) 배포에서 문제를 야기한다. 아카시는 매니페스트의 내용을 불필요한 참여자에게 노출하지 않고, 워크로드 정의를 자율적으로 배포해야 하기 때문이다.

이 문제를 해결하기 위해, 우리는 임대에 참여한 주체들이 필요할 때 서로 매니페스트를 배포하는 피어-투-피어 파일 공유 방식을 고안했다. 이 프로토콜은 TLS 연결을 통한 오프체인 방식으로 실행되며, 각 참여자는 자신이 받은 매니페스트의 해시를 계산하여 블록체인 기반 분산 데이터베이스에 저장된 배포 버전과 비교함으로써 이를 검증할 수 있다.

개인적이고, 안전하며, 자율적인 매니페스트 배포를 제공하는 것 외에도, 피어-투-피어 프로토콜은 대규모 매니페스트를 다수의 데이터센터에 빠르게 배포할 수 있게 한다.

4.2 오버레이 네트워크

기본적으로 워크로드의 네트워크는 격리되어 있으며, 어떤 것도 이에 연결할 수 없다. 이는 보안상 안전하지만, 실제 애플리케이션에는 적합하지 않다. 예를 들어 간단한 웹 애플리케이션을 생각해보자. 최종 사용자의 브라우저는 웹 계층 워크로드에 접근할 수 있어야 하며, 웹 계층은 데이터베이스 워크로드와 통신해야 한다. 또한, 웹 계층은 데이터베이스와 동일한 데이터센터에 호스팅되지 않을 수도 있다.

아카시 네트워크에서 클라이언트는 매니페스트 내에서 연결 토폴로지를 정의함으로써 워크로드 간 및 워크로드와의 통신을 선택적으로 허용할 수 있다. 데이터센터는 이 토폴로지를 사용하여 방화벽 규칙을 구성하고 필요에 따라 개별 워크로드 간의 보안 네트워크를 생성한다.

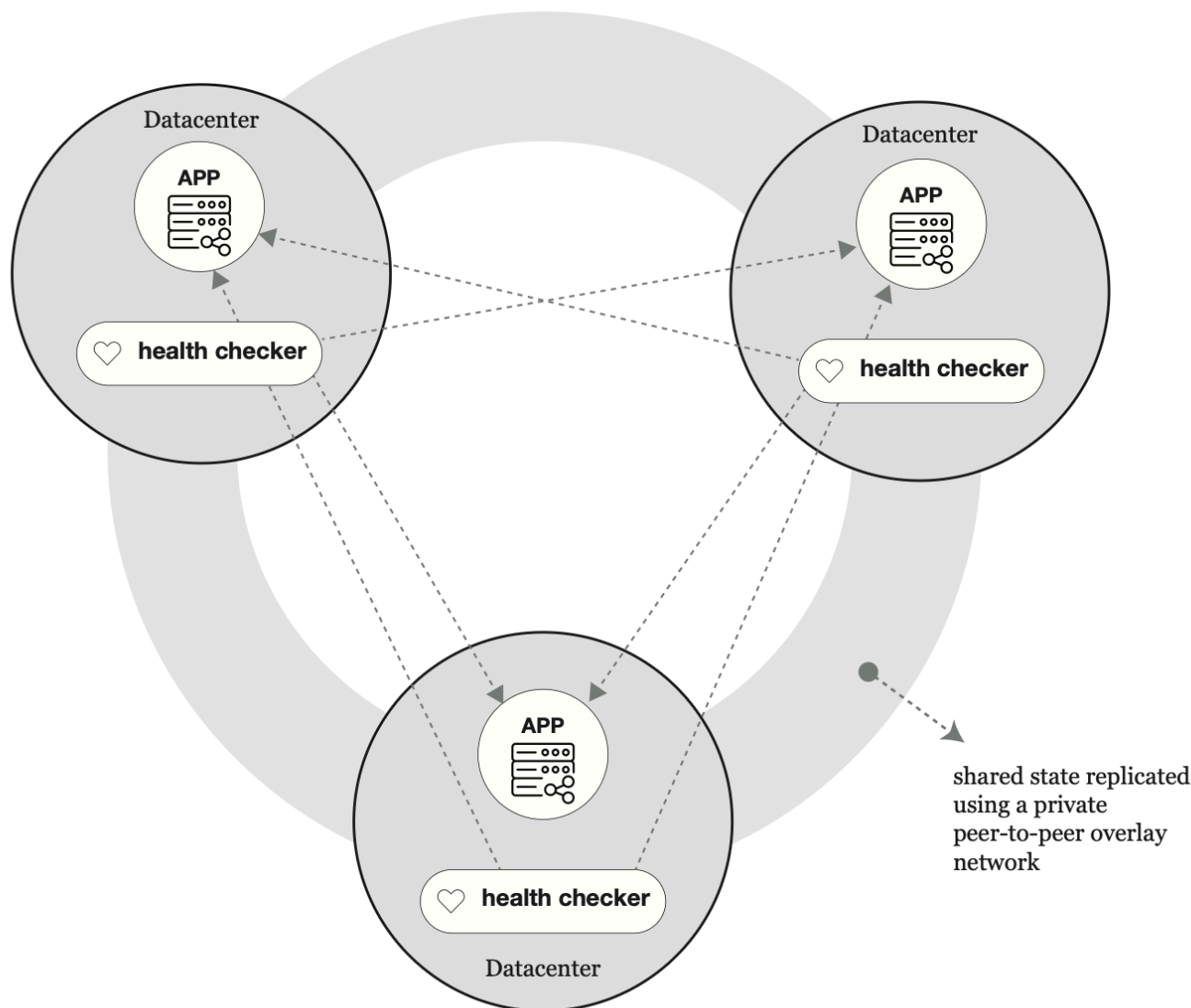


그림 3: 아카시 오버레이 네트워크 그림

보안적인 데이터센터 간 통신을 지원하기 위해, 프로바이더는 mTLS 터널을 통해 워크로드를 서로에게 노출한다. 각 워크로드 간 연결은 고유한 터널을 사용한다.

이러한 터널을 설정하기 전에, 프로바이더는 필요한 각 터널에 대해 TLS 인증서를 생성하고 해당 피어 프로바이더와 이를 교환한다. 각 프로바이더의 루트 인증서는 블록체인 기반 분산 데이터베이스에 저장되어, 피어가 수신한 인증서의 진위 여부를 검증할 수 있도록 한다.

인증서 교환이 완료되면, 프로바이더는 인증된 터널을 설정하고 워크로드의 네트워크를 여기에 연결한다. 이 모든 과정은 워크로드 자체에 대해 투명하게 처리되며, 워크로드는 안정적인 주소와 표준 프로토콜을 통해 서로 통신할 수 있다.

5 자동화

클라우드 인프라의 동적 특성은 운영 관리에 있어 축복이자 저주이다. 새로운 자원을 필요할 때 즉시 프로비저닝할 수 있다는 점은 축복이지만, 폭발적으로 증가하는 관리 오버헤드와 그 자원들의 복잡성은 저주이다. *데브옵스(DevOps)* — 배포를 프로그래밍적으로 관리하는 실천 방식 — 의 목표는 클라우드 인프라의 강점을 활용하여 이러한 고통 지점을 완화하는 것이다.

아카시 네트워크는 데브옵스 엔지니어가 고도로 자동화된 배포를 생성할 수 있는 단순하면서도 강력한 도구 세트를 제공하기 위해 처음부터 설계되었다. 이 도구 세트는 비관리형(non-management) 애플리케이션을 가능하게 하는 기본 요소들 — 범용 워크로드와

오버레이 네트워크 — 로 구성되며, 이를 활용하여 자율적이고 자기 관리(self-managed)되는 시스템을 구축할 수 있다.

아카시에서의 자기 관리형 배포는 스스로 자신의 배포를 관리하는 워크로드를 생성하는 단순한 문제이다. 데브옵스 엔지니어는 프로바이더가 배포에 참여하거나 떠날 때 DNS 엔트리를 갱신하는 워크로드, 웹 계층 애플리케이션의 응답 시간을 테스트하는 워크로드, 그리고 (클라이언트가 정의한 권한과 제약에 따라) 다양한 입력 지표에 기반해 인프라를 확장 또는 축소하는 워크로드를 사용할 수 있다. "관리 계층(management tier)"은 배포를 위해 모든 데이터센터에 분산될 수 있으며, 전역 상태는 보안 오버레이 네트워크를 통해 실행되는 분산 데이터베이스에 의해 유지된다.

5.1 예시: 지연시간 최적화 배포



그림 4: 단일 데이터센터 배포로 인해 전 세계에 분산된 최종 사용자들에게 더 높은 지연시간이 발생하여 성능 저하가 일어나는 그림

많은 웹 기반 애플리케이션은 *지연시간에 민감(latency-sensitive)*하다. 애플리케이션 서버의 응답 시간이 짧을수록 최종 사용자의 경험은 극적으로 향상된다. 이러한 애플리케이션의 현대적 배포는 최종 사용자에게 이미지를 비롯한 정적 콘텐츠를 빠르게 제공하기 위해 *콘텐츠 전송 네트워크(CDN)*를 사용한다.

CDN은 콘텐츠를 지리적으로 사용자와 가까운 곳에 배포함으로써 지연시간을 줄여준다. 아카시 네트워크에서의 배포는 이러한 접근 방식을 단순히 재현할 뿐 아니라 이를 능가할 수 있다. 아카시는 클라이언트에게 애플리케이션 사용자의 가까운 곳에 동적 콘텐츠를 배치할 수 있는 능력을 제공하기 때문이다.

아카시에서 자기 관리형 동적 전송 네트워크(dynamic delivery network)를 구현하기 위해, 데브옵스 엔지니어는 배포에 관리 계층(management tier)을 포함시켜 클라이언트의 지리적 위치를 모니터링할 것이다. 이 관리 계층은



그림 5: 최종 사용자와 가까운 데이터센터에 워크로드와 상태를 동적으로 분산하여 네트워크 성능이 개선되는 그림

전 세계에 걸쳐 데이터센터를 추가하거나 제거하며, 사용자 활동이 높은 지역에서는 더 많은 자원을 프로비저닝하고, 사용자 참여가 낮은 지역에서는 더 적은 자원을 프로비저닝한다.

5.2 예시: 머신러닝 배포

머신러닝 애플리케이션은 대규모 데이터셋을 포함하는 연산을 병렬화하기 위해 다수의 노드를 사용한다. 이들은 "배치(batch)" 단위로 작업을 수행하며, 일정한 상태의 용량(steady state capacity)은 요구되지 않는다.

아카시에서 머신러닝 애플리케이션은 단일 데이터센터 내에서 자원을 선제적으로 조달하기 위해 관리 계층을 사용할 수 있다. 머신러닝 작업이 시작되면 관리 계층은 해당 작업을 위해 노드 수를 "확장(scale up)"할 수 있으며, 작업이 완료되면 이를 위해 프로비저닝된 자원을 해제할 수 있다.

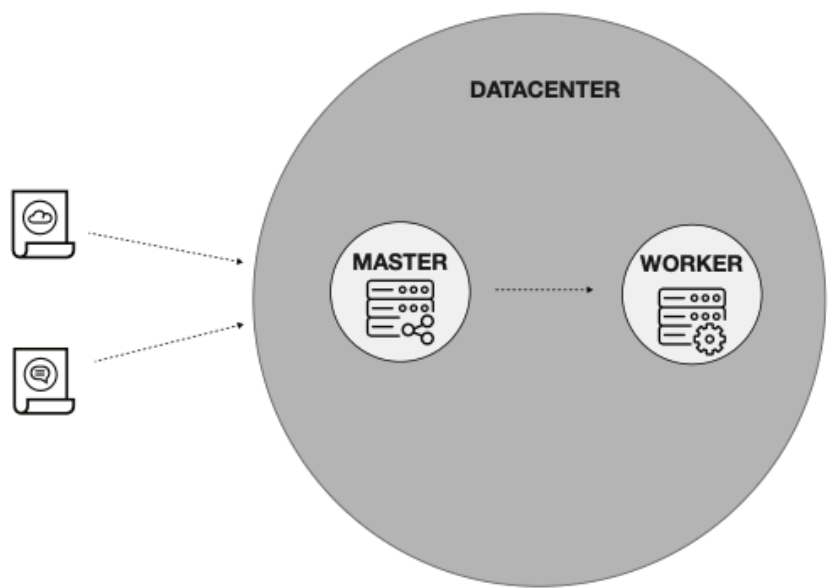


그림 6: 부하가 적을 때 단일 마스터와 단일 워커 노드로 실행되는 머신러닝 배치 작업

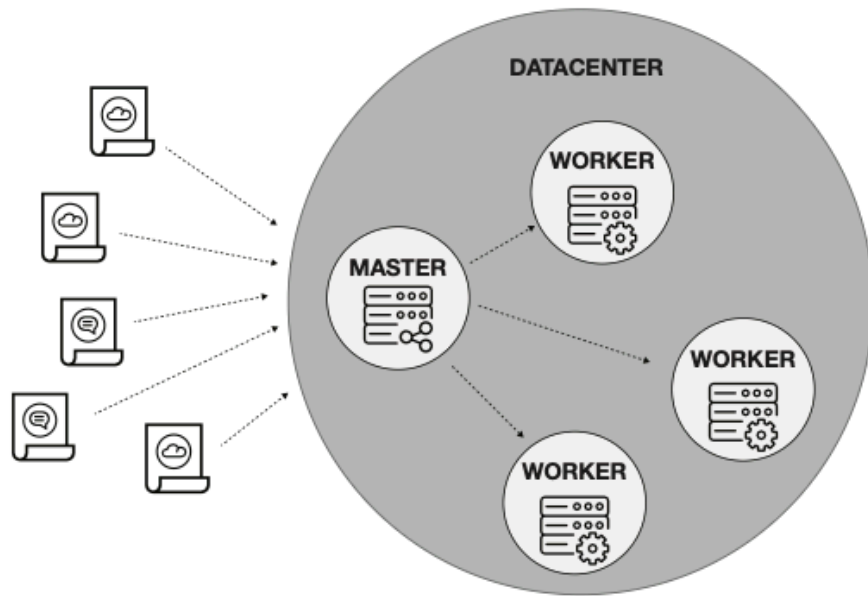


그림 7: 부하가 있을 때 단일 마스터와 다수의 워커 노드로 실행되는 머신러닝 배치 작업

레퍼런스

[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-c](https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-)
[loud-](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](https://www.forrester.com/report/Predictions+2018+Cloud+Computing+Accelerates+Enterpr)
[+Enterpr](https://www.forrester.com/report/Predictions+2018+Cloud+Computing+Accelerates+Enterpr)

[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-ener>

[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.

[Kooimey and Taylor(2015)] Jonathan Kooimey 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_DataSupports30Percent

[Liu(2011)] Huan Liu. A measurement study of server utilization in public clouds. 2011. URL

<http://ieeexplore.ieee.org/document/6118751/>.