Ara — グローバル分散化されたインフラストラクチャはデジタルファイルとコンテンツの支払いを用い、支払い、ユーザー ID、ホスティング、ストリーミング再生、デジタルファイルとコンテンツの所有権（ドラフト）として使用されています

Eric Jiang, Charles Kelly, Joseph Werle, Tony Mugavero, Vanessa Kincaid

最終更新 2018 年 11 月 12 日 (部分的 **)

Abstract

今のインターネットは、その当初の姿が亡霊となってよみがえったかのように、少数の大企業によって支配される状況になってしまいました。これらの大企業が、世界規模の情報の流れ、そのコスト、消費の方法やタイミングを制御して、情報に対する全面的な支配権を行使しています。*Ara* は、このような状況を解消するために生まれた、分散化プラットフォームおよびプロトコルスイートです。*Ara* は、今までにない所有権証明（*Proof-of-Ownership*）システムによるデジタルファイル／コンテンツのライセンス供与および販売を通じて、グローバルなデータ配信をハンドリングするとともに、ネイティブの *Ara* トークンによるこれら資産の購入をサポートします。こうした処理を実行する過程で、*Ara* プラットフォームはユビキタスな分散型のユーザー *ID* およびウォレットシステムも利用します。これにより、ユーザーが個人情報の所有権を持ち続けることが可能になります。*Ara* は実質的に、インターネットにおける情報のホスティングと配信の方法、ならびに消費者による情報の利用とその支払い方法をめぐる、最新のメンタルモデルです。これは企業にとって新しいパラダイムをもたらすだけではありません。ネットワークへの参加とホスティングにより報酬を得て、システムに貢献できるという点で、消費者にも新しいパラダイムをもたらします。ピアツーピア（*P2P*）ファイル共有、所有権とライセンス供与のためのブロックチェーン技術、分散型コンピューティングがすべて組み合され、*1* つの効率的な分散化システムが成立します。

多くの人が *Ara* の恩恵を享受します。消費者は、使用していないストレージ、帯域幅、処理能力を利用して、*Ara* トークンを獲得することができます。このトークンは、*IT* の世界の *Airbnb* のようなもので、これを使ってコンテンツを購入できます。企業は *P2P* テクノロジーを利用して情報を配信することで、コストを節約できます。これは消費者や他の企業のコスト削減にもつながります。誰もがデータセンターとしてネットワークに参加し、報酬を得ることができます。デジタルクリエイター、ゲーム／ソフトウェア開発者、映画／ *TV* スタジオ、出版社は、*Ara* トークンを使ってライセンス取得済みコンテンツをネットワークに公開し、より大きい収益を上げ、その残りをファンにホスティング費用として与えることにより、作品から得られる収益をさらに増やすことができます。要するに、ウィンウィンウィンの関係です。消費者は報酬を獲得し、パブリッシャーは収入を増やし、企業は収益性を改善できます。こうしたメリットはすべて、分散化された完全にニュートラルな方法で達成されます。情報とコンテンツを配信する会社を、仲介会社が抑圧することが一切ありません。

** 本ペーパーは 2018 年 6 月発行のホワイトペーパーを部分的に更新したものです。より完全な最新版を数か月以内にリリース予定です。

注： Ara については活発な研究開発活動が続けられています。本ペーパーに記載された情報は変更される場合があります。最新版はhttps://ara.oneで公開されています。ご意見や提案はhello@ara.oneまでお寄せください。

ディレクトリ

1. はじめに **2**
1.1 背景 . . . 2
1.2 概要 . . . 3
1.3 プラットフォームサービス . . . 3

2. プラットフォームの概要 **5**
2.1 AraID . . . 5
2.1.1 分散化アイデンティティ . . . 5
2.2 分散化コンテンツ配信ネットワーク (DCDN) . . . 6
2.2.1 Ara ファイルシステム (AFS) . . . 6
2.2.2 トークンの用途 . . . 7
2.3 Ara プロトコルスイート . . . 8
2.3.1 報酬とインセンティブ . . . 8
2.3.2 ファイル配信 . . . 9
2.3.3 スマートコントラクト . . . 9

3. 今後の開発 **12**
3.1 モジュール . . . 12
3.2 Ara ネームシステム (ANS) . . . 12

4. 謝辞 **12**

略語 **13**

参考文献 **13**

付録 **15**

I. 成熟した Ara プラットフォームトークンの経済（草稿） **15**
概要 . . . 15
Ara トークン . . . 15
機能 . . . 16
市場力学 . . . 16
インセンティブ構造 . . . 17
ネットワーク効果 . . . 18
参考文献 . . . 18

II. DCDN コスト分析 by Lester Kim **19**
アップローダーの利益の最大化 . . . 19
ディストリビューターのコストの最小化 . . . 21
例 . . . 23
参考文献 . . . 24

III. Ara 報酬予測額の分析 by Lester Kim **26**
ネットワークモデル . . . 26
例 . . . 27
参考文献 . . . 27

1. はじめに

1.1 背景

現在のハイパーメディア環境は、旧態依然としています。情報収集サイトやアプリストアが、コンテンツクリエイターに対する掌握力を強めています。従来型のコンテンツ配信ネットワークは、非効率的なだけでなく高コストです。クラウドコンピューティングは集中化し、少数のゲートキーパーが選ばれるようになりました。データを保存しているのは、そのデータの所有者ではなく、データから利益を得る人々になっています。こうした状況から、コンテンツパブリッシャーおよびクリエイターは価格を上げざるを得ず、低速で割高なシステムのために必要となるコストを、消費者に負担させています。これが、パブリッシャー、消費者、クリエイターにとっての価値が損なわれる結果になっています。

2021 年には、インターネットトラフィック全体の 80% 以上がビデオで占められると予測されています [3]。ファイルサイズも膨張し、コンテンツの配信に必要なコストも高騰を続けています。4K、VR、AAA ゲームも、このようなトレンドを後押ししています。消費者の立場から見れば、トランザクション型コンテンツやサブスクリプションに支払う料金が上がっているだけではありません。無料のコンテンツを視聴するには、複雑で押しつけがましい広告のシステムに対処せざるを得ません。こうした要因は、プライバシーの問題を悪化させ、コンテンツ所有者に何十億ドルもの損失を余儀なくさせています [15][11][4]。また、広告をスキップまたは削除する、広告ブロッカーなどのツールが広まる結果にもなっています。消費者が広告を回避し、コンテンツ所有者が損失を取り戻そうとする動きの中で、広告ブロッカーのブロッカーが出現したり、サブスクリプションサービスの料金がさらに高騰したりしています。まさに悪循環です。

ピアツーピア（P2P）ファイル分散アーキテクチャは、このような非効率性へのアンチテーゼとして出現しました。P2P は、Napster などの集中型サーバーを組み込んだハイブリッドソリューションから、Gnutella、最終的に BitTorrent のような完全な分散化ソリューションに進化しました。現在、P2P ファイル配信は非常にコスト効率に優れているため、Microsoft などの企業も（自社の Azure インフラストラクチャではなく）P2P を利用し、Windows 10 の配信コストを節約しています。

ただし、P2P ファイル共有ネットワークはコスト効率に優れているとはいえ、公衆環境で使用される場合、フリーライド（ただ乗り）、プライバシー、ハッキング、ブラックマーケットなど、悪い噂が絶えないのが現状です。アップロードされる 1 つ 1 つのコンテンツに、果たしてその権利があるのかどうか、信頼性はまったくありません。また、シードされたコンテンツが、コンテンツ所有者の意図した通りに配信されているかどうかを確かめる術もありません。コンテンツを保存・共有する行為への報酬もないため、ユーザーが配信システムにシードを残しておくインセンティブがありませんでした。ピアは往々にして自分でコンテンツを消費するだけで、他のピアに向けてそのコンテンツを広める動きに参加し続けることもありません。この状況を打開するため、P2P アーキテクチャは、インセンティブメカニズムの導入を開始しました。バーター戦略、レピュテーションシステム、独自通貨などです。ところが、これらのメカニズムにも固有の問題があり、Sybil 攻撃やホワイトウォッシング攻撃の標的になっています。

1.2 概要

このホワイトペーパーでは、コミュニティ主導の非集中・分散型コンピューティングおよびコンテンツ配信プラットフォーム、Ara を紹介します。Ara は、世界中のあらゆるデバイスの使われていない処理能力、ストレージ、帯域幅キャパシティを利用することにより、これらのデバイスを、グローバルなスーパーコンピューター、データベース、配信ネットワークに組み込みます。これらのデバイスが、Ara ネットワーク、すなわち誰もが参加してその恩恵を享受することのできるエコシステムを形成します。

このネットワークは基本的に、互いにオーバーラップする消費者、サービスリクエスター、サービスプロバイダー、ソフトウェア開発者のコミュニティで構成されます。これらの人々は、それぞれ独自のインセンティブを求めてこのネットワークを採用します。Ara を利用すると、サービスリクエスターは膨大なコンピューティングリソースと、絶えず拡大する分散型サービスのライブラリを容易に活用できます。スマートフォン、ラップトップ、ゲーム機などのデバイスをすでに所有しているサービスプロバイダーは、未使用のリソースを貸し出すことで、収益化を開始できます。報酬を得るための唯一の要件は、アカウントを有効にするための少額の預入金を送金することだけです。この預入金はいつでも引き出すことができます。ただし、報酬を得て、それと引き換えにする必要があります。ソフトウェア開発者は、前例のないスケールの Ara エコシステムを活用して大量のコンピューティングタスクを実行し、リクエスターとプロバイダーが参加できる斬新な分散型サービスを開発できます。その一方で、消費者は普段通りに日常生活を送りながら、番組を視聴したり好きな音楽を聴いたりすることで報酬が得られます。

このように、コンピューティングリソースが余っている人なら誰でも、すぐにサービス履行者として活動を開始し、コンテンツの配布を支援して報酬を得ることができます。また、リモートのリソースを探している人なら誰でも、Ara の分散化サービスをリクエストして、従来のクラウドコンピューティングサービスと比べれば何分の 1 かのコストで、強化されたセキュリティ、ファイルの可用性、配信速度を利用することができます。Ara には、インフラストラクチャの購入と管理にまつわる負担がありません。そのため、あらゆるタイプのコンテンツクリエイターが恩恵を享受する立場にあります。たとえばインディ系アーティストなら、レコード会社に頼らなくても新譜アルバムを自主出版することができます。大手のメディア複合企業なら、情報収集サイトを経由せずに視聴者をカバーできるようになります。Ara はネットワークのメンバーが提供するリソースに依存します。ネットワークが広がれば広がるほど、堅牢で効率的なネットワークになります。

1.3 プラットフォームサービス

Ara プラットフォームは、次の 3 つのコアサービスおよびシステムで構成されています。

1. **AraID:** AraID　Ara プラットフォーム上のすべてのエージェントおよびコンテンツに関する、安全で分散化された検証可能なグローバルアイデンティティを確立し、データに対する支配権を、そのデータの正当な所有者が行使できるようにします。

2. **分散化コンテンツ配信ネットワーク (DCDN):** DCDN Ara の基盤となるピアツーピアの安全な分散型ファイルシステムおよびストレージネットワーク（Ara ファイルシステム：AFSs）の役割を果たし、コンテンツの完全性、インセンティブ、バージョン管理、分散化アイデンティティをサポートします。
3. **プロトコルスイート:** Ara はセキュアなプロトコルスイートを通じて接続されます。これらのプロトコルによって、DCDN、AraID、イーサリアムブロックチェーンの間のトラストレスな相互運用性が実現されます。

2. プラットフォームの概要

コンフリクトフリーファイルシステムネットワークCFSNetは、Ara のピアツーピア分散型ファイルシステム、AraID、DCDNのバックボーンです。基盤となるマークルツリー構造 [1] および対象イベント同期可能レジャープロトコルSLEEPファイル形式を活用するCFSNetは、従来のファイル転送（クライアント／サーバーと P2P の両方）をめぐる多くの懸念事項を解決し、暗号により保証されたコンテンツ完全性のほか、バージョン管理やリビジョン履歴を提供することで、IPFS など既存のテクノロジーをさらに改良しています。このネットワークは、CFSと呼ばれる一連の分離したファイルシステムで成り立っています。さらに、各CFSインスタンスはファイルシステム階層標準FHSのサブセットを実装します [8]。FHSはパーティションをサポートし、各ディレクトリが独自のアクセスレベルを持った自己完結的なCFSアーカイブとして存在することが可能です。これらのパーティションの中で、AFSは`//home` および`//etc` パーティションを使用し、それぞれAFSコンテンツおよびメタデータを保存します。各CFSパーティションは、作成時に生成される一意の`Ed25519` 32 バイトパブリックキーを使用して、ネットワーク全体でパブリックに識別可能です。CFSのパブリックキーは、ファイルシステムへの読み取り専用アクセスを許可します。ファイルシステムに含まれるコンテンツの更新と公開は、プライベートキーの所有者にしか認められません。

2.1 AraID

AraID は、Ara プラットフォーム上のすべてのユーザーおよびコンテンツに対応する、安全で検証可能な分散化表記を作成および解決する役割を果たします。W3C の分散識別子DID[14] 仕様に完全準拠する AraID は、DID記述子オブジェクトDDOを使用してユーザーおよびコンテンツを表記します (図 *1* を参照)。DDOは、サービスエンドポイントや所有者が制御するプライベートコミュニケーションチャネルを含めて、認証と承認の方法およびその他のアイデンティティ属性を定義する、シンプルなJSON-LD ドキュメントです [14]。DDOには個人識別可能情報PII[14] が保存されないため、これらのサービスエンドポイントおよびコミュニケーションチャネルは、PII を取得するための安全な方法を識別します。そのため、プライベートデータやオンラインアイデンティティに対する、エンティティの自己主権が認められます。

2.1.1 分散化アイデンティティ

Ara プラットフォーム上のすべてのユーザーおよびコンテンツに対し、次のような形式の AraID が生成されます：

- `did:ara:ee93189c629cdaf94 9fd57bac5b005b916936d2a5c6806 40fd1aedc8315730a0`

AraID は、分散アイデンティティ基盤システムの一部分として、DID（上記の `ara`）の 2 番目のコンポーネントによって表されるユニバーサルパーサー `method` 実装しています。この `method` はドライバーとも呼ばれ、Ara プラットフォームにおけるDIDおよびDDOの解決方法を定義します。DIDは、インターネット URI とは違って一元的な登録機関や管理を必要とせず、TCP/IP やDNSに見られるような非インジェクティブ（非単射）、非サージェクティブ（非全射）の関係ではなく、DDOとバイジェクティブ（全単射）な対応関係を形成します。

AraID セキュリティという最も重要な問題は、分散パブリックキーインフラストラクチャーDPKI[12] を使用して暗号により保守されます。ここでは `Ed25519` パブリックキーが、DIDの `id` 部分（上記の `ee9318`）、および対応するDDOが保存されているCFSのパブリックキーの両方として使用されます。これらのドキュメントに含まれる `publicKey` プロパティに、デジタルシグネチャ、暗号化、およびその他の暗号処理に使用される各種のキーが入っています。アイデンティティが作成される時点で、所有者アイデンティティのキー、および対応するイーサリアムアカウントのパブリックキーが、この配列に書き込まれます。

AFS AraID の場合は、関連するコンテンツメタデータを含む`//etc` パーティションのパブ

リックキーも保存されます。このキーはAFS DDOに保存されるので、AFS DIDを持っているリクエスターなら誰でも解決可能です。

新しいアイデンティティが生成される時点で、1 つのニーモニックフレーズを使用してキーペアがシードされます。このニーモニックは所有者が保管し、プライベートキーを簡単に保守できます。エンティティはDIDの所有権を容易に確認することができ、アカウントを復元するのにプライベートキーは不要です。

アイデンティティのアーカイブと解決

アイデンティティが作成されると、そのアイデンティティはまずローカルに書き込まれます。したがってローカルな解決では、キャッシュをチェックしてから、ネットワークにフォールバックできます。これに対し、アイデンティティをリモートで解決するには、最初にそのアイデンティティをアーカイブする必要があります。Ara ではアーカイバーノードが動作します。このノードの役割は、今後の解決に備えて、これらのアイデンティティを保存することです。

アーカイバーノードと同様、Ara ではリゾルバーノードも動作します。このノードは、要求されたDDOについてアーカイバーに問い合わせます。リゾルバー要求は、まずディスクに保存されている可能性のあるアイデンティティでローカルな解決を試み、その後、問題の AraID をアーカイブしているネットワーク上のリモートアーカイブに到達します。

```
{
    'ddo': {
        '@context': 'https://w3id.org/did/v1',
        'id': 'did:ara:ee9318...',
        'authentication;: [{
           'type': 'Ed25519SignatureAuthentication2018',
           'publicKey': 'did:ara:ee9318...#owner'
        }],
        'publicKey': [{
           'id': 'did:ara:ee9318...#eth',
           'type': 'Secp256k1VerificationKey2018',
           'owner': 'did:ara:ee9318...,
           'publicKeyBase58': 'H3C2AVvLMv6gmMNam...'
        }],
        'service': {
           'ens': 'https://etherscan.io/enslookup',
        }
        ...
    }
}
```

伝説 1: *DDO* の例

イーサリアムアカウント

各アイデンティティは、それぞれ 1 つのイーサリアムアカウントおよび対応するイーサリアムウォレットとともに作成されます。これらは、アイデンティティの作成時に生成されるランダムなニーモニックを使って復元可能です。イーサリアムアカウントおよびアイデンティティそれ自体が、このニーモニックを使って確定的に作成されるので、ユーザーはこのニーモニックだけを使用して、イーサリアムアカウントおよびウォレットを含む完全なアイデンティティを復元できます。

AraID は、パブリックキー暗号化で裏付けられた任意のアカウントをサポートするよう設計されています。したがって、AraID でサポートされる暗号通貨アカウントのタイプは、特定の通貨に限定されません。どんな暗号通貨にも、簡単に関連付けることができます。

2.2 分散化コンテンツ配信ネットワーク (DCDN)

DCDNは、拡張性の高い非集中型のハイパーメディアおよびデジタル資産の分散を目的とした Ara のソリューションです。DCDNのコアは、コンテンツとそれに関連付けられたメタデータを収容するCFS実装、Ara ファイルシステム（AFSs）のネットワークで構成されています。

2.2.1 Ara ファイルシステム (AFS)

AFSは、Ara 固有のニーズとゴールを達成するように作られたCFSのフレーバーです。AFSでは、CFSで実装される 2 つの既存のパーティション、`//home` および`//etc` パーティションを活用します。これらのパーティション（FHS [8] のサブセットとして実装）は、それぞれ生のバイナリデータおよびコンテンツを保存しています。`//home` パーティションは、ユーザーがAFSのコンテンツを購入するか、アクセスを許可された場合にのみアクセス可能です。一方、メタデータを含む`//etc` パーティションは、コンテンツの所有権とは無関係にアクセス可能です。AFS有者は、リクエスターによるメタデータの解析を可能にするため、メタデ

ータに関するスキーマを定義することができます。メタデータの構造化方法については、プロトコルによって厳密な標準が強制されるわけではありません。ただし、分散化サービス間の相互運用性を最も適切にサポートするための既存のパラダイムに従って、Schema.orgを基準として推奨します。

AFSを初めて作成した時点で、BIP39[5] ランダム 12 ワードのニーモニックフレーズを使用して AraID が作成されます。生成される DID は、AFS のパブリックキーとして使用します。対応するDDOの `authentication` プロパティが修正され、所有者のDIDが含まれるようになります。これにより、所有者のDIDを解決することで、AFSの所有者を判別できます。

AFSは、システムに導入するコンテンツごとに作成されます。これは映画のように 1 つのファイルの場合もあれば、ゲームのようにファイル集合の場合もあります。コンテンツの所有者を暗号により検証するには、2 組のバイトバッファをイーサリアムブロックチェーンに書き込みます。最初の組は `metadata.tree` エントリであり、データストレージ層に含まれるデータのシリアライズされたマークルツリーを表します。2 番目の組は、シリアライズされたツリーのルートノードのシグネチャを含む `metadata.signatures` ファイルです。

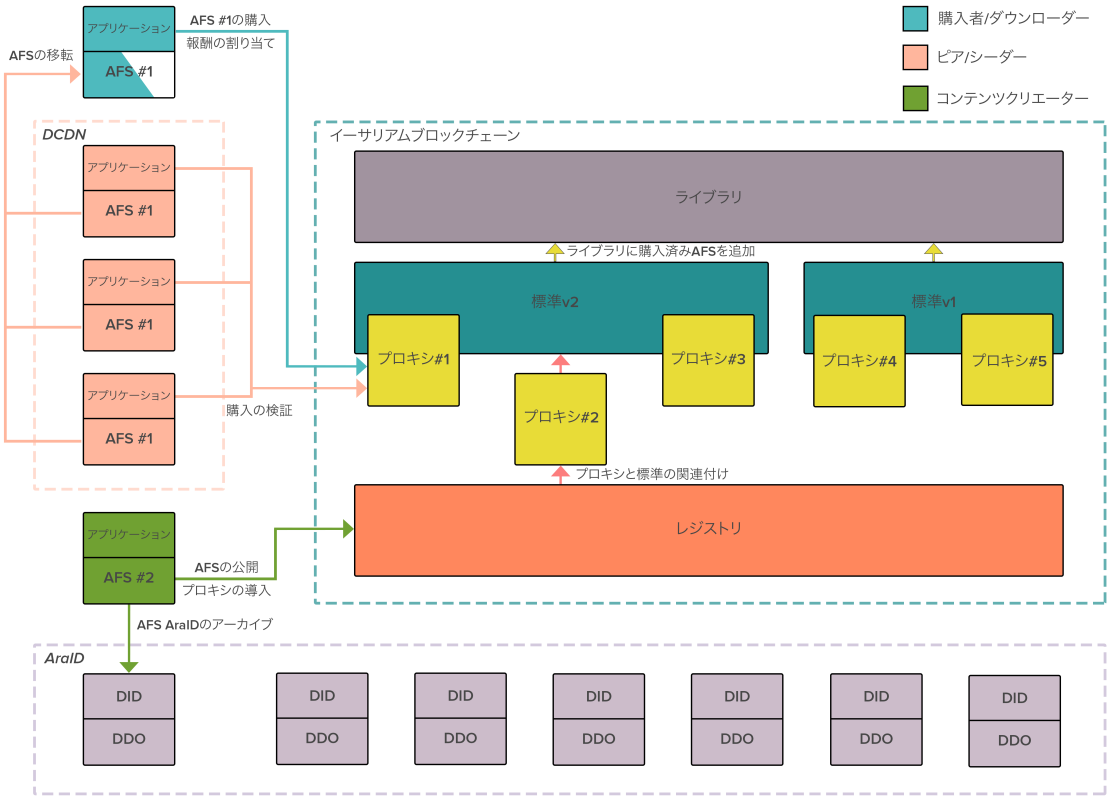
2.2.2 トークンの用途

初期の段階では、Ara トークンを所有している場合、DCDN に対して次のことが認められます。

1. ネットワークに含まれるコンテンツの購入とダウンロードが可能。
2. P2P ファイル配信に参加し、預入金として Ara デポジットを送信することにより、任意のコンテンツで報酬を得ることが可能。

2.3 Ara プロトコルスイート

以下の各節では、このプラットフォームのコアプロトコルを定義し、システムの各部分について詳しく説明し、これらの部分どうしの相互運用性について説明します。



伝説 2: *Ara* プロトコルの図解

2.3.1 報酬とインセンティブ

BitTorrent など、従来のピアツーピアファイル共有システムは、人々の利他的な行動に依存しています [7]。ネットワークから自分がダウンロードするのと同じくらい多くの情報をアップロードするようピアに促すような、効果的なインセンティブに欠けていました。その結果、リーチャー（ダウンロードした分散化ファイルを利用する人々）が、スウォーム（分散化ファイルのダウンロードとアップロードを行うすべてのピアの群れ）の中で多数派を占めるようになり、不均衡が生じています。この不均衡は、フリーライド（ただ乗り）問題の一形式 [6] であり、健全なネットワークにとって望ましいものではありません。リーチャーは多くの場合、ネットワークから恩恵を受けるばかりで何も見返りを与えないため、他の人々が犠牲を強いられます。Ara は、ネットワークのこのような非効率性を緩和するため、ダウンロードが行われるたびに、そのダウンロードをシードしたピアに対する報酬として、少額のコストを適用することで、インセンティブメカニズムとしての報奨システムを実現します。決済モデルによっては、このコストを報酬の割り当て分としてコンテンツの総コストに組み込むことができます。「無料」コンテンツの場合は、従来の広告やデータ収集（現時点で「無料」のコンテンツに対するユーザーの支払い方法）を、このコスト

で置き換えることが可能です。ダウンロードのたびに報酬が発生するので、時間が経つにつれてネットワーク参加者が受け取る報酬が増え、その後、ダウンロードのための前払いが行われます。

2.3.2 ファイル配信

ファイル配信は、Ara のコンテンツ配信ネットワークを支え、参加者が報奨を得るための主要なメカニズムです。Ara のファイル配信プロトコルは、4 ステップのハンドシェークから始まります。

1. コンテンツ要求者である Alice が、何らかのネットワーク検出プロトコル（CFSNet では、mDNS や BitTorrent などいくつかの戦略が実装されています）を使用して、あるコンテンツに対するダウンロード要求をブロードキャストします。
2. ライセンス検証者／コンテンツ配信者である Bob が、このブロードキャストを受信し、ファイルが使用可能である旨を応答します。
3. Alice は応答のプール（スウォーム）の中からピアを選択し、自分の DID とともに中間パブリックキーを含むメッセージを送信します。
4. Bob は Alice のメッセージを暗号により検証し、基の AFS のライセンスを彼女が購入済みであることを確認します。

ハンドシェークが完了すると、ファイル転送が開始されます。

2.3.3 スマートコントラクト

初期の段階で、Ara はイーサリアムメインネット上で運用が開始されます。Ara のプロトコルスイートで中心的なコンポーネントの役割を果たすのが、スマートコントラクトです。これらのスマートコントラクトは、DCDN、AraID、アプリケーション層の相互運用を仲介し、次の情報を含むプラットフォームのすべての非過渡的なプロパティおよびエンティティが、イーサリアムブロックチェーンに確実に登録されるようにします。

- 公開コンテンツ
- 購入
- 報酬
- Ara 残高

Ara のスマートコントラクトアーキテクチャは、セキュリティと更新性を目標に設計されています。AFS の売買、報酬のハンドリングと分散、支払いの処理とシステム内でのルーティングに関する発展的な概念をサポートするために、Ara では公開される AFS ごとにプロキシコントラクトを導入します。プロキシはレジストリコントラクトを通じて導入され、特定のバージョンの `AFS` 標準に関連付けられます。`AFS` 標準は AFS のビジネスロジックを定義します。1 つ 1 つの AFS に `AFS` 標準全体を導入すると、コストが高くつき、更新が難しくなります。AFS は基本的に特定の標準に固定されます。プロキシアーキテクチャによって、1 つの AFS の寿命全体にわたって 1 つのプロキシを導入することが可能になり、そのプロキシが参照する `AFS` 標準のバージョンを変えることで更新できます。プロキシアーキテクチャによって、登録済みのプロキシアドレス（すなわち、有効な AFS コンテンツ）でなければ、ライブラリコントラクトでユーザーライブラリに追加できないことも保証されます。

AFS 標準

AFS は `AFS` 標準によって、イーサリアムブロックチェーンにおける定義済みの、構造化された、自己完結的なプレゼンスを確保できます。この標準には、購入と報酬に関するメソッドのほか、ツリーおよびシグネチャファイルをメタデータ SLEEP レジスタから保存するためのメソッドも含まれます。メタデータ SLEEP レジスタは、AFS に含まれるコンテンツに関するメタデータ（ファイル名、サイズ、権限など）を保存します。これに対し、コンテンツ SLEEP レジスタは、ファイルの生バイナリコンテンツを保存します。メタデータレジスタの内部で、ツリーファイルは、コンテンツレジスタ内のデータを構成するシリアライズされたマークルツリーを表します。シグネチャファイルは、シリアライズされたツリーの署名付きルートを保存します。CFS の場合、これらのファイルはディスクに保存されディスクから読み取られますが、AFS では、

イーサリアムブロックチェーンに書き込まれ、イーサリアムブロックチェーンから読み取られます。AFS は自分自身のプロキシを通じてこの標準とやり取りするので、多くの異なる AFS 標準が共存する可能性があります。コンテンツクリエイターは、各自のニーズに最も合った標準を選ぶことができます。

プロキシの使用によってロジックがストレージから分離され、AFS 標準がその標準バージョンを使用するすべての AFS のロジック層としての役割を果たし、各プロキシが 1 つの AFS のストレージ層としての役割を果たします。最低でも AFS 標準は、価格、購入、報酬、ストレージの機能実装を強制する AFS 標準抽象クラスを実装する必要があります。

最も基本的な（デフォルトの）AFS 標準の場合、価格は AFS の所有者だけが変更できます。購入時には、この価格が購入者の Ara ウォレットから所有者の Ara ウォレットに転送されます。基本の texttttAFS 標準は、報酬予算に関するサポートを強制的に実施します。報酬予算はダウンロードに先立って送信する必要があります。ダウンロードが完了すると、参加しているピアの間で予算が割り振られます。その後、ピアは参加に必要な残高の預入金から引き出していない報酬を使うことができます。

AFS 標準では、コンテンツクリエイターの自由裁量により、非常に多くのカスタマイズ可能なコマース制御もサポートされます。

1. **ロイヤルティ**: 購入による収益を、多くの異なる Ara アカウントの間で、パーセント単位で分散するようにカスタマイズできます。
2. **大量購入**: 購入数量に基づく段階別の価格を設定できます。
3. **再販条件**: 購入したコンテンツを、コンテンツクリエイターが指定する最低再販価格で、複数回にわたって再販することができます。
4. **所有権の移転**: AFS の所有者は、別のイーサリアムアドレスに所有権を簡単に移転できます。
5. **事前注文**: ダウンロード可能になる前に、コンテンツを購入できます。購入者は報酬予算を事前に送信するので、使用可能になった時点ですぐに AFS のダウンロードを開始できます。
6. **希少性**: コンテンツクリエイターは、AFS の最大販売回数を定義できます。この回数が終わると、その AFS はリストから消えて購入不可になります。

これらのコマース制御によって、あらゆるタイプのコンテンツクリエイターが、各自のニーズに合った独特のビジネスモデルや収益モデルを、正確な仕様通りに定義することが可能になります。従来の手段では実施するのが難しい、興味深い新モデルを、さまざまな制御を組み合わせて作り上げることができます。たとえば、音楽のリミックスを楽しんでいる人の場合、一定の再販条件と最低再販価格を設定されたオリジナル曲のトラックをアーティストから購入し、曲をリミックスして、リミックスバージョンを販売しながらアーティストに支払いを続けることができます。従来、このようなタイプの制御を組み合わせて使うには、途方もなく長い時間と、莫大な資金、法律専門家の関与が必要となり、小規模なコンテンツクリエイターにとっては参入障壁が高すぎました。しかし、これからは誰でも無料で利用可能です。

レジストリ

レジストリコントラクトは、プロキシアーキテクチャの一部分として、主に次の 2 つの機能を提供します。

1. プロキシの工場としての役割を果たします。
2. すべてのバージョンの AFS 標準を追跡します。

AFSを最初に公開する時点で、レジストリによってそのAFSのプロキシが導入され、指定の AFS 標準とAFSとの関係が確立されます。プロキシは、コールがあった時点で該当する AFS 標準のアドレスをレジストリに問い合わせ、そのアドレスに委任してコールを処理し、結果がプロキシに返されます。

ライブラリ

Ara ネットワークは、ユーザーが購入またはその他の方法によってアクセス権を持つAFSについて、ライブラリコントラクトを利用して正統な真実の情報源を作成します。コンテンツを購入した時点で、`AFS` 標準の購入機能により、ライブラリコントラクトに含まれる購入者のライブラリにAFSs DIDが自動的に追加されます。これにより、ユーザーのライブラリに関する情報を必要とするサービスは、コントラクトでその情報を照会することができます。ブロックチェーンに保存されたAFS DIDにより、どのサービスでも基のコンテンツを解決できます。ライブラリによって、登録済みのプロキシでなければ、対応するAFSをユーザーのライブラリに追加できないことが保証されます。そのため、どのユーザーも他のユーザーのライブラリを無断で改ざんすることはできません。

3. 今後の開発

3.1 モジュール

Ara プラットフォームの最上部で実行できる分散型サービスは、基本的に、特定のタイプのサービスに限定されません。モジュールは、モジュール API を実装する非集中／分散型サービスであり、このプラットフォーム上で相互に置き換え可能な形で使用することができます。ERC-20 トークン標準によってイーサリアムのトークンが他のアプリケーションでも再利用できるのと同じように [13]、モジュール API により、このインターフェイスを実装しているすべての分散型サービスが、プラットフォーム全体で使用可能になります。そのため、Ara の報酬、購入、支払いシステムを利用する能力のある分散型サービスのエコシステムの構築を専門とする、デベロッパーコミュニティを容易に形成できます。これは本質的に、報酬が得られる分散型サービスエコノミーの成立を意味します。モジュールで報酬システムの利用を希望する場合、そのモジュール独自の報酬メカニズムや手法を定義する、スマートコントラクト API の追加実装も必要になります。各モジュールのスマートコントラクトは、分散型サービスの使用を通じて蓄積される報酬の独自の割り当て分を保存し、コントラクトに従ってその報酬を分散します。

3.2 Ara ネームシステム (ANS)

ANSは、ドメインネームシステムDNS [9] と同じように、Ara ネットワーク上で証明書を登録、照会、有効化、無効化するための分散化された方式です。DNSがインターネットのアプリケーション層の一部分に存在し、人間が判読できる URI を基の IP アドレスに解決して、ユーザーエージェント（例：Web ブラウザ）が要求されたコンテンツを取得およびレンダリングできるようにするのに対し、ANSは、人間が判読できる名前をDIDに解決し、最終的にDDOに解決します。ANSは、2 番目の解決フェーズのために、アイデンティティアーカイバーとリゾルバーを内部で使用し、DIDからDDOを提供します。ANSは基本的に、人間が判別できる URI のアーカイバーでありリゾルバーでもあります。Ara の最上部にある Web ブラウザーのコンテキストにおいて、ホスト名からDDOを提供するのは、ANSの応用例の 1 つです。

各種のレコードタイプを識別するため、DNSが独自のレコードを分類するのと同様、各レコードに `TYPE` リソースレコードが保存されます [16]。`TYPE` フィールドは数値で表されます。したがって今後、さらに別のタイプのレコードをANSに保存することが可能です。以下の表で、レコード `TYPE` について説明します：

TYPE	Value	Description
USR	00	User
PCT	01	Published Content

ANSを構成する各スーパーノードハブは、HyperDB インスタンスを実行します [2]。HyperDB のように分散型で非常にスケーラブルなデータベースは、ANSのようなシステムに適した、いくつかの機能が備わっています。第一は、HyperDB におけるツリーの使用です。各ノードがその子ノードのプレフィックスとなっている検索ツリーです。ツリーに名前を保存することで、データベースに何千ものエントリが存在しても、迅速で低コストのルックアップが可能です。ツリーにおけるルックアップは `O(n)` で、`n` は検索対象のキーの長さです。HyperDB は、ベクタークロックも利用します。ベクタークロックは、分散型システムにおけるイベントの因果関係を追跡し、ノードが脱同期になる事態を防止します [10]。

4. 謝辞

このペーパーは、Littlstarおよびtoken Foundryの協力によって制作されました。Logan Dwight、Andrew Grathwohl、Brandon Plaster の各氏に特に深く感謝します。

略語

AFS Ara File System. 4–7, 10, 11

ANS Ara Name System. 12

CFS Conflict-Free File System. 5, 6

CFSNet Conflict-Free File System Network. 5

DCDN Decentralized Content Delivery Network. 4–6

DDO DID Descriptor Object. 5–7, 12

DID Decentralized Identifier. 5–7, 11, 12

DNS Domain Name System. 5, 12

DPKI Decentralized Public Key Infrastructure. 5

FHS Filesystem Hierarchy Standard. 5, 6

PII Personally-Identifiable Information. 5

SLEEP Syncable Ledger of Exact Events Protocol. 5

参考文献

[1] Code for Science Buus, Ogden. Sleep - syncable ledger of exact events protocol. https://github.com/datproject/docs/blob/master/papers/sleep.pdf, Aug 2017.

[2] Mathias Buus. Hyperdb. https://github.com/mafintosh/hyperdb, Aug 2017.

[3] Cisco. Cisco visual networking index predicts global annual ip traffic to exceed three zettabytes by 2021. https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1853168, Jun 2017.

[4] Stewart Clarke. Piracy set to cost streaming players more than $50 billion, study says. http://variety.com/2017/tv/news/piracy-cost-streaming-players-over-50-billion-1202602184/, Oct 2017.

[5] Palatinus et al. Mnemonic code for generating deterministic keys. https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki, Sept 2013.

[6] Russell Hardin. The free rider problem. https://plato.stanford.edu/entries/free-rider, May 2003.

[7] Ahamad Jun. Incentives in bittorrent induce free riding. https://disco.ethz.ch/courses/ws0506/seminar/papers/freeriding_incentives.pdf, Aug 2005.

[8] The Linux Foundation LSB Workgroup. Filesystem hierarchy standard. https://refspecs.linuxfoundation.org/FHS_3.0/fhs-3.0.pdf, Mar 2015.

[9] Paul Mockapetris. Domain names - implementation and specification. https://tools.ietf.org/html/rfc1035, Nov 1987.

[10] Multiple. Vector clock. https://en.wikipedia.org/wiki/Vector_clock.

[11] Stephen E. Siwek. The true cost of sound recording piracy in the us economy. https://www.riaa.com/wp-content/uploads/2015/09/20120515_SoundRecordingPiracy.pdf, Aug 2007.

[12] Rebooting the Web-of Trust. Decentralized public key infrastructure. http://www.weboftrust.info/downloads/dpki.pdf, Dec 2015. Accessed on 2018-04-19.

[13] Buterin Vogelsteller. Erc-20 token standard. https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md, Nov 2015.

[14] W3C. Decentralized identifiers. https://w3c-ccg.github.io/did-spec, Apr 2018. Accessed on 2018-05-01.

[15] Music Business Worldwide. Why does the riaa hate torrent sites so much? https://www.musicbusinessworldwide.com/why-does-the-riaa-hate-torrent-sites-so-much/, Dec 2014.

[16] Inc ZyTrax. Dns resource records (rrs). http://www.zytrax.com/books/dns/ch8/, Oct 2015.

付録

I. 成熟した Ara プラットフォームトークンの経済（草稿）

概要

従来型のクラウドサービスは、社内でインフラストラクチャを購入して管理するのと比べて、柔軟性、俊敏性、コスト節約効果に優れていることから、脚光を浴びるようになりました。多くのクラウドサービスが、長期のコミットメントと契約を組み込んだ（顧客単位で個別に交渉することも多い）、複雑であいまいな価格モデルを使用しているのは、よく知られているところです。このような価格モデルは、クラウドサービスの本来の利点とされる柔軟性や俊敏性を、台無しにするものです。[1] 近年、P2P CDN が勢いを増してきました。新興の SaaS が、ビデオ配信のためのハイブリッドソリューションを自称しています。これらの SaaS は、ビデオ視聴者のマシンを活用することで、よりシンプルな価格モデルの、非常にスケーブルなソリューションを確立しています。しかし、集中化の根本的な原因をいくつか残しています。具体的にいうと、ユーザーを「二級市民」と見なすビジネスモデルに原因があります。ユーザーから同意を得ることもなく、金銭的な対価も与えずに、ユーザーのマシンを利用しているからです。Ara では、さらに一歩進んで、マシンの利用に対してネットワーク参加者に報奨が与えられます。

既存の従来型クラウドインフラストラクチャのコストに取って代わるために、Ara プラットフォームは、ネイティブのプロトコルユーティリティトークン、Ara トークンを実装しています。Ara プラットフォームでは、このトークンを使用して、健全で正直なネットワーク行動に対する暗号通貨経済的なインセンティブを創出し、コンテンツ消費者とクリエイターの間に、より直接的なエンゲージメントを実現するとともに、プラットフォームの採用を促進します。Ara トークンは、ネットワークの参加者が提供する価値をカプセル化したものと見なすことができます。報酬としてトークンを得るたびに、ネットワークの有用性が少しずつ増えることになります。

Ara トークン

Ara の SDK を使って開発した分散型サービスは、モジュールと呼ばれ、Ara ネットワーク上で購入、販売、要求、履行が可能です。モジュールのタスクは、ネットワークのピアにアウトソーシングされます。ピアがそのタスクを正常に完遂すると、Ara トークンで報酬を受け取ります。開放的で競争のある市場に発展させるために、Ara では、サービスリクエスターは自分が要求するサービスに対する報酬の割り当て（報奨金）を定義することができ、サービスプロバイダーは自分が了承する最小限の報奨金を定義することができます。人間の知性を必要とするクラウドソーシングタスクのマーケットプレイスである Amazon Mechanical Turk と同じように、Ara は、分散型コンピューティングとネットワーキングタスクをアウトソーシングするためのマーケットプレイスを成立させます。各モジュールは、分散型トランスコーディングのような 1 回限り／オンデマンドのタスク

の場合もあれば、P2P マルチプレイヤーゲームサーバーのような継続的／反復的サービスの場合もあります。

機能

このネットワーク全体にわたって、Ara トークンはさまざまな方法で使用可能です。

- 消費者の場合、娯楽用のデジタルコンテンツから、新規モジュールへの参加まで、あらゆる種類の購入を Ara トークンで行うことができます。
- サービスリクエスターは、Ara トークンを使ってジョブ要求を開始し、そのジョブを正常に完遂した場合の報奨金を設定することができます。
- サービスプロバイダーは、報酬と引き換えにタスクを履行するコミットとして、Ara トークンを預け入れることができます（この預入金は要求あり次第、返金されますが、その場合、プロバイダーは報酬を得ることはできなくなります)。
- 開発者は Ara トークンを使って、ネットワークに新しいモジュールを導入できます。

これらの人々の役割は、いずれも互いに大きくオーバーラップしています。サービスプロバイダーは、タスクを履行した報酬として得た Ara トークンを使って新しいコンテンツを購入できます。開発者も、モジュールの購入によって得た Ara トークンを使って新しいジョブ要求を開始できます。

市場力学

ネットワークのメンバーは、完全な主体性を持って、どのタスクまたはサービスに参加するかを決定します。そのため、このネットワークは自由市場を形成し、経済的均衡が実現されます。モジュールごとに独自の経済が成立する可能性があります。これは、各モジュール固有のジョブに対するリクエスターとプロバイダーの行動によって統制されます。たとえば分散型ビデオトランスコードは、その性質上、ビデオ制作者にとって緊急性のあるジョブです。そのため、分散型ビデオトランスコードへの需要とは無関係に、価格は非融通性を帯びる結果になります（すなわち、ビデオ制作者はサービスプロバイダーにどれくらい報酬を支払うべきかについて、相対的に無関心です)。したがって、この場合は売り手市場になります。分散型トランスコードのサービスプロバイダーが報酬の割り当てを有利に決定することができ、報奨金が上がる傾向があります。同様に、P2P ゲームサーバーモジュールは、リクエストが非常に多いにも関わらず、相対的にサービスプロバイダーが少ない場合があります。この場合も売り手市場になり、報奨金が上がります。逆に、分散型マシンラーニングモジュールは、リクエストする人数が少ないにも関わらず、選べるサービスプロバイダーは豊富にある場合が考えられます。相対的に低需要なので、最低報奨金の要件を高く設定しているために、機会を逃すプロバイダーが多くなると予測されます。この場合、買い手市場になり、報奨金が下がります。

報奨金の設定は、モジュールの要件ではない点に注意してください。また、報奨金をどのようにセットアップするかについては、標準化されたモデルは存在しません。目標は、サービスリクエスターとサービスプロバイダーのインセンティブを調整し、あらゆるタイプのインセンティブモデルをサポートするとともに、世

界各地で異なるインフラストラクチャ／ネットワーキング機能の固定費用にうまく適応することです。

インセンティブ構造

サービスリクエスターとサービスプロバイダーの間におけるインセンティブの調整を、このモデルがどのようにサポートするか具体的に理解するため、双方の経済的利益について俯瞰してみます。サービスリクエスターは、最も低いコストで自分のリクエストを履行してほしいと考えています。一方、サービスプロバイダーは、報酬の高いサービスができるだけ多くなるよう最適化したいと考えています。言い換えると、リクエスターとプロバイダーが価格について合意する限り、ネットワークの有用性を最大化することが、双方にとって最大の利益になります。したがって、報奨金とタスクとのバランスが最適なサービスが、最も履行される可能性が高くなります。そのため、競争力のある報奨金と、分散型サービス設計において効率性が高まるようなイノベーションの両方を促進する方向へ、市場圧力が働きます。

このプラットフォーム上で実行可能な分散型サービスは非常に多様なため、Unit-of-Work-Rewarded（*UWR* 報酬の対象となる作業単位。すなわち報奨金が分割されて支払いが行われる作業の基本単位）と、報奨金モデル（報奨金の支払い方を管理する条件）の決定は、柔軟に行わざるを得ません。たとえば P2P マルチプレイヤーゲームサーバーの場合は、リクエストの数を *UWR* として使用する可能性があります。そのサーバーモジュールを呼び出すゲーム開発者は、サブスクリプション単位の定期的な報奨金モデルが、最も合理的だと考える可能性があります。一方、分散型トランスコーディングサービスは、1 分あたりトランスコードされたバイト数を *UWR* として使用する可能性があります。ビデオ制作者は、トランスコードごとに報奨金を支払う可能性があります。

サービスプロバイダーは、Ara トークンを差し出して参加し、報酬を得ることができます。サービスリクエスターは、報奨金と同じように、サービスプロバイダーがサービスに関わるために差し出すべき最小の預入金を決定できます。最小の預入金の額は、そのサービスで要求されるコミットメントのレベルと、サービスが正常に完了した場合に報奨金とともに返されるレベルを表す目安になります。サービスの *UWR* を決定する際には、その一環として、履行を検証するための証拠も定義しなければなりません。

別の方法として、サービスプロバイダーは、特定のサービスにリソースを専用化して、それに対するサブスクリプション料金を設定することができます。このような専用化プロバイダーをスーパーノードと呼びます。スーパーノードの預入金は、サブスクリプションが終了するまでスマートコントラクトに預託されます。スーパーノードは、他の専用化されていないプロバイダーと比べて信頼性が高いので、彼らがサブスクリプション料金をどのように設定するかによって、市場力学をコントロールできる可能性があります。たとえばコロンビアのボコタにあるスーパーノードは、米国のロサンゼルスにあるスーパーノードよりもハードウェアやインターネットの費用が高いため、料金を高く設定すると考えられます。

ネットワーク効果

ネットワークに新しいコンテンツを導入する際、DCDN スーパーノード（コンテンツの冗長性と可用性を高めることに特化した Ara ノード）の呼び出しを行います。1 つのファイルに対する報奨金が一定であると仮定すると、そのファイルから得られる潜在的な報奨金を共有する DCDN（およびインセンティブが固定されたすべての通貨ベース P2P ファイル共有システム [2]）内の任意のピアによる、ファイル可用性のわずかな向上効果は、劣モジュラ関数を使ってモデル化することができます。これは、リターンが徐々に減少する関数であることが直感的に分かります。この特性から、DCDN スーパーノードは、コンテンツの可用性が高まるにつれ、そのコンテンツをホスティングする機会費用が増加するのに対抗するような、サブスクリプション報奨金モデルを採用することができます。

コンテンツパブリッシャーは、特定の地域で呼び出してサブスクライブするスーパーノードの数を、コンテンツ単位で多くすることも、少なくすることもできます。この方法により、世界規模でコンテンツがどの程度、容易に利用できるようにするかを、パブリッシャーが完全に自由に決定できます。たとえば大手のメディア配信業者が、世界中で使用可能なスーパーノードをすべて呼び出し、世界中のオーディエンスをサポートしたい場合にも、これで対応できます。一方、独立系のコンテンツクリエイターが、自分の主要オーディエンスはヨーロッパだと判断した場合、ヨーロッパのスーパーノードを優先することも可能です。コンテンツパブリッシャーは、各コンテンツダウンロードへの報酬の割り当ても決定できます。このようにして、希望するレベルの参加（すなわち、ファイルの可用性）を達成するのに最適な、報酬の割り当てとスーパーノードの分布が成立します。

参考文献

[1] Enterprise Strategy Group (2015, June), *Price Comparison: Google Cloud Platform vs. Amazon Web Services*, https://cloud.google.com/files/esg-whitepaper.pdf

[2] M. Salek, S. Shayandeh, and D. Kempe, *You Share, I Share: Network Effects and Economic Incentives in P2P File-Sharing Systems* https://arxiv.org/pdf/1107.5559.pdf

II. DCDN コスト分析 by Lester Kim

はじめに

潜在的なパートナーのストリーミングコストを計算するには、そのパートナーが消費者に向けて一定時間 B（単位：秒）あたりに配信しなければならないデータの量 T 単位：バイト）を知る必要があります。アップローダーノードのグループが N 個あると仮定し、ここでは $N \in \mathbb{N}$ と仮定します。$\forall n \in \{1, ..., N\}$ グループ n の平均帯域幅は b_n（単位：ノードあたりのバイト/秒）です。q_n をグループ n ノードの数量と仮定します。$\vec{b} = [b_1 ... b_N]^\top$ および $\vec{q} = [q_1 ... q_N]^\top$ と仮定します。したがって、$\frac{B}{T}$ による制約下で 1 秒あたりに配信されるバイト数は、次の通りです。

$$g(\vec{q}) = \vec{b} \cdot \vec{q} = \frac{B}{T}. \tag{1}$$

分散 $C(\vec{q})$ のコストを最小化するための最適な $\vec{q^*}$ を求める必要があります。$\vec{p} = [p_1 ... p_N]^\top$ とすると（ここで p_n がグループ n のノードあたりの価格とする）、次のようになります。

$$C(\vec{q}) = \vec{p} \cdot \vec{q}. \tag{2}$$

アップローダーの利益の最大化

$\vec{p}$ を決定するため、収益を最大化する企業の行動を考えてみましょう。f を生産関数とします。エネルギー入力は E（単位：kWh）で、出力は q（単位：ノード）です。この生産関数は、次のようにモデル化されます。

$$f(E) = AE^\alpha \tag{3}$$

ここで、A は生産の要素（ノード数/kWh$^\alpha$）であり、$\alpha \in [0, 1]$ は、生産の弾性（入力のパーセント増加に対する出力のパーセント増加）です [1]。

ノードがデータのアップロードを開始した時点における電力の増加を、P（単位：kWh/s）とします。これにはネットワークインターフェイスコントローラ（NIC）を使ったデータの送信が含まれますが、（電源オフまたはスタンバイモードの）マシンの電源をオンにするエネルギーも含まれる場合があります。各ノードに P の電力がある場合、いくらかの E で、1 つのノードが $\frac{E}{P}$ 秒にわたって動作できます。ただし、処理を完了するまでの時間定数が T なので、$\frac{E}{PT}$ 個のノードが存在しなければなりません。したがって、次のようになります。=

$$A = \frac{D}{(PT)^\alpha} \tag{4}$$

ここで、D は総要素生産性（単位：ノード）です。

p をノードの価格とし、p_E を（1 kWh あたりの）エネルギーの価格とします。この企業の利益関数 π は、次の通りです。

$$\pi(q, E) = pq - p_E E. \tag{5}$$

帯域幅のコストは無視しています。なぜなら、月[1]ではなく分という短い期間では、帯域幅は固定コストだからです。

必要な出力を q 以上として、この企業の利益を最大化する必要があります。

$$\max_{q,E} \ \pi(q, E) \quad \text{s.t. } f(E) \geq q. \tag{6}$$

これを解決するには、ラグランジュの方程式を次のようにします。

$$\mathcal{L}(q, E, \lambda) = pq - p_E E - \lambda(AE^\alpha - q). \tag{7}$$

偏導関数を取り、0 にセットすると、次のようになります。

$$\frac{\partial \mathcal{L}}{\partial q} = p + \lambda = 0 \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial E} = -p_E - \lambda A \alpha E^{\alpha-1} = 0 \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = q - AE^\alpha = 0. \tag{10}$$

これらの一次条件を解決すると、次のようになります。

$$q^* = \left(\frac{\alpha A^{\frac{1}{\alpha}} p}{p_E} \right)^{\frac{\alpha}{1-\alpha}}. \tag{11}$$

これを書き換えると（q からアスタリスクを取ると）、次のようになります。

$$p = \frac{p_E}{\alpha} \Big(\frac{q^{1-\alpha}}{A} \Big)^{\frac{1}{\alpha}}. \tag{12}$$

クリエイターはこの公式を使って、希望する数のノードを確保するには、p をどれくらいにすれば良いかを判断できます。

(12) で分かるように、q の観点から最適な利益は、次の通りです。

$$pq = \frac{p_E}{\alpha} \Big(\frac{q}{A} \Big)^{\frac{1}{\alpha}}. \tag{13}$$

[1]帯域幅のコストを含めるとしても、1 秒あたりのコストは、エネルギーの大きさと同じ順序になります。ニューヨーク市の場合、50 MBps のコストは \$3 x 10^{-5}/秒です [3]。

ディストリビューターのコストの最小化

企業の収益は、顧客（クリエイター）から見ると支出なので、(2) を次のように書くことができます。

$$C(\vec{q}) = \frac{p_E}{\alpha} \sum_{n=1}^{N} \left(\frac{q_n}{A_n} \right)^{\frac{1}{\alpha}}. \tag{14}$$

クリエイターのコスト最小化の問題は、次の通りです。

$$\min_{\vec{q}} \ C(\vec{q}) \quad \text{s.t. } g(\vec{q}) \geq \frac{B}{T}. \tag{15}$$

ラグランジュの方程式は、次の通りです。

$$\mathcal{L}(\vec{q}, \lambda) = C(\vec{q}) - \lambda(g(\vec{q}) - \frac{B}{T}). \tag{16}$$

一次条件は、次の通りです。

$$\frac{\partial \mathcal{L}}{\partial \vec{q}} = \frac{\partial C}{\partial \vec{q}} - \lambda \frac{\partial g}{\partial \vec{q}} = 0 \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = \frac{B}{T} - g(\vec{q}) = 0. \tag{18}$$

(14)、(4)、(1) から、

$$\frac{\partial C}{\partial \vec{q}} = \frac{p_E T \vec{P}}{\alpha^2} \circ (\vec{q}^{\circledcirc(1-\alpha)} \oslash \vec{D})^{\circ \frac{1}{\alpha}} \tag{19}$$

$$\frac{\partial g}{\partial \vec{q}} = \vec{b} \tag{20}$$

ここで、$(\vec{D}, \vec{P}) = ([D_1 ... D_N]^\top, [P_1 ... P_N]^\top)$ です。”$\circ$”、”$^\circ$”、”$\oslash$” は、それぞれアダマール（エントリごとの）積、累乗、除算です [4]。したがって、$\forall m, n \in \{1, ..., N\}$ であり、以下になります。

$$\frac{P_m^\alpha q_m^{1-\alpha}}{b_m^\alpha D_m} = \frac{P_n^\alpha q_n^{1-\alpha}}{b_n^\alpha D_n}. \tag{21}$$

したがって、次のようになります。

$$b_m q_m = \left(\frac{b_m D_m P_n^\alpha}{P_m^\alpha b_n D_n} \right)^{\frac{1}{1-\alpha}} b_n q_n. \tag{22}$$

(22) と (18) を組み合わせると、次のようになります。

$$\boxed{\vec{q}^* = \frac{B\vec{b}^{\circ -1}}{T\kappa} \circ (\vec{b} \circ \vec{D} \oslash \vec{P}^{\circ \alpha})^{\circ \frac{1}{1-\alpha}}} \tag{23}$$

$$\boxed{C^* = \frac{p_E T}{\alpha} \left(\frac{B}{T\kappa^{1-\alpha}} \right)^{\frac{1}{\alpha}}} \tag{24}$$

ここで、

$$\kappa \equiv \sum_{m=1}^{N} \left(\frac{b_m D_m}{P_m^{\alpha}} \right)^{\frac{1}{1-\alpha}}. \tag{25}$$

ケース: $\alpha = 1$

$\alpha = 1$ の場合、(23) と (24) は次のようになります。

$$q_n^* = \begin{cases} \frac{B}{|\Upsilon| T b_n} & n \in \Upsilon \\ 0 & n \notin \Upsilon \end{cases} \tag{26}$$

$$C^* = \frac{p_E B P_n}{b_n D_n} \qquad \text{any } n \in \Upsilon \tag{27}$$

ここで、

$$\Upsilon \equiv \left\{ n \in \{1, ..., N\} \;\middle|\; n = \underset{1 \le m \le N}{\arg\max} \frac{b_m D_m}{P_m} \right\}. \tag{28}$$

$\forall n \in \Upsilon$、グループ n の各ノードは、$\frac{B}{|\Upsilon| q_n^*} (= b_n T)$ のデータを配信し、報酬として $\frac{p_E P_n T}{D_n}$ 以上を受け取ります。ただし、$\vec{q}^*$ には複数の解決策があります。たとえば、任意の $n \in \Upsilon$ で、グループ n は、$\frac{B}{T b_n}$ ノードを採用することにより、すべての処理を引き受けることができます。

例

ニューヨーク市における次のような条件下で、

$$\alpha = 1 \tag{29}$$
$$B = 1 \text{ GB} \tag{30}$$
$$N = 2 \tag{31}$$
$$p_E = \$0.2321/\text{kWh [5]} \tag{32}$$
$$T = 1 \text{ s} \tag{33}$$
$$\vec{b} = \begin{bmatrix} 100 \text{ MB/s} \\ 1 \text{ MB/s} \end{bmatrix} \text{ [6]} \tag{34}$$
$$\vec{D} = \begin{bmatrix} 1 \text{ node} \\ 1 \text{ node} \end{bmatrix} \tag{35}$$
$$\vec{P} = \begin{bmatrix} 200 \text{ W} \\ 2 \text{ W} \end{bmatrix} \text{ [7][8]} \tag{36}$$

クリエイターの $\vec{q}^*$ および C^* の例を求めてみましょう。この場合、

$$\vec{q}^* = \begin{bmatrix} 5 \text{ nodes} \\ 500 \text{ nodes} \end{bmatrix} \tag{37}$$
$$C^* \approx \$1.29 \text{ x } 10^{-4}. \tag{38}$$

これは AWS Cloudfront のオンデマンド価格 ($0.020/GB - $0.085/GB) と比べて、155 分の 1 〜 659 分の 1(99.35% - 99.85%) という安さです [9]。グループ 1 の各ノードは 100 MB を処理し、グループ 2 の各ノードは 1 MB を処理します。グループ 1 および 2 の各ノードに必要なコストは、それぞれ $1.29 x 10^{-5} および $1.29 x 10^{-7} 以上です。

これを踏まえて、Netflix を潜在的なパートナーと仮定します。2017 年、Netflix のコンテンツは 1 日あたり平均 1 億 4,000 万時間以上、視聴されました [10]。Netflix ビデオは平均 1 GB/時です [11]。Ara プラットフォームでは、年間 51.1 エクサバイトへ [12] の支出額が、わずか年間 660 万ドル（1 秒あたり 0.2106 ドル）で済みます。Netflix のストリーミング費用を $0.03/GB [13] と推定して、Ara なら年間 15 億ドル（1 秒あたり 46.61 ドル）節約できます。Ara ネットワークを使用した場合、2017 年における Netflix の純利益 5 億 5,890 万 [14] ドルは、ほぼ 4 倍になります。(マンハッタンには 166 万人が住んでおり [15]、そのうち 287,008 人が Netflix ユーザーで[2]、1 日あたり 321.45 TB をストリーミングしています（3.72 GB/秒に相当）。この場合、1 日あたり 41.47 ドルのグループ 2 ノードが、3,721 個必要です）。

[2]2018 年第 1 四半期末の時点で、Netflix 加入者は米国で 5,671 万人、世界全体で 1 億 2,500 万人でした [16]。米国の人口は 3 億 2,800 万人なので [17]、この比率で計算すると、マンハッタン在住の Netflix 加入者は、287,008 人と推定されます。

参考文献

[1] Wikipedia (2018, April 22), *Cobb–Douglas production function*, https://en.wikipedia.org/wiki/Cobb%E2%80%93Douglas_production_function

[2] Wikipedia (2018, June 5), *Total factor productivity*, https://en.wikipedia.org/wiki/Total_factor_productivity

[3] Spectrum (2017, December 29), *Broadband Label Disclosure*, p.2, https://www.spectrum.com/content/dam/spectrum/residential/en/pdfs/policies/Broadband_Label_Disclosure_Charter_122917.pdf

[4] Wikipedia (2018, March 10), *Hadamard product (matrices)*, https://en.wikipedia.org/wiki/Hadamard_product_(matrices)

[5] Electricity Local (2018, June 19), https://www.electricitylocal.com/states/new-york/new-york

[6] Wikipedia (2018, June 19), *Bandwidth (computing)*, https://en.wikipedia.org/wiki/Bandwidth_(computing)

[7] Energuide.be (2018, June 19), *How much power does a computer use? And how much CO2 does that represent?*, https://www.energuide.be/en/questions-answers/how-much-power-does-a-computer-use-and-how-much-co2-does-that-represent/54/

[8] R. Sohan, A. Rice, A. W. Moore, and K. Mansley, "Characterizing 10 Gbps Network Interface Energy Consumption,"*The 35th Annual IEEE Conference on Local Computer Networks (LCN) Short Papers*, University of Cambridge, Computer Laboratory, July 2010, https://www.cl.cam.ac.uk/ acr31/pubs/sohan-10gbpower.pdf.

[9] Amazon Web Services Pricing (2018, June 19), *Amazon Cloudfront Pricing*, https://aws.amazon.com/cloudfront/pricing

[10] L. Matney (2017, December 11), "Netflix users collectively watched 1 billion hours of content per week in 2017," *Techcrunch*, https://techcrunch.com/2017/12/11/netflix-users-collectively-watched-1-billion-hours-of-content-per-week-in-2017

[11] K. Hubby (2017, May 23), "The surprising amount of data Netflix uses," *The Daily Dot*, https://www.dailydot.com/debug/how-much-data-netflix-use/

[12] Wikipedia (2018, June 20), *Exabyte*, https://en.wikipedia.org/wiki/Exabyte

[13] D. Rayburn (2009, July), "Stream This!: Netflix's Streaming Costs," *Streaming Media (June/July 2009)*, http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/Stream-This!-Netflixs-Streaming-Costs-65503.aspx

[14] Netflix (2018, January 1), p.40, https://ir.netflix.com/static-files/20c3228d-bf1f-4956-a169-c8b76911ecd5

[15] Wikipedia (2018, July 5), *Manhattan*, https://en.wikipedia.org/wiki/Manhattan

[16] Statistica (2018, July 5), *Number of Netflix streaming subscribers in the United States from 3rd quarter 2011 to 1st quarter 2018 (in millions)*, https://www.statista.com/statistics/250937/quarterly-number-of-netflix-streaming-subscribers-in-the-us/

[17] United States Census Bureau (2018, July 5), *U.S. and World Population Clock*, https://www.census.gov/popclock/

III. Ara 報酬予測額の分析 by Lester Kim

ネットワークモデル

一連のノードからなる Ara ネットワークを完全なグラフ [1] $G = (V, E)$ で表してみましょう。ここで V には N 個の頂点が含まれており、それぞれの頂点がノードを表します。E には $\frac{N(N-1)}{2}$ 個のエッジが含まれており、各エッジがノード間の通信チャネルを表します。C は、すべてのノードが希望する何らかのコンテンツの集合（デジタルエンターテインメントファイル）であると仮定します。サブセット $S \subseteq V$ には、C を持っているすべてのノードが含まれていると仮定します（すわなち $S = \{v \in V : C \in v\}$）。

時間 $t \in \mathbb{N}$ と仮定します。$t = 0$ のとき、$|S| = 1$ となり、1 つの $v_0 \in V$ だけがコンテンツ C を持っています。したがって、ネットワークの他のノードに C のコピーを配信できる頂点は 1 つだけです。他のすべての $N - 1$ ノードが C を希望しており、v_0 には 1 つのノードに C を配信する十分な帯域幅があると仮定します。$t = 0$ から $t = 1$ に変化すると、$|S|$ は 1 から 2 に増加します。一般に、時間 t では、次のようになります。

$$|S| = \begin{cases} 2^t & 0 \leq t < \log_2 N \\ N & t \geq \log_2 N. \end{cases} \tag{39}$$

$|S| = N$ は $t = \lceil \log_2 N \rceil$ で始まっている点に注意してください。

$\forall s \in S, s$ は、C を一部の $v \in V \setminus S$ に配信します。ただし、v が s に対し、金額 p を支払う場合に限られます。M は、エンターテインメント配信のためのネットワークの総予算であると仮定します。これを N 個のノードで均等に割ると、$p = M/N$ になります。

$t = 0$ のときは、$v_0 \in S$ だけが p を一部の $v_1 \in V \setminus S$ から受信します。その後、$t = 1$ のとき、$v_0, v_1 \in S$ はそれぞれ p を一部の $v_2, v_3 \in V \setminus S$ から受信します。$t < \lfloor \log_2 N \rfloor$ のとき、$|S| = 2^t$ の中の S ノードはそれぞれ、p を 2^t の中の $V \setminus S$ ノードから受信します。$t = \lfloor \log_2 N \rfloor$ のときは、$|S| > \frac{N}{2}$ となり、C の供給側のほうが需要側より多くなります。この場合、$N - 2^t$ から S 個のノードがランダムに選ばれ、C を配信します。$t = \lceil \log_2 N \rceil,\ S = V$ となります。

このモデルで、v_0 は最低でも次の報酬を獲得します。

$$\frac{M \lfloor \log_2 N \rfloor}{N}; \tag{40}$$

v_1 は最低でも $\frac{M(\lfloor \log_2 N \rfloor - 1)}{N}$ を獲得します。v_k は最低でもを獲得します。

$$\frac{M(\lfloor \log_2 N \rfloor - \lceil \log_2 (k+1) \rceil)}{N}. \tag{41}$$

k が v_k 以上で獲得する最大の $\frac{M}{N}$ は、次の場合です。

$$\lfloor \log_2 N \rfloor - \lceil \log_2 (k+1) \rceil \geq 1 \tag{42}$$

これは、次のことを意味します。

$$\log_2(k+1) \leq \log_2 \frac{N}{2}. \tag{43}$$

したがって、$\frac{M}{N}$ 以上で獲得する k の最大値は、$k = \lfloor \frac{N}{2} \rfloor - 1$ です。平均すると、それぞれ次の報酬を獲得します。

$$\frac{M - \frac{M}{N}}{2^{\lceil \log_2 N \rceil - 1}} = \frac{M(1 - \frac{1}{N})}{2^{\lceil \log_2 N \rceil - 1}}. \tag{44}$$

分子は $M - \frac{M}{N}$ です。v_0 のエンターテインメント予算を除外するためです。これは $t = 0$ の時点で C を持っていたからです。分母は $2^{\lceil \log_2 N \rceil - 1}$ です。なぜなら、$t = \lceil \log_2 N \rceil - 1$、$|S| = 2^{\lceil \log_2 N \rceil - 1}$、およびこの時点で、$S$ はプロセス全体を通じて報酬を得る可能性のあるすべてのノードで構成されているからです。つまり、報酬を得られないノードが、$N - 2^{\lceil \log_2 N \rceil - 1}$ 個あることを意味します。

例

米国では約 80% の人々が、インターネットアクセス可能なコンピューターを所有しています [2]。米国の人口は 3 億 2,700 万人なので [3]、インターネットに接続するデバイスを持っている米国人は $(0.8)(327\text{M}) = 261.6\text{M}$ 人と推定されます。1 人 1 台のデバイスを所有していると仮定すると、$N = 261.6\text{M}$ となります。米国におけるエンターテインメント消費総額は、年間 7,340 億ドルです [4] [5]。今後、この支出額のほとんどがデジタルになると仮定します。ただし、インターネットにアクセスする米国人の 80% の予算だけを含めることにします。したがって、これらの人々による支出額は $(0.8)(734\text{B}) = \$587\text{B}$ です。この支出額の 10% が、ディストリビューションコストに充てられると仮定します。この場合、$M = (0.1)(\$587\text{B}) = \$58.7\text{B}$ となります。この場合、$p = M/N = \$58.7\text{B}/261.6\text{M} = \$224.39$ となります。(44) によると、年間の平均報酬額は、1 ノードあたり $437.35 です。(40) によると、ほとんどの v_0 が獲得できる金額は $6282.87 です。したがってこの例から、最初にコンテンツを共有するピアが、報酬を最も多く得ると予測されます。

参考文献

[1] Wikipedia (2018, June 19), *Complete graph*, https://en.wikipedia.org/wiki/Complete_graph

[2] C. Ryan and J. M. Lewis, "Computer and Internet Use in the United States: 2015," *American Community Survey Reports* U.S. Census Bureau, September 2017 https://www.census.gov/content/dam/Census/library/publications/2017/acs/acs-37.pdf

[3] World Population Review (2018, June 18) *United States Population 2018* http://worldpopulationreview.com/countries/united-states-population/

[4] SelectUSA (2018, August 22), *MEDIA AND ENTERTAINMENT SPOTLIGHT*, https://www.selectusa.gov/media-entertainment-industry-united-states

[5] Bureau of Labor Statistics (2017, August 29), *CONSUMER EXPENDITURES–2016* https://www.bls.gov/news.release/cesan.nr0.htm