Beyond Byzantium: The Players of Ludos

Eric Harris-Braun

(Dated: 2024-07-01)

Introduction

The difficulty of distributed coordination is often described in the literature using the frame of the Byzantine Generals Problem. This frame starts from the assumption of a set of coordinating nodes trying to agree on a single data reality at a given time across all the nodes despite some of them being faulty, and then coordinating based on that consensus. The frame is understood to be necessary by the nature of distributed systems whereby different nodes will experience the world differently for many different reasons.[1] Solutions typically involve mechanisms for comparing state transition proposals from multiple nodes to confirm a consensus reality, usually through leader-selection, where for a given state transition, one node gets to apply its particular data-reality[2].

A careful reading of the "The Byzantine Generals Problem" papers[3] reveals one source of this frame: a mandate to build control systems for making critical automated decisions in contexts such as in nuclear power plant control and interpreting radar data about possible nuclear strikes. The Generals in the problem are understood to be sources of input (either from sensors or human agents) that are sent to Lieutenants who are understood to be the computers carrying out the orders indicated by those inputs. The critical insights of these papers are 1: proofs that simple majority voting on data from redundant systems is insufficient to overcome the fault cases, and 2: provision of algorithms they prove to solve the problem under various restrictions of how many nodes must not be faulty for correct decisions to be reached.

The core axiom (though not explicitly stated as such) of the Byzantine Generals Problem is that coordination starts AFTER "consensus on state", i.e. that the Lieutenants can't execute their plan until they have followed the consensus algorithm and arrived at single data reality[4]. This axiom at first blush seems sensible, and it is carried over into Blockchain-based distributed ledger systems which are often explicitly described as solutions to the Byzantine Generals problem.[5] This axiom gives rise to the fundamental data and process architecture of a single globally advancing chain of blocks that miners or stakers select into existence. The chain of blocks is the single data reality with miners and stakers acting as selectors of that expanding reality using the various proof-of-work/stake algorithms. Unfortunately this axiom does not actually correctly represent the realities of large scale distributed coordination, furthermore its use as a starting point results in some significant consequences that emerge from the resulting architecture:

1. Negative or zero scaling: Adding new miners or stakers to the network does not increase the capacity of the network as a whole to do coordinative work. In the case of proof-of-work reality selection, new miners only increase the amount of wasted energy used while achieving the same throughput in transactions-per-second. In the case of proof-of-stake reality selection (either chain-based or BFT leader-selection based) though there is less redundant burned energy work, there is still no overall throughput increase as the number of stakers increases.
2. Inherent centralizing dynamics: Blockchain reality selection (via mining or staking) produces an inherently centralizing network effect through "rich get richer" power concentration. The more you mine or stake, the more rewards you get, which you can spend on mining rigs or further staking to

[1] These differences can arise from the fundamental physical properties of the medium carrying messages between nodes and the way these media may introduce noise in messages or delay message transmission which results in nodes receiving messages in different orders. They may also arise because of intentional malicious behavior of nodes sending false information or simply not sending messages.

[2] The appropriateness of these leader-selection algorithms as solutions to distributed coordination is argued differently according to the particular selection algorithm. In the case of blockchain proof-of-work the argument is that the pure computational probabilistic nature of finding the cryptographic result and the energy cost of doing so ensures that collusion can only happen above 50% malicious nodes and then the cost of that collusion is higher than any value that can be gained from it. In the case of proof-of-stake algorithms, the loss of the "stake" is argued to be the incentive for non-malicious behavior when validating and presenting a given block for inclusion. These arguments often ignore the ways in which these selection algorithms defeat the very premise of distributing collaboration as they re-introduce centralizing dynamics.

[3] See: *The Byzantine Generals Problem*, Leslie Lamport, Robert Shostak, and Marshall Pease https://lamport.azurewebsites.net/pubs/byz.pdf and *Reaching Agreement in the Presence of Faults* Marshall Pease, Robert Shostak, and Leslie Lamport https://dl.acm.org/doi/pdf/10.1145/322186.322188

[4] In *Reaching Agreement...* this single data reality is called "interactive consistency" as it is about the vector of "Private Values" sent by each node.

[5] E.g. https://cointelegraph.com/blockchain-for-beginners/how-does-blockchain-solve-the-byzantine-generals-problem and https://medium.com/swlh/bitcoins-proof-of-work-the-problem-of-the-byzantine-generals-33dc4540442

> continually increase your rewards. These dynamics increase the influence of large scale participants. Fundamentally, you cannot successfully operate a decentralized system using a consensus algorithm which centralizes the power and wealth within that system.

Thus, the aspiration of such systems for decentralizing planetary scale coordination appears to us as extremely unlikely to be realized without a fundamental ontological shift.

In the spirit of the Byzantine Generals Problems, we offer a story to aid in discovering a new starting point from which to design systems for distributed coordination.

The Players of Ludos

Imagine a civilization, which for flavor, perhaps lived somewhere close to Byzantium, but nomadic, more in harmony with nature and great lovers of playing board games (like chess, checkers or go), whom we will call the Players of Ludos. Imagine that these highly independent and egalitarian nomadic bands with their long tradition of playing would gather for tournaments played in a large arena with many simultaneous boards and players.

Now at some point, because these Players of Ludos so loved their games, they decided they would like to keep their inter-band playing going year round, even as bands would be on the move, living in harmony with the land, as they did. And thus they devised to send written correspondence by messengers out to other bands with game moves. As you might expect, they soon realized that their game playing broke down, because messages between bands were not always reliably delivered. Sometimes just because the messengers were just lazy or distracted and would fail to deliver messages, sometimes the messengers would fail to protect the message from the rain which would cause the ink to run and garble the messages, and sometimes nefarious and overly serious game players wanting to affect the outcome of these games would purposefully make changes to messages! These failures resulted in players in different bands not seeing the same game reality and making incorrect moves.

At first the Players of Ludos believed that to solve this problem, they had to replicate their experience of the tournaments where players could simultaneously look at all the board's states before making moves. To this end they created a drawing, one for each band, of the "virtual arena" that they would be synchronized across all the bands. However, they knew that to do this would be complicated by the fact that messages of lists of moves sent between bands would never arrive at the same time or order. Because of their deep egalitarian ethics, they couldn't just elect one band as the authoritative sender of update-messages, rather they wanted a way of choosing different band's board-state-update messages over time in a way that was random and fair. They did indeed find several ingenious methods of choosing which band's proposal would be the "real" one for the next round of moves.[6] But they soon realized that starting with the assumption of having a single agreed upon arena drawing was actually an unnecessary starting assumption, and that an entirely different approach would make it possible to coordinate the games much more efficiently. We won't go into just how deeply inefficient their original "ingenious" solutions were (unless you care to read the footnote above) and just how much these solutions created the very inequality between bands that they were trying to avoid in the first place!

The heart of the new approach was just to do a few things:

1. Require individual players to keep track of their own moves
2. Require individual players to validate and keep track of a portion of other players moves
3. Require players to respond to requests of the moves of other players they are keeping track of..

But it all hinged on a few special abilities that the Players of Ludos used:

1. They developed a way to unforgeably sign any document.
2. They had a ingenious method to create a "fingerprint" of each move that looked exactly as if it were a human fingerprint, and just like a human fingerprint was different from all other fingerprints of other moves, AS WELL as being different from all other human fingerprints
3. They also had an even more ingenious method of very easily being able to tell if one fingerprint was similar to another, i.e. they could group themselves into "fingerprint neighborhoods" according to those similarities.
4. Finally they created a way to locate players by ensuring that all the fingerprint neighborhoods overlap

[6] One of these methods they called Proof-of-Wait whereby they would feed a small but very hard-shelled gourd to their pack animals. Now the seeds of these gourds had a very interesting property when passing through the animal's digestive tract. It so happened that most often these gourds would not be digestible. But occasionally and very randomly but also very predictably (on average once every 10 days) a gourd would be digested, and fascinatingly the seeds of the gourd would be transformed by the digestive juices of the pack animal into a unmistakable and otherwise unreproducible color which faded away in just a couple days. It also so happened, that this gourd had exactly as many seeds as there were nomad bands. Thus, the ingenious Players knew that they could send one gourd seed along with a proposal to update the board drawings, and that would unequivocally and randomly select one band's proposal, and they would be able to do so, on average every 10 days, and you couldn't cheat because the seed color faded before the next update! Of course it could happen randomly that two bands would get a gourd at the same time, but they just agreed to just use the proposal that had the longest list of moves. This worked, but it kept the pace of games quite slow, and moreover it involved sorting through a lot of pack-animal dung, which sadly some players actually came to believe was valuable work!

in such a way that it was guaranteed that you could always either send a message to someone in a neighborhood that's closer to the destination than your own, or know that you are in the neighborhood that's responsible for holding data of a given fingerprint.

Given these abilities this is how the Players coordinate:

1. When making a move in a game they take a fingerprint of the move, and then write a small document which contains the fingerprint of the move, the date, and the fingerprint of the previous document. They called these documents "Actions". Note how storing the fingerprint of the previous document creates an unbreakable Action chain.
2. They also "publish" the move by sending out a few messages to other players as follows:
    1. Send the full move and action documents to players whose human fingerprint is in the neighborhood of the move document, and of the action document. These players will be able to respond to requests for the actions and moves.
    2. Send the action document to the players whose human fingerprint is in the neighborhood of the publisher's human fingerprint. These players will be able to respond to requests of player move history. This is important because sometimes validation of a given move requires the ability to check the history of a player's previous moves.
3. Any player receiving a published document checks to see if all of the data in the document they receive conforms to the rules of the game (this may involve making requests of other players, for example to retrieve the history of previous moves). They then sign and send a receipt confirming the validity or invalidity of the published document back to the player that sent the document as well as to the other players in the same neighborhood who also should have received the original document.
4. Players periodically gossip with other players in their neighborhood about what they've heard about, validating and updating their records accordingly. Thus, players who cheat, including by changing their history's and reporting different moves to different players, will be found out because all moves must be signed, and the history of the moves is baked into the actions. Remember that each action contains the fingerprint of the previous action. Players who receive actions by gossip (i.e not as a result of publishing as in step 2.b) will eventually be able to detect any actions that show contradictory histories as soon as they see two different actions that use the same fingerprint of a previous action.
5. Finally, players who receive notices of cheating players, or who observe the cheating directly, may, depending on the particular rules of the game, simply drop communication with offending players, or give them warnings, as there are some circumstances where players may have sent conflicting messages accidentally, in which case they can send corrections.

With these simple steps all players could confidently recreate the state of the boards. Every move is signed and validated, and players receive confirmation from other players in similar neighborhoods about the correctness of all the moves. Players can request copies of moves they are interested in (both the specific moves themselves, and the actions that provide a history of the moves, by requesting them from players in the correct neighborhood of the moves. Of course it's not guaranteed that at any given moment in time if you ask a neighborhood for any given move that it will have reached the nodes who receive your request, but eventually they always will, and the answers returned will converge on the same reality.

Thus the Players of Ludos greatly improved the speed of their play. Adding a new band or new players did not slow down the play, in fact it actually increased the overall resiliency of the play. Furthermore they realized that they could increase the complexity of play from turn based board games, to arbitrarily complex games. As long as every player of a game started with the same rule-set, and they could deterministically validate any move, and they could get notification of cheaters, the system worked perfectly well for coordination.

There's a second phase to our story that relates to scaling up distributed coordination which we offer here in the broadest of strokes. It goes like this: a clever player created general rules for a "tournament" game, which could reference any other type of game that had winners and losers, thus allowing games to compose with other games. Because tournament winners gained social status in their society, they soon realized that many other of their social interactions could be encoded similarly as games. They even realized they could replace their monetary system with an accounting "game" where players simply recorded the granting and receiving credit with each other. All this further allowed them to live into their commitment to both independence and egalitarian ethics.

New Axioms for Distributed Coordination

This story is meant to elucidate what happens if we start from a different ground when thinking about distributed coordination. Our story only focuses on the first part of coordination, the low level mechanisms of distributed coordination but points to what's possible when one treats that capacity as something to assemble larger capacities. And so, from this fanciful account, we would like to offer somewhat less fancifully, two axioms[7] from which to

[7] The term "axiom" is often understood as being a statement that is self-evident and not proven but rather assumed. As in the cases of the parallel axiom of Euclidean geometry it seemed natural to be able assume that such lines never meet. Similarly it might seem

start when considering building systems for large scale distributed coordination:

1. Coordination arises from agents starting from the same ground-rules and acting as soon as parties can confirm that actions or interactions conform to those ground rules. (Thus, in our frame, coordination looks like a swarm of agents that converge on a direction, rather than all agents proceeding in lockstep agreement.)
2. Coordination is grammatic[8]. It comes from two forms of embodiment[9] by groups of independent agents: 1) embodying a collective understanding of the shape of interaction (the "ground-rules") or what you could think of as a geometry of the space of play, which removes enough uncertainty that it's worth risking to play. 2) embodying an ability to compose different coordinative subsystems that have different ground-rules.

Axiom 1 arises from the insight of not fighting against what's true about physical reality. Namely that different nodes in a network of interaction will experience different realities. In the real world there is no such thing as simultaneity, which means there is no such thing as global temporal ordering of events out of which to build a global state of a system. Instead, any coordination system must align its ontology with the truth that **global state actually does not exist**. Thus, we start with what does actually exist: local temporal state. This local state can be shared with, and validated by, others in conformity with pre-defined ground-rules. In so doing we can still achieve difficult and complex coordination safely (including in the context of problems on the scale of global monetary transactions) without the costs and bottlenecks that arise from starting from the ontology of a single shared global state. Holochain is an implementation of a system using this alternate frame.

Axiom 2 arises from the insight that systems for successful large scale coordination demand the property of anti-fragility, that is, they must perform better under perturbation[10]. Coordination happens in the context of fundamentally dynamic environments in which the coordinating elements are changed by the fact of their coordination. Coordination is a co-evolutionary context. We claim by this axiom that what meets the challenge of anti-fragility in such contexts is composable sub-systems, in which the composition comes out of a grammatics that embodies the actual dimensionality of the problem subdomains (i.e. their geometry), and by which agents in that context can react powerfully to perturbations because the available composability is dimensionally aligned.

The old axiom that coordination starts after consensus on state, leads system designers to figure out how to implement machinery for **Global Consensus**. Our new axioms lead us, instead, to implement tooling for **Scaling Consent**.

Holochain is built starting from the above two axioms and thus delivers on coordination at scale with out global consensus.

natural to simply assume that coordinated action begins after consensus on state. We hope here that it has been sufficiently demonstrated that taking on perhaps surprising axioms (as was done the "parallel lines allways meet" of spherical non-Euclidean geometry), provides a similar expansion of understanding and possibility.

[8] Think of the term "grammatic" as a way to generalize from the usual understanding of grammar which is linguistic. Where grammar is often understood to be limited to language, grammatics points to the pattern of creating templates with classes of items that can fill slots in those templates. This pattern can be used for creating "grammars" of social interaction, "grammars" of physical structures (we would call Christopher Alexander's "A Pattern Language" for architecture an example of grammatics) and so on.

[9] Insofar as our compute-powered platforms are meant to solve problems in particular domains, it follows that the ways those problems show up in the platform actually meet the dimensionality of the problem space. By this I mean that the independent variables or ontological entities that are part of the problem space are reflected in the compute system. That reflection I call **embodiment** in the system. A generalized platform for creating applications that solve problems must therefore embody this higher-level dimensionality of the problem space of "generalized application creation" itself, and it must do so in an evolvable manner. The use of the term "geometry" here is similarly intended to help elucidate the notion of dimensionality, in that geometries distinguish independent directions of motion and the relations between them.

[10] *Antifragile: Things that Gain from Disorder*, Nassim Nicholas Taleb, 2012.