Distributed Cloud Computing on Trusted Hardware

Ankr

Chandler Song
Stanley Wu
Song Liu
Ryan Fang
Quan-Lai Li

June 1st, 2018

Abstract

In this paper, we present Ankr, a computing-power-based decentralized network with a native data feed system. We first review a novel mining scheme based on the Proof of Useful Work (PoUW) consensus on an SGX-enabled trusted execution environment (TEE). We then elaborate how Ankr can use this platform to enable Distributed Cloud Computing (DCC) on the Blockchain environment. Next, we describe a Native Authenticated Oracle Service (NOS) for smart contracts using trusted hardware. We conclude by discussing applications in the Ankr ecosystem for real-world business applications.

1 Introduction

Since the advent of Bitcoin, the world has witnessed the rapid progress of and improvements to blockchain technology. However, various obstacles continue to prevent the mass adoption of blockchain technology for real-world business applications. The biggest challenges are:

- **Resource-Inefficiency** Massive wastage of energy and computing power
- **Data-Insufficiency** Lack of reliable and efficient data feed services
- **Low Scalability** Low throughput due to one-chain implementation
- **Inequity** The majority of resources lie in the hands of the few
- **Exposure** Lack of confidentiality for otherwise private information

Ankr resolves these challenges through the introduction of a resource-efficient blockchain framework that truly enables distributed cloud computing and provides a user-friendly infrastructure for business applications. Ankr's key innovations and improvements include:

- **Novel Incentive Scheme** based on useful computing workload
- **Distributed Computing Power** composed of serverless and stateless computing units
- **Reliable Oracle Service** for connecting to existing businesses

- **Security and Confidentiality** guaranteed by both trusted hardware and cryptographic primitives
- **Speed and Scalability** enabled by Plasma-chain implementation

2 Background

2.1 Consensus Protocol

Bitcoin's underlying consensus, Proof of Work (PoW), addresses two underlying problems in decentralized cryptocurrency design: how to choose consensus leaders and how to allocate rewards in a fair manner among participants. However, Bitcoin and other cryptocurrencies which rely on PoW lack any real-world benefits beyond serving as an alternative payment method. In spite of this limited functionality, Bitcoin's network currently uses more electricity than the entire country of Iceland, with energy consumption projected to scale to the amount consumed by the population of Denmark by the year 2020.

Other popular consensuses such as Practical Byzantine Fault Tolerance (PBFT) or Proof of Stake (PoS) are essentially waste-free, but they restrict participation or require participants to lock in the stakes of the blockchain. Further, PBFT and PoS are often complex and arbitrary, making it extremely difficult to update and adjust the original setup.

2.2 Trusted Hardware

Intel SGX (Software Guard Extensions) is a new set of instructions that permit execution of an application inside a hardware enclave. This mechanism protects the application's integrity and confidentiality against certain forms of hardware and software attacks, including hostile operating systems. By processing isolated executions in SGX, system calls are guaranteed to execute correctly and securely.

SGX allows the generation of authentication that remotely proves the validity of an operation. When an enclave is created, the CPU produces a hash of its initial state. The software in the enclave may at a later time request a report which includes a measurement and supplementary data provided by the process. The report is digitally signed using a hardware-protected key to produce proof that the software is running in an SGX-protected enclave.

Such proof can be verified in a remote system, and SGX verifies proof using a group signature.

SGX was introduced in 2015 with the sixth generation Intel Core microprocessors. Ankr's technology uses the SGX CPU mining platform to greatly reduce the barrier to entry for miners, minimizing the likelihood of monopolies forming with mining pools.

Other companies including ARM and Nvidia are also investing the TEE solution. For example, TrustZone, offered by ARM, is a simplified version of TEE specialized for mobile or tablet device. Nvidia, on the other hand, published Trusted Little Kernel (TLK) for TEGRA as the first GPU-based approach towards TEE. These efforts, along with Intel SGX, represent the complete TEE device portfolio which will expand the foundation of Ankr's technology.

2.3 Distributed Computing

Several projects are trying to utilize blockchain technology to provide distributed computing services. They can be primarily categorized into:

- **Distributed Smart Contracts** Code execution in blockchain is currently decentralized but not distributed. Therefore, every node in Ethereum redundantly executes the same code and maintains the same public state. A natural improvement to overcome this performance pitfall is to distribute the execution of smart contracts. For example, **Dfinity**, which incorporate Proof of Stake with a verifiable random function, appoints only certain candidate node(s) in the chain with the right to mine the block and hence execute the smart contracts. This method improves the throughput of executions but still suffers from the limitation of smart contracts especially when there is no Internet connection.

- **External Computing Containers** Another idea is to combine the blockchain with external computing containers. Projects like **Golem**, **SONM** and **iExec** share the same vision for this infrastructure. However, their respective technical design and go-to-market strategies differ. Golem and SONM are both integrated with **Docker** and have their own niche markets. Golem is aiming to attract regular 3D rendering users, and SONM is approaching fog and edge computing. IExec, on the other hand, focuses on using the **Desktop Grid Network** to build a decentralized cloud. While innovative, these projects do not focus on fixing POW mechanics.

- **Efficient consensus** The third way is to utilize the computing power in the consensus protocol. With the help of trusted hardware such as Intel SGX or ARM TrustZone, platforms can develop trust of correctness protocols upon the attestation mechanism of Trusted Execution Environment (TEE) providers. This allows a significant portion of computing resources to be released for useful workloads. **Ankr**'s Proof of Useful Work (PoUW) utilizes practical computing needs instead of hash computing as the deciding factor to generate new blocks in the chain. This approach allows miners to get income not only from mining new blocks and transaction fees but also from computing subtasks of useful work for clients.

2.4 Data Feed

Currently, solutions for data feed systems are undesirable:

- **Centralized Oracle Service** A centralized oracle is the mostly commonly used approach in many blockchains. Such a system is the antithesis of decentralization as it fails to provide tamper resistance and security. Almost all centralized oracles rely on off-chain notarization, which can create potentially troublesome results.

- **Manual Human Input** Many proposals focus on the full adoption of manual human input. Though decentralized and flexible, such an approach is not only time-inefficient, but also resource-intensive.

- **Maneuvering Data Sources** Even though TLS-N provides digitally signed data sources, such a method would require all legacy systems and websites to change their infrastructure accordingly.

3 Proof of Useful Work

The Proof of Useful Work (PoUW) consensus is able to achieve a high-security standard without wasting energy. In this scheme, the participants with CPU computing power can execute useful computations under the

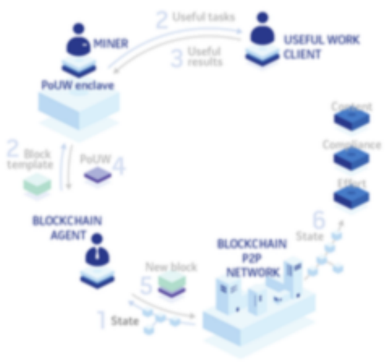

Figure 1: Architecture Overview of PoUW Consensus

trusted hardware's proctoring to earn rewards within the blockchain ecosystem. In contrast, PoW only allows a reward to be claimed after miners have solved complex hashing problems. Such a setup is due to the lack of a Trusted Execution Environment(TEE). A SGX-protected hardware enclave can act as a trusted proctor for a CPU's activities, testifying to miners' useful computations and providing proof for mining rewards.

3.1 PoUW Components

As depicted in Figure 1, Ankr has a mining scheme composed of three components: **miners**, **useful work providers**, and **blockchain agents**.

- **Blockchain Agent** Blockchain agents serve as the liasion between the PoUW mining scheme and the rest of blockchain system. They directly interact with the blockchain's P2P network via RPC requests. A blockchain agent will collect transactions and generate a block template without proof of useful work. Once the miner provides proof of userful work and embeds it into the template, the blockchain agent will validate the attached proof, publish the block to the blockchain, and then receive the corresponding rewards.

- **Useful Work Provider** The major functionality of useful work providers is to supply miners with useful work tasks and receive the task results.

A useful work task contains two parts: **a PoUW enclave** and **some task inputs**. Any SGX-compliant code can be transformed to a PoUW enclave by using the tools that Ankr will provide.

- **Miners** Miners are the centerpiece of the entire consensus protocol. A miner takes a useful work task and the template mentioned above as inputs. Miners can then launch an Intel SGX enclave to load and run this useful work task. At the end of execution, the results of the task will be returned to the useful work providers. The miner will need to finish this cycle by metering the useful work task and also by deciding whether this work successfully won the consensus leadership or not. If yes, the miner will generate a PoUW which includes the following two parts and is attached to the block template for publishing.

  1. An Intel SGX-produced attestation to prove the PoUW enclave's compliance with Ankr's mining algorithm
  2. Another attestation to demonstrate the task was finished at a given difficulty level, which will be used as the baseline for the next iteration of new block

Algorithm 1 PoUW Mining Algorithm as in Figure 1

```
1: procedure MineLoop
2:     while true do
3:         block template ← blockchain agent
4:         (hash, difficulty) ← process(template)
5:         task ← useful work client
6:         PoUW ← enclave(task, hash, difficulty)
7:         if PoUW ≠ ⊥ then
8:             new block ← block template, PoUW
```

3.2 Block Generation

Ankr's miners statistically simulate the process of block generation in a Proof of Work system. In Bitcoin, each iteration of hash calculations can be modeled as a Bernoulli trial and the entire block generation process can be viewed as a Poisson random process. Similarly, in PoUW, the enclave of a miner

treats each instruction as a Bernoulli trial and then the mining time of PoUW can be treated and tweaked in the same manner as in PoW.

To decide which instruction deserves the reward, the PoUW enclave generates a random number by using SGX's random number generator (SRNG) and checks whether this number is smaller than the desired difficulty. To avoid the prohibitive overhead of checking for each individual instruction, Ankr divides a useful work task into smaller sub-tasks of short durations (e.g., 10 seconds). After each subtask, the enclave calls SRNG to check whether at least one instruction gained the consensus leadership. If so, the enclave generates the corresponding attestation including the difficulty level.

3.3 Useful Work Metering

At Ankr, the effort of a useful work task is metered on a per-instruction basis, which is much more accurate than variables such as task running time, number of tasks, task file size, etc. While the alternative approach of using CPU cycles might initially seem more accurate, this method is vulnerable to manipulation as the CPU cycle counts could be accessed through the CPU's performance counters. Moreover, counters are incremented even when the enclave in question is swapped out. As a consequence, a multi-enclave environment will cause a dramatically inflated CPU cycle count for measurement. In conclusion, CPU instruction counting, although not perfect, is still a better proxy to estimate the working effort of a task.

3.4 Proof Attestation and Block Validation

3.4.1 Intel SGX Remote Attestation

Attestation is the process of demonstrating that a piece of software has been properly instantiated on the platform. In Intel SGX, it is the mechanism by which another party can gain confidence that the correct software is securely running within an enclave on an enabled platform. To do this, the Intel SGX architecture produces an attestation assertion that can be verified by a remote instance.

When an enclave is created by SGX, the CPU will generate a hash of its initial state called measurement. Later, the code inside enclave can request a report consisted by this measurement and other supplementary data provided by the code. This report is digitally signed by a hardware-protected key to

prove that the code have been successfully been running on an SGX enclave. This proof, called quote, along with measurement, and supplementary data are part of attestation which can be verified remotely.

3.4.2 Two-Layer Hierarchical Attestations

The PoUW attestation follows a two-layer hierarchical attestations.

- **Useful Work Attestation** Useful work attestation is created by a useful work enclave. This attestation contains the prefix hash and difficulty from the blockchain template. This type of attestation can prove:

  1. A useful work task was properly run inside a miner's enclave
  2. This useful work task was mined for the block template with the same prefix hash and difficulty

- **Compiler Checker Attestation** PoUW miners include a tool called the compiler checker, which detects and confirms whether the useful work tasks conform to the requirements of PoUW. The compiler checker runs in the trusted environment as well and generate an attestation containing the measurement of useful work enclave if the task meets all the following requirements:

  1. The *text* section is non-writable, which means it will not allow attempts to rewrite iteself at runtime for the purpose of security.
  2. The dedicated register is reserved for instruction counting in order to correctly meter the effort.
  3. The runtime is correctly linked.
  4. The only entry point is the PoUW runtime.

3.4.3 Block Validation

Besides the exisiting block validation mechanisms in Bitcoin or Ethereum, PoUW also requires verification of the attestations in each block. In general, the hierarchical attestation approach is used to obtain attestation to code that no one knows in advance. And the measurement of the compiler checker enclave is hard-coded into any portion that needs to verify PoUW. Therefore, block validation works as instructed and the new block in question will be successfully validated only when both checks pass.

1. The measurement of the compiler checker enclave and the compiler checker attestation prove that an enclave has successfully passed the compiler checker.

2. The measurement of this enclave and the useful work attestation prove that this enclave was properly run in the miner's enclave.

3.5 PoUW Outlook

PoUW opens up new opportunities for Ankr as miners' computing power within the network can be used towards almost all on-chain computation. For instance, if a client submits a private datagram request through our oracle service, **the miner will use its resources to help encrypt and decrypt the parameters in the datagram's request**. Excessive computational resources can be monetized and sold to internal or external applications for useful work. These useful work tasks range from training a neural network to hosting web services. Unlike Bitcoin, Ankr rewards **every user for contributing his or her computing resources**. On the other hand, only one user during a computing cycle is lucky enough to have permission to generate a block and receive an extra reward.

With the widespread use of SGX-enabled CPUs after 2015, the PoUW protocol has the potential to unlock massive idle SGX-enabled CPU computing power for useful work computations. We envision that useful work can come from computation tasks both on and off the blockchain. This will enable distributed cloud computing, as otherwise idle CPUs can now generate value for owners with little cost. Furthermore, under such a distributed computing framework, the task requestors will have access to a cheaper cloud computing service compared to traditionally centralized cloud computing services. Currently, cloud service giants like Amazon, Microsoft, Google and Alibaba operate with large profit margins (30%) due to their monopolization of the market.

4 Distributed Cloud Computing

4.1 Blockchain Technology and Distributed Cloud Computing

4.1.1 Background

The advancement of internet technology has generated massive troves of data spanning text, audio, video, etc. However, most of this data is neither structured nor relevant to each other. Processing the data in series becomes increasingly resource-inefficient and cannot be tolerated by the rapid velocity of business development.

- **Distributed computing and parallel computing** Distributed computing is a solution that completes huge computing tasks through a plurality of geographically diverse host computers (clusters) instead of through a single supercomputer. Parallel computing refers to the parallel processing of multiple CPUs. Parallel computing can improve computational efficiency with the premise that program algorithms should be designed in parallel as much as possible.

- **Virtualization** Virtualization is a method of dividing resources for cloud computing. It includes two aspects: physical resource pooling and resource pool management. There are also two types of virtualization. One is to virtualize multiple physical resources into one "large" logical resource layer. The other is to divide one physical resource into multiple "small" logical units.

Currently, cloud computing mostly adopts the first type of virtualization, which are primarily deployed on distributed clusters to process massive data and provide on-demand IT services for a huge amount of access. DCC represents a superior method of rapidly processing massive amounts of data. This solution proposes to use more hardware in exchange for processing time, which requires distributing data on multiple computers and processing them simultaneously (in parallel). However, most IT companies are not capable of building their own cloud distributed systems. Instead, they use the products from existing cloud service providers to focus on their own products and business.

With help of flexible dev tools, DCC can help developers to rapidly deliver services or products according to their specs. The new foundation of app-building will rely on decentralizing micro-services and synchronizing the delivery of small tasks after execution. As the price of CPUs declines, complex applications (e.g. CGI rendering, scientific computing, machine learning, etc.) will become affordable and accessible to everyone in the form of cloud services.

4.1.2 Grid Computing and Volunteer Computing

Grid computing is the collection of computer resources from multiple locations to reach a common goal. Generally speaking, the grid can be thought of as a distributed system in which each node set performs a different task or application. This can thus be considered a solution to the overwhelming data problem mentioned above.

However, the major source of the grid is currently from volunteers, either from individuals or academic organization. For example, the Berkeley Open Infrastructure for Network Computing (BOINC), is a common platform for various academic projects seeking public volunteers. Its goal is to turn a heterogeneous, high-churn, untrusted pool of consumer computers into a reliable, predictable, trusted job processing system for scientists or researchers. While technically savvy, the platform continues to face the following problems:

- **Sustainable Computing Source** With volunteer computing, nodes are likely to go "offline" from time to time, as their owners use their resources for their primary purpose. Additionally, the lack of any material incentive makes it difficult to acquire a stable computing supplier pool.

- **Source Distribution** The distribution of BOINC, for instance, leans toward wealthier areas such as North America and Western Europe. However, areas that lack funding and depend highly on volunteer computing to conduct research experiments do not have many computing suppliers to engage with.

4.1.3 Advantages of DCC on Blockchain

A P2P network allows application owners and individual users (both are requesters) to rent computing power from other users (suppliers). Currently, the computing resources in popular blockchain networks such as Bitcoin or Ethereum, are more than sufficient to process high throughput computing tasks, thus providing an affordable alternative to supercomputers or large corporate cloud computing. However, cloud computing resources are primarily controlled by centralized cloud service providers and are subject to rigid operation models. A decentralized cloud computing platform can incorporate a blockchain-based payment system (such as Ethereum), which can allow for direct payment among operators (requesters), sellers (suppliers) and software developers.

4.2 Operation Model

4.2.1 Token Economic Model

Here is a rough outline of the token economic model.

- **Requesting Node** This node is generally performed by a number of merchants or scientific research institutes with computational needs. Since the computers in their possession cannot meet their current computing needs, supercomputers or other cost-effective computational resources such as globally-distributed computing are used for this purpose. The requesting node may be required to categorize the code or data according to certain specifications before sending jobs into the network. Alternatively, other nodes may exist to complete this task.

- **Categorizing or Dispatching Node** This node is used to classify the task and data according to some specification and then dispatch them to the appropriate processing node. The purpose of this step is to preprocess the job's metadata, characteristics, and priorities, thus choosing the optimal node to handle them.

- **Processing Node** This node is dedicated to processing tasks or data. Since there are thousands of types of tasks or data transferred from the previous two type of nodes, methods for processing specific data or models that should be used are also different. For some complicated

projects, the requirements for participation are higher. Here are two examples:

1. In the medical field, the processing of data such as medical images generally requires personnel with expertise. Additionally, not everyone should have the access or permission for manual inputs.
2. In the field of AI and machine learning, data training and processing are usually coded by experienced developer due to the specificity of data processing models such as KNN or decision trees. Additionally, single ordinary CPUs or GPUs are no longer enough to meet this computing need.

- **Verification node** This node judges and filters the processing results from one or more processing nodes. This in general only requires a general type of CPU or computing power. Basically, the same data processing task will be sent to multiple data processing nodes, so that voting can be performed later to determine the satisfactory output. Though this will result in data or computing redundancy, the accuracy of the final result can be highly protected. Besides, working with the reputation mechanism can effectively reduce the impact of compromised nodes.

4.2.2 Ankr Token System

Ankr tokens serve as a means to both store and transfer value. At all times, Ankr tokens can be used to incentivize engagement with the Ankr network and can be used for the computation fees of PoUW. All mining participants of Ankr's DCC can earn Ankr tokens by contributing their computing power. Such an ecosystem is a virtuous cycle: the more people participate in useful work computations, the more Ankr tokens they will earn; the more tokens they have, the more services they can acquire in the ecosystem, triggering further computation needs.

Ankr tokens can be mined by contributing idle computing power for useful work computation and will be charged for computation and transactions that happen on the blockchain. This will create a truly self-sustained ecosystem and create a "decentralized world computer." The decentralized computer will cost users Ankr token to use, but it will be much cheaper than centralized solutions such as Amazon cloud or Google cloud, because the decentralized

cloud will utilize otherwise wasted computing power and would not charge a high markup like the Internet giants.

Moreover, human capital for running the decentralized computer will be much lower, as the decentralized solution won't spend a large amount of money for administration, marketing and high executive compensation. In other words, the "decentralized computer" will provide the strongest computing power at the lowest cost.

4.2.3 Reputation System

The blockchain DCC is no longer contributed by volunteers. A new mechanism needs to be established to measure the contribution of each node and allocate more tasks to give out more rewards.

One possibility is to measure the number of tasks completed. However, projects operated on distributed cloud computing in the future may vary greatly. For example, one task may take 1 hour to process, yet another task may take 20 hours. This would result in the same rewards for the same number of tasks but totally different workloads. This is obviously not a feasible solution. Similarly, this method does not do a good job of measuring the CPU time, as it is difficult to accurately record the amount of calculations actually performed by each user per individual heuristic. A multi-heuristic algorithm should be built to achieve fair and accurate reward distribution.

A working reputation system should consider the following aspects to calculate the contribution of a node:

- **Performance Test** The resource usage and cost of performance equipment varies a lot in the network. Therefore, a standardized performance measurement should be used for different equipment to obtain a reasonable performance judgment for further calculation.
- **The Number of Correct Results** The key concern of a DCC service is the quality of the result. It is not acceptable if the result is incorrect or adversely generated by malicious users. The number of correct results will be a good direct indicator of a participants' reputation. More rewards will be issued to the honest, while punishments will be delivered to malicious users.

Problems that may occur:

- **Inaccurate Performance Test** This is more obvious when crossing platforms. For example, a computer may have extremely different measurements on a Windows system when compared to a Linux system.
- **Lower Cheating Barrier** The DCC needs to be open-sourced to the public. It is possible for the user to download the source code, in order to modify and build the logic in their own favor.

4.3 Architecture Overview

Ankr strives to build a resource-efficient blockchain framework that truly enables Distributed Cloud Computing (DCC) and provides user-friendly infrastructure for business applications. To achieve this, Ankr thoroughly investigated the trend of DCC's most popular use patterns and underlying technologies or platforms. The goal is to provide a smooth on-hand experience for cloud users and minimize the switching cost to integrate Ankr's DCC into their solutions.

PoW was invented with a lack of trusted execution environments (TEE). Therefore, Bitcoin requires every miner, even cheaters, to solve a mathematically difficult task (hash calculations) in order to be the next consensus leader. However, PoUW, with the help of TEE, releases the computing power of each individual miner for tasks of general purposes. Furthermore, these miners form a large-scale network of independent *serverless* computing units, on top of which Ankr has the capability to build a DCC service.

Serverless architecture is a heated design paradigm in the cloud. By using serverless architecture, the developer can focus on the core logic of the business needs without worrying about managing and operating the servers. According to Amazon Web Service (AWS), "A serverless architecture is a way to build and run applications and services without having to manage infrastructure. Your application still runs on servers, but all the server management is done by AWS."

At Ankr, we present a three-layer design of Ankr's Distributed Cloud Computing and how these layers are organized to operate together as a system.

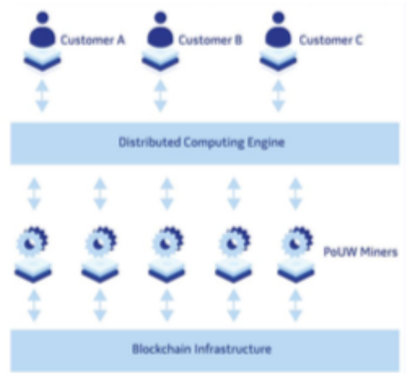

Figure 2: Ankr's DCC 3-layer Design

4.3.1 Blockchain Infrastructure

Blockchain infrastructure is the lowest level of Ankr's DCC design. This layer provides the routine functionality, except mining, of blockchain such as serving RPC requests, maintaining blocks or transactions, etc. All dApps built upon Ankr's blockchain are mostly directly involved with this layer.

Due to the inefficiency of Ethereum, we also utilize the multi-chain structure to improve the throughput of smart contracts, which will be discussed in detailed in section 6.

4.3.2 PoUW Miners

PoUW miners are the execution layer of the stack. As described in section 3, the specific functions of this layer among the DCC are:

1. Fetch block templates with transactions from the Blockchain Infrastructure layer.
2. Fetch useful work from the Distributed Computing Engine layer and perform the work inside the TEE.
3. If a miner successfully gains the privilege to publish the next block, this miner will attach the PoUW to the block and push it to the Blockchain Infrastructure layer for publishing.

4.3.3 Distributed Computing Engine

The Distributed Computing Engine is the most important layer of the three. The main purpose of this layer is to accept the useful work from the cloud customers and dispatch it among the PoUW miners in Ankr's discretionary manner.

Most DCC customers use existing open source engines like Spark to develop solutions at a large scale such as machine learning, model training, etc. Ankr would like to provide a similar development experience in order to lower the switching cost and learning curve for Ankr's customers.

- **Job Dispatcher and Scheduling** The initial thought of the job dispatch mechanism is to enable a fair scheduler. Under fair sharing, the node in this layer will keep a hash map in the memory to keep track of all the miners it is managing and assign jobs among miners in a "round robin" fashion, so that all miners get a roughly equal share of jobs in terms of quantity. Once this has been established, further heuristics such as job size (storage size), bidding price (tips) and even IP address can be also taken into consideration to adjust the priority of jobs for a better user experience and prevent the centralization of mining power that PoW currently does.

- **Serverless on Spark** To remove the operational complexity of customers, the next generation of Spark is headed toward serverless computing as well. Databricks, the inventor of Spark, has launched the first phase of its serverless product, called Serverless Pool, which allows users to run a pool for serverless workloads in their own AWS accounts. Although still in its early stage, this initiative brings on board a practical direction to integrate Ankr's computing power from PoUW miners with popular open source tools by introducing a proper adaption layer.

- **Privacy and Security** As illustrated in section 5, NOS presents an authenticated and secure way to transfer the off-chain data to the on-chain smart contract. A similar but not identical mechanism could be employed here utilizing Intel SGX to make sure that the useful work submitted by customers (as one type of data) can be securely transmitted to the enclaves in the miners. In the interim, TLS and certificate should be also used to strengthen the inter-layer communication.

4.4 Security Model

Ankr will have its own blockchain which could be forked from Ethereum, even though Ankr's consensus protocol can be easily swapped into either Bitcoin or Ethereum as a new consensus framework. Ethereum is preferred, from a customers' point of view, for supporting smart contracts. As a result, the security level of Ankr will be maintained at the same level as Ethereum.

In general, security is carefully designed for Ankr's Distributed Cloud Computing.

4.4.1 Enclave Security

SGX permits the execution of trustworthy code in a totally isolated, tamper-free environment. Users also have the option of attesting remotely that outputs represent the result of such execution. Additionally, on-chip digital random number generator (SRNG) is used to increase the security level.

4.4.2 Data Source Security

The data source providers can choose different security levels based on the requirements. Ankr supports all security levels from non-encryption to high confidentiality. The following measures are taken for high-level security:

- In transmission, providers can choose TLS 1.2/1.3 and PFS(Perfect Forward Secrecy). Provider certificate will be verified. Providers can choose client certificate verification as well, but its own provider node should be configured in this case.

- In data encryption, the data will be encrypted with parameters. Therefore, even the TLS end-point cannot see the clear text of the data. The data symmetric key and MAC are encrypted with the public key of targeted enclave. Only that enclave can decrypt the data. When one enclave wants to transfer such data to another enclave, the first enclave will encrypt the clear text with the public key of the second enclave and then send it to the second enclave. A simple version of TLS record protocol is designed for confidentiality and integrity of Ankr data.

In this way, only enclaves have the chance to read the clear texts of data, even in compromised operation systems.

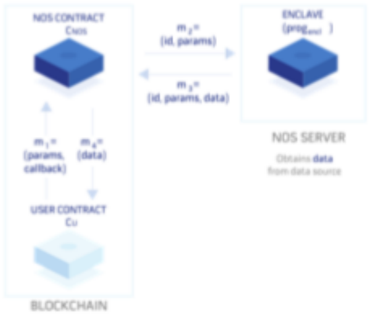

Figure 3: Data Feed Model

5 Native Authenticated Oracle

We introduce the Native Oracle service (NOS), an authenticated data feed system built on trusted hardware.

5.1 Components

NOS consists of three components: the **NOS Smart Contract**, the **Enclave** and the **Relay**. The Enclave and Relay reside on the NOS server, and the NOS smart contract is executed on the blockchain.

- **NOS Smart Contract** The NOS Smart Contract functions as the front-end of the NOS system. It provides an API to interact with any form of smart contract in various programming languages. Concretely, a NOS Smart Contract accepts a datagram request from a blockchain-based smart contract and responds with the corresponding datagram. C_{NOS} also provides monetary management similar to Ethereum's gas. Additionally, it fixes many existing issues with Ethereum's gas, which will be addressed later.

- **Hardware Enclave** The Enclave is responsible for ingesting datagram requests from the blockchain. It queries external HTTPS-enabled internet data sources, and returns a datagram as a digitally signed message. The Enclave should be assumed to be secure as it lacks network access

Figure 4: Enclave and Server

and is completely isolated from any operating system and software, whether hostile or not.

- **Relay** The Relay handles bidirectional network traffic on behalf of the Enclave.

5.2 Protocol

- **Initiate** The smart contract sends a datagram request to the NOS Contract.

- **Monitor and Relay** The Relay monitors the NOS Contract and delivers any incoming requests to the Enclave.

- **Fetch** The Enclave communicates with the data source via HTTPS with the given parameters. After obtaining the datagram, it forwards the datagram to the NOS Contract via the Relay.

- **Response** The NOS Contract returns the datagram to the smart contract.

5.3 Data Flow

A datagram request comes in the form of $m_1 = (params, callback)$. The parameter specifies the data source url, any specific requirement for content and the expected delivery time of the response. Then, the NOS Contract

forwards $m_2 = (id, params)$ to the Enclave where the id is a uniquely generated number. The NOS Contract then receives $m_3 = (id, params, data)$ from the NOS server where data is the requested datagram. After checking and confirming the consistency of parameters in both the request and response, NOS Contract generates $m_4 = (data)$ to user's smart contract. Messages between layers in the system require authentications using digital signatures.

5.4 Improved Security and Confidentiality

- **Private Datagram Request** Not all transactions have to be publicly visible. NOS supports permission-based user confidentiality by encrypting parameters in the datagram requests. Only users with permission will be able to see the data responses.
- **Gas Sustainability** Ethereum requires transaction initiators to pay gas costs. Such a design has the risk of malicious users triggering calls to steal gas, causing the depletion of gas and potential denial-of-service attack on the application level. NOS Protocol ensures that an honest system will not run out of money and that an honest requester will not pay excessive fees.
- **Hardware Code Minimization** The Trusted Computing Base (TCB) is a hybrid of an on-chain and off-chain computing environment. Computation of smart contracts on blockchain are slow, costly and transparent. To establish an almost perfectly secure communication between components, we need to minimize the code in the TCB. The logic behind this is that theoretically smaller code bases tend to be harder to attack. In the TCB, code sizes in the Enclave and the TC Contract have been minimized, with the Enclave having only around 2000 lines of C or C++ code and the Contract having only about 100 lines of Solidity code.

5.5 Use Cases

A single NOS host can handle around 65 transactions per second. Furthermore, NOS is easily parallelized across many hosts, as separate NOS hosts can serve requests with no interdependency. For comparisons, Ethereum handles between 20 to 30 transactions per second while Bitcoin handles around 7 transactions per second.

- **Website API Data Scraping** NOS can easily transfer data in bulk in JSON or XML from HTTPS-enabled websites to blockchain. For example, MLS is a trusted source for all real estate data. The API supports data of recent sales, public and private schools, demographics, home values and market trends. NOS will enable close-to-real-time data delivery from off-chain entities to on-chain environments.
- **Legacy System Migration** NOS has the ability to migrate data from traditional SaaS database systems using a Transport Layer Security (TLS) protocol.
- **Modern Application Interface** NOS system can connect blockchain with modern applications such as Facebook messenger and WeChat.

6 Future Work

6.1 Scalability

Ethereum processes all smart contracts on one chain in serial which bottlenecks throughput and dramatically reduces usability, especially when there are massive contracts and complicated data on the chain. We will look closely into how we plan to scale up blockchain through the adoption of Plasma and Sharding.

6.1.1 Plasma

Plasma is a protocol designed to solve scalability issue by building a tree structure of blockchains where various application chains (Child or Plasma Chains) are connected to a single root chain (Main Chain). The main chain serves as the backbone of the entire system, while each child chain is tailored to the need of a specific application. Basically, each block of the main chain contains references to the boundary of child chains. As new blocks are generated on the child chain, the main chain will create new blocks to reflect the new boundary, with cross-chain communication implemented as a message system to keep the isolation of chains.

The efficiency of the main chain can be significantly improved by offloading some transactions from the main chain to Plasma chains, especially if proper incentives are given to Plasma operators. Another advantage is the

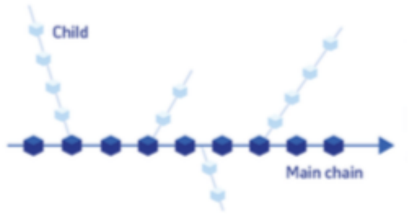

Figure 5: Multi-chain structure

flexibility of Plasma chain implementation, so long as it can be effectively cross-checked by contract on a parent chain. With new cryptographic approaches one can extend Plasma implementation with transactions utilizing ring signatures of zkSNARKs, a zero-knowledge verification mechanism to validate the correctness of a computation without executing the code, for confidentiality of end users.

Each child chain can be tailored to serve different purposes based on technical need. Application-specific smart contracts will be stored on the child chain, and the main chain will be used for consensus and distributed computation. Because the main chain can tap into a distributed global computing power, transactions on the child-chain will be calculated at a speed much faster than transactions on a traditional one-chain structure. Transactions can even be reversed if a participant within the child blockchain is proven to have acted maliciously.

At Ankr, we believe the main chain will also provide a native authenticated data feed service for off-chain data to relay to each child chain. Currently, existing oracle solutions operate separately from the blockchain framework and are limited in compatibility. We propose an user-friendly universal API for each child chain to connect to off-chain entities. Existing business can build decentralized autonomous applications on the child chain with powerful computing power and native data feed service provided by the main chain. The different tailored needs can be categorized as follows:

- Low transaction volume but high transaction amount, e.g. Real Estate
- High transaction volume but low transaction amount, e.g. E-commerce
- Real-time requests and responses, e.g. Prediction Market

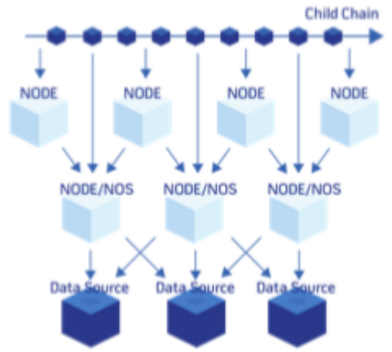

Figure 6: Multi-chain with NOS

6.1.2 Sharding

Sharding is another scaling solution that uses shards, or micro-chains, to process separate types of transactions on the congested blockchain. By classifying transactions on the chain, only a group of nodes need to verify the relevant transaction. Sharding removes the need for the entire network of nodes to process every individual transaction, thereby increasing TPS on blockchains.

Together, the two scaling solutions can compound the network's TPS while retaining security on popular blockchain protocols like Ethereum, creating a decentralized ecosystem that can potentially support a significant number of transactions per second. So if there is a 100x from Sharding and a 100x from Plasma, those two would theoretically give the underlying blockchain a 10,000x scalability gain, which essentially means this chain will be powerful enough to handle the majority of applications that people will try to do on the network.

6.2 Storage

Big data analytics is one of the use cases of Ankr's technology. Even though Ankr's DCC is friendly to data locality analytics, some computing customers also need high throughput data-intensive computing. IPFS is one of the

solutions, but Ankr needs features like short-time storage, meta-data storage, self-destruct, and storage rewards. Therefore, Ankr has developed a plan to build its own storage system on top of the IPFS kind of distributed file systems.

6.3 Mobile & GPU

As discussed in section 2, the future progress of TEE on mobile or GPU environments will enable Ankr to expand potential computing resources to new devices or hardware.

- **Mobile**, although limited by battery and individual computing capabilities, can provide a huge number of computing candidates for high throughput jobs such as data processing.
- **GPU**, on the other hand, has much more computing power than a CPU and can be a good candidate for the task requiring supercomputer or for intense computing capability like deep learning.

In short, along with Intel SGX, TEEs on the variety of platforms will give Ankr the capability to offer computing services with most suitable hardware, which will be very helpful for improving the usability of its product. Ankr will closely monitor the progress of chip manufacturers and incorporate these possibilities into Distributed Cloud Computing as early as possible.

6.4 Smart Identity and Credit System

Ankr's embedded oracle service on the blockchain enables off-chain assets and data to be digitized, tokenized, and written into smart contracts with an immutable log of all transactions. This allows users on the blockchain to then establish identities as smart contracts within the ecosystem (e.g., houses, cars, bonds). Each user exists individually on the chain, and the platform has the potential to record all user events via the blockchain's smart contracts. In the future, the connected Internet of Things will enable a truly innovative credit system that allows both sides of a transaction to be transparent and untampered.

Figure 7: Programmable Interfaces

Figure 8: Context within the Distributed Cloud

6.5 Programmable Interfaces

A well-designed public API is crucial for business adoption and usability. Ankr's blockchain does not change the existing PRC and thus is compatible with the majority of existing DApps. A Ethereum DApp developer would be able to work on Ankr's blockchain directly without any modification of their code. Beyond that, Ankr will provide extra RPC and CLI to support DCC for computing customers. These RPC and CLI will provide functions

including but not limited to submiting, retrieving, monitoring useful work tasks.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[3] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[4] D. P. Anderson, "Boinc: A system for public-resource computing and storage," in *proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*. IEEE Computer Society, 2004, pp. 4–10.

[5] D. P. Anderson, E. Korpela, and R. Walton, "High-performance task distribution for volunteer computing," in *e-Science and Grid Computing, 2005. First International Conference on*. IEEE, 2005, pp. 8–pp.

[6] D. P. Anderson, C. Christensen, and B. Allen, "Designing a runtime system for volunteer computing," in *SC 2006 Conference, Proceedings of the ACM/IEEE*. IEEE, 2006, pp. 33–33.

[7] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson, "Cost-benefit analysis of cloud computing versus desktop grids," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*. IEEE, 2009, pp. 1–12.

[8] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol." in *NSDI*, 2016, pp. 45–59.

[9] F. Zhang, I. Eyal, R. Escriva, A. Juels, and R. Van Renesse, "Rem: Resource-efficient mining for blockchains." *IACR Cryptology ePrint Archive*, vol. 2017, p. 179, 2017.

[10] Intel Corporation, "Intel® software guard extensions programming reference," 329298-002us edition, 2014.

[11] ——, "Intel® software guard extensions sdk," https://software.intel.com/en-us/sgx-sdk, 2015.

[12] ——, "Intel software guard extensions enclave writer's guide," https://software.intel.com/sites/default/files/managed/ae/48/Software-Guard-Extensions-Enclave-Writers-Guide.pdf, Accessed: 2017-2-16.

[13] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town crier: An authenticated data feed for smart contracts," in *Proceedings of the 2016 aCM sIGSAC conference on computer and communications security*. ACM, 2016, pp. 270–282.

[14] Oraclize, "The provably honest oracle service," http://www.oraclize.it, 2016.

[15] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," Ph.D. dissertation, 2016.

[16] M. Castro, B. Liskov *et al.*, "Practical byzantine fault tolerance," in *OSDI*, vol. 99, 1999, pp. 173–186.

[17] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," 2017.

[18] "Randhound and randherd," http://www.oraclize.it, Accessed: 2017-2-16.

[19] J. Poon and V. Buterin, "Plasma: Scalable autonomous smart contracts," *White paper*, 2017.