Hypernet Protocol

Distributed High-Performance Computing Powered by Blockchain Technology

Hypernet is a protocol for parallel computing over a decentralized, dynamic, distributed network utilizing idle computational power. It provides an API suite to efficiently perform global data operations in parallel that are robust to limited bandwidth and peer dropout using Distributed Average Consensus (DAC) over a decentralized network of devices.

Abstract

The Hypernet Foundation is launching the Hypernet protocol in order to make high-performance parallel computing a consumer good. Currently, only large corporations can truly benefit from massively parallel computing by exploiting their oligopoly on consolidated computing power. However, abundant processing power from the industrial Internet and data from ubiquitous, inexpensive sensing technologies will soon make decentralized computing more powerful and less expensive than centralized models. Hypernet takes advantage of this trend through a new parallel programming model that unites the power of blockchain technology with the simplicity and robustness of distributed average consensus to enable high-performance computing on the decentralized, heterogeneous, dynamic networks of the future. This technology will finally empower consumers to take on ambitious, novel, massively parallel projects in big data, AI, and modeling with the tools to compete against the likes of Google, Amazon, and Facebook – and win.

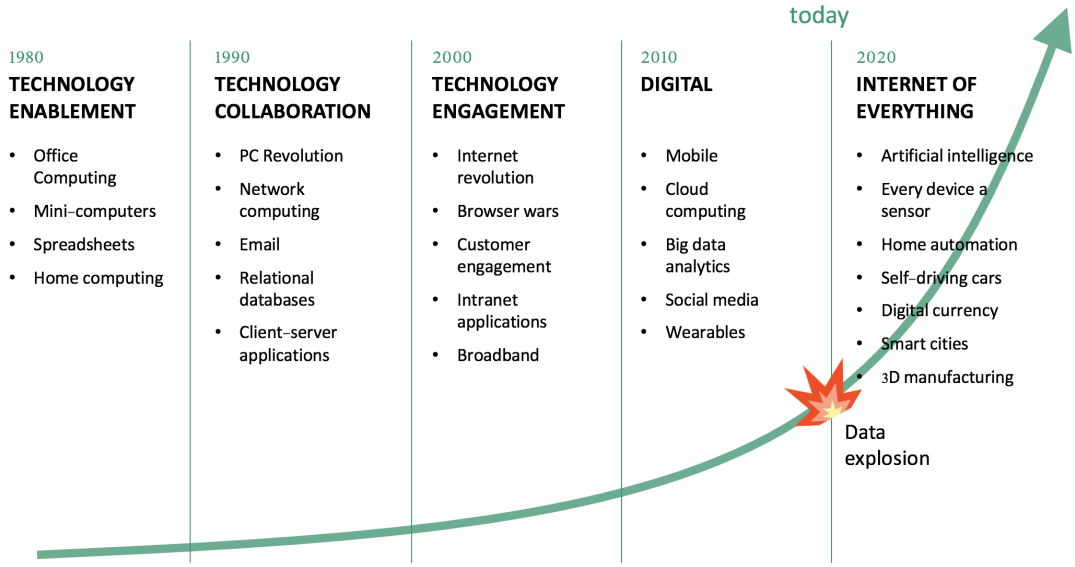
Tuesday 27th March, 2018: Version 1.1

Contents

1 Introduction

1.1 Background

An extraordinary amount of data is generated every second of every day. The production of information is growing so fast that Cisco is anticipating that by 2020, 15 billion terabytes of data will be generated per year [2]. We are entering a new Internet age, the Internet of Everything, where everything (e.g. devices, sensors, people) is connected and the amount of data is exploding. Leveraging this ocean of data provides opportunities in various verticals; for example, untethered augmented reality, real-time consumer data analytics, environmental disaster relief intelligence, or intelligent traffic flow.



Acting on this opportunity is certainly a challenge that will require significant quantities of computing power; Hypernet is answering the call to provide that power. Data alone is no more than digital exhaust unless it can be processed and analyzed. Currently, the amount of information being created far out paces the resources available to anatomize the data and make it useful. The Hypernet protocol makes it simple and efficient to tap into latent computational power by providing the gateway to the billions of devices that exist in the world capable of contributing unused capacity to solving some of humankinds greatest challenges.

Up to this point there has been no effective solution to harness the world's unused computational capacity (found in smartphones, tablets, embedded devices, and personal computers) to function as a unified dispersed global supercomputer that operates efficiently, securely and resiliently. The current hub-and-spoke model of funneling vast data streams to a server farm to be processed before closing the loop on what decision is to be made and what action must be taken is inherently inefficient and is a serious limiting factor to all big data opportunities. Additionally, simply porting existing centralized cloud ar-

chitecture solutions to a heterogeneous decentralized network is not workable and is not poised to gain from the inherent characteristics of such a network. An approach specific to distributed computation must be developed, and that is the precise aim of the Hypernet.

1.2 Overview

In order to bring big data to the edge and leverage the massive latent computer power already found in the digital wild, Hypernet is developing a block-chain based resource scheduling protocol, a containerized execution environment, and an API based on the principle of distributed average consensus (DAC) to provide the capability for true parallel public resource computing, not just grid computing. Hypernet's API naturally facilitates efficient collective operations on distributed data even under dynamic network scenarios. This API suite will allow developers to incorporate data-center-scale computational capability into their distributed applications.

Development of a robust decentralized scheduling protocol based on blockchain technology coupled with a distributed computing API based on DAC will allow many essential ingredients in scientific computing to be not only feasible, but efficient to deploy on dispersed, dynamic networks, enabling new business models and new data processing techniques. Dot products, matrix-vector products, linear and nonlinear least-squares regression, and various methods of solving systems of simultaneous equations in parallel can be encompassed in a single framework. Furthermore, an approach based on global data reductions allows the API to feel natural to developers of high performance computing (HPC) codes which rely on the Message Passing Interface (MPI) standard.

This white paper summarizes the structure and operation of the Hypernet computational ecosystem. In addition, it presents the essential mathematical components of DAC (stability and convergence), discusses its proposed implementation in an API for Hypernet, and details how it is incorporated into several essential numerical tools for the deployment of HPC applications on dispersed dynamic networks.

2 Problem

The current computing paradigm is centralized. Data, the life-blood of the modern economy, is generated in the real world and then forwarded to a remote data fusion center. Only after the data has been centralized can inference begin. This hub-and-spoke model is doomed to obsolescence with the proliferation of modern sensing technology.

Cloud computing was the savior technology five years ago, when local devices were incapable of providing the requisite compute power or storage space necessary for meaningful tasks. In centralized cloud computing, for a single data request or transfer, data needs to move through an average of twenty network gateways, each delaying actionable intelligence. This delay was acceptable when society was just learning how to benefit from

real-time data acquisition, but now it constitutes a prohibitive bottleneck.

As we are entering a new Internet Age, the production of data and subsequent data traffic is exploding. Real time data collection represents the new status-quo. Today's cloud computing will likely fall far short of meeting the needs of digital businesses [1], as sending data back and forth to centralized servers is inefficient, unpractical, and costly. The amount of data produced by the Internet of Everything, the need for local interaction and real-time analysis will push computing towards the data, closer to sensors and things [1].

In the near future, network congestion is expected to be compounded by the billions of online devices that are attempting to communicate with the cloud. Sensor proliferation will swamp existing infrastructure. Insufficient bandwidth creates bottlenecks and delays in computing tasks. Andreessen Horowitz has already predicted the end of cloud computing in 2016 [4]. Indeed, the role of centralized cloud computing will shift toward coordination, aggregation, archiving, and traditional back-office processing [1].

Secondly, storing and processing large amounts of data in the cloud is costly. As much as 85% of the price of computing is associated with infrastructure, hardware, maintenance, personnel, and cooling [3]. These costs are unnecessary and unsustainable as there are already millions of untapped computing devices in the world that could be leveraged for their computing power.

Furthermore, today's HPC market is dominated by only a few major players. It is a market that is dominated by a few large corporations, such as Amazon Web Services or Microsoft Azure. This leaves little choice for consumers of computing power.

Moreover, consumers face severe data security threats and trust issues, having personal information stored in the cloud. Data breaches and cyber attacks are becoming more frequent. This is a serious problem for corporations using the cloud and their customers whose sensitive personal information, such as credit and financial identity, are stored in centralized data repositories.

A new technology that enables high-performance computing in a secure, rapid and robust manner for the billions of connected things, devices, and sensors is needed.

3 Solution: Hypernet Ecosystem

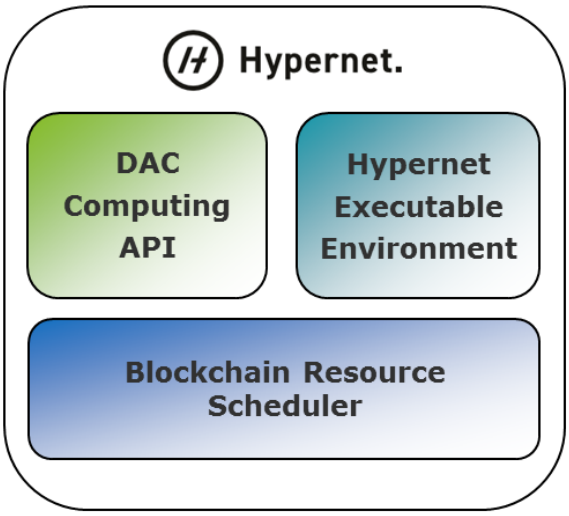
3.1 Overview

The Hypernet Foundation is partnering with developers to create a solution to the problem of centralized cloud computing: the Hypernet protocol, the first public resource computing network capable of providing true parallel computing. Together with its applications, the Hypernet ecosystem is expected to be a decentralized platform that connects individuals and companies who need computing power with those who want to sell time on their idle networked machines. Hypernet achieves this by leveraging the principles of distributed optimal control and cryptographic blockchain technology.

The Hypernet software infrastructure has three primary components:

- Blockchain Resource Scheduler
- Distributed Average Consensus Computing API
- Hypernet Executable Environment

Sustaining a global computational ecosystem on top of this infrastructure requires participants (sellers) to offer compute resources on individual devices and clients (buyers) to submit jobs to the network. The prices for compute time on seller hardware will be dictated by the market and depending on the job offer could also be scaled by the hardware specifications on each device. The blockchain scheduler facilitates the matching of buyers and sellers and provides a platform for smart contracts to be signed between the two parties for services rendered to be paid in Hypertoken. The executable environment offers a containerized, modular environment to keep buyer's executables isolated from the seller's hardware as well as giving control to the seller over how much of the machine's resources are devoted to the Hypernet.



3.2 Hypernet Participants

The Hypernet ecosystem represents a two-sided market consisting of buyers of computing power and sellers of computing power. A buyer is anyone who wants to run a compute task on the Hypernet. This group will be comprised of consumers with a variety of high performance computing needs, such as users of computationally intensive smartphone applications, research scientists, or Wall Street financial analysts. A seller is anyone with a computer that can be connected to the Internet.

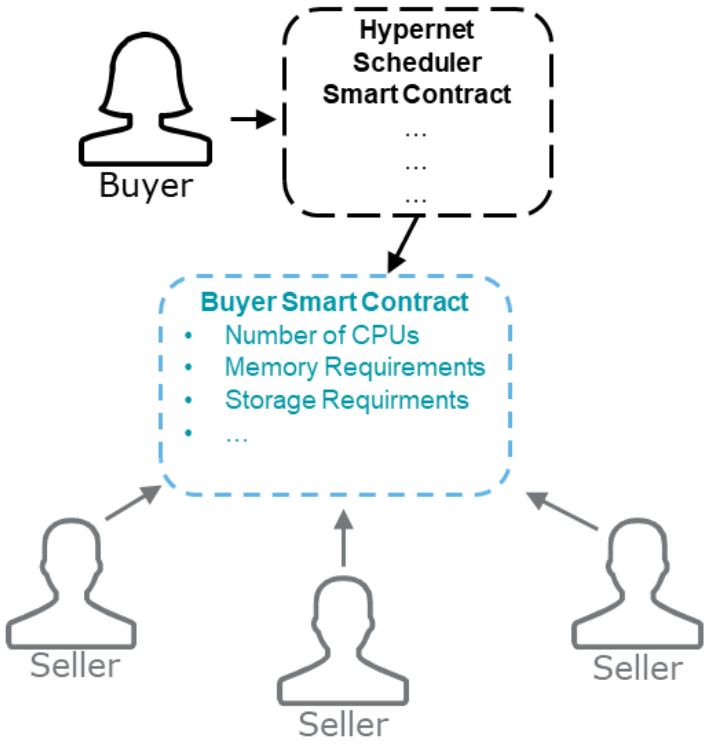
Buyers will purchase compute time on nodes with some minimum technical specifications (memory and disk space for example) via a smart contract initialized with the buyer's signature. Sellers with viable hardware will offer their devices for a given price

and be matched with contracts as they enter the market. If the price and hardware of a seller match a buyer's specifications, the two will be matched by adding the seller's signature to the smart contract. The marketplace for the computation will be governed by the free market, so the price for compute time will fluctuate based on supply and demand. Buyers can increase their priority by offering to pay more and sellers can win more bids by offering reasonable prices for their services.

The Hypernet ecosystem utilizes smart contracts (generated through the blockchain scheduler) to facilitate resource allocation, finalize secure payment, and to allow users to generate custom contracts of their own. Smart contracts are thus a key component of the distributed resource scheduler that will be signed by multiple participants during a transaction to buy compute time. As the needs of the community and the technological landscape change, the blockchain scheduling protocol will allow for the creation of more sophisticated smart contracts.

3.3 Job Placement

The buyer will place an order by calling the Hypernet smart contract scheduler which is a factory for creating smart contracts. The buyer will provide the scheduler with the minimum requirements of their job, such as the number of CPUs, amount of memory, storage, network bandwidth, and the price they are willing to pay. The scheduler factory will then automatically produce a smart contract for the buyer which sellers can sign if the specifications and pricing match what they are willing and able to provide. Sellers are made aware of new job contracts via the broadcast of Ethereum network events.



Once the contract has been signed by the requisite number of sellers, it is published and the committed sellers will cooperate on the order until completion of the job.

3.3.1 Checkpoints

Buyers may include checkpoints in their code they submit to the participating sellers. Checkpoints are a standard technique for providing suitable restart locations for long-running calculations in which the code may fail before completion but after having performed a substantial amount of work. When sellers successfully reach a buyer specified checkpoint, they will be paid for the work they have done up to that point.

3.3.2 Process Replication

Due to the inherently unreliable nature of dynamic networks, process replication will be a primary means to ensure that the loss of a compute node does not compromise an entire job. In addition, process replication offers a means for peers to check that other peers are indeed contributing to the calculation in which they are participating. When a group of peers that are assigned the same data reach a checkpoint, the peers can ensure that no participant has cheated by hashing their current result (which should be the same across all peers) with a piece of public information (such as their process ID) and sharing this with the group. This procedure is similar to the proof-of-spacetime scheme used in other decentralized projects.

3.4 Cohesion Through Consensus

3.4.1 Robustness Through Cooperative Decentralization

Nakamoto consensus is the logical framework that currently underlies all distributed cryptographic ledgers [5]. It is this principle that endows the distributed ledger architecture with its inherent resilience to network uncertainty and malicious attacks. Similarly, consensus in the context of weighted distributed average is of particular importance in the fields of multi-agent estimation and control. It is an attractive principle for performing reductions on distributed data in that it inherently bypasses the need for sophisticated routing protocols and overlay topologies for complicated networks. This section will present some of the basic mathematical theory of the DAC principle. Descriptions of a few fundamental use cases are also given for distributed memory applications in order to highlight the diverse utility of DAC as a parallel programming model in the Hypernet.

3.4.2 Theoretical Background

Consider a calculation in which $N_{process}$ agents must agree on their average value. The continuous time model for the local agent state governed by the DAC algorithm is given by the feedback model

$$\begin{gathered} \dot{x}_i(t) = u_i(t) \\ x_i \in \mathbb{R}^n \\ i \in \{1, ..., N_{process}\} \end{gathered}$$

where $x_i(t)$ is the numerical state of process i at time t, $\dot{x}_i(t)$ is the time derivative of the state, and $u_i(t)$ represents a particular consensus feedback protocol. In this white paper, the simple Nearest Neighbor protocol is considered for illustrative purposes

$$u_i(t) = \sum_{v_j \in \mathcal{N}_i} (x_j(t) - x_i(t))$$

where $\mathcal{N}_i$ is the neighbor set of process i. The global system can be written as a dynamical system of equations of the form

$$\begin{gathered} \dot{x}(t) = -Lx(t) \\ x \in \mathbb{R}^{nN_{process}} \\ L \in \mathbb{R}^{nN_{process} \times nN_{process}} \end{gathered}$$

where L is the graph Laplacian matrix. It is straightforward to show that in the case of a connected network, the unique and universally convergent equilibrium state of this system is

$$x_i(\infty) = \frac{1}{N_{process}} \left[x_1(0), ..., x_{N_{process}}(0) \right] \mathbb{1}$$

where $\mathbb{1} \in \mathbb{R}^{N_{process}}$ is the vector of all ones. This result means that the agents in the network not only come to an agreement on a value, but a particular unique value: the average of the initial conditions of the agents on the network.

The rate at which $x_i(k)$ converges to $x_i(\infty)$ for this feedback controller is proportional to the smallest nonzero eigenvalue of system Laplacian L. Furthermore, it can be proven that the equilibrium state can be attained under dynamic, directional topologies with time delays [6]. This notion of consensus satisfies the definition of a distributed protocol since each process requires communication only with a set of neighboring processors and there is no need for a fusion center or centralized node with global information. It is in this sense that consensus is exploited in the Hypernet ecosystem to achieve a variety of useful tools for distributed computing.

3.4.3 Example Applications

Distributed dot products:

The dot product is one of the most important primitive algebraic manipulations that must be available to a parallel computing application. Without a method for computing distributed dot products, critical parallel numerical methods (such as conjugate gradients, Newton-Krylov, or GMRES) for simulations and machine learning are not possible.

Applying DAC as described in the previous subsection, dot products can be performed on distributed data in a trivial manner by first performing the dot product on the local sub-vectors, and then performing consensus on the resulting local scalar values. After consensus, the result is then multiplied by the number of processes in the computation.

Distributed matrix-vector products:

Arguably as fundamental as the distributed dot product, distributed matrix-vector products are essential for most iterative numerical schemes, such as fixed point iteration or successive approximation. Consider the case in which a matrix is partitioned column-wise (where each process owns one or more columns of the global matrix) and a matrix-vector product must be performed. DAC is applicable to both scalar and vector quantities, so this operation can easily be performed by first weighting the local columns appropriately (local matrix-vector product) and then performing average consensus on the result-

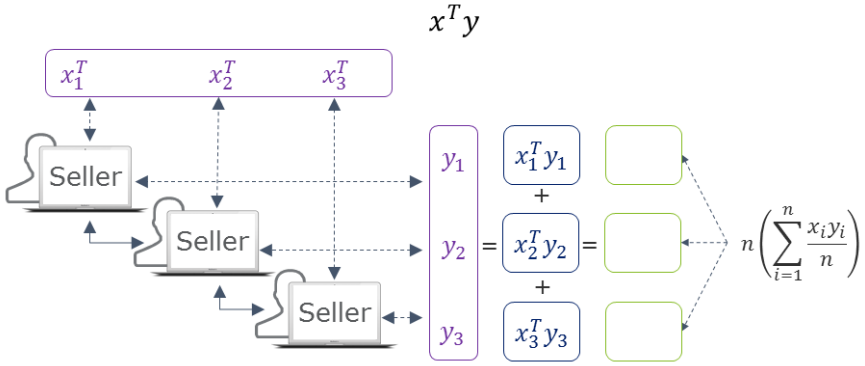


Figure 1: Data locality and numerical procedure for computing a distributed dot product.

ing local vectors. The consensus result is then multiplied by the number of processes in the computation.

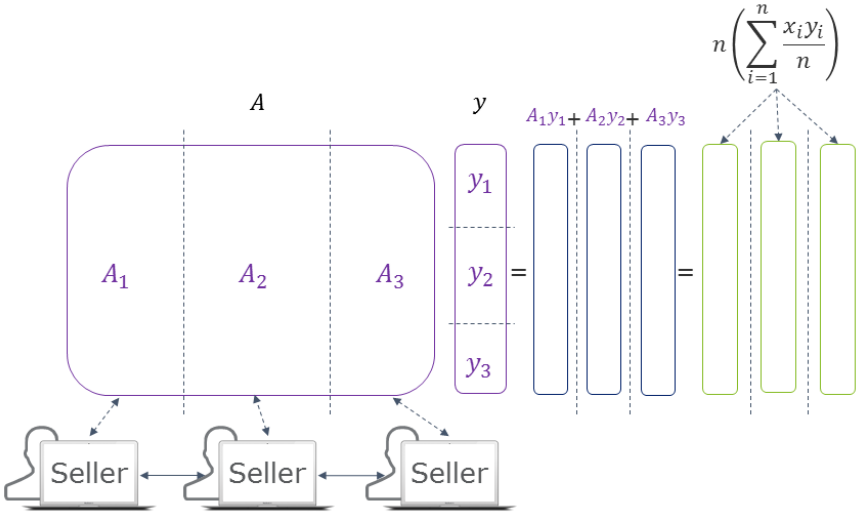


Figure 2: Data locality and numerical procedure for computing a distributed matrix-vector product.

Distributed least squares:

Least squares is one of the most important regressions used by scientists and engineers today. It is one of the main numerical ingredients in software designed for maximum likelihood estimation, image reconstruction, neural network training and much more.

Consider the problem of finding the least-squares solution to an overdetermined system of equations.

$$Ax = b$$
$$A \in \mathbb{R}^{(nN_{process}) \times M}$$

Under the appropriate assumptions, the solution to this problem is given by the pseudo inverse.

$$x = (A^T A)^{-1} A^T b$$

In many parallel computing applications the sensing matrix, A, is distributed row-wise and the least-squares solution, x, is solved for locally on each computational node since it is small in comparison to the total number of measurements. This means that each process in the network owns a few rows (measurements) of the sensing matrix A and the target b. The least squares solution can be recovered by a straightforward application of DAC.

Let A_i and b_i be the rows of the sensing matrix and target vector owned by the compute node i. Each process calculates the products, $A_i^T A_i$ and $A_i^T b_i$ in its local memory. Then DAC is performed on the quantity $A_i^T b_i$ and $A_i^T A_i$, which are both small compared to the total number of observations in A. The result of the DAC operation, $\frac{1}{n}\sum_{i=1}^{n} A_i^T A_i$, which is present on every node, is multiplied by the number of processes in the computation leaving every node with a copy of $A^T A$ which can be inexpensively and efficiently factorized locally to obtain the least squares fit to the global data set.

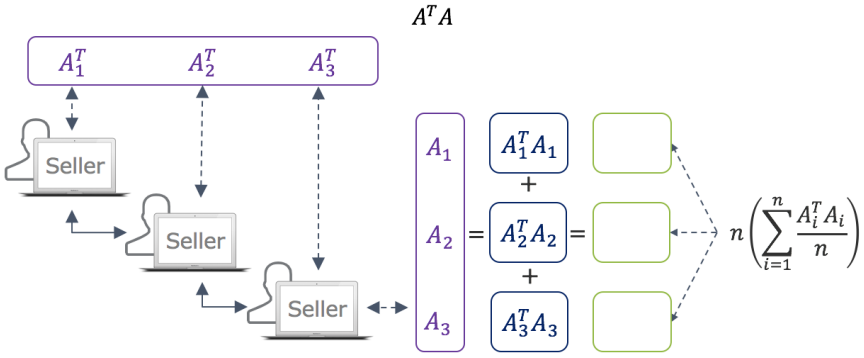


Figure 3: Data locality and numerical procedure for computing a Grammian matrix required for least squares regression.

Decentralized Bayesian parameter learning:

Many industrial applications benefit from having a data driven statistical model of a given process based on prior knowledge. Economic time series, seismology data, and speech recognition are just a few big data applications that leverage recursive Bayesian estimation for refining statistical representations. DAC is a natural fit for facilitating recursive Bayesian estimation on distributed data sets.

Again consider the scenario of n dispersed computational nodes. Every node is trying to estimate a quantity, x, via a probability distribution, $\pi(x) = p(x \mid y_{1:n})$. Each node $i \in \{1, ..., n\}$ makes an observation, y_i, that is related to the quantity of interest through a

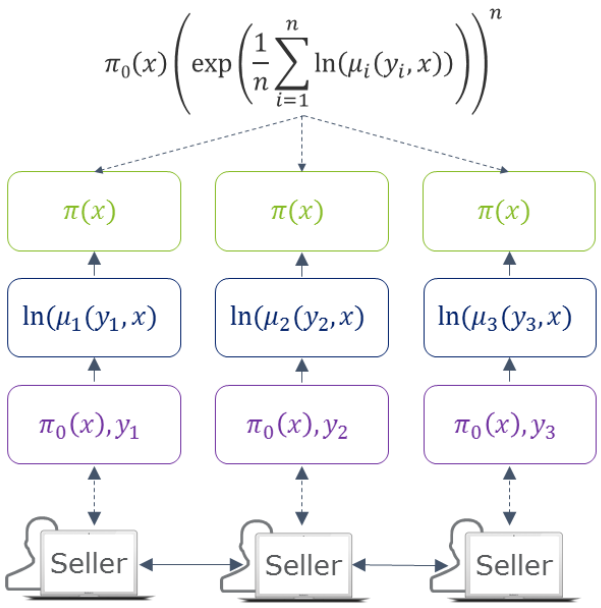


Figure 4: Data locality diagram for distributed Bayesian parameter estimation.

predefined statistical model $\mu_i(y_i, x)$. Under mild conditions, the Bayesian estimate of x is proportional to

$$\pi(x) \propto \pi_0(x) \prod_{i=1:n} \mu_i(y_i, x)$$

where $\pi_0(x)$ is the prior distribution based on past knowledge. The posterior estimate, $\pi(x)$, conditional on the distributed measurements can be easily computed using a DAC approach by rewriting the product term in the form of an average quantity.

$$\pi(x) \propto \pi_0(x) exp(\frac{1}{n} \sum_{i=1:n} ln(\mu_i(y_i, x)))^n$$

Leveraging DAC to compute the global average of the distributed measurement functions allows each node to consistently update its local posterior estimate without direct knowledge or explicit communication with the rest of the global data set.

Hypernet will adopt Hypertokens, a decentralized ERC-20 compliant cryptocurrency to be initiated by The Hypernet Foundation for transactions on its ecosystem. A total of 100,000,000 Hypertokens will be created. The tokens will not be mined but will be released upon the initialization of the Hypertoken protocol.

3.5 Network Currency

Hypertoken will serve as a mechanism for buyers to pay sellers for their compute time. Additionally, Hypertokens provide a means for users to stake network peers and track their reputation in the ecosystem. The ability to associate a reputation with buyers and sellers will be crucial for the health of the Hypernet ecosystem.

3.6 Collateral

In order to discourage malicious behavior, both buyers and sellers must post collateral to join a job contract. Buyers submitting a job to the network post the full cost of the job into the smart contract when it is created. Sellers joining into a job contract must also post collateral. The amount of collateral required by a seller is a parameter that is set by the buyer, thus providing a quality-of-service function. As sellers reach checkpoints in the job they are running, the collateral for the work they have completed will be returned to them.

3.7 Voting

In the event that a major change must be made to the Hypernet protocol in the future, the revision will be subject to a voting process. The number of votes a peer has is proportional to the number of Hypertokens associated with their account.

4 Technical Roadmap

Figure 5 shows the expected development time-line for The Hypernet Foundation software stack.

4.1 Blockchain Resource Scheduler

The primary feature of the Hypernet ecosystem is the resource scheduler. The Hypernet Foundation will work with developers to create the scheduler in the form of a graphical dashboard (with a command line option for those who prefer a more utilitarian aesthetic). This scheduler will perform the same function that a workload manager provides in a static linux cluster. On the buyer side, the scheduler will take as input the program to be distributed to the sellers, as well as the amount of computational assets desired and the required runtime. Buyers will also specify, through the scheduler, any necessary data files that are to be forwarded to sellers and what is to be done with any files generated by their binaries on seller machines (i.e. a forwarding address, a payment for temporary storage,

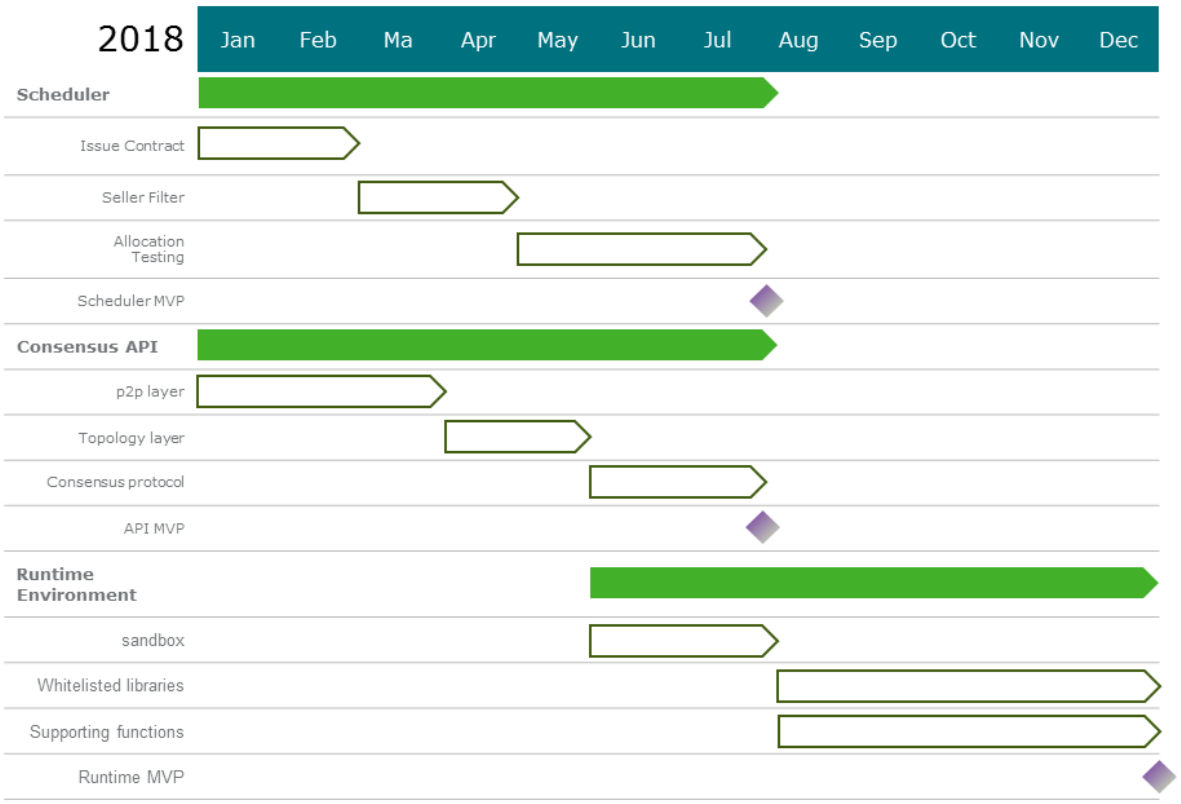


Figure 5: Timeline for developing key technology stacks.

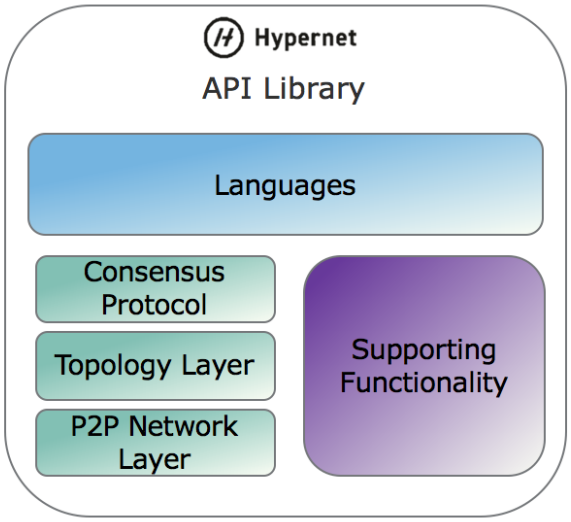
deletion on completion, etc). Once a buyer has finalized the parameters of their job, the scheduler will generate the required smart contract to be executed on the blockchain.

Sellers will be able to use the scheduler to specify the terms of use of their hardware (when the hardware can be used, how much memory and disk are to be available, and how much of their CPU time they are willing to offer). The scheduler will then handle the matching of the seller's terms to available contracts currently on the ledger.

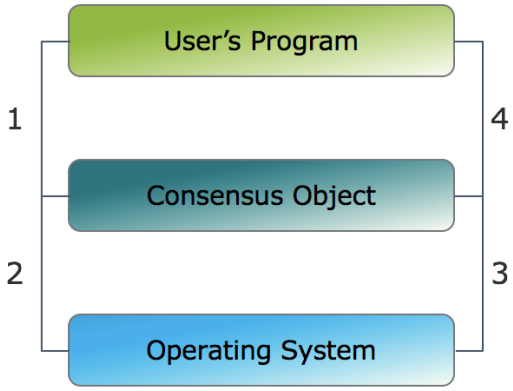
4.2 Hypernet Consensus Infrastructure

The Hypernet Foundation will work with third parties to develop an API library for leveraging distributed average consensus in the Hypernet. The API will have bindings in coding languages that are of importance in scientific computing, large-scale optimization, and machine learning such as python, and C++. Additional bindings will be made available as the need arises.

Development of the API library will consist of two primary silos: consensus p2p infrastructure, and supporting functions. The consensus infrastructure can be partitioned into three layers, beginning with a p2p networking layer that handles establishing a tcp/ip connection with a participating compute node. This networking layer will be utilized by a topology layer that manages the set of neighboring peers participating in the consensus job. The final layer will implement the average consensus feedback protocol on top of the local topology for reaching distributed average consensus. Sitting alongside this functionality will be supporting functions for developing software for public resource computing, such as data types and read/write functionality.



The API's operational concept is given below. First the user program will initiate the particular consensus object (e.g. distributed dot product, least-squares, user-defined functional, etc.). Next the consensus object will signal the operating system to begin an asynchronous connect. The operating system will indicate when the connection is established and return control to the consensus object. The consensus object will return to the user's program once convergence on the particular data structure has been reached.



4.3 Hypernet Executable Environment

In order to facilitate security and compatibility within a system composed of heterogeneous hardware, The Hypernet Foundation is expected to feature a sandbox environment for running executables to isolate them from the seller's hardware. The environment will include white-listed libraries that are trusted to be run on the Hypernet. The runtime environment will also provide a DAC-based progress querying function so buyers can monitor the progress of their jobs.

5 Conclusion

Hypernet is built from the ground up to provide a truly decentralized environment for extremely large scale parallel computing tasks. By employing the principle of Distributed Average Consensus, the power of every device can be tapped to its fullest to facilitate new science, power smart cities, and enable large scale data processing. From armchair AI developers to Wall Street software analysts, the expectation for low latency high performance computing is quickly out pacing the abilities of centralized clusters. Not only is the technology itself powerful, but it offers an elegant remedy to an important market inefficiency; processor cycles are routinely wasted, but paradoxically, are in very high demand. We have shown that we can salvage these cycles and re-purpose them for consumers – and better yet, we can do it in parallel. The Hypernet is the first parallel computing infrastructure aimed at providing a simple API for leveraging Distributed Average Consensus, making it extraordinarily well positioned to disrupt the high performance computing and data analytics market in the long term. Therefore, Hypernet is ready to revolutionize the parallel processing paradigm and everyone can be a part of it.

REFERENCES

[1] Thomas J. Bittman. "Maverick Research: The Edge Will Eat the Cloud". In: *Gartner* (September 2017).

[2] Cisco Global Cloud Index. *Forecast and Methodology, 2015-2020 White Paper*. 2016.

[3] Christos Kozyrakis. *Computer Systems Architecture, Virtual Machines and Data Center Introduction, Stanford Course*. 2014.

[4] Peter Levine. *The End of Cloud Computing*. 2016.

[5] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. 2008.

[6] Reza Olfati-Saber and Richard M Murray. "Consensus problems in networks of agents with switching topology and time-delays". In: *IEEE Transactions on automatic control* 49.9 (2004), pp. 1520–1533.